A Joint Dictionary-Based Single-Image Super-Resolution Model

by

Jun Hu

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Image super-resolution technique mainly aims at restoring high-resolution image with satisfactory novel details. In recent years, leaning-based single-image super-resolution has been developed and proved to produce satisfactory results. With one or some dictionaries trained from a training set, learning-based super-resolution is able to establish a mapping relationship between low-resolution images and their corresponding high-resolution ones. Among all these algorithms, sparsity-based super-resolution has been proved with outstanding performance from extensive experiments. By utilizing compact dictionaries, this class of super-resolution algorithms can be efficient with lower computation complexity and has shown great potential for the practical applications.

Our proposed model, which is known as Joint Dictionary-based Super-Resolution (JDSR) algorithm, is a new sparsity-based super-resolution approach. Based on the observation that the initial values of Non-locally Centralized Sparse Representation (NCSR) model will affect the final reconstruction, we change its initial values by using results of Zeyde's model. Besides, with the purpose of further improvement, we also add a gradient histogram preservation term in the sparse model of NCSR, and modify the reference histogram estimation by a simple edge detection based enhancement so that the estimated histogram will be closer to the ground truth. The experimental results illustrate that our method outperforms the state-of-the-art methods in terms of sharper edges, clearer textures and better novel details.

Acknowledgements

It is a great pleasure to thank all who have helped me during my research.

I would give my immeasurable appreciation to my supervisor, Prof. Jiying Zhao, whose encouragement, patience and guidance benefit me much in the accomplishment and success of my study.

I'm also grateful to all my lab colleagues who supported me during my study.

Special thanks to my family and my girlfriend Ya Xing for their unfailing love. They have always been the biggest support to my life and study.

This thesis is dedicated to my family.

Table of contents

List of tables

List of figures

Nomenclature

Abbreviations

ANR	Anchored Neighborhood Regression
ASDS	Adaptive Sparse Domain Selection
CSR	Centralized Sparse Representation
GHP	Gradient Histogram Preservation
GR	Global Regression
HR	High-Resolution
JDSR	Joint Dictionary-Based Super-Resolution
K-SVD	K-Singular Value Decomposition
LARS	Least Angle Regression
LLE	Locally Linear Embedding
LR	Low-Resolution
LS	Unconstraint Least Squares
MAP	Maximum A Posterior
MOD	Method of Optimal Dictionary
MP	Matching Pursuit

MRF	Markov Random Field
MSE	Mean Squared Error
NCSR	Nonlocally Centralized Sparse Representation
NE	Neighbor Embedding
NLM	Non-Locally Mean
NNLS	Non-negative Least Squares
NEDI	New Edge-Directed Interpolation
OMP	Orthogonal Matching Pursuit
PCA	Principle Component Analysis
PSNR	Peak Signal-to-Noise Ratio
QCQP	Quadratically Constrained Quadratic Programming Problem
QP	Quadratic Optimization Problem
RGB	A Cartesian coordinate notation by defining R, G, and B as three primary colour lights – Red, Green, and Blue respectively
RMSE	Root-Mean Squared Error
SR	Super-Resolution
SSIM	Structural Similarity
s.t.	Subject To
YCbCr	A Cartesian coordinate notation by defining Y as brightness, Cb and Cr are two chrominance components

Chapter 1

Introduction

Images can often be divided into high-resolution (HR) ones and low-resolution (LR) ones. High-resolution images, within which the pixel densities are high, are often required and desired in many imaging applications. Compared with low-resolution images, HR ones provide clearer and more adequate details for object texture and boundary. These advantages consequently leads to a better viewing experience for human, and more abundant data for pattern recognition algorithms. For example, we can easily distinguish the interested objects from others in an image with clearer object boundary. As a result, we can achieve better performance in pattern recognition, or image segmentation by using HR images. Besides, an HR medical image will often help the doctor give a correct diagnosis. In practice, although current sensors can satisfy most imaging applications, the resolution level is still required to be improved to fit the future demands.

1.1 Resolution enhancement

Considering hardware improvement, resolution enhancement can be achieved by the following ways [1]. The most direct way, is to reduce the size of each pixel by sensor manufacturing techniques. In other words, the spatial image resolution can be enlarged by increasing the number of pixels per unit area. However, the decrease of pixels' size also leads to a decrease of efficacious lights, as a result of which, shot noise will be produced to degrade the image quality. Hence, there is a limitation of reducing the pixel size, which is an important concern in many commercial applications. Another method is to enlarge the size of chip. In this approach, capacitance will be enlarged [2], which will result in a limitation of charge transfer rate. And high cost for high precision sensors also makes it less efficient. And these disadvantages are still unsettled.

Except for the improvement on hardware, it is also promising and cost saving to recover HR image from its LR versions based on digital image processing techniques. Generally, we call this kind of approaches as image super-resolution (SR) [3–12]. Because of its low-cost but outstanding performance, it has raised many interests and become one of the most active research directions in image reconstruction. Based on the number of the given LR images, image SR can be categorized into single-image SR and multiple-frame image SR. Here, we only discuss the former one in this thesis. As the given LR image is normally regarded as a distorted version of original HR image with smaller size, it is necessary to scale it up in the process of super-resolution, which is known as interpolation. In the next section, we will introduce the regular interpolation methods in detail.

1.2 Interpolation

In general, three interpolation algorithms are most commonly used in the scenario of super-resolution: Nearest Neighbour, Bilinear, and Bicubic interpolations.

Nearest Neighbour interpolation is the most basic and easiest way to interpolate the unknown pixels into the given LR image. The core idea of this method is to insert new pixels whose values are equal to those of their nearest neighbor. In Figure 1.1,

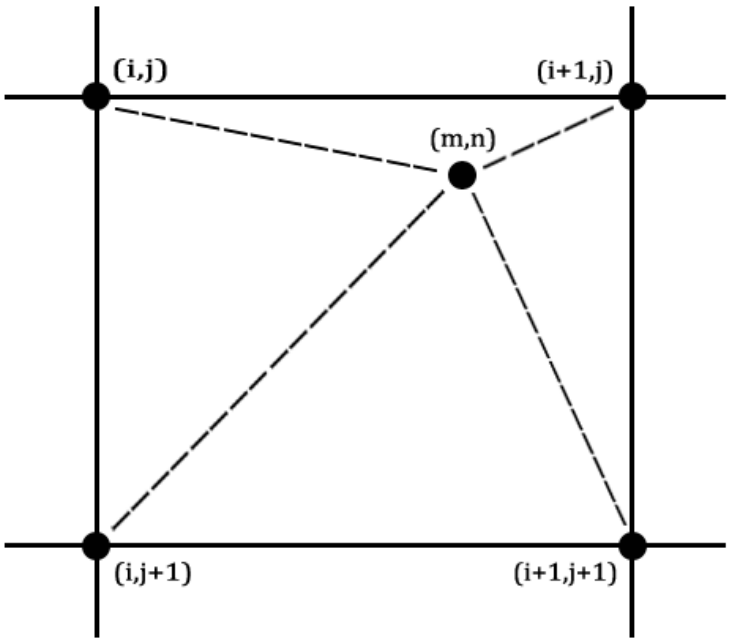


Figure 1.1: Nearest neighbor interpolation

the values of points at the position (i, j), $(i + 1, j)$, $(i, j + 1)$ and $(i + 1, j + 1)$ are known, and the point at the location (m, n) is to be interpolated. Considering the

definition of Nearest Neighbor interpolation, we firstly calculate the distance from (m, n) to the other four points, then select the one with minimum distance, which is denoted as:

$$\begin{aligned} D_{min} = \min\{&D[(m,n),(i,j)], D[(m,n),(i+1,j)], \\ &D[(m,n),(i,j+1)], D[(m,n),(i+1,j+1)]\} \end{aligned} \tag{1.1}$$

where D is the distance. Clearly, in Figure 1.1, $D_{min} = D[(m, n), (i + 1, j)]$, and the value of the interpolated point will be equal to that of point $(i + 1, j)$.

The second method is called Bilinear interpolation. In Figure 1.2, $P(x, y)$ is the point to be interpolated based on $P_1(x_1, y_2)$, $P_2(x_2, y_2)$, $P_3(x_1, y_1)$ and $P_4(x_2, y_1)$. With an unknown function f, the values of f at these four corner points are assumed to be known, denoting as $f(P_1)$, $f(P_2)$, $f(P_3)$ and $f(P_4)$. In order to obtain the value

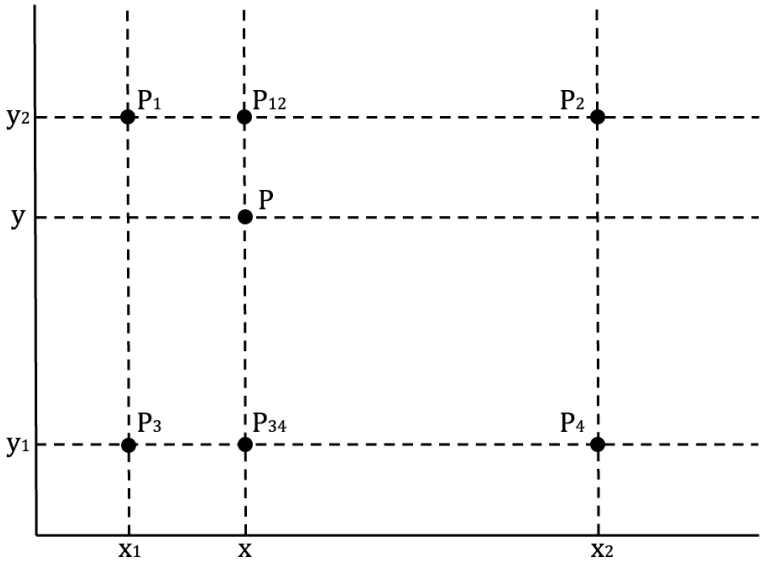


Figure 1.2: Bilinear interpolation

of f at the point P, we firstly do the interpolation in x direction. The values of f at points P_{12} and P_{34} can be calculated as:

$$f(P_{12}) = \frac{x_2 - x}{x_2 - x_1} f(P_1) + \frac{x - x_1}{x_2 - x_1} f(P_2) \tag{1.2}$$

$$f(P_{34}) = \frac{x_2 - x}{x_2 - x_1} f(P_3) + \frac{x - x_1}{x_2 - x_1} f(P_4) \tag{1.3}$$

Then, we can obtain $f(P)$ by interpolating between $P_{12}(x, y_2)$ and $P_{34}(x, y_1)$ again in y direction.

$$\begin{aligned} f(P) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} (& f(P_1)(x_2 - x)(y - y_1) + f(P_2)(x - x_1)(y - y_1) \\ & + f(P_3)(x_2 - x)(y_2 - y) + f(P_4)(x - x_1)(y_2 - y)) \end{aligned} \tag{1.4}$$

This solution can also be re-written as:

$$f(P) = a_0 + a_1 x + a_2 y + a_3 xy \tag{1.5}$$

Here, a_0, a_1, a_2 and a_3 are the coefficients, which can be computed as:

$$a_0 = -f(P_1)x_2 y_1 + f(P_2)x_1 y_1 + f(P_3)x_2 y_2 - f(P_4)x_2 y_2 \tag{1.6}$$

$$a_1 = f(P_1)y_1 - f(P_2)y_1 - f(P_3)y_2 + f(P_4)y_2 \tag{1.7}$$

$$a_2 = f(P_1)x_2 - f(P_2)x_1 - f(P_3)x_2 + f(P_4)x_1 \tag{1.8}$$

$$a_3 = -f(P_1) + f(P_2) + f(P_3) - f(P_4) \tag{1.9}$$

In addition, another method called Bicubic interpolation is commonly used in image processing with a better performance [13]. It is an extension of cubic interpolation [14], by inserting data on a two dimensional grids. Compared with Nearest Neighbor and Bilinear interpolation, Bicubic interpolation produces smoother edges at the cost of higher computational complexity. Similar with Bilinear interpolation, the value of f at the point $P(x, y)$ can be written as:

$$f(P) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{1.10}$$

Here, we assume the function values f and the derivatives f_x, f_y and f_{xy} at the points P_1, P_2, P_3 and P_4 are known. For convenience, we simplify the coordinates of these four corner points as $P_1(0, 1)$, $P_2(1, 1)$, $P_3(0, 0)$ and $P_4(1, 0)$. Thus, we have:

$$f_x(P) = \sum_{i=1}^{3} \sum_{j=0}^{3} a_{ij} i x^{i-1} y^j \tag{1.11}$$

$$f_y(P) = \sum_{i=0}^{3} \sum_{j=1}^{3} a_{ij} j x^i y^{j-1} \tag{1.12}$$

$$f_{xy}(P) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij} i j x^{i-1} y^{j-1} \tag{1.13}$$

If we rewrite the equations above in a matrix form as follows:

$$\mathbf{f} = \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} \tag{1.14}$$

$$\mathbf{a} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{1.15}$$

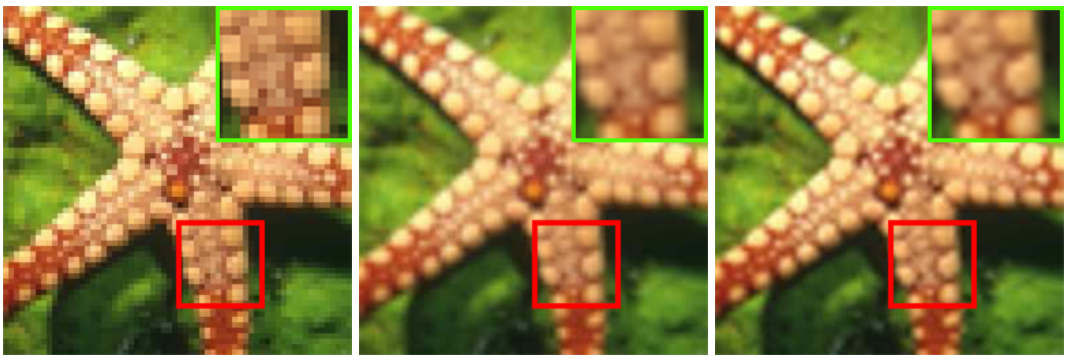
we can rewrite Equ. (1.11) - Equ. (1.13) as:

$$\mathbf{f} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{a} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 \end{bmatrix} \tag{1.16}$$

In other words, we can obtain the coefficients $\mathbf{a}$ by:

$$\mathbf{a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \mathbf{f} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{1.17}$$

Overall, from above, nearest neighbor interpolation is not suitable for the natural images with complex textures since the value of inserted new point is equal to its nearest neighbor. As for Bilinear, although it computes the value of inserted point with its four neighbors, it may still lead to blurring effects. Compared with them, Bicubic interpolation inserts new pixels by considering not only their neighbors, but also the derivatives of the neighbors, leading to better results. Figure 1.3 gives the visual comparison of images interpolated by these three interpolation methods with the same scaling factor. From left to right, the images stand for results of Nearest Neighbor, Bilinear and Bicubic, respectively. Visually, we are able to tell that Bicubic interpolation produces the image with the best quality. Therefore, it is used to interpolate the LR images in most super-resolution algorithms.



(a) Nearest Neighbor (b) Bilinear (c) Bicubic

Figure 1.3: Visual comparison of the results from 3 interpolation methods with scaling factor = 4

1.3 Super-resolution

Both Image interpolation and image super-resolution is a class of algorithms which are able to improve the quality of LR images. However, from the last section, we have

seen that image interpolation only focuses on upsampling the low-resolution images, and results in an insufficient estimation. Therefore, it is unreasonable to regard the LR observed images to be aliased only by downsampling. As a matter of fact, the undesirable effects caused by other degradations such as blurring and addictive noise usually exists as well. Hence, super-resolution may involve in de-aliasing all these undesirable effects, leading to a more complicated process. Except for the upsampling, *i.e.* interpolation, SR algorithms may contain deblurring and denoising processes as well. By doing this, they are able to recover better high-frequency components and preserve the characteristics of generic images.

Since image super-resolution was first proposed by Tsai *et al.* in 1980s [15], a variety of SR algorithms have been raised to solve this highly ill-posed problem. Basically, these algorithms can be classified into three categories: interpolation-based, reconstruction-based and learning-based super-resolution [16].

Interpolation-based super-resolution [17, 18] is regarded as a set of the fundamental and naive methods. Either fixed-function kernels [13] or adaptive-structure kernels [19–21] are used to estimate the unknown pixels, leading to a high-resolution image. New edge-directed interpolation (NEDI), proposed by Li *et al.* [19], is one of the classical interpolation-based super-resolution methods. With the assumption that generic images are able to be modeled as a locally stationary Gaussian process, it uses covariance-based adaptive interpolation for the edges and utilizes Bilinear interpolation for the rest as its simplification. In Figure 1.4, $Y_i, i = 1, 2, 3, \cdots, 9$ stands for the pixels of an LR image. $X_j, j = 1, 2, 3, 4$ and X' represent the HR pixels to be estimated. Basically, NEDI estimates pixel by its four neighbors. As for Y_5, we can

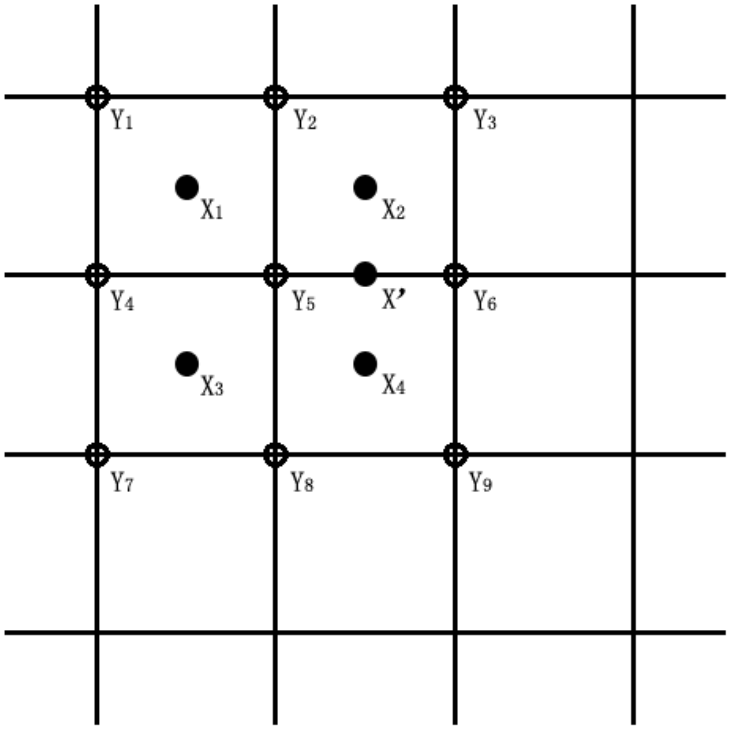


Figure 1.4: New edge-directed interpolation

calculate it by:

$$Y_5 = \sum_{i,k} A_k Y_i + \varepsilon, \quad i = 1, 3, 7, 9, k = 1, 2, 3, 4 \tag{1.18}$$

Here, ε is the estimation error, and A_k is the model parameters. In order to minimize the ε, we can solve it by selecting a good A_k:

$$\hat{A}_k = \min[Y_5 - \sum_{i,k} A_k Y_i]^2, \quad i = 1, 3, 7, 9, k = 1, 2, 3, 4 \tag{1.19}$$

In this approach, we can search for a good set of model parameters $\hat{A}_k$ that best

match the entire LR image. In the procedure of super-resolution, we can estimate the unknown pixels by using this set of model parameters. In Figure 1.4, $X_j, j = 1, 2, 3, 4$ can be estimated by their neighbors. For example, as for X_4, it can be estimated by:

$$X_4 = \sum_{i,k} \hat{A}_k Y_i, \quad i = 5, 6, 8, 9, k = 1, 2, 3, 4 \tag{1.20}$$

When it comes to the missing point between two LR pixels X', it can be obtained by rotating the spatial positions of its neighbors, *i.e.*, $\{X_2, X_4, Y_5, Y_6\}$, by 45 degrees with a scaling factor of $\sqrt{2}/2$. Although interpolation-based super-resolution algorithms are simple and low-cost, these methods normally generate unsatisfactory results with degradations.

Reconstruction-based SR [22–25] assumes the LR image can be obtained from HR image by a series of degradations, such as blurring, downsampling and noising (normally regard as additive zero-means Gaussian white noise). With this assumption, reconstruction-based SR uses different kinds of prior knowledge, gradient prior [26–28] and similarity redundancy [29, 30] for instance, to enhance the LR image. Compared with interpolation-based SR, this family of methods have the ability to sharp the edges and suppress the aliasing artifacts efficiently, but the HR images can not be restored with satisfying novel details, especially when the scaling factor is large.

The third category is the learning-based SR [31–41], which is also the class of methods exploited in this thesis. Basically, these methods capture the prior knowledge from training datasets to learn the underlying mapping relationships between LR images and HR images. Such mapping relationship is then used to estimated new HR images from LR input. According to the mapping formulations, learning-based

SR includes neighbor-based, regression-based and sparsity-based methods. Neighbor-based methods often require searching in a vast reference dataset for similar patterns. Fressman *et al.* proposed a learning strategy via a Markov random field (MRF) solved by belief propagation [42]. Chang *et al.* raised a new locally linear embedding (LLE)-based neighbor-embedding (NE) super-resolution [43]. As for regression-based model, it establishes the direct mapping relationships between HR images and their corresponding LR images [44–46]. However, the quality of final estimation relies heavily on the number of neighbors, and the qualities and number of prototypes. And when the number of neighbors is fixed, blurring effects became inevitable during reconstruction. Therefore, Yang *et al.* came up with a new sparsity-based super-resolution method [47]. A joint over-completed LR-HR dictionary pair was trained for sparse reconstruction. Later, some efforts [48–50] have been made for cost saving while preserving the SR estimation quality. Based on the same framework in [51], Zeyde *et al.* used the dimensionality reduction based on principle component analysis (PCA) and K-SVD dictionary training with orthogonal matching pursuit (OMP) to improve Yang's model in both computational time and restoration quality [52]. Timofte *et al.* introduced an anchored neighborhood regression (ANR) model and its extreme case called Global Regression (GR) model for super-resolution by leaning a neighbor-based dictionary [53]. However, an over-completed dictionary with high redundant information is potentially unstable [54]. Therefore, a PCA-based subspaces dictionary training technology was utilized in many algorithms. An adaptive sparse domain selection (ASDS) model proposed by Dong *et al.* decomposed images into subspaces by K-means clustering and PCA for dictionary training, and recovered HR image by using piecewise autoregressive (AG) models and a non-local self-similarity constraint as the regularization term [55]. And they further proposed an improved model named

non-local centralized sparse representation (NCSR) [56]. With the idea of reducing sparse coding noise, it leads to an impressive performance and has been proved to be one of the state-of-the-art SR algorithms. Except for the methods mentioned above, recent researches extended the conventional super-resolution approaches with some new ideas [57–61]. Cheng *et al* proposed a novel image super-resolution method via group sparse representation, regarding RGB channels as a group of three atoms instead of only considering the luminance channel [62]. Qi *et al.* improved the SR reconstruction quality by using 2D sparse model so that the intrinsic 2D structure and spatial correlation inherence will not be ignored [63]. Zhang *et al.* extended Yang's model by PCA-based subspaces dictionary training and replacing the back-projection constraint with the non-local mean (NLM) constraint [16]. From the experimental results, these methods are capable to improve the reconstruction ability of sparsity-based image super-resolution by producing better recovered HR images.

1.4 Thesis structure

In this thesis, we focus on the sparsity-based single-image super-resolution, and propose a new joint dictionary-based method, which is called Joint Dictionary-based Super-Resolution (JDSR) model. In our proposed model, two existing algorithms are integrated together and an extra gradient histogram preservation processing is also added into our SR scheme. Compared with some existing state-of-the-art super resolution algorithms, our proposed method is able to recover better high-resolution natural images with clearer textures, sharper edges and better novel details. The rest of the thesis is organized as follows. Chapter 2 introduces the basic idea of sparsity-based super-resolution. Chapter 5.1 illustrates three sparsity-based single-

image super-resolution algorithms in detail. Chapter 4 presents our proposed super-resolution model JDSR. Chapter 5 gives our experimental results and Chapter 6 concludes this thesis.

Chapter 2

Related work

2.1 Observation model

As for still images, observation model reflects the relationship between the desired HR image and the observed LR images [1]. Consider the desired HR image X written in lexicographical notation as $[X_1, X_2, X_3, ..., X_N]^T$ has the size of $D_1N_1 \times D_2N_2$, where $X_i, i = 1, 2, 3, \cdots, N$ stands for each pixel in X, and $N = D_1N_1 \times D_2N_2$. Here, X is assumed to be an ideal high-resolution image sampled at or above Shannon-Naquist rate from a continuous scene, and D_1, D_2 are regarded as the downsampling factor in the horizontal and vertical directions, respectively. Therefore, the observed LR image Y has the size of $N_1 \times N_2$, and it can be written in lexicographical notation as $[Y_1, Y_2, Y_3, ..., Y_M]^T$, where $Y_j, j = 1, 2, 3, \cdots, M$ stands for each pixel in Y, and $M = N_1 \times N_2$.

During the acquisition of the observed LR image, blurring is another important factor to be considered. It may either be caused by the optical system, such as

out of focus and diffraction limit, or the movement of an object's position occurred during the exposure time of camera. Besides, the point spread function (PSF) of the LR sensor modeled as linear space invariant or linear space variant can also lead to spatial averaging. Figure 2.1 gives an example of LR sensor PSF by using regular spatial averaging operator. a_i, where $i = 1, 2, 3, \cdots, 9$, represents the values of high-resolution grid, and b_1 is the value of corresponding low-resolution grid obtained by $\frac{\sum a_i}{9}, i = 1, 2, 3, \cdots, 9$.

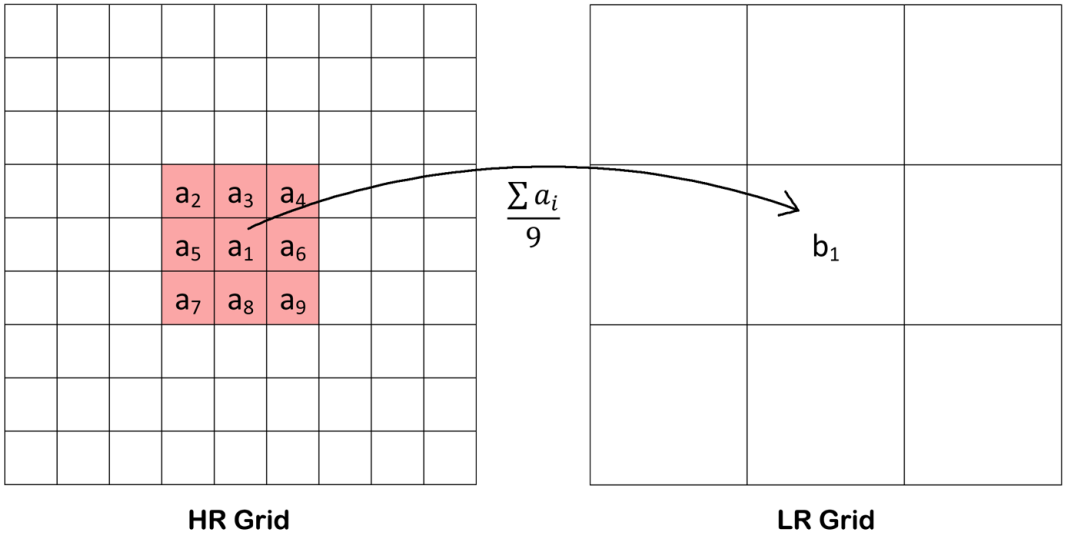
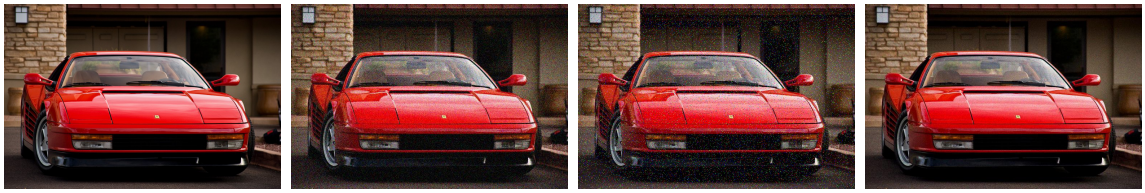


Figure 2.1: Spatial averaging operator affects high-resolution image

In addition, noise is another undesirable but inevitable distortion produced by digital camera. Considering the probability density function (PDF), noise can be categorised and treated differently, such as Gaussian noise, salt-and-pepper noise, shot noise and so on. Gaussian noise is one of the most commonly used noise in image processing. It is often caused by poor illuminance, high temperature and electronic circuit noise. For simplification, Gaussian noise in image processing is

always regarded as a zero-mean addictive Gaussian white noise, whose power spectral density is distributed uniformly. As for salt-and pepper noise, it is often caused by analog-to-digital converter errors and bit errors in transmission, and can be eliminated by median filtering. Shot noise, *e.g.*, Poisson noise, is normally caused by statistical quantum fluctuations. As it follows Poisson distribution, it is able to be estimate as a Gaussian distribution at low intensity levels. Figure 2.2 presents the results by adding these three types of noise into the same natural image.



(a) Original (b) Gaussian (c) Salt-and-Pepper (d) Poisson

Figure 2.2: Image with different noises

The observation model of single-image super-resolution can be established by combining the degradations mentioned above. Figure 2.3 gives the detailed structure of the model, which can be express as Equ. (2.1) mathematically. In Equ. (2.1), Y and X represent the LR image and HR image, respectively, and $\mathbf{B}$, $\mathbf{D}$ and $\mathbf{v}$ stand for blurring matrix, downsampling matrix and addictive noise, respectively.

$$Y = \mathbf{B}\mathbf{D}X + \mathbf{v} \tag{2.1}$$

For simplification, we can replace D and B by a degradation matrix $\mathbf{H}$, and the observation model can be re-written as Equ. (2.2).

$$Y = \mathbf{H}X + \mathbf{v} \tag{2.2}$$

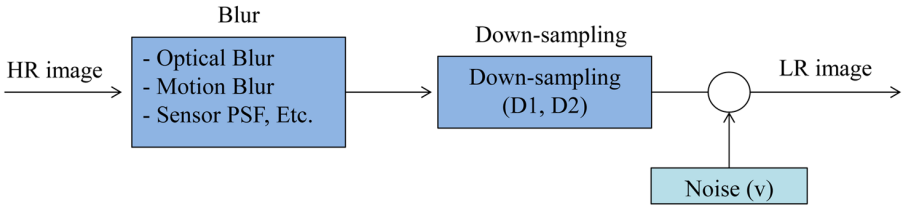


Figure 2.3: Observation model for single-image super-resolution

2.2 Sparse-Land model

Considering $X \in \mathbb{R}^N$ is a signal in n dimension space. In Sparse-Land model, it can be represented as $X = D\alpha$, where $D \in \mathbb{R}^{N \times M}$ is a set of redundant bases with m parameters ($N < M$), and $\alpha \in \mathbb{R}^M$ stands for the sparse vector or sparse code with most coefficients equal or close to zero [64]. In the other words, for any signal, it can be expressed as a linear combination of several bases in D, as shown in Figure 2.4. The sparsest representation of X can be found by solving Equ. (2.3), where $\|\cdot\|_0$

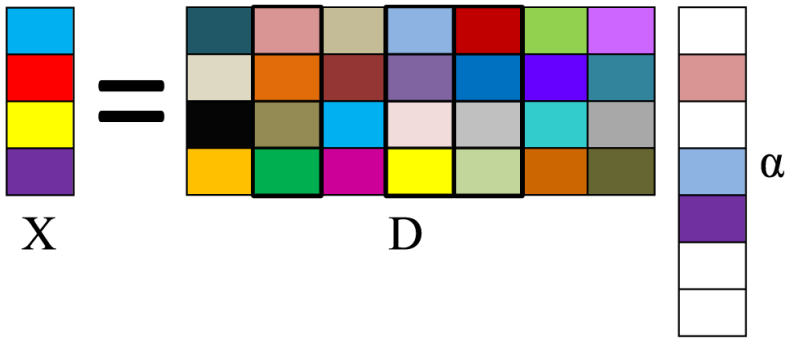


Figure 2.4: Sparse representation model

counts the number of nonzero entries of α, $\|\cdot\|_2$ stands for the standard Euclidean

norm or l^2 norm, and s.t. is short for "subject to".

$$\min \|\alpha\|_0 \quad s.t. \quad \|X - D\alpha\|_2^2 \leqslant \xi \tag{2.3}$$

The optimization problem above is NP-hard in general and can not be solved directly. [65] suggested that if the sparse code α is sufficiently sparse, Equ. (2.3) can be modified to a l^1-minimization problem, as shown in Equ. (2.4):

$$\min \|\alpha\|_1 \quad s.t. \quad \|X - D\alpha\|_2^2 \leqslant \xi \tag{2.4}$$

where $\|\cdot\|_1$ denotes as the standard l^1 norm. By using Lagrange multipliers, Equ. (2.4) can be re-written as:

$$\min_{\alpha} \|X - D\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{2.5}$$

Here, λ is used to balance the sparsity of sparse code and the fidelity of approximated signal.

With the similar concepts, as for an image signal $X \in \mathbb{R}^N$ in n dimension space, it can also be expressed as the product of D and α. Here, $D \in \mathbb{R}^{N \times M}$ is called dictionary, and it is normally over-completed as for represent generic images with different textures, *e.g.*, $M \gg N$. Each column of dictionary is called an atom. α is sparse vector or sparse code. Therefore, with the acknowledge of dictionary D and its sparse code α, we can easily reconstruct the image by using Equ. (2.6)

$$\hat{X} = D\alpha \tag{2.6}$$

2.3 Sparsity-based image super-resolution model

As we know, in Sparse-Land model, an image signal $X \in \mathbb{R}^N$ can be denoted sparsely by $\alpha \in \mathbb{R}^M$ over an over-completed dictionary $D \in \mathbb{R}^{N \times M}$, *e.g.*, $X = D\alpha$. Combining the observation model established before, the observed LR image Y can be presented as:

$$Y = \mathbf{H}D\alpha + \mathbf{v} \tag{2.7}$$

As downsampling, blurring and noising in super-resolution model have been predefined to be linear transforms normally, Equ. (2.7) can be re-written as:

$$Y = \mathbf{L}D\alpha \tag{2.8}$$

where $\mathbf{L}$ is a projection matrix, and the observation model becomes a sparsity-based compressed sensing model. Rauhut *et al.* have shown that in a sparsity-based compressed sensing model, a signal can be recovered almost perfectly from its random samples with high probability as long as D satisfies an appropriate near-isometry condition [66]. It makes sparsity-based image super-resolution feasible and meaningful.

In the history of single-image super-resolution, Yang *et al.* were the first to come up with the idea of super-resolving image by sparse representation [51]. A conventional sparsity-based image super-resolution model can be mainly divided into two stages: dictionary training and super-resolution. Dictionary is very important in image reconstruction, and often trained locally. Assume $x_i = \mathbf{R}_i X \in \mathbb{R}^n$ denotes an HR image patch of size $\sqrt{n} \times \sqrt{n}$ extracted at the location i, where $\mathbf{R}_i$ is an extracting

matrix. In Sparse-Land model, it can be expressed as $x_i = D_h\alpha_i$. Considering the observation model established before, for an LR image patch $y_i = \mathbf{R}_i Y$ extracted from the corresponding LR image Y at the location i, it can be re-written as:

$$y_i = \mathbf{H}_i D_h \alpha_i + \mathbf{v}_i \tag{2.9}$$

where $\mathbf{H}_i$ is a local degradation operator. Scaling up the LR image by a simple interpolation operator $\mathbf{P}$, we will obtain a mid-resolution image Z with the same size of X. Mathematically, the local patch z_i of the mid-resolution image Z at the location i can be formulated as:

$$z_i = \mathbf{P}_i \mathbf{H}_i D_h \alpha_i + \mathbf{P}_i \mathbf{v}_i \tag{2.10}$$

where $\mathbf{P}_i$ is the local interpolation operator. For more simplification, we replace $\mathbf{P}_i\mathbf{H}_i$ and $\mathbf{P}_i\mathbf{v}_i$ by $\mathbf{Q}$ and $\mathbf{v}'$. Thus, Equ. (2.11) can be re-written as:

$$z_i = \mathbf{Q} D_h \alpha_i + \mathbf{v}' \tag{2.11}$$

implying that

$$\|z_i - \mathbf{Q} D_h \alpha_i\|_2 \leqslant \epsilon \tag{2.12}$$

Here, ϵ is related to the noise $\mathbf{v}'$.

From above, it is necessary to enforce the sparse code of low-resolution image over the dictionary $D_l = \mathbf{Q}D_h$ to be the same as that of its corresponding high-resolution image. As a result, in the process of super-resolution, the sparse code of LR image in

terms of D_l can be directly used with D_h to recover the corresponding HR image, as shown in Equ. (2.13)

$$\hat{X}_0 \approx D\alpha \tag{2.13}$$

In the sparsity-based super-resolution model, Equ. (2.13) is called sparse prior.

Bringing the observation model back into consider, in the sparsity-based super-resolution model, it can be seen as reconstruction constraint, because the HR image produced by the sparse representation may not satisfy this relationship. Therefore, with the sparse prior, we normally regularize it by utilizing this reconstruction constraint globally to remove the artifacts produced during the sparse prior, so that the final image will be more consistent and natural. Equ. (2.14) is formulated to solve this regularization problem:

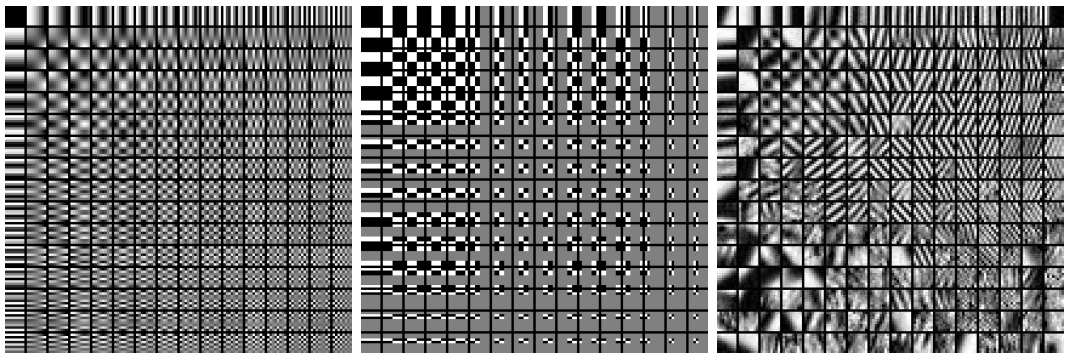
$$\hat{X} = arg\min_{X}\{\|Y - HX\|_2^2 + \lambda R(X)\} \tag{2.14}$$

Here, Y is the given LR image, $\|Y - HX\|_2^2$ is called fidelity term, and $R(X)$ is called regularization term that can be determined by various image priors. λ is a positive constant.

2.4 Dictionary training

In the previous section, we have discussed that traditionally two over-completed coupled dictionaries D_l and D_h will be used to solving the sparsity-based super-resolution problem. Compared with some analytical dictionaries, shown in Figure 2.5 (a) and

(b), the learning-based dictionary has a higher adaptive ability for different types of data, shown in Figure 2.5 (c). To train a dictionary as Figure 2.5 (c), we utilize the



(a) DCT dictionary (b) Haar dictionary (c) Learning-based dictionary

Figure 2.5: Comparison of analytical dictionary and learning-based dictionary

following formulation:

$$D = \min_{D,\alpha} \|X - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \quad s.t. \quad \|D_i\|_2^2 \leqslant 1, i = 1, 2, 3, \cdots, K \tag{2.15}$$

where, D is a dictionary with K atoms, $\|\cdot\|_1$ is the l^1 norm used to enforce the sparsity of α, and $\|\cdot\|_2$ is the l^2 norm constraints on the column of D for removing the scaling ambiguity. This is known as a L1-regularized least squares problem, or Lasso problem [67], which is a linear regression regularizing its coefficients by l^1-norm. Previous studies [68, 69] on this formulation have indicated that it is not convex in both D and α, while is convex when one of them is fixed. Therefore, an alternative strategy can be denoted to help solve this optimization problem. The detailed steps are shown as follows:

I. Initialize D, and normalize each atom of D;

II. Fix D, and update α by the following formulation:

$$\alpha = \min_{\alpha} \|X - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{2.16}$$

III. Fix α, and updated D by the following formulation:

$$D = \min_{D} \|X - D\alpha\|_2^2 \quad s.t. \quad \|D_i\|_2^2 \leqslant 1, i = 1, 2, 3, \cdots, K \tag{2.17}$$

IV. Iterate Step **II** and Step **III** until converge.

Apparently, solving the ill-posed problem formulated as Equ. (2.15) is time consuming, and varieties of algorithms have been proposed to solve this problem. In order to search for the best sparse codes over a fixed dictionary, Lee *et al.* raised a feature-sign search algorithm by "guessing" the signs of coefficients first [70]. We can re-wrote Equ. (2.16) by replacing D and α with $\sum_j D_j$ and $\sum_j \alpha_j$:

$$\alpha = \min_{\alpha} \left\| X - \sum_j D_j \alpha_j \right\|_2^2 + \lambda \sum_j \|\alpha_j\|_1 \tag{2.18}$$

Thus, after "guessing" the signs of the coefficients α, we are able to replace $\|\alpha_j\|_1$ with α_j (if $\alpha_j > 0$), $-\alpha_j$ (if $\alpha_j < 0$), or 0 (if $\alpha_j = 0$). Considering the non-zero coefficients, Equ. (2.18) will be simplified as a standard unconstraint quadratic optimization problem (QP) and can be solved efficiently. If the initial "guess" is incorrect, this algorithm is capable to systematically refine it by entering into the next iteration. Compared with some other algorithms, such as genetic QP solvers (*e.g.* CVX) [71], interior point method [72], a modification of least angle regression (LARS) [73] and grafting [74], the feature-sign search algorithm has been proved to be able to achieve

better performance with shorter time. As for Step **III**, Lee *et al.* regard Equ. (2.17) as a quadratically constrained quadratic programming (QCQP) problem, and can be solved by using L2-Lagrange duel. Compared to gradient descent with iterative projection [71], Lee's algorithm has fewer optimization variables, and is able to converge faster.

Besides, there are still many other traditional algorithms such as Maximum Likelihood Method, Method of Optimal Dictionary (MOD) [75], Maximum A Posterior Probability Approach (MAP), K-Singular Value Decomposition (K-SVD) [76] and so on. Compared with the others, K-SVD has been proved to be simpler and efficient, and it is flexible with any pursuit method. In the next chapter, we will introduce K-SVD algorithm in detail, using Matching Pursuit (MP) [77] and Orthogonal Matching Pursuit (OMP) algorithm [78] to search for the sparse code.

2.5 Quality evaluation

In image processing, quality evaluation is a very important issue which will reflect the ability of algorithms. Many approaches have been designed to measure the qualities of the processed images compared with ground truth. In single-image super-resolution, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) are the most commonly used methods due to their efficiency and simplicity. In our experiments, we use PSNR and SSIM as well, for evaluating the recovered qualities of HR images by using our proposed algorithm.

PSNR, which is short for Peak Signal-to-Noise Ratio, is the most commonly used method in both signal processing and image processing. In image processing, PSNR

can be defined simply via mean squared error (MSE), which can be formulated as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_g(i,j) - I_d(i,j)]^2 \tag{2.19}$$

Here, I_d stands for the degraded monochrome image with size $m \times n$, and I_g is its corresponding ground truth. Extended to the signal processing, I_g and I_d represent two signals, and MSE results are able to reflect the similarity of these two signals. Therefore, it has been used in our experiments to measure the similarity of estimated reference histogram and real histogram.

Based on the MSE, PSNR can be defined as:

$$PSNR = 10 \cdot log_{10} \left[\frac{(MAX_I - MIN_I)^2}{MSE} \right] \tag{2.20}$$

Here, MAX_I and MIN_I stand for the largest and smallest possible pixel values of image I. In general, MAX_I is equal to be 255 if each pixel of image is represented by 8 bits per sample, and MIN_I is equal to be 0. However, in the scenario of image super-resolution, the images are normally not monochrome but colorful. In this case, we are able to calculate the PSNR in R, G, B channel individually, and computer the average afterwards. From the definition, we can easy understand that I_d has a better quality if the value of PSNR is higher.

Except for PSNR, SSIM is another method used in our experiments, which was proposed by Wang *et al.* in 2004 [79]. As its name, it mainly aims at measuring the structural similarity of two images, and it has been proved to be more consistent with human visual perception than PSNR and MSE. As the description in [79], the SSIM of two image signal x and y can be computed by the combination of their luminance

comparison, contrast comparison and structure comparison. It can be formulated as:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{2.21}$$

In Equ. (2.21), $l(x,y)$, $c(x,y)$ and $s(x,y)$ stand for the results of luminance comparison, contrast comparison and structure comparison of x and y, and α, β and γ are the positive parameters. In order to make it simple, α, β and γ are set to be 1, and SSIM can be easily expressed as;

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.22}$$

Here, μ_x and μ_y denote as the mean intensity of x and y, formulated as:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.23}$$

$$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{2.24}$$

and σ_x and σ_y represent the standard deviation of x and y, formulated as:

$$\sigma_x = \left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right]^{\frac{1}{2}} \tag{2.25}$$

$$\sigma_y = \left[\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2\right]^{\frac{1}{2}} \tag{2.26}$$

and C_1 and C_2 are two constants which can be calculated as:

$$C_1 = (K_1 L)^2 \tag{2.27}$$

$$C_2 = (K_2 L)^2 \tag{2.28}$$

Here, L is the dynamic range of the pixel values, and K_1 and K_2 are set to be 0.01 and 0.03 by default. As for regular images with each pixel represented by eight bits per sample, L can be calculated as $2^8 - 1 = 255$. Usually, for evaluating the quality of images, SSIM is only applied on luminance channel, and the larger the result is, the better the recovered image will be.

2.6 Summary

In this chapter, we have established the mapping relationship between HR image and its corresponding LR image by observation model. Besides, the basic concepts of Sparse-Land model have been illustrated briefly and applied in the area of image super-resolution based on the observation model. As the most important part of learning-based super-resolution, dictionary training has also been formulated as an optimization function, and solved by an iterative strategy. At last, we provided a brief introduction on three quality evaluation methods that have been used in our experiments.

Chapter 3

Literature review

In Chapter 2, we have established the observation model and given a detailed introduction on the basic concepts of sparsity-based image super-resolution model. Based on these models, many implementations have been designed and achieved to recover the unknown HR image from its LR version. In this chapter, we will focus on the sparsity-based super-resolution models that used for comparison in our experiments, and introduce Yang's, Zeyde's and NCSR models in detail since they have been used directly in our proposed algorithm. Besides, we will also introduce three neighbor embedding based super-resolution algorithms briefly because they are used for comparison in the experiments as well.

3.1 Yang's super-resolution algorithm

Although single image super-resolution has been studied for decades, using Sparse-Land model to super-resolve the low-resolution images is a new idea proposed by Yang

et al. several years ago. By training two coupled dictionaries from a training set, the connection between HR images and their corresponding LR images are established. Then, the input LR image can be recovered by sparse prior locally and reconstruction constraint globally. And they call the model as SCSR, which can also be divided into two stages: dictionary training and super resolution.

As we mentioned before, in a conventional sparsity-based image super-resolution model, dictionary training can be done offline by a patch-based method. And in this phase, two dictionaries D_h and D_l are trained by the images in the training set we have prepared before, over the same sparse representation. Normally, a training set is filled with variety of high-resolution images. Therefore, it is necessary to produce HR-LR image patch pairs at the beginning for the future training.

An HR-LR image patch pair, is formed by an HR image patch and its corresponding LR image patch. Generally, the sizes of both patches in one pair are allowed to be different or the same. In SCSR model, Yang found it works better when they have the same size. Hence, considering the observation model, we manually produce the LR images by employing the downsampling, blurring and additive noise operators on the given HR images first, and use Bicubic interpolation afterwards so that each image pair is able to have the same size. Then, each image pair will be extracted into small patch pairs with the same size processed by raster-scan order from left to right and top to bottom, and reconstruct them as vector form. At last, the HR images in training set will be pre-processed to be two coupled training matrixes in vector form, and we can call them as HR training matrix and LR training matrix. For simplification, we use $X = \{x_1, x_2, x_3, \cdots, x_N\}$ and $Z = \{z_1, z_2, z_3, \cdots, z_N\}$ to be the HR image and its corresponding interpolated LR image in vector form, respectively.

Among them, x_i and $z_i, i = 1, 2, 3, \cdots, N$ refer to the HR and LR image patches at the location i, respectively.

In last chapter, we have shown the objective function for single dictionary training as Equ. (2.15). In SCSR, two coupled dictionaries D_h and D_l are required to be trained for the future super-resolution. Therefore, the objective functions for HR dictionary and LR dictionary are formulated independently as:

$$\min \|\alpha\|_1 \quad s.t. \quad \|X - D_h\alpha\|_2^2 \leqslant \xi_1 \tag{3.1}$$

$$\min \|\alpha\|_1 \quad s.t. \quad \|Z - D_l\alpha\|_2^2 \leqslant \xi_2 \tag{3.2}$$

By using Lagrange multipliers, they can be re-written as:

$$D_h = \min_{\{D_h,\alpha\}} \|X - D_h\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3.3}$$

$$D_l = \min_{\{D_l,\alpha\}} \|Z - D_l\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3.4}$$

As we enforce the sparse code in Equ. (3.3) and Equ. (3.4) to be the same, Yang *et al.* proposed a joint dictionary-training method by combining Equ. (3.3) and Equ. (3.4) as:

$$\min_{\{D_h,D_l,\alpha\}} \frac{1}{N}\|X - D_h\alpha\|_2^2 + \frac{1}{M}\|Z - D_l\alpha\|_2^2 + \lambda(\frac{1}{N} + \frac{1}{M})\|\alpha\|_1 \tag{3.5}$$

where N and M represent the dimensions of HR and LR image patches in vector

form, respectively. $\frac{1}{N}$ and $\frac{1}{M}$ here are used to balance the Equ. (3.3) and Equ. (3.4). For simplification, Equ. (3.5) can be re-written as:

$$\min_{\{D_h, D_l, \alpha\}} \|X_c - D_c\alpha\|_2^2 + \lambda_c\|\alpha\|_1 \tag{3.6}$$

$$X_c = \begin{bmatrix} \frac{1}{\sqrt{N}}X \\ \frac{1}{\sqrt{M}}Z \end{bmatrix} \quad D_c = \begin{bmatrix} \frac{1}{\sqrt{N}}D_h \\ \frac{1}{\sqrt{M}}D_l \end{bmatrix} \quad \lambda_c = \lambda\left(\frac{1}{N} + \frac{1}{M}\right) \tag{3.7}$$

Clearly, it is inevitable to store huge amount of image information if we directly use intensities of image pixels. Hence, before training, we can pre-process the images to remove variability so that the training matrixes will be generally applicable. As human is more sensitive to high-frequency signals, we don't have to store all the possible low-frequency component values. So a linear feature extraction operator F can be employed to remove the low-frequency signals in images. Generally, F can be seen as a high-pass filter and the objective functions based on the feature maps can be expressed as:

$$D_h = \min_{\{D_h, \alpha\}} \|F_1X - F_1D_h\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3.8}$$

$$D_l = \min_{\{D_l, \alpha\}} \|F_2Z - F_2D_l\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3.9}$$

By denoting the joint dictionary-training method, the combining objective function

can be re-written as:

$$\min_{\{D_h, D_l, \alpha\}} \|F_c X_c - F_c D_c \alpha\|_2^2 + \lambda_c \|\alpha\|_1 \tag{3.10}$$

$$F_c = \begin{bmatrix} \frac{1}{\sqrt{N}} F_1 \\ \frac{1}{\sqrt{M}} F_2 \end{bmatrix} \quad X_c = \begin{bmatrix} \frac{1}{\sqrt{N}} X \\ \frac{1}{\sqrt{M}} Z \end{bmatrix} \quad D_c = \begin{bmatrix} \frac{1}{\sqrt{N}} D_h \\ \frac{1}{\sqrt{M}} D_l \end{bmatrix} \quad \lambda_c = \lambda \left(\frac{1}{N} + \frac{1}{M} \right) \tag{3.11}$$

From above, it is easy to find that we normally use different feature extraction operators for HR images and LR images. The selection of F has been discussed a lot in the past. Freeman *et al.* utilized a Gaussian low-pass filter to extract the high-frequency parts from both HR images and LR images [42]. Sun *et al.* used a bilateral filter for low-resolution images feature extraction, and subtract low-resolution image directly from its corresponding high-resolution images to obtain high-frequency parts of HR images [80]. In SCSR, Yang *et al.* extract HR feature map locally by subtracting the mean values from the original HR patches, and use first- and second-order gradient operators globally, proposed by Chang *et al.* [43], to remove the low-frequency parts in LR images. Due to its simplification and efficiency, first- and second-order gradient operators have been used widely in image feature extraction. Given an image patch with size 5×5 as shown in Figure 3.1. $W_i, i = 1, 2, 3, \cdots, 25$ represent for the values in each pixel. Considering the central pixel W_{13}, the first- and second-order can be expressed as Equ. (3.12) and Equ. (3.13):

$$\nabla W_{13} = \begin{bmatrix} (W_{14} - W_{13}) + (W_{13} - W_{12}) \\ (W_{18} - W_{13}) + (W_{13} - W_8) \end{bmatrix} = \begin{bmatrix} W_{14} - W_{12} \\ W_{18} - W_8 \end{bmatrix} \tag{3.12}$$

W_1	W_2	W_3	W_4	W_5
W_6	W_7	W_8	W_9	W_{10}
W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
W_{16}	W_{17}	W_{18}	W_{19}	W_{20}
W_{21}	W_{22}	W_{23}	W_{24}	W_{25}

Figure 3.1: An image patch with size 5×5

$$\nabla^2 W_{13} = \begin{bmatrix} (W_{15} - W_{13}) + (W_{13} - W_{11}) \\ (W_{23} - W_{13}) + (W_{13} - W_3) \end{bmatrix} = \begin{bmatrix} W_{15} - 2W_{13} + W_{11} \\ W_{18} - 2W_{13} + W_8 \end{bmatrix} \tag{3.13}$$

For simplification, they are normally expressed as four 1-D filters:

$$f_1 = [-1, 0, 1] \tag{3.14}$$

$$f_2 = [-1, 0, 1]^T \tag{3.15}$$

$$f_3 = [-1, 0, -2, 0, 1] \tag{3.16}$$

$$f_4 = [-1, 0, -2, 0, 1]^T \tag{3.17}$$

Hence, for each patch at location i, we can obtain four feature maps by adopting these filters. They are then concatenated in vector form used to be trained later. Besides its simplification, this method also considers the neighboring information of each patch, so that higher compatibility can be promoted between patches during the reconstruction.

With both HR and LR training matrixes having been prepared, the iterative strategy for dictionary training mentioned in last chapter can be employed to search for the best coupled dictionaries and their corresponding sparse codes. In SCSR model, Yang *et al.* used the feature-sign search algorithm and Lagrange dual, proposed by Lee *et al.* [70], to solve the l^1-minimization problem. We have briefly introduced these two algorithms in last chapter, and here we will give a more detailed description.

The objective function of feature-sign search algorithm can be written as Equ. (2.18), while $\sum_j D_j$ and $\sum_j \alpha_j$ represent the dictionary and sparse code, where $j = 1, 2, 3, \cdots, n$. The main theory of this algorithm is replacing $\|\alpha_j\|_1$ with α_j (if $\alpha_j > 0$), $-\alpha_j$ (if $\alpha_j < 0$), or 0 (if $\alpha_j = 0$) by "guessing" the signs of the coefficients α. Therefore, we can set $\theta_j^i \in \{-1, 0, 1\}$ to stand for the sign of i-th coefficient (by "guessing") in j-th column of sparse matrix, and use an active set A to store the corresponding indices i satisfying $\theta^i \neq 0$. Assume $\hat{D}$ is the submatrix of D with columns corresponding to A, and $\hat{\alpha}$ and $\hat{\theta}$ are subvectors of α and θ, respectively, corresponding to A as well. After a current guess of A, $\hat{\theta}$ and $\hat{\alpha}$, the analytical solution $\hat{\alpha}_{new}$ can be calculated by solving the unconstraint quadratic problem formulated as:

$$\min_{\hat{\alpha}} \left\| X - \hat{D}\hat{\alpha} \right\|_2^2 + \lambda \hat{\theta}^T \hat{\alpha} \tag{3.18}$$

Equally, solving Equ. (3.18) means minimizing the objective value formulated as:

$$\min_{\alpha} \quad 0.5 * \alpha^T D^T D\alpha - \alpha^T D\alpha + \lambda|\alpha| \tag{3.19}$$

and $\hat{\alpha}_{new}$ can be calculated as:

$$\hat{\alpha}_{new} = (\hat{D}^T\hat{D})^{-1}(\hat{D}^T X - \lambda\hat{\theta}/2) \tag{3.20}$$

In order to obtain the lowest objective value, we update $\hat{\alpha}$ with $\hat{\alpha}_{new}$ by using a discrete line search, and renew A and θ as well. All the steps above are operated iteratively until convergence. The detailed algorithm is shown as Algorithm 1.

By fixing the sparse codes, we can update the dictionary by using a Lagrange dual. Considering the objective function as:

$$\min \|X - D\alpha\|_2^2, \quad s.t. \quad \sum_{i=1}^{n} D_{i,j}^2 \leqslant 1, \forall j = 1, 2, \cdots, K \tag{3.21}$$

We can obtain the Lagrangian by:

$$\mathbb{L}(D, \zeta) = trace((X - D\alpha)^T(X - D\alpha)) + \sum_{j=1}^{K} \zeta_j \left(\sum_{i=1}^{n} D_{i,j}^2 - 1 \right) \tag{3.22}$$

where ζ_j is a non-negative dual variable. Therefore, by minimizing Equ. (3.22) over D, we can calculated the Lagrange dual by:

$$\mathbb{D}(\zeta) = trace(X^T X - X\alpha^T(\alpha\alpha^T + \Lambda)^{-1}(X\alpha^T)^T - \Lambda) \tag{3.23}$$

Here, $\Lambda = diag(\zeta)$. Thus, by using Newton's method or conjugate gradient, we can

Algorithm 1 Feature-sign search algorithm

Objective: Search for the best sparse code α of a given signal X over a fixed dictionary D by solving the objective function as:

$$\min_\alpha \|X - D\alpha\|_2^2 + \lambda |\alpha|_1$$

Initialization: $\alpha = 0$, $\theta = 0$, $A = \{\}$;
while $1 \leq j \leq n$:

1. For zero coefficients in α, computer $i = argmax|\frac{\mathrm{d}\|X-D\alpha\|_2^2}{\mathrm{d}\alpha_j^i}|$:
2. Update A and θ by the following constraints:
   if $\frac{\mathrm{d}\|X-D\alpha\|_2^2}{\mathrm{d}\alpha_j^i} > \lambda$, set $\theta_j^i = -1$, and $A = \{i\} \bigcup A$;
   if $\frac{\mathrm{d}\|X-D\alpha\|_2^2}{\mathrm{d}\alpha_j^i} < -\lambda$, set $\theta_j^i = 1$, and $A = \{i\} \bigcup A$;
3. Compute analytical solution $\hat{\alpha}_{new}$ by solving an unconstraint quadratic problem formulated as Equ. (3.18):

$$\hat{\alpha}_{new} = (\hat{D}^T\hat{D})^{-1}(\hat{D}^T X - \lambda\hat{\theta}/2)$$

4. Update $\hat{\alpha}$, A and θ by performing a discrete line search according to $\hat{\alpha}_{new}$;
5. Check the optimality conditions:
   if $\forall\alpha_j^i \neq 0$, $\frac{\mathrm{d}\|X-D\alpha\|_2^2}{\mathrm{d}\alpha_j^i} + \lambda sign(\alpha_j^i) \neq 0$, back to Step 4;
   if $\forall\alpha_j^i = 0$, $\frac{\mathrm{d}\|X-D\alpha\|_2^2}{\mathrm{d}\alpha_j^i} > \lambda$, back to Step 3;

end while
Return α;

update D with Equ. (3.24) by maximizing $\mathbb{D}(\zeta)$:

$$D^T = (\alpha\alpha^T + \Lambda)^{-1}(X\alpha^T)^T \tag{3.24}$$

As for the feature-sign search algorithm, compared with some other algorithms, such as genetic QP solvers(*e.g.* CVX), interior point method a modification of least angle regression (LARS) and grafting, it has been proved to be able to achieve better performance with shorter time. And when it comes to the Lagrange dual, it has fewer optimization variables, and is able to converge faster, compared with gradient descent with iterative projection. Besides, the combination of the feature-sign search algorithm and the Lagrange dual performs the better than any other combinations with the algorithms mentioned above. For more details, please refer to [70].

After the coupled dictionaries trained, we can go into the next stage: super-resolution. With a given low-resolution image, we can firstly use the bicubic interpolation, and extract the patches based on the feature maps generated by the same feature extraction operator mentioned above, and reconstruct them in vector form. Then, with the trained LR dictionary D_l, the feature-sign search algorithm is used again to search for the best sparse code. And the objective function is:

$$\alpha = \min_{\alpha} \|X - D_l\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{3.25}$$

In this stage, we still enforce the sparse representation of HR and LR image patches over dictionary D_h and D_l to be the same. Hence, with the acknowledge of α and D_h, we can estimate the HR image by $\hat{X} = D_h\alpha$, and add the mean value m of each LR patch in order to recover image from feature maps to original intensity maps, remarked

as $X_0 = \hat{X} + m$. However, we have discussed that the result reconstructed only by the sparse representation may not satisfy the reconstruction constraint. Therefore, instead of pursuing HR image recovered perfectly by sparse representation α, we allow it to satisfy the reconstruction constraint more, leading to a less sparse α. This idea can be implemented by a large optimization problem formulated as below:

$$X = argmin\{\|Y - \mathbf{H}X\|_2^2 + \gamma \sum_{i,j} \|\alpha_{ij}\|_0 + c \sum_{i,j} \|D_h \alpha_{ij} - \mathbf{L}_{ij} X\|_2^2 + \tau\rho(X)\} \quad (3.26)$$

In Equ. (3.26), α_{ij} stands for the sparse code for the (i, j)-th patch of X, $\mathbf{H}$ is the degradation matrix, $\mathbf{L}_{ij}$ represents a projection matrix choosing the (i, j)-th patch of X, and $\rho(X)$ is a penalty term encoded other prior knowledge of HR image. All the γ, c and τ are constants. Clearly, from above, we didn't take any prior knowledge except sparse representation, so there is no need to incorporate the forth term in Equ. (3.26). Besides, when the α is known and fixed, the third term in Equ. (3.26) penalizes the difference between X and the HR image X_0 that recovered from sparse representation. Thus, it can be re-written as:

$$c \sum_{i,j} \|D_h \alpha_{ij} - \mathbf{L}_{ij} X\|_2^2 \approx c \|X_0 - X\|_2^2 \quad (3.27)$$

Overall, the regularization function can be simplified as:

$$X = argmin\{\|Y - \mathbf{H}X\|_2^2 + c \|X_0 - X\|_2^2\} \quad (3.28)$$

This is called back-projection algorithm, utilized in SCSR model for the further regularization after knowing the sparse prior. Clearly, the fidelity term is $\|Y - \mathbf{H}X\|_2^2$,

and the regularization term is $\|X_0 - X\|_2^2$. c is a small positive constant. Using gradient descent method to update X iteratively according to Equ. (3.28), and we can get the final estimated HR image. In the k-th iteration, the update equation can be expressed as:

$$X_k = X_{k-1} + \chi[\mathbf{H}^T(Y - \mathbf{H}X_{k-1}) + c(X - X_0)] \tag{3.29}$$

where χ is the step size of the gradient descent.

For most situations, input LR images are assumed to be free of noise. Therefore, the SCSR model is required to be modified if we need to deal with noisy images. Since the connection between Maximum A Posterior (MAP) and sparse representation has been established [81]. we can formulate Equ.(2.5) into MAP problem, the sparse code α can be obtained by:

$$\tilde{\alpha} = \max P(\alpha) \times P(X|\alpha, D) \tag{3.30}$$

In Equ.(2.5), $P(\alpha)$ and $P(X|\alpha, D)$ can be calculated as:

$$P(\alpha) = \frac{1}{2b} exp\left(-\frac{\|\alpha\|_1}{b}\right) \tag{3.31}$$

$$P(X|\alpha, D) = \frac{1}{2\sigma^2} exp\left(-\frac{1}{2\sigma^2}\|D\alpha - X\|_2^2\right) \tag{3.32}$$

Here, b stands for the variance of the Laplacian prior on α, and σ is the standard deviation of noise of X. By using negative log likelihood, Equ.(3.30) can be written

as:

$$\min_{\alpha} \|X - D\alpha\|_2^2 + \frac{\sigma^2}{b} \|\alpha\|_1 \tag{3.33}$$

Compared with Equ.(2.5), it is easy to find $\lambda = \sigma^2 b$. Therefore, the higher noise level the given images have, the larger λ is required. Yang *et al.* suggest the empirical λ for noisy images should be one tenth of the standard deviation.

The algorithm of super-resolution stage can be summarized as Algorithm 2.

Algorithm 2 Detailed algorithm of super-resolution stage in SCSR model

Required: The given LR image Z, and two coupled dictionaries D_h and D_l

1. Pre-process the LR image with Bicubic interpolation operator, first- and second-order gradient method and patch extraction operator
2. Search for the best sparse code with D_l by the Feature-Sign Search algorithm, and the objective function can be expressed as:

$$\alpha = \min_{\alpha} \|X - D_l\alpha\|_2^2 + \lambda \|\alpha\|_1;$$

3. Compute the mean value m of each LR patch, and compute the estimated HR image $\hat{X}$ by:

$$\hat{X} = D_h\alpha$$

4. Computer the sparse prior by adding the mean values into the estimated HR image, as $X_0 = \hat{X} + m$;
5. Using back-projection to regularize the X_0 so that it will more satisfies the reconstruction constraint:

$$X = argmin\{\|Y - \mathbf{H}X\|_2^2 + c\|X_0 - X\|_2^2\}$$

Return X;

3.2 Zeyde's super-resolution algorithm

In the last section, we have introduced that Yang *et al.* proposed an SCSR model, giving a new direction to solve the single image super-resolution problem via sparse representation. Based on his model, Zeyde *et al.* raised an improvement with less computation time and better estimation results [52]. As a conventional single image super-resolution model, Zeyde's model also includes two main stages: dictionary training and super-resolution. In the following, we will discuss them in detail.

3.2.1 Construction of training patch pairs

In Zeyde's model, the single image super-resolution is regarded as a scaled-up algorithm, in which, dictionary training is viewed as a training phase, and super-resolution is seen as a testing phase. The former produces two coupled over-completed dictionaries D_h and D_l from the matching HR-LR patches, and the latter scaled a test LR image up into an HR image by using these two dictionaries. Therefore, we start our discussion from the training phase.

The overall structure of the training phase can be summarized into two steps: patch-pairs construction and dictionary training. The same as Yang's model, a training set with lots of HR images will be prepared for the patch-pairs construction, marked as $\{X_j\}$. These high-resolution examples are firstly downsampled and blurred, leading to the corresponding low-resolution images $\{Z_j\}$, and then scaled up back to the original size by some interpolation algorithms. The interpolated images can be expressed as $\{Y_j\}$. Next, we will perform the feature extraction and patch extraction on both HR and interpolated LR images, keeping the high-frequencies and decom-

posing the full images into small patches. Instead of applying the patch extraction before feature extraction, we firstly extract the feature information or high-frequencies from the full images in order to avoid boundary problems, and then decompose the matching feature maps into small patches.

The feature extraction methods on HR images and LR images are still different. For HR images, Zeyde's model mainly aims at removing low-frequencies by computing the difference image $E_j = X_j - Y_j$. Clearly, E_j contains edge and texture information, which allow training phase focus on establishing the relation between high-resolution feature maps and low-resolution ones. For LR images, they are filtered by K high-pass filters, remarked as $f_i, i = 1, 2, 3, \cdots, K$. Hence, we will obtain K filtered images from each LR image Z_j through $f_i \bigotimes Z_j$. For simplification, here Zeyde still inherit the first- and second-gradient operator written as:

$$f_1 = [-1, 1] \tag{3.34}$$

$$f_2 = [-1, 1]^T \tag{3.35}$$

$$f_3 = [-1, -2, 1] \tag{3.36}$$

$$f_4 = [-1, -2, 1]^T \tag{3.37}$$

After feature extraction, the LR images also need interpolation for the later patch extraction. Equally, we can filter the interpolated LR image Y_j by zero-padded filters

with the form as:

$$f_1 = [0, 0, -1, 0, 0, 1] \tag{3.38}$$

$$f_2 = [0, 0, -1, 0, 0, 1]^T \tag{3.39}$$

$$f_3 = [-1, 0, 0, -2, 0, 0, 1] \tag{3.40}$$

$$f_4 = [-1, 0, 0, -2, 0, 0, 1]^T \tag{3.41}$$

The number of zero depends on the scale factor of image. Equ. (3.38)-Equ. (3.41) present the filters with scaling factor equaling to 3.

With all the feature maps extracted, small matching patches are extracted in the same way as Yang's model and written in vector form then. If we assume the LR patches in vector form are $\tilde{P}_l$, the corresponding HR patches in vector form are P_h, and the size of each patch to be $\sqrt{n} \times \sqrt{n}$, the dimensions of each HR patch and LR patch are n and $4n$. Apparently, the dimensions of LR patches are much higher than HR patches. This is one of the reasons that Yang's model is very time consuming. Therefore, it will be much efficient if we can project the patch vectors into a subspace while preserving most of their average energy. Here, Zeyde's model utilized the Principle Component Analysis (PCA) to achieve the dimensionality reduction. By applying the projection matrix L, the reduced LR feature vectors can be obtained

by:

$$P_l = L\tilde{P}_l \tag{3.42}$$

So far, we have accomplished to construct the patch-pairs $\{P_h, P_l\}$ for the dictionary training.

3.2.2 K-SVD dictionary training algorithm

With the training patch pairs prepared, we are ready to train for the two coupled dictionaries. The objective function of low-resolution dictionary training is the same as that in Yang's model, which can be written as:

$$\alpha = \min_{\alpha} \|P_l - D_l\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{3.43}$$

In Zeyde's model, this optimization formula will be solved by Orthogonal Matching Pursuit algorithm [78] and K-SVD dictionary training procedure [76], which will be introduced in detail next. As the assumption of enforcing LR images and their corresponding HR ones share the same sparse representation is still available, the sparse code α trained from above can be utilized in constructing the high-resolution dictionary D_h. And the optimization problem to train D_h can be formulated as:

$$D_h = \min_{\alpha} \|P_h - D_h\alpha\|_2^2 \tag{3.44}$$

Hence, a straightforward least-square solution of D_h can be obtained by:

$$D_h = P_h \alpha^T (\alpha \alpha^T)^{-1} \tag{3.45}$$

Through this approach, we can only train for LR dictionary and its corresponding sparse codes, leading to more time saving and less computation complexity.

3.2.2.1 K-means clustering

K-SVD dictionary training algorithm [76] is designed as a direct generalization of K-means by Aharon *et al.*. Due to its simplicity, efficiency and flexibility, it has been widely used with different pursuit methods to search for an over-completed dictionary. Therefore, we firstly give a brief introduce of K-means clustering, not only K-SVD is a direct generalization of it, but also it will be used in NCSR model which we will introduce in detail in the next section.

As for a wide set of vectors $X = \{x_i\}_{i=1}^n \in \mathbb{R}^N$, by given a codebook with K codewords where $K \ll n$, each vector can be represented by one of the codewords using nearest neighbor assignment. It is apparent that this vector quantization (VQ) method will lead to an efficient description of given signals as clusters. If we use $C = [c_1, c_2, \cdots, c_K]$ to represent the codebook matrix with each column to be one codeword, for x_i, the vector quantization process can be written as $x_i \approx Ce_j$, where $0 < j \leq K$. Here, e_j is a trivial basis determined by:

$$\forall j \neq t, \|x_i - Ce_j\|_2^2 \leqslant \|x_i - Ce_t\|_2^2 \quad s.t. \quad 0 < j \leq K, 0 < t \leq K \tag{3.46}$$

Assume we use $E = [e_1, e_2, \cdots, e_K]$ to be the coefficient matrix, for all the signals, the

mean squared error (MSE) of this process of VQ can be expressed as:

$$MSE = \sum_{j} \|x_i - Ce_j\|_2^2 = \|X - CE\|_2^2 \tag{3.47}$$

Therefore, this VQ problem becomes a l^2-minimization problem, trying to find the best codebook matrix C while minimizing the MSE. It can be formulated as:

$$C = \min_{C} \|X - CE\|_2^2 \tag{3.48}$$

From Equ. (3.48) and Equ. (2.5), it is clear that there is an intriguing relationship between K-means and sparse representation. Similar to the methods of solving Equ. (2.15), K-means can be employed by the following iterative strategy [82]:

I. Initialize $C \in \mathbb{R}^{N*K}$;

II. Partition the given signal X into K parts, written as $S = (s_1, s_2, \cdots, s_K)$. For each of them, it can be obtained by:

$$s_k = \{i | \forall k \neq j, \|x_i - c_k\|_2^2 \leqslant \|x_i - c_j\|_2^2\} \tag{3.49}$$

III. For k-th column of C, it can be updated by:

$$c_k = \frac{1}{|s_k|} \sum_{i \in s_k} x_i \tag{3.50}$$

IV. Iterate Step **II** and Step **III** until converge.

Apparently, E in Equ. (3.48) can be seen as sparse code in Sparse-Land model with only one non-zero entry, and the codebook matrix C can be viewed as a set of

redundant bases as we described in previous sections. In previous section, we have mentioned that in Sparse-Land model, a signal can be approximately represented by the linear combination of some atoms in dictionary. Therefore, K-means is an extreme sparse representation, with the constraint of only one non-zero coefficient in each column of sparse code matrix.

3.2.2.2 Matching pursuit and orthogonal matching pursuit

From above, we know that sparse representation can be seen as a generalization of vector quantization. In Sparse-Land model, a given signal X is allowed to be represented by a linear combination of dictionary atoms. Different from K-means clustering, the coefficient vector in sparse representation is allowed to have more than one non-zero entries with arbitrary values. Mathematically, it can be formulated as Equ. (3.51):

$$\min_{D,\alpha} \|X - D\alpha\|_2^2, \quad s.t. \quad \forall i, \|\alpha_i\|_0 \leqslant T_0 \tag{3.51}$$

where T_0 is a positive integer. This is the objective formulation of K-SVD algorithm, and can be solved by updating D and α iteratively. However, different from the iterative strategy mentioned in dictionary training, K-SVD updates only one column of dictionary and the corresponding row of sparse code at the same time in each iteration, which will help reduce the training time, to some extension. Therefore, before the updating, we need to search for the sparse code by using the initial dictionary (can be initialized by the same method as K-means) first. As K-SVD is flexible to work with any pursuit algorithm, we choose Matching Pursuit algorithm (MP) [77] or Orthogonal Matching Pursuit algorithm (OMP) [78] due to its simplicity.

Matching Pursuit (MP) algorithm is one of the most commonly used pursuit algorithm raised by Mallet and Zhang [77]. It mainly aims at decomposing a given signal into a linear expansion of waveforms chosen from a redundant dictionary for the best matching. For a fixed dictionary $D = \{d_i\}_{i=1}^{K}$ with each column normalized in a Hilbert space $\mathbb{H}$, we can define V as the closed linear span of the dictionary vectors, and W is the orthogonal component of V in $\mathbb{H}$. Mathematically, V can be written as $V = \overline{Span\{d_i\}}$. MP algorithm is described as an iterative algorithm reconstructing the given signal X by the form of Equ. (3.52):

$$\mathbf{P}_V X = \sum_i \alpha_i D_i \tag{3.52}$$

Here, $\mathbf{P}_V$ is the orthogonal projection operator onto V, and α_i is the coefficient. This is the basic representation of signal constructed by MP approximation, and can be solved by an iterative algorithm. In detail, in the first iteration, the given signal X can be decomposed into orthogonal projection on some atom of dictionary, remarking as d_{r_1}, and a residue vector R_1, as shown in Equ. (3.53):

$$X =< X, d_{r_0} > d_{r_0} + R_1 \tag{3.53}$$

Clearly, R_1 is orthogonal to d_{r_0}. Thus, we have:

$$\|X\|^2 = | < X, d_{r_0} > |^2 + \|R_1\|^2 \tag{3.54}$$

Apparently, minimizing the residue R_1 is the key objective to select d_{r_0} from the dictionary. Therefore, from Equ. (3.54), it is clear that minimizing the residue R_1 is equivalent to maximizing $| < X, d_{r_0} > |^2$. In the other words, the best match atom

of the dictionary can be found by:

$$| < X, d_{r_0} > | \geqslant \rho \cdot sup_{i \in \{1,2,3,\cdots,K\}} | < X, d_i > | \tag{3.55}$$

Here, $\rho \in (0, 1]$ is an optimality factor, and $sup_{i \in \{1,2,3,\cdots,K\}} | < X, d_i > |$ refers to the supremum of $| < X, d_i > |$ when $i = 1, 2, 3, \cdots, K$. As the Matching Pursuit is an iterative algorithm, the residue R_1 is then regarded as the new signal to be decomposed. Hence, in kth iteration, the original signal X can be decomposed as:

$$X = \sum_{n=0}^{k-1} < R_n, d_{r_n} > d_{r_n} + R_k \tag{3.56}$$

Here, R_0 represents the original signal X, and $< R_n, d_{r_n} >$ is the coefficient in the sparse code. However, A non-ignorable shortcoming of MP algorithm is that it can not be well converged by any finite number of steps. That's because the result of each iteration is suboptimal. Considering $V_k = \overline{Span\{d_{r_0}, d_{r_1}, d_{r_2}, \cdots, d_{r_{k-1}}\}}$, Equ. (3.56) can be re-written as:

$$X = X_k + R_k \tag{3.57}$$

where X_k stands for the result of kth iteration. Therefore, X_k will be optimal if and only if $R_k \in V_k^{\perp}$. However, MP algorithm can only guarantee that $R_k \perp d_{r_{k-1}}$. Hence, in general, X_k is a suboptimal result in kth iteration. As a result, after finite number of steps, the residue may still be very large, although asymptotic convergence is guaranteed.

In order to solve this problem, Pati *et al.* proposed an refinement of Matching

Pursuit algorithm named Orthogonal Matching Pursuit (OMP) [78]. Compared with MP, OMP requires that the residue in each iteration is orthogonal to all the atoms of dictionary used for approximation in the previous iterations. Therefore, we need to set up a k^{th}-order model according to the Equ. (3.56) as follows:

$$X = \sum_{n=1}^{k} a_n^k d_n + R_k, \quad s.t. \quad < R_k, d_n >= 0, n = 1, 2, 3, \cdots, k \tag{3.58}$$

Here, d_n refers to the atom of dictionary selected in nth iteration. Hence, the $(k+1)^{th}$-order model updated from k^{th}-order model can be written as:

$$X = \sum_{n=1}^{k+1} a_n^k d_n + R_{k+1}, \quad s.t. \quad < R_{k+1}, d_n >= 0, n = 1, 2, 3, \cdots, k+1 \tag{3.59}$$

In OMP algorithm, an auxiliary model is required to build for the dependence of the current chosen atom of dictionary on the previous atoms. For example, as for d_{k+1}, the auxiliary model is built as:

$$d_{k+1} = \sum_{n=1}^{k} b_n^k d_n + \gamma_k, \quad s.t. \quad < \gamma_k, d_n >= 0, n = 1, 2, 3, \cdots, k \tag{3.60}$$

Apparently, $\sum_{n=1}^{k} b_n^k d_n = \mathbf{P}_{V_k} d_{k+1}$, and $\gamma_k = \mathbf{P}_{W_k} d_{k+1}$. W_k is the orthogonal component of V_k, as we defined before. Therefore, a correct update for $(k+1)^{th}$-order model will be:

$$a_n^{k+1} = a_n^k - \Psi_k b_n^k, \quad n = 1, 2, 3, \cdots, k \tag{3.61}$$

$$a_{k+1}^{k+1} = \Psi_k \tag{3.62}$$

where

$$\Psi_k = \frac{< R_k, d_{k+1} >}{\|d_{k+1}\|^2 - \sum_{n=1}^{k} b_n^k < d_n, d_{k+1} >} = \frac{< R_k, d_{k+1} >}{\|\gamma_k\|^2} \tag{3.63}$$

The detailed steps of OMP algorithm are shown as Algorithm 3:

Figure 3.2 provide a comparison of OMP and MP algorithm. Given the same dictionary, with the increase of iteration number, OMP algorithm converges much faster than MP, and the residue of OMP after finite number of iterations is much smaller than MP. The residue here is calculated by regular l^2-norm. As for more general results, we choose several different signals to compare the performance of the two algorithms, and four of them have been shown as below: Based on the faster and better convergence, in Zeyde's model, OMP algorithm is selected for the initial sparse coding.

3.2.2.3 K-SVD detailed updating description

As we have described above, the optimization problem formulated as Equ. (2.15) can be solved by freezing any of α or D and updating another until convergence. For many K-means generalization methods, such as Maximum Likelihood Method, Method of Optimal Dictionary (MOD) and Maximum A-Posterior Probability Approach (MAP), this iterative strategy is a general way to search for the best dictionary. However, different from those methods, K-SVD updates only one column of the dictionary at a time. With the acknowledge of the initial sparse code, K-SVD fixes all the columns in the dictionary except one, remarked as d_k, and updates it and its corresponding sparse coefficients by singular value decomposition (SVD) in order to minimize the

Algorithm 3 Orthogonal matching pursuit algorithm

Objective: Search for the best sparse code α of a given signal X over a fixed dictionary D by solving the objective function as:

$$\min_{\alpha} \|X - D\alpha\|_2^2, \quad s.t. \quad \forall i, \|\alpha_i\|_0 \leqslant T_0$$

Required: A given signal $X = \{x_i\}_{i=1}^{N}$, a fixed dictionary D with each column normalized and T_0 to be the maximum number of non-zero entries in each column of sparse vector;
while the iteration $1 \leq t \leq T0$ **do**:

1. Compute $< R_k, d_n >$ $(d_n \in D \setminus \Lambda_k)$, and find the best match $d_{n_{k+1}}$ that satisfies:

$$| < R_k, d_{n_{k+1}} > | \geqslant \rho \cdot sup_{i \in \{1,2,3,\cdots,K\}} | < R_k, d_i > |, \quad s.t. \quad 0 < \rho \leqslant 1$$

2. Re-order D by interchanging the atoms in the position $k+1$ and n_{k+1};
3. Compute $\{b_n^k\}_{n=1}^{k}$ by Equ. (3.60), and compute $\{a_n^{k+1}\}_{n=1}^{k+1}$ by Equ. (3.61) - Equ. (3.63);
4. Update the model by:

$$X_{k+1} = \sum_{n=1}^{k+1} a_n^{k+1} d_n$$

$$R_{k+1} = X - X_{k+1}$$

$$\Lambda_{k+1} = \Lambda_k \cup \{d_{k+1}\}$$

end while
Return α;

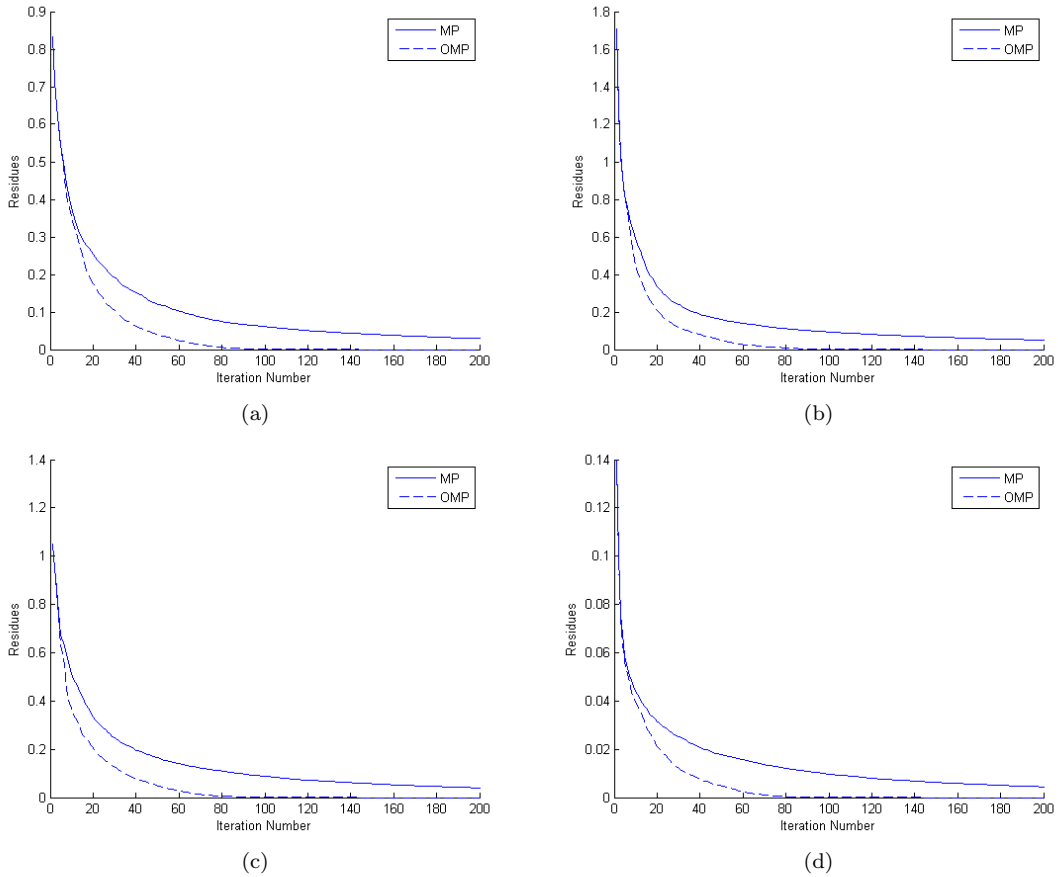


Figure 3.2: Comparison of the convergence capability of OMP and MP algorithm with different input signals

MSE. The initial sparse code here represents the best sparse code searched with the initial dictionary by pursuit algorithms mentioned in the last subsection.

Algorithm 4 K-SVD algorithm

Objective: Search for the best dictionary D and its corresponding sparse code α according to the given signal X by solving the objective function as:

$$\min_{D,\alpha} \|X - D\alpha\|_2^2, \quad s.t. \quad \forall i, \|\alpha_i\|_0 \leqslant T_0$$

Required: A given signal $X = \{x_i\}_{i=1}^N$ and an initial dictionary D_0 with each column normalized;
while the iteration $1 \leq t \leq T1$ **do**:

1. For each x_i, using any pursuit algorithm to search for the best sparse code α_i that satisfying the objective function as below:

$$\min_{\alpha_i} \|x_i - D\alpha_i\|_2^2, \quad s.t. \quad \|\alpha_i\|_0 \leqslant T_0$$

2. For each column $k = 1, 2, 3, \cdots, K$, define $\omega_k = \{i | 1 \leqslant i \leqslant K, \alpha_T^k(i) \neq 0\}$;
3. Decompose $D\alpha$ by K rank-1 matrixes and fix $K-1$ of them to calculate the overall error E_k by:

$$X - \sum_{j \neq k} d_j \alpha_T^j$$

4. Use the constraint ω_k to find E_k^R;
5. Apply SVD on E_k^R by:

$$E_k^R = U \Delta V^T$$

Update d_k by $U(:, 1)$ and update α_T^k by $\Delta(1, 1)V(:, 1)$;

end while
Return D and α;

For the detailed algorithm, we firstly recall the objective function formulated as

Equ. (3.51). With an initial dictionary D, searching for a good sparse code by pursuit algorithms has been introduced in detail in the last subsection. This is the first stage of K-SVD algorithm. After D and α fixed, we can turn to the second stage to update them together. As the K-SVD mainly aims at updating only one column of the dictionary at a time, we can use $\sum_{j=1}^{K} d_j$ to represent D with d_j to be the column of D, and replace α with $\sum_{j=1}^{K} \alpha_T^j$ with α_T^j to be the row of coefficients corresponding to d_j. Hence, the penalty term in Equ. (3.51) can be re-written as:

$$\|X - D\alpha\|_2^2 = \left\| X - \sum_{j=1}^{K} d_j \alpha_T^j \right\|_2^2 \tag{3.64}$$

Clearly, the multiplication $D\alpha$ has been decomposed to the sum of K rank-1 matrixes, among which $K-1$ columns of D are assumed to be fixed. In order to update the k-th column d_k, we keep modifying the penalty term in Equ. (3.51) as:

$$\left\| X - \sum_{j=1}^{K} d_j \alpha_T^j \right\|_2^2 = \left\| (X - \sum_{j \neq k} d_j \alpha_T^j) - d_k \alpha_T^k \right\|_2^2 = \left\| E_k - d_k \alpha_T^k \right\|_2^2 \tag{3.65}$$

Here, α_T^k stands for the row of coefficients corresponding to d_k, and E_k is the error for all the N samples when k-th atom is removed. Then, K-SVD allows us to use singular value decomposition (SVD) method to find the closest rank-1 matrixes to estimate E_k while minimizing the error efficiently.

However, as we know, α_T^k is the coefficients in sparse vectors, that might include lots of non-zero entries. Therefore, the updated α_T^k is likely to be filled if we decompose the error directly. An easy solution has been introduced in K-SVD model to solve this problem efficiently. Define ω_k as the indices of samples using k-th atom whose

corresponding α_T^k is non-zero. Equally, ω_k can be expressed as:

$$\omega_k = \{i | 1 \leqslant i \leqslant K, \alpha_T^k(i) \neq 0\} \tag{3.66}$$

By using ω_k, we define another matrix Ω_k as the extraction matrix, with the size of $N \times |\omega_k|$. In the position $(\omega_k(i), i)$ of Ω_k, the values are set to be 1, and we fill the values in the other positions with 0. By multiplying α_T^k by Ω_k, we can obtain the subvector of α_T^k with all the values are non-zero. We remark this subvector as α_T^R. Similarly, the multiplication $E_k^R = E_k\Omega_k$ is also the submatrix of E_k by using k-th atom. By multiplying Ω_k, the Equ. (3.65) will become:

$$\left\|E_k\Omega_k - d_k\alpha_T^k\Omega_k\right\|_2^2 = \left\|E_k^R - d_k\alpha_T^R\right\|_2^2 \tag{3.67}$$

After we calculate E_k^R, we can decompose it by SVD, as $E_k^R = U\Delta V^T$, and define $U(:,1)$ to be the updated d_k, and $\Delta(1,1)V(:,1)$ to be the updated α_k^R. The detailed algorithm for the whole K-SVD has been shown as Algorithm 4.

3.2.2.4 Computational complexity

Dictionary training is a time consuming procedure. With different number of training patches, dictionary size or the size of each patch, it may cost differently in computational complexity. Suppose we have a training set with K patches, and the dictionary $D \in \mathbb{R}^{N\times M}$. Clearly, the size of each patch should be $\sqrt{N} \times \sqrt{N}$, and the number of atoms in dictionary is M, which is also known as the dictionary size. In [83], Rubinstein *et al.* have provided the computational time of the standard OMP algorithm,

expressed as:

$$T_{omp} = 2KLNM + 2KL^2N + 2KL(M+N) + KL^3 \tag{3.68}$$

Here, L is the pre-defined number of non-zero entry in sparse codes, which is also the number of iterations. Therefore, if K, L and N are all fixed, the computational complexity of OMP algorithm can be written as $O(M)$.

As for K-SVD updating procedure, it performs M intensive singular value decomposition (SVD) steps, as we described before. Generally, the computational complexity of each SVD step is $O(min\{N^2K, NK^2\})$ [84] since the decomposed matrix has the size $N \times K$. In the scenario of image super-resolution, the number of training patches is normally much larger than the size of each patch, resulting in $K \gg N$. Therefore, here the computational complexity of each SVD step is $O(NK^2)$. Due to the K intensive singular value decomposition steps, the final K-SVD computational complexity per iteration can be expressed as $O(NK^2M)$. Clearly, if N and K are fixed, the complexity of K-SVD can also be expressed as $O(M)$.

The entire K-SVD with OMP algorithm includes OMP sparse coding and SVD updating. Therefore, combining the computational complexity above, we are able to say that the computational complexity of K-SVD with OMP algorithm can be approximately expressed as $O(M)$, which means when the number of iterations, the size of each patches and the number of training patches are fixed, the computational time of dictionary training will be approximately linear to the dictionary size. In Chapter 5, we will also provide an experiment to prove this conclusion.

3.2.3 Image super-resolution

In the testing phase, we firstly extract the feature map from the test LR image with the same method in training, and decompose it into patches with same size. In order to suppress the boundary artifacts, we extracted patches with some overlapping and average the values of those during reconstruction. After dimensionality reduction, we use OMP as well, to search for the best sparse codes. Combining the sparse codes and high-resolution dictionary, we can easily reconstruct the high-resolution feature map. Then, we regard the interpolated LR image as the low-frequency components, adding to the HR feature map to obtain the estimated HR image.

Since we have mentioned before, after obtaining the sparse prior, we normally need to regularize it so that it can satisfy the reconstruction constraint more. In Yang's model, a back-projection method is used globally to solve this problem, leading to a final reconstructed results. However, Zeyde holds a total different opinion with Yang. In his view, as the degradation operator $\mathbf{H}$ is singularity, the objective function of back-projection method does not has a single minimum. As a results, we can not converge to the correct high-resolution image. Overall, the back-projection method may not help improve the PSNR of final results, but may even adding more artifacts. And using interpolated LR image as the low-frequency components of HR image will help balance the sparse prior satisfies the reconstruction constrain more. Thus, the sparse prior here would stand for the final estimated high-resolution image.

Algorithm 5 shows the summary of Zeyde's testing phase. Compared with Yang's method, Zeyde's model save much computation time by denoting the PCA dimensionality reduction and K-SVD dictionary, and remove the back-projection step for adding potential artifacts. Besides, as for the low-frequency components of HR image,

Zeyde's model replace the mean value of each patches with the full low-resolution image. It not only simplifies the computation, but also help guarantee the sparse prior satisfying the reconstruction to some extension. The experimental results of these two models will be shown in the next chapter.

Algorithm 5 Detailed algorithm of the testing phase in Zeyde's model

Required: The given LR image Z, and two coupled dictionaries D_h and D_l

1. Pre-process the LR image with Bicubic interpolation operator, remark the results as Y, and perform the first- and second-order gradient method and patch extraction operator on it.
2. Search for the best sparse code with D_l by the OMP algorithm, and the objective function can be expressed as:

$$\alpha = \min_\alpha \|X - D_l\alpha\|_2^2 + \lambda \|\alpha\|_1;$$

3. Compute the estimated HR feature map $\hat{X}$ by:

$$\hat{X} = D_h\alpha$$

4. Computer the sparse prior by adding the interpolated LR image, as $X = \hat{X} + Y$;

Return X;

3.3 Non-locally centralized sparse representation algorithm

We have introduced the conventional single-image super-resolution models in the previous sections. Both of them include two stages: dictionary training and super-resolution. The former is a training stage and can be done offline, and the latter is a

testing stage. However, there are many other image super-resolution algorithms with different structures, and some of them can also achieve satisfying results. Here, we take NCSR model as an example to introduce a different single image super-resolution model.

NCSR model, which is short for Non-locally Centralized Sparse Representation Model, is raised by Dong *et al.* [56]. As an sparsity-based super-resolution model, NCSR still includes sparse prior and reconstruction regularization. However, different from the conventional model, NCSR does not have a training set, but use input image and its scale-up versions to train the dictionaries. Therefore, the dictionary training can be viewed as an adaptive and iterative process and can not be done offline. Besides, it applies the PCA-based sub-dictionary training so that there is no need to use over-completed dictionaries. As its outstanding performance, it has been proved to be one of the best state-of-the-art super-resolution algorithms. In the following, we will give a more detailed introduction.

3.3.1 Establishment of sparse model

As we mentioned before, in Sparse-Land model, an HR image can be re-expressed as the multiplication of a dictionary and its corresponding sparse codes. With a known dictionary, we can search for the sparse codes α_x by solving the function formulated as Equ. (3.69):

$$\alpha_x = \min_{\alpha} \|X - D\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{3.69}$$

where D stands for the dictionary, and X represents the HR image. The estimated HR image can be reconstructed by $X = D\alpha_x$. However, the observed image is often a degraded version of original HR image according to the observation model. Thus, the sparse codes searching can be solved by Equ. (3.70):

$$\alpha_y = \min_{\alpha} \|Y - \mathbf{H}D\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{3.70}$$

Here, Y stands for the observed LR image, and $\mathbf{H}$ represents the degradation matrix. The estimated HR image can be reconstructed by $\hat{X} = D\alpha_y$. Consider the difference of two sparse codes $v_\alpha = \alpha_y - \alpha_x$, which is called Sparse Code Noise (SCN) here. Apparently, the smaller v_α is, the higher quality of reconstructed image we can obtain because $\hat{X} - X = D\alpha_y - D\alpha_x$. So the α_y are expected as close as α_x, which is the key idea of NCSR model.

Before we continue the modeling of NCSR, we first re-write the objective function formulated as Equ. (3.69) in a more detailed form. In the last two sections, we have mentioned that the image needs to be pre-processed by decomposing into patches first. Therefore, as for an image signal X, we can re-write it as $\{x_i\}_i$. Each $x_i = \mathbf{R}_i X$ stands for the patch of size $\sqrt{n} \times \sqrt{n}$ at the location i of image X. Here $\mathbf{R}_i$ is an extraction matrix extracting the patch x_i from X at the location i. So for each patch x_i, the optimization problem can be re-written as:

$$\alpha_i = \min_{\alpha} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \tag{3.71}$$

Here, α_i is the sparse vector corresponding to x_i. Then for the entire image, the sparse codes can be represented as $\alpha = \{\alpha_i\}_i$. Thus, the reconstructed image $\hat{X}$ over

α can be obtained by a straightforward least square solution as:

$$\hat{X} = (\sum_{i=1}^{N} \mathbf{R}_i^T \mathbf{R})^{-1} \sum_{i=1}^{N} (\mathbf{R}_i^T D\alpha_i) \tag{3.72}$$

Here, N is the total number of patches. For simplification, we can re-write Equ. (3.72) as:

$$\hat{X} = D \circ \alpha \tag{3.73}$$

Therefore, with an observed LR image Y, Equ. (3.70) can be re-expressed as:

$$\alpha = \min_{\alpha} \|Y - \mathbf{H}D \circ \alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{3.74}$$

In a conventional super-resolution model, such as SCSR and Zeyde's model, the sparse priors are often obtained by two coupled dictionaries D_h and D_l. However, NCSR proposed a different super-resolution strategy so that we only use the interpolated LR image as the sparse prior, which can also be called initial HR image or initial value. In fact, experiments have shown that the qualities of initial values will influence the final estimations, and we will discuss it in detail in the next chapter.

Recalling the definition of SCN, it is difficult to measure it directly as the sparse codes of original HR image α_x is unknown. Hence, we may replace α_x with some good estimation, denoted as β, we could calculate v_α by $\alpha_y - \beta$. Therefore, combining with SCN, in order to suppress v_α and improve the objective function Equ. (3.74), an Centralized Sparse Representation Model (CSR) [55] can be used. So Equ. (3.74)

can be extended as:

$$\alpha = \min_{\alpha}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + \lambda \sum_i \|\alpha_i\|_1 + \gamma \sum_i \|\alpha_i - \beta_i\|_p\} \tag{3.75}$$

Here, β_i is some good estimation of α_i, λ and γ are positive constants, and p can be 1 or 2 (Here, p is set to be 1). In the CSR model, although the sparsity of α_i is required, the SCN v_α can also be suppressed by centralizing the sparse codes to β.

As other super-resolution algorithms, dictionary training is also a very important step in NCSR. Good dictionaries will help improve the quality of the final reconstructed image. In general, two coupled dictionaries learned from a training set are required to be over-completed, so that they can be used to recover varieties of images. However, it has been proved that it is unstable for sparse coding with an over-completed dictionary [54]. Besides, redundancies exist within qualities of patches extracted from examples. Therefore, NCSR utilizes K-means clustering algorithm to partition all the patches into K clusters, and denotes a PCA-based sub-dictionary training method to train one compact dictionary for each cluster. Since all the patches in one cluster are similar, there is no need to train an over-completed dictionary. Instead, we can utilize the principle components of each cluster to be the compact dictionaries [85], leading to a more stable and sparser representation.

In addition, we have mentioned that NCSR selects different training set from SCSR and Zeyde's model. As Glasner *et al.* shown that many redundancies exist in a image within the same scale or across different scales [86]. Thus, NCSR employed an iteration strategy, using initial HR image and its scaled versions as the training set, to the further clustering and dictionary training. After each iteration, the initial HR image will be updated, and can be used for the next dictionary training.

In a conventional image super-resolution model, λ in Equ. (3.75) is used to ensure the sparsity of α, which means only a small number of atoms will be selected from the trained over-completed dictionary. But in NCSR, each patch is encoded by a PCA sub-dictionary selected adaptively. Clearly, this has enforced the coefficients of this patch over the other sub-dictionaries to be zero. In the other words, this PCA-based sub-dictionary training method has actually ensured the sparsity of α, as a result of which, the regularization term $\|\alpha\|_1$ can be removed. So the objective function used for regularization can be re-written as:

$$\alpha = \min_{\alpha}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + \gamma \sum_i \|\alpha_i - \beta_i\|_1\} \tag{3.76}$$

And there is only one regularization term left, which is $\|\alpha_i - \beta_i\|_1$. Next, we will discuss the way to search for the good estimation β.

3.3.2 Sparse codes updating

As we defined above, β is some good estimation of α_x. Various methods of estimating β can be chosen depending on the prior knowledge of α_x we have. Clearly, if we have large amount of images that similar to the original image X, we are able to estimate β directly from them. However, this is often not available in practice. On the other hand, the sparse coding coefficients of each image are strongly correlated, and generic images are often contain many non-local redundancies [87]. Thus, it is allowed to estimate β directly from the input image. Here, we choose Non-Local Means method (NLM) for its simplification and efficiency. For each patch x_i, we first search for the non-local similar patches to it in a large window centered at pixel i. We can define Ω_i

to be the set of similar patches to the patch x_i (including patch x_i). Then, β_i can be computed as the weighted average of the sparse codes corresponding to these similar patches in Ω_i. Mathematically, it can be written as:

$$\beta_i = \sum_{q \in \Omega_i} \omega_{i,q} \alpha_{i,q} \tag{3.77}$$

Here, $\omega_{i,q}$ is the weight, computed by:

$$\omega_{i,q} = \frac{1}{W} exp(- \left\| \hat{x}_i - \hat{x}_{i,q} \right\|_2^2 / h) \tag{3.78}$$

where $\hat{x}_i$ and $\hat{x}_{i,q}$ are the estimations of x_i and $x_{i,q}$, computed by $\hat{x}_i = D\hat{\alpha}_i$ and $\hat{x}_{i,q} = D\hat{\alpha}_{i,q}$. W is the normalization parameter and h is a pre-defined scale. It is obvious that the weight is set to be inversely proportional to the distance between x_i and $x_{i,q}$. And as we use PCA-based orthogonal sub-dictionaries, the sparse codes $\hat{\alpha}_i$ and $\hat{\alpha}_{i,q}$ can be easily computed as $\hat{\alpha}_i = D^T \hat{x}_i$ and $\hat{\alpha}_{i,q} = D^T \hat{x}_{i,q}$.

Equ. (3.76) can also be solved by adopting an iteration strategy. Initially, β_i is set to be 0, *i.e.*, $\beta_i^{-1} = 0$. Equ. (3.76) can be re-written as:

$$\alpha = \min_{\alpha} \{ \left\| Y - \mathbf{H} D \circ \alpha \right\|_2^2 + \gamma \sum_i \left\| \alpha_i \right\|_1 \} \tag{3.79}$$

Thus, the sparse codes α^0 can be obtained from the initial HR image by $\alpha^0 = D^T X^0$. As we discussed before, this initial HR image can be seen as the sparse prior and obtained by interpolating the input image directly. With the initial sparse code α^0, we can search for the similar patches to computer the NLM estimated sparse codes, denoted as β^0. Based on the β^0, we can update α again by fixing β^0. Thus, in the k-th

iteration, the sparse codes α^k can be obtained by solving the optimization problem formulated as:

$$\alpha^{k+1} = \min_{\alpha}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + \sum_i \sum_j \gamma_{i,j} \left\|\alpha_i(j) - \beta_i^k(j)\right\|_1\} \tag{3.80}$$

Here, $\gamma_{i,j}$ stands for the value of γ in the position (i, j), $\alpha_i(j)$ and $\beta_i(j)$ represent for the j-th element of α_i and β_i. In this model, we utilize the surrogate algorithm in [88] to solve the problem formulated as Equ. (3.80). For the j-th element of α_i, the shrinkage operator is:

$$\alpha_i^{k+1}(j) = S_\tau(v_{i,j}^k - \beta_i(j)) + \beta_i(j) \tag{3.81}$$

where $S_\tau(\cdot)$ is the classical soft-thresholding operator, $\tau = \gamma_{i.j}/c$, and c is an auxiliary parameter. $v^k = K^T(Y - K \circ \alpha^k)/c + \alpha^k$, where $K = \mathbf{H}D$ and $K^T = D \circ \mathbf{H}$. This iteration will be carried out until convergence. Algorithm 6 present the summary of NCSR algorithm.

3.3.3 Super-resolution of noisy images

In Equ. (3.80), the regularization parameter γ is used to balance the fidelity term and the centralized sparsity term. In order to deal with noisy images, γ is required to be modified as well. As the connection between Maximum a Posterior (MAP) and sparse representation has been proved in [81], NCSR model extends this connection from local sparsity to non-locally centralized sparsity to help set γ. Before illustration, we define $\theta = \alpha - \beta$ for convenience. Therefore, the MAP estimation of θ can be

computed as:

$$\theta = \max_{\theta} logP(\theta|Y) = \max_{\theta}\{logP(Y|\theta) + logP(\theta)\} \tag{3.82}$$

Here, $logP(Y|\theta)$ is the likelihood term. If we assume θ and β are independent, the likelihood term can be modeled as Gaussian distribution:

$$logP(Y|\theta) = logP(Y|\alpha,\theta) = \frac{1}{\sqrt{2\pi}\sigma_n} exp\left(-\frac{1}{2\sigma_n^2}\left\|Y - \mathbf{H}D \circ \alpha\right\|_2^2\right) \tag{3.83}$$

Here, σ_n stands for the standard deviation of addictive noise. As for $logP(\theta)$, θ reflects the deviation of the estimated sparse coding coefficients β from the unknown α, which can be seen as SCN associated basically. In [56], Dong *et al.* has shown that the SCN signal can be well modeled by independent and identically distributed Laplacian distribution. Thus, we can model this prior probability as:

$$logP(\theta) = \prod_i \prod_j \frac{1}{\sqrt{2}\sigma_{i,j}} exp\left(-\frac{|\theta_i(j)|}{\sigma_{i,j}}\right) \tag{3.84}$$

Here, $\theta_i(j)$ stands for the j-th element of θ_i, and $\sigma_{i,j}$ represents the standard deviation of $\theta_i(j)$. Combining with Equ. (3.83) and Equ. (3.84), Equ. (3.82) can be re-written as:

$$\theta = \min_{\theta}\{\left\|Y - \mathbf{H}D \circ \alpha\right\|_2^2 + 2\sqrt{2}\sigma_n^2 \times \sum_i \sum_j \frac{1}{\sigma_{i,j}}|\theta_i(j)|\} \tag{3.85}$$

Therefore, if we have searched for a good estimation of α, *i.e.* β, we can obtained α by solving the optimization problem formulated as below:

$$\alpha = \min_{\theta}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + 2\sqrt{2}\sigma_n^2 \times \sum_i \sum_j \frac{1}{\sigma_{i,j}}|\alpha_i - \beta_i|\} \tag{3.86}$$

Compared with Equ. (3.80), we can conclude $\gamma_{i,j}$ can be computed by:

$$\gamma_{i,j} = \frac{2\sqrt{2}\sigma_n^2}{\sigma_{i,j}} \tag{3.87}$$

In general, non-locally similar patches are used for computing θ, which is utilized then to obtain $\sigma_{i,j}$. $\gamma_{i,j}$ can be updated in each iteration after we update θ, or in several iteration for saving the computation time.

3.4 Anchored neighborhood regression and global regression

In the last three sections, we have introduced the three single image super-resolution algorithms in detail, which will be directly used in our proposed method. From this section, we will provide brief introductions on some other SR algorithms that we used for comparison in our experiments. And in this section, we will firstly introduce the ANR model and its extreme case called GR model. Motivated to design a fast example-based super-resolution algorithm, Timofte *et al.* proposed a new method by combining sparse learned dictionaries with neighbor embedding, called Anchored Neighborhood Regression(ANR) [53]. It mainly aims at anchoring the embedding neighborhood of a low-resolution patch to the nearest atom in the dictionary by

Algorithm 6 Detailed algorithm of NCSR model

Input:
low-resolution input image Y;
Pre-defined parameters γ, δ, c (for updating α), $T1$, $T2$ and initial β to be 0;
Output:
the final recovered HR image $\hat{X}_N$;

Interpolate the given LR image to obtain the initial HR image $\hat{X}_N^{(1)}$;
while the outer-loop iteration $1 \leq t_1 \leq T1$ **do**:

- Computing or updating $\{D_k\}$ by K-means clustering and PCA;
- **while** the inner-loop iteration $1 \leq t_2 \leq T2$ **do**:

  1. Updating $\hat{X}_N^{(t_2+1/2)}$ by fidelity term:

  $$\hat{X}_N^{(t_2+1/2)} = \hat{X}_N^{t_2} + \delta \mathbf{H}^T (Y - \mathbf{H}\hat{X}_N^{t_2+1/2})$$

  2. Computing υ^{t_2} by using $\{D_k\}$ and $\hat{X}_N^{t_2}$;
  3. Calculating non-local mean β_i of $\alpha_i^{(t_2+1/2)}$;
  4. Re-weighting γ for better l^1-norm regularization;
  5. Updating $\alpha_i^{(t+1)}$ by iterative shrinkage operator:

  $$\alpha_i^{(t_2+1)}(j) = S_\tau(\upsilon_{i,j}^{t_2} - \beta_i(j)) + \beta_i(j)$$

  6. Reconstructing $\hat{X}_N^{(t_2+1)} = D \circ \alpha^{(t_2+1)}$;

- **end while**

end while
Return $\hat{X}_N$;

pre-computing the corresponding embedding matrix. Considering the size of neighborhood, it also has a global case which is known as Global Regression(GR) [53]. For simpler formulation, we start with the Global Regression.

From the last two sections, we can easily see that the least squares problem are normally constrained by using l^1 norm of the coefficients in sparsity-based super-resolution algorithms, which is time consuming. Therefore, Global Regression is reformulated as Ridge Regression, which is a least squares problem by using l^2 norm of coefficients instead for the regularization, written as:

$$\alpha = \|y - D_l\alpha\|_2^2 + \lambda \|\alpha\|_2 \tag{3.88}$$

In Equ. (3.88), α is the coefficient vector, y stands for the input low-resolution feature and D_l refers to the low-resolution dictionary. The algebraic solution of the coefficient vector α can be written as:

$$\alpha = (D_l^T D_l + \lambda I)^{-1} D_l^T y \tag{3.89}$$

By enforcing the coefficient vector of an HR patch to be the same as the that of its corresponding LR patch, we can recover the HR patches by:

$$x = D_h\alpha \tag{3.90}$$

where x stands for the recover HR patch, and D_h represents the high-resolution dictionary. Clearly, the entire procedure is quite similar to that of conventional sparsity-based super-resolution algorithms, such Yang's and Zeyde's algorithm. And as a

matter of fact, the two coupled dictionaries are also trained by K-SVD with OMP algorithms, the same as that used in Zeyde's model. The only difference between GR and conventional sparsity-based super-resolution model is that the least squares problem of the former is constrained by using l^2 norm but the latter uses l^1 norm. Combining Equ. (3.89) and Equ. (3.90), we can establish the relationship between the input low-resolution image with the recovered high-resolution image by:

$$x = D_h(D_l^T D_l + \lambda I)^{-1} D_l^T y \tag{3.91}$$

If we use P_G to represent $D_h(D_l^T D_l + \lambda I)^{-1} D_l^T$, the mapping function can be written as:

$$x = P_G y \tag{3.92}$$

where P_G is a projection matrix and can be computed offline.

Clearly, a global solution is obtained with the entire dictionary, which includes all the features from the training images. Compared with Global Regression, Anchored Neighborhood Regression use the local neighborhoods with a specific size instead of the entire dictionary, leading to more than one projection matrix and neighborhoods. With a sparse learned dictionary from K-SVD and OMP as well, we search for K nearest neighbors of each atom d_i by correlation between the dictionary atoms instead of the Euclidean distance. Then, based on the neighborhoods of d_i, a separate projection matrix P_{Gi} can be computed. Similar to the global regression, the projection matrix can be obtained offline, and the procedure of super-resolution can be

expressed as:

$$x_i = P_{Gi} y_i \tag{3.93}$$

where y_i is the feature map of input data. From extensive experiments, compared with Yang's and Zeyde's model, this approach has been proved to have better reconstruction ability with less computational time.

3.5 Adaptive sparse domain selection super-resolution

In Section 3.3, we have given a detailed introduction on NCSR model, which has been used in our proposed method. Considering the structures, NCSR is not the only one which is different from the conventional model. Here, we will provide another example named adaptive sparse domain selection super-resolution model. Since it is just used for experimental comparison, we will only give a brief introduction on it.

Adaptive sparse domain selection super-resolution model, which is generally called ASDS model [55] for convenience, was also proposed by Dong *et al.* Similar to the NCSR model, the main structure of ASDS is interpolating the given LR image with Bicubic interpolation first, and then updating the quality by an energy function with fidelity term and regularization term. In NCSR, the regularization term is designed based on suppressing the Sparse Coding Noise. By updating sparse code α iteratively, the recovered image will be closer to the ground truth. Different from NCSR, ASDS model utilizes two regularization terms. One of them is designed based on a piecewise autoregressive model trained from the training set, and the other is designed based

on the non-local self-similarity. The formulation can be written as:

$$\alpha = \min_{\alpha}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + \gamma \cdot \|(\mathbf{I} - \mathbf{A})D \circ \alpha\|_2^2 + \eta \cdot \|(\mathbf{I} - \mathbf{W})D \circ \alpha\|_2^2 + \sum_{i=1}^{N}\sum_{j=1}^{n}\lambda_{i,j}\|\alpha_{i,j}\|_1\} \tag{3.94}$$

Here, $\mathbf{H}$ is the degradation matrix, $\hat{X} = D \circ \alpha$ has the same definition as NCSR model where D is also trained by the same PCA-based sub-dictionary training method, and $\lambda_{i,j}$ is the weight assigned to $\alpha_{i,j}$. Referring to the regularization terms, the first one is designed based on the AR model, which can be trained from the training set, and the second one is designed based on the non-local self-similarity. As for the former, since the autoregressive model can well model the local area of a natural image, an AR model will be trained for each cluster of the training dataset by using all the samples in it. A square window is used here so that the AR model will be able to predict the central pixel of the window through its neighborhoods. Considering the k-th cluster C_k, the vector of AR model parameters $\mathbf{a}_k$ can be obtained by solving a least squares problem formulated as:

$$\mathbf{a}_k = \min_{\mathbf{a}} \sum_{\mathbf{c}_i \in C_k} (c_i - \mathbf{a}^T \mathbf{U}_i)^2 \tag{3.95}$$

Here, c_i is the central pixel of image patch $\mathbf{c}_i$, and $\mathbf{u}_i$ denotes the vector including all the neighborhood pixels of c_i in the pre-defined window. Since all the AR models $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k\}$ are prepared, for a given local patch, one or several AR models can be selected to regularize the solution. The adaptive selection of AR models is the same as the selection of the sub-dictionaries as we mentioned in NCSR model. With the selection of k-th AR model $\mathbf{a}_k$, we can expect the error between the central pixel

x_i of the given patch $\mathbf{x}_i$ and predicted central pixel by using $\mathbf{a}_k$ and $\mathbf{U}_i$ to be small, where $\mathbf{U}_i$ is the neighborhoods of x_i. Mathematically, it can be formulated as:

$$\min \sum_{\mathbf{x}_i \in C_k} \left\| x_i - \mathbf{a}_k^T \mathbf{U}_i \right\|_2^2 \tag{3.96}$$

We can simplify this constraint as:

$$\min \|(\mathbf{I} - \mathbf{A})D \circ \alpha\|_2^2 \tag{3.97}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{A}$ follows as:

$$\mathbf{A}(i,j) = \begin{cases} a_i & x_j \in \mathbf{G}_i, a_i \in \mathbf{a}_k \\ 0 & otherwise \end{cases} \tag{3.98}$$

This constraint can be added into the ASDS model, as shown in Equ. (3.95).

From above, we have established the AR model for local regularization of each image patch. In addition, since natural images contain repetitive patterns, we are able to provide another regularization term based on the non-local redundancies. For each image patch x_i, we search for its m similar patches in the entire image, denoted as $Q_i = \{x_i^1, x_i^2, x_i^3, ..., x_i^m\}$. As the non-local similarity model in NCSR, we use the weighted average $\sum_{j=1}^m w_j x_i^j$ to predict x_i, where w_j is the weight related to the patch x_i^j and can be computed as the same way in NCSR. Clearly, based on the quantities of redundancies in images, we would expect the prediction error $\left\| x_i - \sum_{j=1}^m w_j x_i^j \right\|_2^2$ to be small. Similar to the AR model, we can simplify this constraint as:

$$\min \|(\mathbf{I} - \mathbf{W})D \circ \alpha\|_2^2 \tag{3.99}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{W}$ follows as:

$$\mathbf{W}(i,j) = \begin{cases} w_j & x_i^j \in \mathbf{Q}_i \\ 0 & otherwise \end{cases} \tag{3.100}$$

Adding this constraint into the ASDS model, and we can have the entire energy function shown in Equ. (3.95).

3.6 Super-resolution via neighbor embedding

Since ANR and GR models are both the combination of sparsity-based and neighbor-embedding based super-resolution, we also use three neighbor-embedding (NE) based super-resolution algorithms for comparison in the experiments. These three algorithms are known as Neighbor Embedding with Locally Linear Embedding [43], Neighbor Embedding with Non-negative Least Squares [89] and Neighbor Embedding with Unconstraint Least Squares [53]. Due to the high similarity, we will introduce the main structure of them and point out the differences.

As we mentioned before, neighbor-embedding based super-resolution is one subcategory in learning-based super-resolution. With quantities of sample patches, HR patch can be reconstructed by the weighted average of its nearest neighbors. Similar to the sparsity-based super-resolution, the weights related with HR patches are enforced to be the same as those of corresponding LR patches. Apparently, in NE-based algorithms, dictionary is also an integral part, denoted as $D = \{X_d, Y_d\}$. Here, $Y_d = \{y_d^i\}_{i=1}^N$ and $X_d = \{x_d^i\}_{i=1}^N$ are the set of LR patch vectors and its corresponding HR patch vectors. With the dictionary prepared, for each patch x_i in input image

X_t, we firstly search for its nearest neighbor via Euclidean distance in X_d, denoted as $\mathbb{N}_t^i$. Then, we compute the reconstruction weights by minimizing the prediction error formulated as:

$$\varepsilon = \min \left\| x_i - \sum_{x_i^j \in \mathbb{N}_t^i} w_j x_i^j \right\|_2^2 \tag{3.101}$$

Here, w_j is the reconstruction weight related to the neighbor patch x_i^j. With the weights calculated, we can recover the HR image by finding the corresponding HR neighbors.

The main differences between these three NE-based algorithms are the constraints. In Neighbor Embedding with Locally Linear Embedding, the minimization function will be solved subject to the constraint $\sum_{x_i^j \in \mathbb{N}_t^i} w_j = 1$. Therefore, it is also called Neighbor Embedding with Constraint Least Squares. In this algorithm, a local Gram matrix G_i is defined as:

$$G_i = (x_i \mathbf{1}^T - \mathbb{N}_t^i)^T (x_i \mathbf{1}^T - \mathbb{N}_t^i)^T) \tag{3.102}$$

where $\mathbf{1}$ is a column vector of ones and $\mathbb{N}_t^i$ is the set of K nearest neighbors of x_i. Based on this, the close-form solution of this constraint least squares problem can be written as:

$$w = \frac{G_i^{-1}\mathbf{1}}{\mathbf{1}^T G_i^{-1} \mathbf{1}} \tag{3.103}$$

Considering the Neighbor Embedding with Non-negative Least Squares, the constraint used in the algorithm is to force $w \geq 0$. With this constraint, it can be

equivalent to a quadratic programming problem expressed as:

$$\min_{w} 0.5w^T Qw + c^T w, \quad s.t. w \geq 0 \tag{3.104}$$

where $w = \{w_j\}$, $Q = (\mathbb{N}_t^i)^T \mathbb{N}_t^i$ and $c = -(\mathbb{N}_t^i)^T x_i$. This problem can be solved by many mathematic methods such as Feature-sign search algorithm and so on. Compared with those two algorithms, Neighbor Embedding with Unconstraint Least Squares is a similar implementation proposed by Timofte *et al.* [53] and can be solved by formulating the problem as a quadratic programming problem without any constraint.

3.7 Summary

In this chapter, we have introduced three sparsity-based single-image super-resolution algorithms in detail, which will be directly used in our proposed method. Among them, Yang's and Zeyde's models are known as the conventional image SR models and can be divided into dictionary training and super-resolution. With two coupled HR-LR dictionaries, it is easy to recover the unknown HR image from its LR version by enforcing the sparse codes over two dictionaries to be the same. In addition, we also introduce another model named NCSR, which is one of the state-of-the-art algorithms in the area of image super-resolution. Different from the conventional model, the procedure of dictionary training in NCSR is adaptive and can not be done offline. During the process of super-resolution, the non-local means of reconstructed image in each iteration are utilized as image prior to suppress the sparse coding noise, so that the sparse codes of unknown HR image can be updated iteratively. Finally,

we can recover the HR estimated image with dictionary and its corresponding sparse codes. Besides, we also introduce 6 SR algorithms briefly as they have been used for comparison of the experimental results.

Chapter 4

Proposed joint dictionary-based super resolution algorithm

In Chapter 5.1, we have presented three existing SR algorithms, among which NCSR is one of the state-of-the-art SR methods. However, NCSR can be further improved as its initial HR value affects the final estimation. Based on this idea, in this chapter, we propose a joint sparsity-based single-image super-resolution framework, by integrating Zeyde's model and NCSR model together. Besides, we inherit the Gradient Histogram Preservation (GHP) model raised by Zuo *et al.* [90], adding a modified GHP term as an extra regularization term in Equ. (3.76) for better estimation. We call our model as JDSR, which is short for Joint Dictionary-based Super-Resolution.

4.1 Contribution

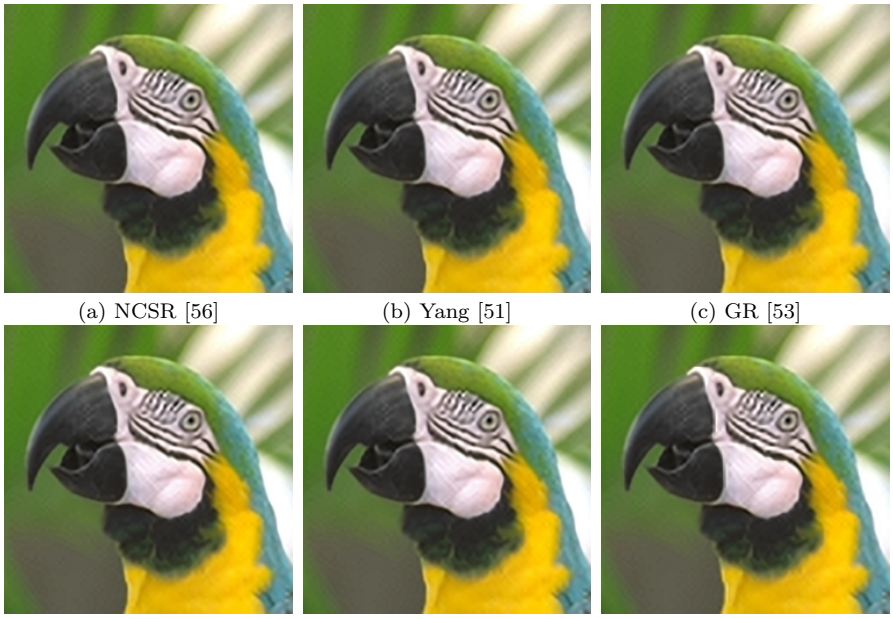
As for our proposed JDSR model, there are two main contributions on improving the estimation quality of final HR images.

Firstly, JDSR is a combination of three existing models. Since we found the initial values of NCSR model will affect the final estimation, many different algorithms have been tested to provide the initial HR value to NCSR. Based on our extensive experiments, we found that Zeyde's model works the best with NCSR, which means NCSR performs the best when the initial HR value is provided by Zeyde's model. Besides, in order to improve the quality of final estimation further, we also add a GHP term in the sparse model of NCSR for better regularization. Based on a reference gradient histogram estimated by a deconvolution model, it enhances the contrast of estimated gradient image by transformation F so that the histogram of transformed gradient image matches the reference gradient histogram.

Secondly, we proposed an edge-detection based enhancement, modifying the reference gradient histogram estimation algorithm so that it is able to perform better when the input image is blurring. From our experiments, we found the reference gradient histogram estimation algorithm we want use can fit well only if the input image is sharp. This makes sense since the observation model established here didn't consider the blurring effects. Besides, blurring images also have weaker gradients, which is not suitable for this GHP model as well. Based on this, we proposed an edge-detection based enhancement, or called ED-based enhancement, to enhance the textures of input image. As a result, the final estimations are closer to the ground truth.

4.2 Motivation

In the last chapter, we have mentioned the performance of NCSR model inherently relies on the initial estimated HR image, *i.e.*, initial value. Here, an experiment is given to verify the influence on the final results from the initial value. We firstly recover a given LR image by 6 super-resolution and interpolation methods, including NCSR, SCSR, Global Regression (GR), Bicubic interpolation, Bilinear interpolation and Nearest Neighbor(NN) interpolation. Then, using the 6 reconstructed images as



(a) NCSR [56] (b) Yang [51] (c) GR [53]

(d) Bi-cubic (e) Bilinear (f) Nearest Neighbor

Figure 4.1: Visual comparisons on the final results of NCSR with different initial estimations

the initial values, we recover the given LR image again by NCSR model. Table 4.1 gives the PSNR(dB)/SSIM of final results under Ratio=3 (magnify LR images by 9 times). And Fig. 4.1 shows the visual comparison of these results. Clearly, from the results above, we can verify that different initial values will lead to different

Table 4.1: PSNR(dB)/SSIM results of NCSR by using different initial values, with ratio=3

Algorithm	NCSR+NCSR	Yang+NCSR	GR+NCSR
PSNR(dB)/SSIM	28.68/0.8905	28.86/0.8916	28.79/0.8910
Algorithm	Bicubic+NCSR	Bilinear+NCSR	NN+NCSR
PSNR(dB)/SSIM	28.49/0.8894	28.42/0.8889	28.17/0.8869

reconstruction results. Besides, only considering the initial estimations, the qualities of them from high to low are ranked as follows: NCSR, Yang, GR, Bicubic, Bilinear, Nearest Neighbor. However, from Table 4.1, the performance of NCSR algorithm with different initial values can be ranked as Yang+NCSR, GR+NCSR, NCSR+NCSR, Bicubic+NCSR, Bilinear+NCSR and Nearest Neighbor+NCSR. This demonstrates that the initial value affects the final quality of recovered image, but not strictly positively related.

In order to improve the performance of NCSR model, one of the feasible approaches is to concatenate another existing SR method before NCSR, so that we can improve the quality of the initial HR image to some extension. Besides, based on the gradient histogram preservation (GHP) proposed by Zuo *et al.* [90], we also add a modified GHP term in the optimization formula used in NCSR model for the further regularization. As our proposed model is still based on sparse representation and dictionary learning, we call it Joint Dictionary-Based Single-Image Super-Resolution (JDSR). From our experiments, we found that using the results of Zeyde's model [52] as the initial HR images leads to the best final estimations.

By doing this, the super-resolved images will have clearer textures, sharper edges and better novel details in visual comparison. And when it comes to PSNR and SSIM, our proposed model is also able to provide satisfactory results. The next section will

give a detailed introduction of our proposed JDSR model, and in Chapter 5, we will show the performance of our proposed model and compare the results with some other SR algorithms.

4.3 Framework of joint dictionary-based super resolution

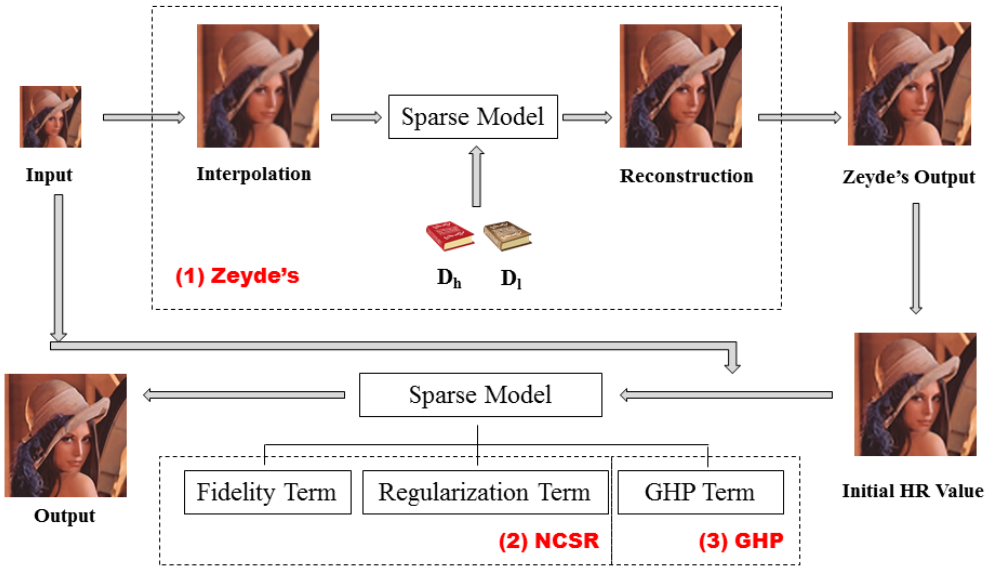


Figure 4.2: Framework of proposed JDSR model

Fig 4.2 shows the framework of proposed JDSR model. With a given LR image, we firstly adopt Zeyde's model to obtain the first-stage reconstructed HR image. Then, we take this image as the initial value, and improve its quality again by using NCSR model. In this way, the final estimated HR images will be better in qualities than using only one model, both in PSNR/SSIM and in visual comparison.

Commonly, image super-resolution algorithms are required to produce fewer artifacts. Therefore, the textures in the estimated HR images are often over-smoothed, leading to much weaker gradients. Apparently, a good estimated HR image without being over-smoothed should have the similar gradient distribution with the corresponding original HR image. Thus, we can adopt the Gradient Histogram Preservation (GHP) method raised by Zuo *et al.* [90] for the better reconstruction.

In the last chapter, we have discussed that the objective function of NCSR model can be formulated by using CSR model:

$$\alpha = \min_{\alpha}\{\|Y - \mathbf{H}D \circ \alpha\|_2^2 + \gamma \sum_i \|\alpha_i - \beta_i\|_1\} \tag{4.1}$$

As the gradient histogram constraint is allowed and can be easily incorporated with any other image prior, we can establish the JDSR model by adding an extra GHP term in the last as:

$$\begin{aligned}\hat{X} = \min_{\alpha} \|Y - \mathbf{H}D \circ \alpha\|_2^2 + \gamma \sum_i \|\alpha_i - \beta_i\|_1 + \mu \left\| F(\nabla \hat{X}) - \nabla \hat{X} \right\|_2^2 \\ s.t. \mathbf{h}_F = \mathbf{h}_r, \hat{X} = D \circ \alpha\end{aligned} \tag{4.2}$$

Here, $\mathbf{h}_r$, which can also be called as reference histogram, is a good estimated histogram of original HR image. F is a transformation operator, and normally to be an odd and monotonically non-descending function. $\mathbf{h}_F$ stands for the histogram of the transformed gradient image $|F(\nabla X)|$, and ∇ denotes as the gradient operator. μ is a positive constant. Equ. (4.2) can also be adopted by the alternating optimization strategy. When the estimated image $\hat{X}$ is given, we can fix it and update F; Similarly, when F is fixed, we can update $\hat{X}$.

When $\hat{X}$ is given, the objective function can be simplified into a sub-problem, formulated as:

$$\left\| F(\nabla \hat{X}) - \nabla \hat{X} \right\|_2^2 \quad s.t. \quad \mathbf{h}_F = \mathbf{h}_r \tag{4.3}$$

To solve this problem, a histogram specification based shrinkage operator can be used to update F. For simplification, we can select the standard histogram specification operator [91] to obtain the only reasonable monotonic transform, remarked as T. It can also be guaranteed that the histogram of $T(|\nabla \hat{X}|)$ to be the same as $\mathbf{h}_r$. Considering the minimizing function Equ. (4.3), the sign of $F(\nabla \hat{X})$ is required to be the same as $\nabla \hat{X}$. Therefore, F can be updated by:

$$F(\nabla \hat{X}) = sgn(\nabla \hat{X}) T(|\nabla \hat{X}|) \tag{4.4}$$

The JDSR algorithm can be summarized as Algorithm 7.

4.4 Reference histogram estimation with edge detection enhancement

From the framework of the JDSR model, it is easy to find that another important issue is finding a good $\mathbf{h}_r$ so that it can be well fitted to the histogram of unknown HR image X. In [90], a regularized deconvolution model is proposed to solve this problem. Firstly, the pixels in the gradient image ∇X are assumed to be independent and identically distributed (i.i.d). Hence, ∇X and ∇Y are i.i.d variables, denote as x and y, respectively. In image denoising, the observation model is normally simplified

Algorithm 7 Framework of proposed JDSR model

Input:
low-resolution input image Y;
initial estimated HR image from Zeyde's model $\hat{X}_Z$;
Pre-define parameters μ, γ, c (for updating α), ρ, ι, $T1$ and $T2$;
Pre-compute the reference histogram of unknown HR image $\mathbf{h}_r$ by using edge-detection based enhancement;
Output:
the final recovered HR image $\hat{X}_J$;

while the outer-loop iteration $1 \leq t \leq T1$ **do**:

- Updating $\{D_k\}$ by K-means clustering and PCA;
- Updating F by the reference gradient histogram $\mathbf{h}_r$;
- Updating $\hat{X}_J^{(t)}$ by GHP term;
- **while** the inner-loop iteration $1 \leq j \leq T2$ **do**:

    1. Updating $\hat{X}_J^{(t+1/2)}$ by fidelity term;
    2. Computing the sparse coefficients of each patch by $\alpha_i^{(t+1/2)} = D_k^T R_i \hat{X}_J^{(t+1/2)}$;
    3. Re-weighting μ for better l^1-norm regularization;
    4. Calculating non-local mean β_i of $\alpha_i^{(t+1/2)}$;
    5. Updating $\alpha_i^{(t+1)}$ by iterative shrinkage operator [56];
    6. Reconstructing $\hat{X}_J^{(t+1)}$;

- **end while**

end while
Return $\hat{X}_J$;

as:

$$Y = X + v \tag{4.5}$$

Here, v is an addictive white Gaussian noise, whose elements can be seen as i.i.d as well. So ∇v is i.i.d variables, denotes as ε. Thus, we have:

$$y = x + \varepsilon \tag{4.6}$$

Apparently, x and ε are independent. As the normalized histogram of image is actually a discrete probability distribution, we can regard it as the probability density function (PDF) of this image. Thus, if we define p_y, p_x and p_ε to be the PDF of y, x and ε, respectively, we can computed p_y by

$$p_y(y) = \int_z p_x(z) p_\varepsilon(y - z) dz \tag{4.7}$$

Thus, we replace p_y, p_x and p_ε with $\mathbf{h}_y$, $\mathbf{h}_x$ and $\mathbf{h}_\varepsilon$, Equ. (4.7) can be re-written as:

$$\mathbf{h}_y = \mathbf{h}_x \bigotimes \mathbf{h}_\varepsilon \tag{4.8}$$

Here, $\mathbf{h}_y$, $\mathbf{h}_x$ and $\mathbf{h}_\varepsilon$ stand for the normalized histograms of y, x and ε, respectively. Clearly, the histogram of the noisy observation y is the convolution of the histograms of original image x and noise ε.

Similar as the Sparse-Land model, in order to estimate the histogram of unknown

HR image, we can model this problem by a deconvolution model formulated as below:

$$\mathbf{h}_r = \min \left\| \mathbf{h}_y - \mathbf{h}_x \bigotimes \mathbf{h}_\varepsilon \right\|_2^2 + \lambda \cdot R(\mathbf{h}_x) \tag{4.9}$$

where λ is a positive constant and $R(\cdot)$ is a regularization term related to the previous image's gradient histogram. In [92], the gradient noise can be well modeled as an i.i.d Gaussian distribution:

$$p_\varepsilon = \frac{1}{2\sqrt{\pi}\sigma} exp(-\frac{\varepsilon^2}{4\sigma^2}) \tag{4.10}$$

And the gradient natural image signal can be well modeled as a hyper-Laplacian distribution [93–95] as:

$$p_x \approx C \cdot exp(-\kappa |x|^\zeta) \tag{4.11}$$

Here, C is the normalization parameter, and κ and ζ are the parameters used in hyper-Laplacian distribution. Besides, $\mathbf{h}_x$ is required to be non-negative potentially. Thus, the regularization term here can be characterized by:

$$R(\mathbf{h}_x) = a \cdot \left\| \mathbf{h}_x - C \cdot exp(-\kappa |x|^\zeta) \right\|_2^2 + b \cdot \left\| \mathbf{h}_x - \mathbf{h}_x^{'} \right\|_2^2 \tag{4.12}$$

where a and b are constant. Substituting Equ. (4.12) into Equ. (4.13), we have:

$$\begin{aligned} \mathbf{h}_r = \min \left\| \mathbf{h}_y - \mathbf{h}_x \bigotimes \mathbf{h}_\varepsilon \right\|_2^2 + \lambda \cdot \left\| \mathbf{h}_x - C \cdot exp(-\kappa |x|^\zeta) \right\|_2^2 + \eta \left\| \mathbf{h}_x - \mathbf{h}_x^{'} \right\|_2^2 \\ s.t. \mathbf{h}_x^{'} \geq 0 \end{aligned} \tag{4.13}$$

where η is also a constant.

This deconvolution problem can be solved by iteratively updating $\mathbf{h}_x$, $\mathbf{h}_x^{'}$, C, κ and ζ. Normally, with the initial setting of κ and ζ, $\mathbf{h}_0$ can be computed by $C{\cdot}exp(-\kappa|x|^{\zeta})$. So $\mathbf{h}_x$ can be updated by:

$$\mathbf{h}_x = \frac{\overline{FFT(\mathbf{h}_\varepsilon)} \bullet FFT(\mathbf{h}_y) + \lambda FFT(\mathbf{h}_0) + \eta FFT(\mathbf{h}_x^{'})}{FFT(\mathbf{h}_\varepsilon) \bullet \overline{FFT(\mathbf{h}_\varepsilon)} + \lambda + \eta} \tag{4.14}$$

Here, "$\bullet$" and "$\frac{*}{*}$" stand for the element-wise multiplication and division, and "$\overline{*}$" represents for the complex conjugate "$*$". Then, $\mathbf{h}_x^{'}$ and C can be updated as:

$$\mathbf{h}_x^{'}(i) = max(\mathbf{h}_x(i), 0) \tag{4.15}$$

$$C = \frac{\sum_i exp(-\kappa|i|^{\zeta})}{\sum_i \mathbf{h}_x(i)} \tag{4.16}$$

By using gradient decent method, in $(l+1)$-th iteration, we update κ and ζ by:

$$\kappa^{(l+1)} = \kappa^{(l)} + \iota \sum_i C|i|^{\zeta^{(l)}} exp(-\kappa^{(l)}|i|^{\zeta^{(l)}}) \times (C \cdot exp(-\kappa^{(l)}|i|^{\zeta^{(l)}} - \mathbf{h}_x(i))) \tag{4.17}$$

$$\begin{aligned} \zeta^{(l+1)} = \zeta^{(l)} + \rho \sum_{|i\neq 0|} \{ & C\kappa^{(l)}|i|^{\zeta^{(l)}} ln|i| exp(-\kappa^{(l)}|i|^{\zeta^{(l)}}) \\ & \times (C\dot{e}xp(-\kappa^{(l)}|i|^{\zeta^{(l)}} - \mathbf{h}_x(i)))\} \end{aligned} \tag{4.18}$$

Here, ι and ρ are pre-defined constants. The iteration will be finished until convergence. However, from extensive experiments, we found that this prediction method can fit well only if the input image is sharp. But in most cases of SR, the low-resolution image is blurring. This is because the gradients of blurred images are weak, which

is not suitable for the reference histogram estimation model we introduced above. Based on this, we proposed a simple edge-detection enhancement, or ED-based enhancement by scaling the gradients in edges so that the estimated reference histogram can be closer to the ground truth. Since we hope the procedure of edge detection to be fast and efficient, we only test some classical edge detection operators such as Roberts, Prewitt, Sobel, Laplacian and Canny. From experimental results, we found Canny detector can search for the clearest and continuous edge information, and it is also robust to noise. Therefore, we choose Canny detector to search for the edge information from input LR images. With edges detected, we apply a simple method by scaling the gradients of the feature maps to enhance the input texture, formulated as:

$$Y_g = Y_g \bigodot e \tag{4.19}$$

Here, $\bigodot$ is defined as:

$$x \bigodot y = axy + x\tilde{y} \tag{4.20}$$

where a is the scaling factor, y is a logical matrix and $\tilde{y}$ stands for the invert of y. In Equ. (4.19), Y_g stands for the gradient map of image, and e is the edge map generated by edge detector. By doing this, the final estimated reference histogram will be closer to the ground truth.

4.5 Summary

In this chapter, we modified the NCSR model and proposed a joint dictionary-based single-image super-resolution model, which is called JDSR. In our model, we replace Bicubic interpolation with Zeyde's model for a better initial value, and add a GHP term for the histogram regularization. As we have illustrated Zeyde's and NCSR model in detail in the last chapter, we only provide the structure of our proposed method and focus more on GHP term and reference gradient histogram estimation. In the next chapter, we will present the performance of reference histogram estimation and our JDSR model qualitatively and quantitatively.

Chapter 5

Experimental results

In this thesis, we present a new single-image super-resolution method, by concatenating Zeyde's model and NCSR model together. Based on this, we also add a GHP term for gradient histogram preservation, resulting in a better final estimation. We call our proposed model as Joint Dictionary-Based Super-Resolution model (JDSR). In our experiments, we only apply our method on color images, by magnifying the input LR image by the ratio of 3 and 4, *i.e.*, super-resolving the input LR image by 9 times and 16 times. We compare the results of our JDSR model with other 10 image super-resolution algorithms. Before we present the performance of our JDSR model, we firstly discuss the parameter setting of Zeyde's model and the gradient histogram estimation of unknown HR images.

5.1 Parameters setting of Zeyde's model

As a conventional single-image super-resolution model, it contains two independent stages: dictionary training and super-resolution. In the dictionary training phase, the training set we use is the same as [51]. For color images, we firstly transform the color channels from RGB to YCbCr, and apply the algorithm only on the Y channel, because people are more sensitive to luminance. Then, we filter the low-frequency components in both HR images and their related interpolated LR images by the methods we discussed before, and extract patches from them with size 9×9 and 12×12, corresponding to the scaling factor of 3 and 4, and arrange them in vector form as the HR-LR training pairs. The related interpolated LR images here are obtained by applying downsampling operator, Gaussian filter with window size of 7 and standard deviation of 1.6 and Bicubic interpolation on the example HR images. In patch extraction, we also set 1 pixel overlapped in both horizontal and vertical directions, to suppress the boundary artifacts. In dictionary training procedure, the λ in sparsity regularization, the iterations of K-SVD and the maximum non-zero entries T_0 of each sparse vector are set to be 0.1, 40 and 3, respectively.

In Zeyde's model, the selection of dictionary size is also very important. Large dictionary size, *i.e.*, large amount of atoms lead to more accurate estimations, as well as increasing of computation cost. Here, we trained coupled dictionaries with size 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192 under the scale factor 4, and compute the average computation time for each size. The high-resolution images used for training are color images, so we only extract Y channel and remove the low-frequency components by the methods we mentioned above. Besides, as the scale factor is equal to 4, the patch size used in these experiments was set to

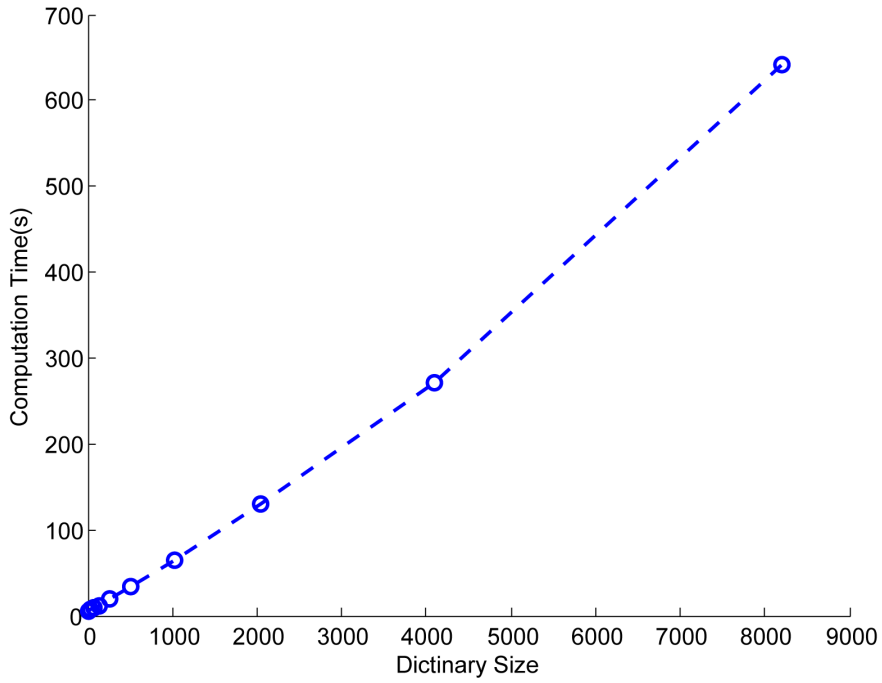


Figure 5.1: Computation time of training dictionaries with different sizes

be 12×12, and the number of patches was set to be 140,000. The algorithm was written in Matlab and run on a desk computer of Core i5-3475S 2.90GHz with 4G memory. From Figure 5.1, the computation time is almost linear to the dictionary size, which also proves the conclusion mentioned in Chapter . In the other words, with the increasing of dictionary size, the procedure of dictionary training will cost more, leading to a longer computation time. By picking one set of the trained dictionaries with different sizes, we apply them on the same input image under the same ratio to test the performance. In Figure 5.2, we present three of our testing images in visual comparison, and give the PSNR(dB)/SSIM results of them in Table 5.1. From our results, we can clearly find that although larger dictionaries normally perform better, the training process cost much more than it deserved. Considering both computation

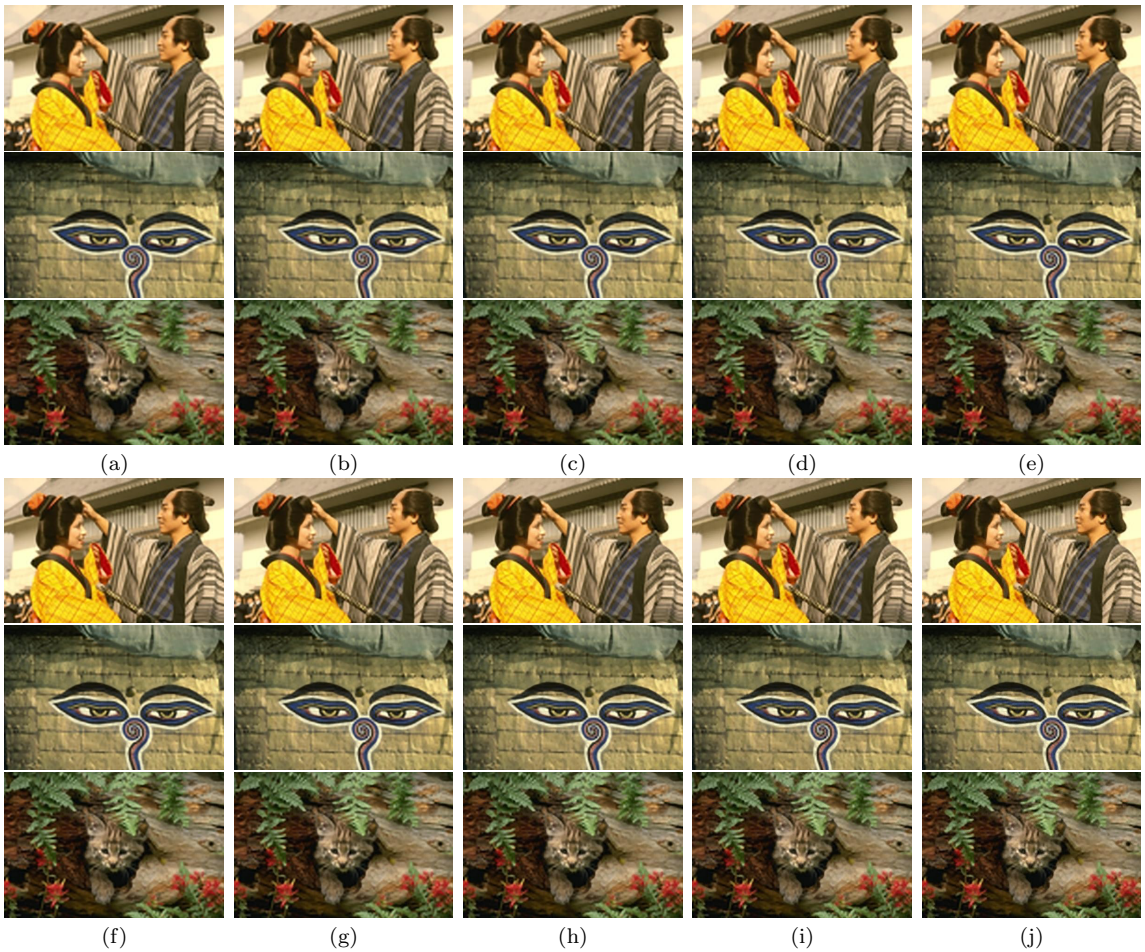


Figure 5.2: Visual comparisons of super-resolved images by Zeyde's model under ratio=4 with different dictionary sizes: (a)dictionary size = 16, (b)dictionary size = 32, (c)dictionary size = 64, (d)dictionary size = 128, (e)dictionary size = 256, (f)dictionary size = 512, (g)dictionary size = 1024, (h)dictionary size = 2048, (i)dictionary size = 4096, (j)dictionary size = 8192.

Table 5.1: PSNR(dB)/SSIM results of super-resolved images by Zeyde's model under ratio=4 with different dictionary sizes

Images \ Dictionary Size	16	32	64	128	256
Japan	22.91/0.6835	23.01/0.6906	23.07/0.6950	23.14/0.7001	23.21/0.7048
Graffiti	21.91/0.5469	21.99/0.5524	22.05/0.5560	22.10/0.5600	22.19/0.5654
Cat	23.38/0.5744	23.43/0.5802	23.48/0.5844	23.52/0.5883	23.55/0.5915
Images \ Dictionary Size	512	1024	2048	4096	8192
Japan	23.27/0.7089	23.33/0.7118	23.35/0.7129	23.41/0.7154	23.41/0.7153
Graffiti	22.23/0.5678	22.28/0.5695	22.31/0.5715	22.34/0.5733	22.36/0.5749
Cat	23.57/0.5941	23.60/0.5960	23.60/0.5968	23.61/0.5985	23.60/0.5973

cost and final estimation, we generally select the dictionary with 512 or 1024 atoms, and in Zeyde's model, we use 1024 as the dictionary size.

In super-resolution procedure, we deal with the input LR image with the same parameters as above, and the final results will be presented and compared with others in the later section.

5.2 ED-based reference gradient histogram estimation

In our proposed JDSR model, in addition to the two existing image super-resolution algorithms, we also add a modified GHP term as a new regularization term to help improve the final estimation. As we discussed in the previous chapter, the constraint of using GHP term is to enforce the transformed gradient histogram of current recovered image equal to the estimated gradient histogram of unknown HR image. Therefore, it is important to estimate a satisfactory gradient histogram of unknown HR image directly from the given input LR image. In this section, we will present some of our

experimental results to show the good performance of reference histogram estimation by using our modified method.

In reference histogram estimation procedure, six parameters are required to be set initially: λ, η, ι, ρ, κ and ζ. As image gradients are assumed to follow the hyper-Laplacian distributions, λ is required to be set as a relatively large constant to ensure the estimated histogram is close to the hyper-Laplacian distributions. In this model, we choose $\lambda = 10$. And η is used to make sure the estimated histogram is non-negative. Thus, it is also required to be relatively large, and we can also set it as 10. κ and ζ are two parameters of hyper-Laplacian distribution and we set them here to be $\kappa \in [0.001, 3]$ and $\zeta \in [0.02, 1.5]$. As for ι and ρ, they stand for the step sizes used in updating κ and ζ by gradient descent algorithm, respectively. Therefore, they are required to be sufficiently small so that the gradient descent algorithm will converge to a local minimum. And in this model, we set both of them to be 0.01.

We have tested large amount of images under different ratios and standard deviation of additive noises, and we only present some of our experimental results here for the space limitation. The LR images are generated by the downsampling operator, Gaussian low-pass filter and an addictive Gaussian white noise. The downsampling operator and Gaussian low-pass filter here are chosen as the same as we use in Zeyde's model, and the standard deviation of Gaussian white noise is set to be 5. The ratio we use here is equal to 4 as well. Figure 5.4 and Figure 5.6 give 4 of all our experimental results of GHP in X direction and Y direction, respectively. In Figure 5.4, each figure represent three gradient histograms of one example image: the red one stands for the gradient histogram of low-resolution input, the blue one denotes the gradient histogram of ground truth, and the black one represents the reference gradient histogram

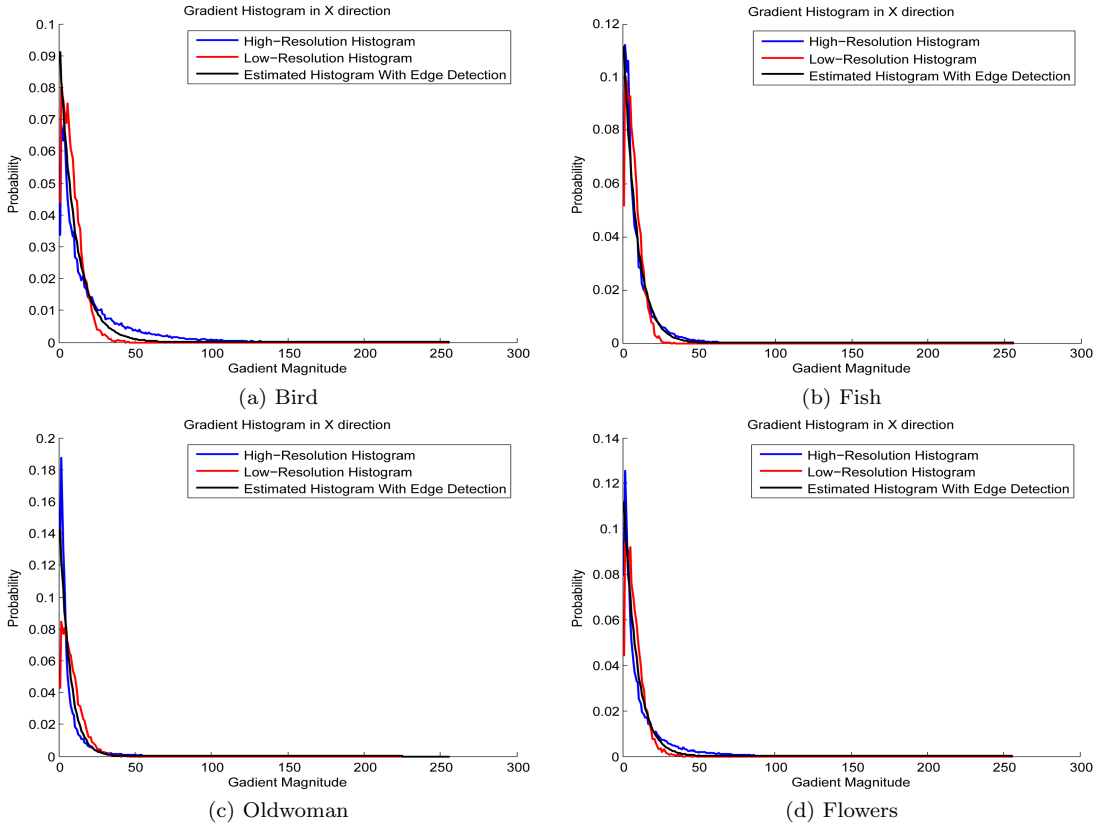
estimated by our proposed method. Figure 5.6 shares the similar idea with Figure 5.4, only replacing X direction with Y direction. Figure 5.5 and Figure 5.7 show the MSEs of ground truth with low-resolution input, reference gradient histogram estimated by original GHP method and our proposed ED-based enhancement method in bar plots. Figure 5.3 provides the corresponding 4 test images we use here. From the results, it is easy to see that our proposed ED-based reference gradient histogram estimation is able to estimate better reference histogram of unknown HR images that closer to the ground truth.

(a) Bird (b) Fish (c) Oldwoman (d) Flowers

Figure 5.3: Test images for reference gradient histogram estimation

5.3 Joint dictionary-based super-resolution

In this section, we will firstly present the parameters setting of our proposed JDSR model, and give some experimental results to compare it with other 10 image super-resolution algorithms. As for the input LR image, we still denote the same method mentioned in Zeyde's model, employing the downsampling operator and Gaussian low-pass filter with standard deviation of 1.6 on the HR image. Here, we set the standard deviation of addictive Gaussian white noise to be 0, and we will discuss the performance of recovering noisy images in detail in the next section.



(a) Bird (b) Fish

(c) Oldwoman (d) Flowers

Figure 5.4: Gradient histogram estimation of unknown high-resolution images in X direction under ratio=4 with standard deviation of noise equal to 5

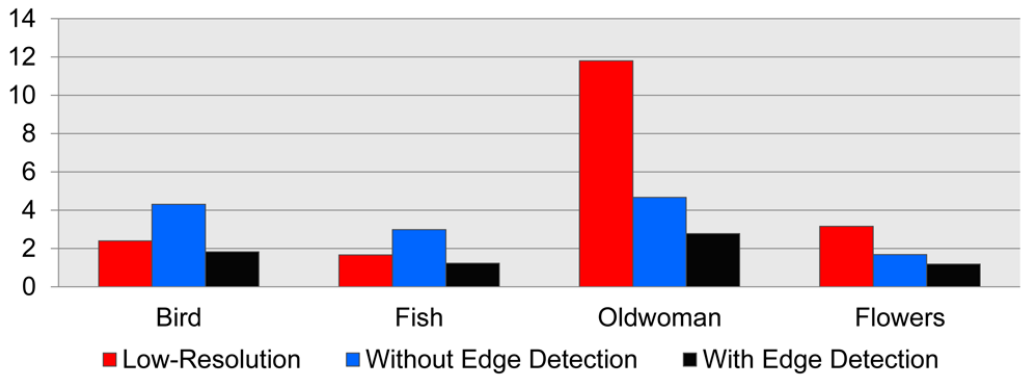
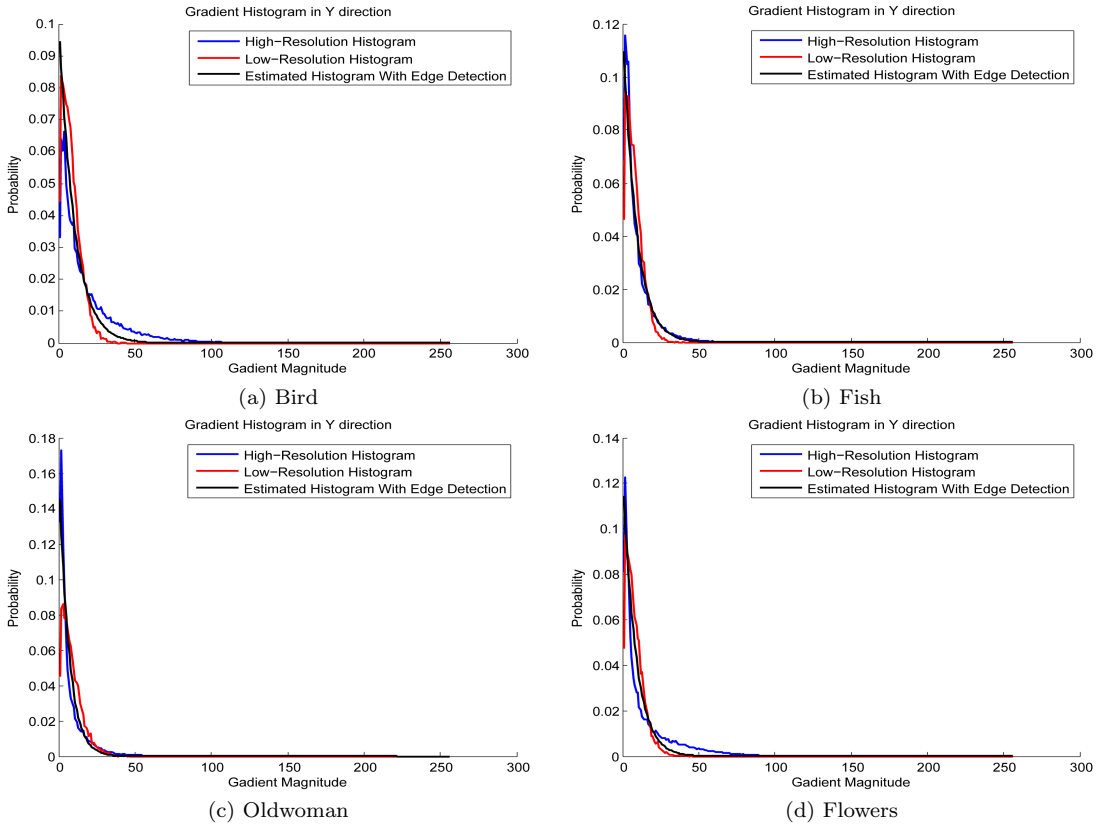


Figure 5.5: MSE results (10^{-5}) of 4 test images in X direction



(a) Bird (b) Fish

(c) Oldwoman (d) Flowers

Figure 5.6: Gradient histogram estimation of unknown high-resolution images in Y direction under ratio=4 with standard deviation of noise equal to 5

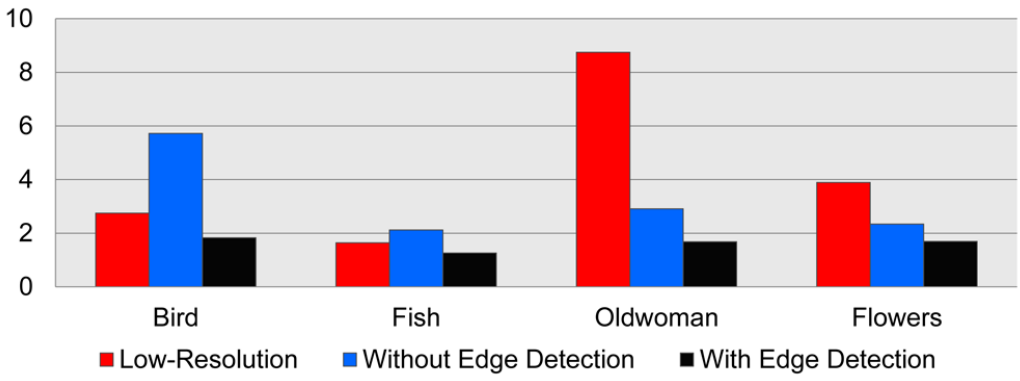


Figure 5.7: MSE results (10^{-5}) of 4 test images in Y direction

In JDSR model, the first stage is Zeyde's model, and all the parameters are set to be the same as we discussed before. As for GHP reference gradient histogram estimation, we also use the parameters exactly the same as we set above to estimate the reference gradient histogram of unknown HR image from the given LR image. The patch size and the number of clusters are set to be 7×7 and 64. Dealing with the image without noise, $\gamma = 7$, $\mu = 0.21$, $c = 1.55$, $T_1 = 4$ and $T_2 = 160$. With the noisy images, γ is adaptively updated by the method mentioned in the last chapter with an initial value of 1, $\mu = 0.03$, and c is set to be 0.21. T_1 and T_2 are the same as above. We have tested large amount of images, and select 10 to present here as examples, as shown in Figure 5.8. Figure 5.9 and Figure 5.10 provide the visual comparison of the results from 11 single-image super-resolution algorithms and ground truth under Ratio=3 and Ratio=4, respectively. In each figure, we present 4 example images selected from all of our test images, and each two rows stand for one example. From left to right and top to bottom, they are denoted as ground truth, results of Bicubic interpolation, Yang's model, Zeyde's model, Anchored Neighborhood Regression (ANR) model, Global Regression (GR) model, Adaptive Sparse Domain Selection (ASDS) model, NCSR model, JDSR model, Neighbor Embedding with Locally Linear Embedding (NE+LLE), Neighbor Embedding with Non-negative Least Squares (NE+NNLS) and Neighbor Embedding with Unconstraint Least Squares (NE+LS), respectively. The last three methods belong to neighbor embedding, and we use them only for comparison. For a fair comparison, we generate the test LR image by the same method as we discussed above. Compared with others, Zeyde's results looks clearer than those achieved by Bicubic interpolation, but they are still not satisfactory enough. As for NCSR model, it can implement much better results than Zeyde's model, with clearer textures and sharper edges. Nevertheless, our proposed JDSR

model can achieve even better than NCSR, and it is also able to overcome some shortcomings of NCSR such as "line-information reconstruction" (see the second example in Figure 5.9 and third example in Figure 5.10). In these "line-information" area, NCSR generally recover the textures with apparent artifacts, while our proposed JDSR model can achieve good results. Table 5.2 provides the PSNR(dB) and SSIM results of 10 test images by using these 11 super-resolution algorithms under Ratio=3. For each image, the first column are PSNR, and the second column are SSIM. And Table 5.3 presents the same things as Table 5.2, but under Ratio=4. For most test images, NCSR achieves the second best results in both visual comparison and PSNR/SSIM. However, there is a special case in our experiments. We can take the image "Plants" as example, shown as the fourth example in Figure 5.10. The PSNR/SSIM of NCSR by using this image are lower than others under both Ratio=3 and Ratio=4. While in visual comparison, it is easy to find the result of NCSR contain less artifacts and smoother and clearer textures. Nevertheless, our proposed JDSR always achieve better than others in all this cases, in both visual comparison and PSNR/SSIM results. Besides, from our experiments, our proposed algorithm is also capable to result a satisfactory reconstruction under with a large scale factor.

Most single-image super-resolution algorithms assume the input LR images are free of noise. However, noise is subjectively existing in the real application and required to be removed. Since we discussed in the previous sections, both SCSR and NCSR model try to remove noises by updating the regularization parameters in the procedure of super-resolution. Therefore, we test the performance of super-resolving noisy LR images by our proposed JDSR model, and compare the results with SCSR, Zeyde's model and NCSR. In our experiments, we also generate the LR

(a) Bike (b) Butterfly (c) Face (d) Flower (e) Hat

(f) Lena (g) Parrots (h) Parthenon (i) Plants (j) Raccoon

Figure 5.8: Images used in testing performance of JDSR model

Table 5.2: PSNR(dB)/SSIM results of super-resolved images by 11 different algorithms under ratio=3

Images	Bicubic	Yang	Zeyde	GR	ANR	ASDS	NCSR	NE+LS	NE+NNLS	NE+LLE	JDSR
Bike	21.51	22.64	22.59	22.20	22.69	23.11	23.16	22.63	22.47	22.60	**23.74**
	0.6793	0.7491	0.7479	0.7269	0.7534	0.7747	0.7791	0.7484	0.7423	0.7494	**0.8024**
Butterfly	22.76	24.36	24.63	23.74	24.60	25.46	26.30	24.53	24.31	24.45	**27.21**
	0.7929	0.8425	0.8604	0.8105	0.8546	0.8795	0.8946	0.8592	0.8485	0.8946	**0.9050**
Face	32.11	32.56	33.04	32.90	33.13	32.58	32.73	32.98	32.84	33.05	**33.24**
	0.8122	0.8156	0.8378	0.8386	0.8411	0.8275	0.8351	0.8363	0.8335	0.8389	**0.8437**
Flower	26.09	27.13	27.26	26.97	27.40	27.59	27.74	27.21	27.06	27.32	**28.28**
	0.7731	0.8144	0.8197	0.8082	0.8244	0.8305	0.8363	0.8187	0.8129	0.8216	**0.8507**
Hat	28.06	29.21	29.34	28.66	29.23	29.19	29.51	29.28	29.13	29.19	**30.56**
	0.8064	0.8306	0.8451	0.8248	0.8425	0.8371	0.8443	0.8435	0.8394	0.8417	**0.8640**
Lena	28.13	29.17	29.31	28.85	29.38	29.46	29.58	29.35	29.19	29.34	**30.19**
	0.8063	0.8256	0.8390	0.8261	0.8417	0.8387	0.8417	0.8394	0.8354	0.8405	**0.8518**
Parrots	26.61	27.84	27.96	27.69	28.07	28.44	28.49	28.03	27.80	27.99	**29.55**
	0.8688	0.8827	0.8953	0.8863	0.8959	0.8898	0.8894	0.8951	0.8915	0.8944	**0.9069**
Parthenon	24.65	25.19	25.37	25.11	25.28	25.48	25.66	25.25	25.11	25.26	**26.07**
	0.6687	0.8975	0.7072	0.6973	0.7053	0.7141	0.7225	0.7027	0.6979	0.7039	**0.7379**
Plants	30.18	31.41	31.67	31.11	31.74	30.47	30.82	31.60	31.51	31.68	**33.40**
	0.8349	0.8623	0.8791	0.8650	0.8821	0.8501	0.8617	0.8787	0.8761	0.8812	**0.9044**
Raccoon	27.05	27.48	27.60	27.63	27.67	27.87	27.89	27.61	27.52	27.63	**28.00**
	0.7001	0.7171	0.7351	0.7451	0.7411	0.7436	0.7521	0.7342	0.7314	0.7389	**0.7521**

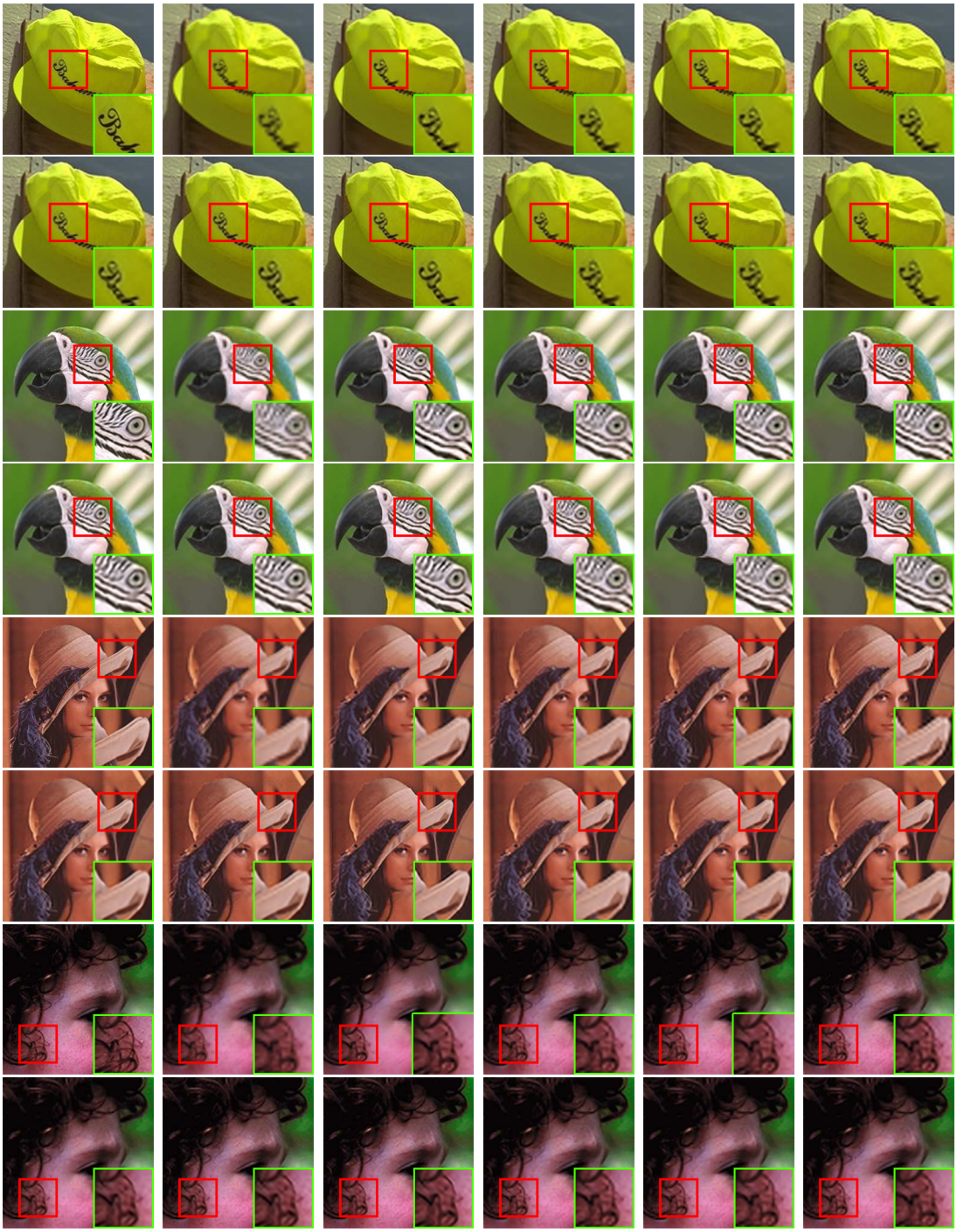


Figure 5.9: Visual comparison of the results from 11 image super-resolution algorithms under ratio=3

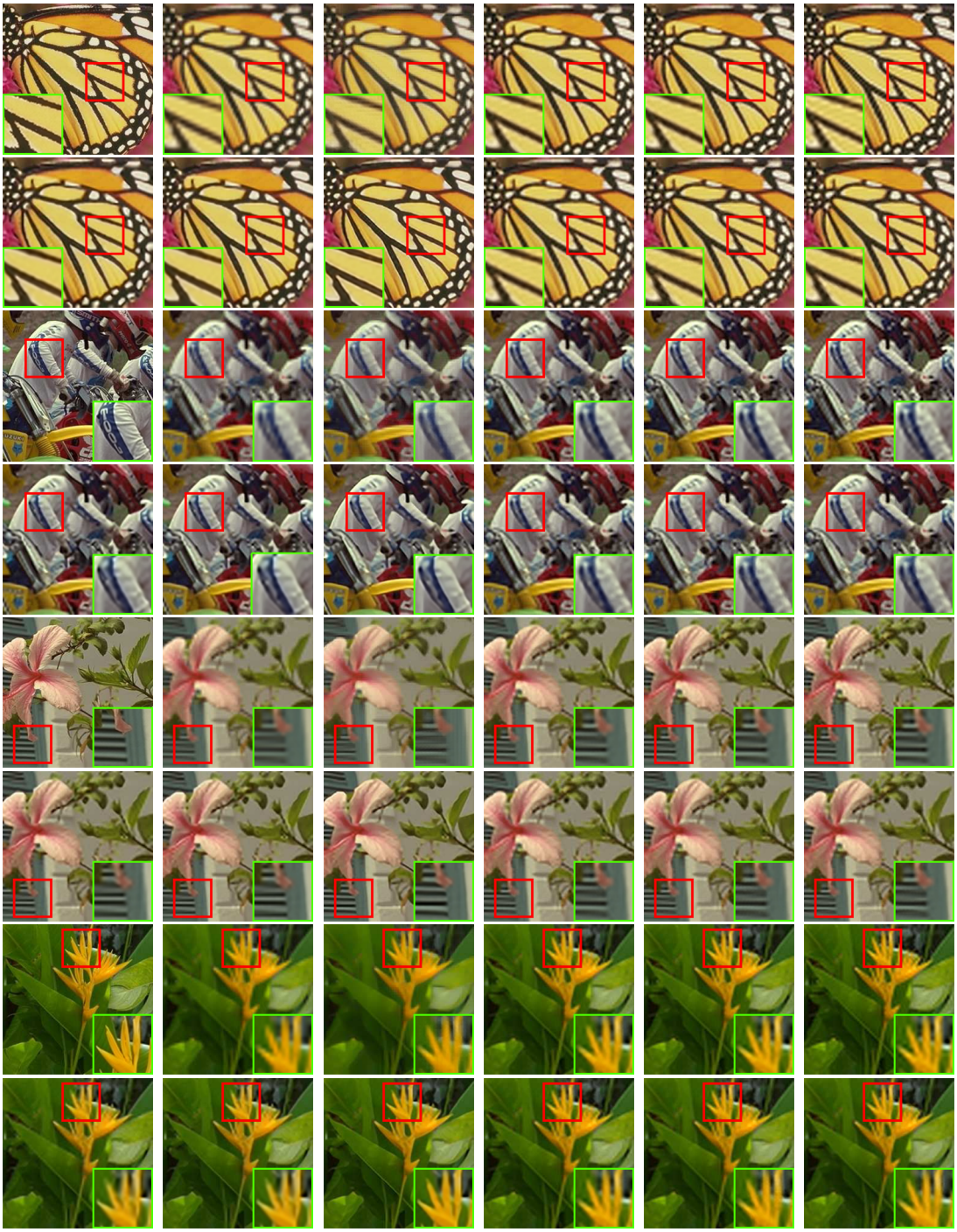


Figure 5.10: Visual comparison of the results from 11 image super-resolution algorithms under ratio=4

Table 5.3: PSNR(dB)/SSIM results of super-resolved images by 11 different algorithms under ratio=4

Images	Bicubic	Yang	Zeyde	GR	ANR	ASDS	NCSR	NE+LS	NE+NNLS	NE+LLE	JDSR
Bike	20.15	20.39	20.98	20.75	21.05	21.10	21.24	20.95	20.89	20.95	**22.02**
	0.5645	0.5847	0.6308	0.6367	0.6138	0.6578	0.6740	0.6282	0.6222	0.6295	**0.7167**
Butterfly	20.78	21.08	22.27	21.76	22.22	22.83	23.41	22.09	22.01	22.09	**25.19**
	0.6981	0.7335	0.7722	0.7184	0.7632	0.8188	0.8417	0.7635	0.7562	0.7583	**0.8853**
Face	30.54	29.97	31.25	31.09	31.31	30.98	31.15	31.21	31.10	31.28	**31.80**
	0.7540	0.7288	0.7770	0.7758	0.7800	0.7746	0.7827	0.7761	0.7736	0.7791	**0.7963**
Flower	24.44	24.47	25.32	25.14	25.38	25.16	25.34	25.25	25.19	25.32	**26.18**
	0.6797	0.6829	0.7276	0.7150	0.7301	0.7305	0.7473	0.7237	0.7206	0.7274	**0.7809**
Hat	26.74	26.79	27.79	27.19	27.64	27.46	27.86	27.65	27.59	27.59	**29.11**
	0.7576	0.7572	0.7941	0.7685	0.7887	0.7854	0.8050	0.7904	0.7865	0.7871	**0.8365**
Lena	26.60	26.57	27.59	27.18	27.61	27.54	27.91	27.58	27.42	27.54	**28.69**
	0.7401	0.7347	0.7745	0.7562	0.7760	0.7803	0.7944	0.7752	0.7689	0.7744	**0.8089**
Parrots	24.77	24.87	25.77	25.52	25.84	25.57	25.62	25.74	25.69	25.73	**26.89**
	0.8190	0.8085	0.8467	0.8365	0.8472	0.8403	0.8468	0.8450	0.8426	0.8455	**0.8732**
Parthenon	23.52	23.42	24.09	23.90	24.00	23.92	24.15	23.99	23.93	23.99	**24.72**
	0.5951	0.5887	0.6291	0.6191	0.6257	0.6320	0.6462	0.6236	0.6207	0.6243	**0.6728**
Plants	28.41	28.41	29.55	29.16	29.63	28.41	28.72	29.50	29.45	29.54	**31.08**
	0.7678	0.7537	0.8120	0.7959	0.8156	0.7850	0.8075	0.8101	0.8076	0.8124	**0.8650**
Raccoon	25.89	25.45	26.23	26.32	26.26	26.23	26.30	26.18	26.16	26.25	**26.57**
	0.6185	0.5816	0.6448	0.6542	0.6490	0.6496	0.6538	0.6421	0.6404	0.6477	**0.6640**

image by the same downsampling operator (here we take Ratio=4 as examples) and Gaussian low-pass filter as above, and add an addictive Gaussian white noise with standard deviation σ of 2,4,6,8 and 10. We have tested large amount of images, and as the space limitation, we only select 5 test images to present here, as shown in Figure 5.11. Figure 5.12 provides the visual comparison of 5 recovered images with different σ. In Figure 5.12, from top to bottom, each column are the results with σ equals to 2,4,6,8 and 10, respectively. And from left to right, each column stands for the ground truth, the results by using SCSR, Zeyde's Model, NCSR and our proposed JDSR, respectively. Table 5.4 gives the PSNR(dB) results of the 5 test images. In visual comparison, SCSR generates blurring recovered images since the regularization parameter used in it are positively related with σ. And for Zeyde's model, the results are still noisy because no denoising measure is designed in it. NCSR model can

reconstruct noiseless image with fewer artifacts and less blurring effects. However, compared with our proposed JDSR model, it can generate less sharpen edges and less clear textures. In Table 5.4, our proposed JDSR model perform the best in PSNRs, from which we can also conclude the outstanding performance of our model in super-resolving noisy LR images.

(a) Boat (b) Corn (c) House (d) Penguin (e) Wall

Figure 5.11: Images used in testing denoising performance of JDSR model

Table 5.4: PSNR(dB) results of super-resolved noisy images with different standard deviations by 4 different algorithms under ratio=4. For each image, top left: SCSR; top right: Zeyde's model; bottom left: NCSR; bottom right: JDSR

Images	$\sigma = 2$		$\sigma = 4$		$\sigma = 6$		$\sigma = 8$		$\sigma = 10$	
Boat	34.45	34.49	34.44	34.46	34.44	34.41	34.43	34.35	34.41	34.27
	36.04	**36.08**	35.86	**35.99**	35.73	**35.97**	35.52	**35.90**	35.25	**35.75**
Corn	33.90	33.96	33.89	33.95	33.88	33.91	33.87	33.87	33.85	33.81
	35.87	**36.16**	35.86	**36.15**	35.64	**35.96**	35.38	**35.79**	35.11	**35.64**
House	34.54	34.63	34.53	34.60	34.52	34.56	34.51	34.50	34.49	34.42
	35.58	**35.72**	35.52	**35.69**	35.40	**35.60**	35.23	**35.52**	35.02	**35.44**
Penguin	37.21	37.51	37.20	37.41	37.19	37.25	37.15	37.05	37.09	36.81
	39.51	**40.39**	39.25	**40.08**	38.71	**39.69**	38.06	**39.31**	37.43	**38.93**
Wall	36.69	36.81	36.68	36.73	36.67	36.32	36.65	36.47	36.61	36.29
	37.54	**37.84**	37.43	**37.72**	37.18	**37.59**	36.86	**37.46**	36.48	**37.34**

5.4 Summary

In this chapter, we provided the parameters setting of our JDSR model, and tested the performance of the reference gradient histogram estimation and super-resolution

Figure 5.12: Visual comparison of the results from 4 image super-resolution algorithms under ratio=4

qualitatively and quantitatively. During the parameters setting of Zeyde's model, we presented the SR results for using different sizes of dictionaries so that we can select one to balance the computation time and estimation performance. As for the histogram estimation, we ploted the histograms of the ground truths and the estimated HR images and calculated their MSEs. The results are satisfactory as MSEs are all very small. When it comes to the super-resolution, our proposed JDSR model can achieve outstanding performance in both visual and PSNR/SSIM comparisons.

Chapter 6

Conclusion

Single-image super-resolution has been studied for decades, during which large amount of algorithms were proposed with satisfactory performance. With small scaling factor, simple interpolation according to the neighbors can work well and efficiently. However, with the increase of ratio, fewer image priors can be utilized to recover the original textures. So far, sparsity-based methods have been proved to provide a good high-resolution estimation by transferring the reconstruction into a minimization optimization. In general, this class of algorithms search for the mapping relationship between HR example images and their related LR versions, and reconstruct the unknown HR image by this prior. Apparently, huge databases are required to be trained and stored, so that these methods are often time consuming.

In this thesis, based on two existing state-of-the-art sparsity-based SR algorithms, we propose a joint single-image sparsity-based super-resolution model. In the procedure, we also add a GHP term as an extra prior to constraint and regularize the recovered HR images. The experimental results have shown that our proposed JDSR

model is capable to recover HR images with clearer textures, shaper edges and better novel details. Besides, our method can also work well under a large ratio (such as Ratio=4, *i.e.*, magnify images by 16 times) and on dealing with noisy input LR images.

References

[1] S. Park, M. Park, and M. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[2] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, "Signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis," *IEE Proceedings I (Communications, Speech and Vision)*, vol. 140, no. 1, pp. 19–25, 1993.

[3] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences-a comprehensive review with directions for future research," *Lab. Image and Signal Analysis, University of Notre Dame, Tech. Rep*, 1998.

[4] S. Chaudhuri, *Super-resolution imaging*. Springer Science & Business Media, 2001.

[5] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 2, pp. 181–186, 1992.

[6] W. Siu and K. Hung, "Review of image interpolation and super-resolution," in

Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), Dec. 2012, pp. 1–10.

[7] N. Nguyen and P. Milanfar, "An efficient wavelet-based algorithm for image superresolution," in *International Conference on Image Processing*, Sep. 2000, pp. 351–354.

[8] S. Kim, N. Bose, and H. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 6, pp. 1013–1027, 1990.

[9] R. Hardie, K. Barnard, J. Bognar, E. Armstrong, and E. Watson, "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system," *Optical Engineering*, vol. 37, no. 1, pp. 247–260, 1998.

[10] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," in *Maximum Entropy and Bayesian Methods*, 1996, pp. 293–308.

[11] S. Borman and R. Stevenson, "Super-resolution from image sequences-a review," in *mwscas*, Aug. 1998, pp. 374–378.

[12] M. Ozkan and M. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Mar. 1992, pp. 169–172.

[13] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[14] R. Bernstein, "Digital image processing of earth observation sensor data," *IBM Journal of Research and Development*, vol. 20, no. 1, pp. 40–57, 1976.

[15] R. Tsai and T. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, no. 2, pp. 317–339, 1984.

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

[17] P. Thevenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of medical imaging, processing and analysis*, pp. 393–420, 2000.

[18] M. Li and T. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1121–1128, 2008.

[19] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[20] W. Tam, C. Kok, and W. Siu, "Modified edge-directed interpolation for images," *Journal of Electronic imaging*, vol. 19, no. 1, pp. 1–20, 2010.

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

[22] 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.

[23] Z. Lin and H. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 83–97, 2004.

[24] B. Morse and D. Schwartzwald, "Image magnification using level-set reconstruction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2001, pp. 333–340.

[25] M. Ben-Ezra, Z. Lin, and B. Wilburn, "Penrose pixels super-resolution in the detector layout domain," in *IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.

[26] R. Fattal, "Image upsampling via imposed edge statistics," in *ACM Transactions on Graphics (TOG)*, Jul. 2007, pp. 95:1–95:8.

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

[28] J. Sun, Z. Xu, and H. Shum, "Gradient profile prior and its applications in image

super-resolution and enhancement," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1529–1542, 2011.

[29] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36–51, 2009.

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

[31] C. Liu, H. Shum, and W. Freeman, "Face hallucination: Theory and practice," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 115–134, 2007.

[32] Q. Wang, X. Tang, and H. Shum, "Patch based blind image super resolution," in *IEEE 10th International Conference on Computer Vision (ICCV)*, Oct. 2005, pp. 709–716.

[33] K. Zhang, X. Gao, X. Li, and D. Tao, "Partially supervised neighbor embedding for example-based image super-resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 230–239, 2011.

[34] Y. Tang, Y. Yuan, P. Yan, and X. Li, "Single-image super-resolution via sparse coding regression," in *IEEE 6th International Conference on Image and Graphics (ICIG)*, Aug. 2011, pp. 267–272.

[35] X. Gao, K. Zhang, D. Tao, and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3194–3205, 2012.

[36] Y. Tang, X. Pan, Y. Yuan, P. Yan, L. Li, and X. Li, "Local semi-supervised regression for single-image super-resolution," in *IEEE 13th International Workshop on Multimedia Signal Processing (MMSP)*, Oct. 2011, pp. 1–5.

[37] X. Lu, H. Yuan, Y. Yuan, P. Yan, L. Li, and X. Li, "Local learning-based image super-resolution," in *IEEE 13th International Workshop on Multimedia Signal Processing (MMSP)*, Oct. 2011, pp. 1–5.

[38] Y. Tang, P. Yan, Y. Yuan, and X. Li, "Single-image super-resolution via local learning," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 1, pp. 15–23, 2011.

[39] X. Gao, K. Zhang, D. Tao, and X. Li, "Joint learning for single-image super-resolution via a coupled constraint," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 469–480, 2012.

[40] G. Mu, X. Gao, K. Zhang, X. Li, and D. Tao, "Single image super resolution with high resolution dictionary," in *IEEE 18th International Conference on Image Processing (ICIP)*, Sep. 2011, pp. 1141–1144.

[41] X. Lu, P. Yan, Y. Yuan, X. Li, and H. Yuan, "Utilizing homotopy for single image superresolution," in *First Asian Conference on Pattern Recognition (ACPR)*, Nov. 2011, pp. 316–320.

[42] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications Magazine*, vol. 22, no. 2, pp. 56–65, 2002.

[43] H. Chang, D. Yeung, and Y. Xiong, "Super-resolution through neighbor embed-

ding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2004, pp. 1–8.

[44] K. Ni and T. Nguyen, "Image superresolution using support vector regression," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1596–1610, 2007.

[45] H. He and W. Siu, "Single image super-resolution using gaussian process regression," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2011, pp. 449–456.

[46] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.

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

[48] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[49] C. Yang and M. Yang, "Fast direct super-resolution by simple functions," in *IEEE Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 561–568.

[50] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 1059–1066.

[51] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[52] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*. Springer, 2010.

[53] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 1920–1927.

[54] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4701–4714, 2009.

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

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

[57] L. Juan, W. Jin, Y. Shen, and L. Jin, "Dictionary learning for image super-resolution," in *33rd Chinese Control Conference (CCC)*, Jul. 2014, pp. 7195–7199.

[58] G. Bhosale, A. Deshmukh, S. Medasani, and R. Dhuli, "Image super-resolution

using dictionaries and self-similarity," in *International Conference on Signal Processing and Communications (SPCOM)*, Jul. 2014, pp. 1–6.

[59] S. Ramakanth and R. Babu, "Super resolution using a single image dictionary," in *IEEE International Conference on Electronics, Computing and Communication Technologies*, Jan. 2014, pp. 1–6.

[60] F. Yeganli, M. Nazzal, M. Unal, and H. Ozkaramanli, "Image super-resolution via sparse representation over coupled dictionary learning based on patch sharpness," in *Proceedings of the 2014 European Modelling Symposium*, Oct. 2014, pp. 203–208.

[61] F. Yeganli, M. Nazzal, and H. Ozkaramanli, "Selective super-resolution via sparse representations of sharp image patches using multiple dictionaries and bicubic interpolation," in *23th Signal Processing and Communications Applications Conference (SIU)*, Jan. 2015, pp. 1957–1960.

[62] M. Cheng, C. Wang, and J. Li, "Single-image super-resolution in rgb space via group sparse representation," *IET Image Processing*, vol. 9, no. 6, pp. 461–467, 2014.

[63] N. Qi, Y. Shi, X. Sun, W. Ding, and B. Yin, "Single image super-resolution via 2d sparse representation," in *IEEE International Conference on Multimedia and Expo (ICME)*, Dec. 2015, pp. 1–6.

[64] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[65] D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[66] K. Rauhut, H.and Schnass and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2210–2219, 2008.

[67] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[68] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[69] J. Murray and K. Kreutz-Delgado, "Learning sparse overcomplete codes for images," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 45, no. 1-2, pp. 97–110, 2006.

[70] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, Dec. 2006, pp. 801–808.

[71] Y. Censor and S. Zenios, *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press, 1997.

[72] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[73] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[74] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, Aug. 2003, pp. 592–599.

[75] K. Engan, S. Aase, and J. Husoy, "Multi-frame compression: Theory and design," *Signal Processing*, vol. 80, no. 10, pp. 2121–2140, 2000.

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

[77] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[78] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, pp. 40–44.

[79] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[80] J. Sun, N. Zheng, H. Tao, and H. Shum, "Image hallucination with primal sketch priors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2003, pp. 729–736.

[81] I. Selesnick and S. L., "Bivariate shrinkage functions for wavelet-based denoising

exploiting interscale dependency," *Multiscale Modeling & Simulation*, vol. 50, no. 11, pp. 2744–2756, 2002.

[82] A. Gersho and R. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012.

[83] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuits," *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[84] M. Holmes, A. Gray, and C. Isbell, "Fast svd for large-scale matrices," in *Workshop on Efficient Machine Learning*, Dec. 2007, pp. 1–2.

[85] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1991.

[86] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE 12th Computer Society Conference on Computer Vision (ICCV)*, Sep. 2009, pp. 349–356.

[87] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[88] I. Daubechies, M. Defrise, and C. De Mol, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[89] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *British Machine Vision Conference (BMVC)*, Sep. 2012, pp. 1–10.

[90] W. Zuo, L. Zhang, C. Song, D. Zhang, and H. Gao, "Gradient histogram estimation and preservation for texture enhanced image denoising," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2459–2472, 2014.

[91] R. Gonzalez, *Digital image processing*. Pearson Education India, 2005.

[92] J. Patel and C. Read, *Handbook of the normal distribution*. CRC Press, 1996.

[93] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in Neural Information Processing Systems*, Dec. 2009, pp. 1033–1041.

[94] T. Cho, C. Zitnick, N. Joshi, S. Kang, R. Szeliski, and W. Freeman, "Image restoration by matching gradient distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 683–694, 2012.

[95] T. Cho, N. Joshi, C. Zitnick, S. Kang, R. Szeliski, and W. Freeman, "A content-aware image prior," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2010, pp. 169–176.