Sparse Representation over Multiple Learned Dictionaries via the Gradient Operator Properties with Application to Single-Image Super-Resolution

Faezeh Yeganli

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical and Electronic Engineering

Eastern Mediterranean University
December 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Cem Tanova
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Prof. Dr. Hasan Demirel
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Prof. Dr. Hüseyin Özkaramanlı
Supervisor

Examining Committee

1. Prof. Dr. Gözde Bozdağı Akar ________________________________

2. Prof. Dr. Aydın Akan ________________________________

3. Prof. Dr. Hüseyin Özkaramanlı ________________________________

4. Prof. Dr. Hasan Demirel ________________________________

5. Prof. Dr. Osman Kükrer ________________________________

ABSTRACT

Single-image super-resolution is an ill-posed inverse problem that requires effective regularization. Super-resolution over learned dictionaries offers a successful framework for efficiently solving this problem exploiting the sparsity as regularizer. It is well acknowledged that the success of sparse representation comes as a direct consequence of the representation power of learned dictionaries. Along this trend, this thesis considers the problem of super-resolution via sparse representation, where representation is done over a set of compact high and low resolution cluster dictionaries. Such an approach inevitably calls for a model selection criteria both in the learning and reconstruction stages. The model selection criteria should have scale-invariance property so that link between low resolution and high resolution feature spaces is properly established.

The main contribution in this thesis is to employ two approximately scale-invariant patch measures for the classification of image patches in the learning and reconstruction stages. These are the sharpness measure and the dominant phase angle defined in terms of the magnitude and phase of the gradient operator, respectively. These measures are empirically shown to have acceptable degrees of scale-invariance. i.e. sharpness measure and the dominant phase angle do not significantly change for two consecutive resolution levels. This invariance to a large extent ensures that model selection is correct in the reconstruction stage where one only knows the low resolution patch. Three super-resolution algorithms are proposed based on selective sparse coding over cluster dictionaries with the proposed measures, applied individually and combined together. In each algorithm, training data is clustered and a coupled dictionary pairs are learned for each cluster. In the learning stage any standard coupled dictionary learning algorithm can be used. In the reconstruction stage, the most appropriate dictionary pair is selected for each low resolution patch and the sparse coding coefficients with respect to the low resolution dictionary are calculated. The link between the low and high resolution feature spaces is the fact that the sparse representation coefficients of the high and low resolution patches are approximately equal. For the case of multiple structured dictionaries this link is also strengthened since the dictionaries are learned for structured feature spaces. Imposing this link, a high resolution patch estimate is obtained by multiplying the sparse coding coefficients with the corresponding high resolution dictionary. Quantitative and qualitative experiments conducted over natural images validate that each of the proposed algorithms is superior to the standard case of using a single dictionary pair, and is competitive with the state-of-the-art super-resolution algorithms.

From the rate-distortion perspective, it is shown that computational complexity (rate) can be reduced significantly without a significant loss in quality. This is achieved due to the fact that the proposed clustering criterion lends itself nicely for identifying the patches that are un-sharp (with low frequency content). Such patches can be handled effectively using simple algorithm (computationally much less complex) such as bicubic interpolation instead of computationally expensive sparse representation. Specifically for a typical image, 73.03 % of the patches can be handled using bicubic interpolation without significant degradation in quality.

Keywords: Single image super-resolution, sparse representation, dictionary learning, sharpness measure, gradient phase angle, coupled dictionaries.

ÖZ

Tek-görüntüden süper-çözünürlüğe sahip bir görüntü elde etme problemi kötü konumlanmış bir problemdir ve etkili bir şekilde düzenlileştirilmesi gerekmektedir. Öğrenilen sözlükler üzerinden süper-çözünürlük, bu problemin etkili bir şekilde çözülmesi için seyreklik kavramından düzenleyici olarak faydalanarak başarılı bir çerçeve sunmaktadır. Seyerek temsiliyetin başarısının öğrenilen sözlüklerin temsiliyet gücünün direkt bir sonucu olduğu aşikardır. Bir eğilim doğrultusunda seyrek temsiliyet ve buna bağlı olarak öğrenilen bu tez çalışması süper-çözünürlük problemini bir dizi yüksek ve düşük çözünürlüklü küme sözlükleri kullanarak geliştirmektedir. Birden fazla sözlük kulanma yaklaşımı kaçınılmaz olarak öğrenme ve yeniden yapılandırma aşamaları olmak üzere her aşamada bir model seçme kriterini gerektirmektedir. Düşük ve yüksek çözünürlüklü öznitelik uzayları arasındaki bağlantının uygun bir şekilde sağlanması için model seçme kriterinin ölçekten bağımsız bir özelliğe sahip olması gerekmektedir.

Bu tez çalışmasının asıl katkısı öğrenme ve geri çatma (yeniden yapılandırma) safhalarında görüntü yamalarının sınıflandırılması maksadıyla iki tane ölçekten yaklaşık bağımsız kriterlerin kullanılmasıdır. Bu kriterler eğim operatörü kullanılarak tanımlanan keskinlik ölçüsü ve baskın faz açısıdır. Bahsi gereçen ölçülerin ölçekten yaklaşık olarak bağımsız oldukları kanıtlanmıştır. Bu veri iki ardışık çözünürlük seviyesindeki görüntü yamalarının hem keskinlik ölçüsü hem de baskın faz açısının önemli ölçüde değişim göstermediği anlamını taşımaktadır. Bu bağımsızlık büyük ölçüde kişinin yalnızca düşük çözünürlüklü görüntü yaması hakkında bilgi sahibi olduğu yeniden yapılandırma model seçiminin (yani yüksek çözünürlülük kümesinden hangisinin kullanılacağı) doğru olduğunu göstermektedir. Önerilen sınıflandırma ölçekleri ile birlikte küme sözlükleri üzerinde seçici seyrek kodlamaya dayalı üç süper çözünürlük algoritması önerilmiş olup her biri tek başına uygulanmış ve daha sonra birbirleri ile birleştirilerek daha sıradüzensel bir sınıflandırmaya dayalı süper-çözünürlük algoritması önerilmiştir. Her bir algoritmada, öğrenme verileri kümlenmiş olup her bir küme için o kümeye ait görüntü yamalarının öznitelik uzayları birbirine bağlantılı bir yöntem kullanılarak alçak ve yüksek çözünürlükte birer sözlük öğrenilmiştir. Öğrenme aşamasında herhangi bir standart bağlantılı sözlük öğrenme algoritması kullanılabilmektedir. Yeniden yapılandırma aşamasında her biri çözünürlüklü yama için en uygun sözlü çifti seçilmiş olup, düşük çözünürlüklü sözlük dikkate alınarak seyrek kodlama katsayıları hesaplanmıştır. Düşük ve yüksek çözünürlüklü öznitelik uzayları arasında bağlantı, düşük ve yüksek çözünürlüklü yamaların seyrek temsiliyet katsayılarının yaklaşık olarak eşit oldukları varsayımdır. Bu varsayım önerilen çoklu yapısal sınıflama yönteminden dolayı güçlenmektedir. Bu bağlantıdan yararlanılarak seyrek kodlama katsayıları ve ilgili yüksek çözünürlüklü sözlükler çarpılarak yüksek çözünürlüklü yama tahminleri elde edilmiştir. Doğal görüntüler üzerinde gerçekleştirilen ve niteliksel denemeler önerilmiş olan herhangi bir algoritmanın tek bir sözlük çiftinin kullanıldığı standart yöntemlere karşı üstünlük sağladığı ve gelişmiş süper-çözünürlük algoritmaları ile rekabet ettiğini onaylamaktadır.

Hız-bozunum teorisi bakış açısından herhangi önemli bir kalite kaybı yaşanmadan hesaplama karmaşıklığının azaltılabildiği gösterilmiştir. Bu ise önerilen kümeleme kriterinin keskin olmayan (düşük frekans içerikli) yamaları kolayca belirlenmesine dayanmaktadır. Bu tür keskin olmayan yamalar, hesaplama açısından yüksek maliyetli seyrek temsiliyet yerine bikübrik ara değer bulma gibi basit algoritmalar (hesaplama açısından karmaşıklık derecesi çok daha düşük) kullanılarak etkili bir şekilde ele alınabilmektedir. Özellikle tipik görüntü için yamaların 96%’lık oranı bikübik ara değer kullanılarak kalitede herhangi önemli bir düşüş yaşanmadan ele alınabilmektedir.

Anahtar Kelimler: Tek görüntü süper-çözünürlük, seyrek temsiliyet, çoklu sözlük öğrenimi, keskinlik ölçüsü, baskın faz açısı, bağlantılı sözlükler.

DEDICATION

Baba, it is for you

Mama, it is because of you

Faegheh, Hanieh and Sepideh, it is dedicated to you

ACKNOWLEDGMENT

I would like to thank my supervisor Prof. Dr. Hüseyin Özkaramanlı, for his willingness to support my work and his guidance throughout my studies which allowed me to develop my skills. His intention and openness gave me enormous encouragement in every stage of the preparation of this thesis.

I would also like to thank to dean of faculty of engineering Prof. Dr. Aykut Hocanin, head of electrical and electronic engineering department Prof. Dr. Hasan Demirel, department faculty members and fellow students for their help and support during my course of study. I thank Dr. Mahmoud Nazzal for the opportunity to work with him as a team member in research projects. I learned a lot from working with him.

I would like to express my love, gratitude and respect towards my parents, MirMahmoud Yeganli and Alvan Damirchy and my sisters Faegheh, Hanieh and Sepideh Yeganli for being beside me during the most important part of my life. I thank them for their love, patience, care and attention.

My warmest regards go to Assist. Prof. Dr. Suna Bolat, Narges Sabeti and all my friends for their friendship and encouragement which enriched my life throughout my studies in so many ways.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

x	A vector signals
D	A dictionary
w	Sparse coding coefficients vector
k	Number of dictionary atoms
M	Number of structured dictionaries
n	Dimension of the signal space
S	Sparsity
$\| \|_2$	Euclidean vector norm
$\| \|_0$	Number of non-zero elements in a vector
$\| \|_F$	Frobenius matrix norm
NP	Non-deterministic polynomial-time
BMP	Basis matching pursuit
MP	Matching pursuit
OMP	Orthogonal matching pursuit
ORMP	Order recursive matching pursuit
BP	Basis pursuit
FOCUSS	Focal underdetermined system solver
ODL	Online dictionary learning
DL	Dictionary learning
W	Matrix of sparse coding coefficients vector
X	Training set of vector signals

r	Sparse coding residual vector
T	The transpose operator
tr	The trace operator
ε	Vector sparse approximation error tolerance
ψ	The blurring and downsampling operator
MOD	Method of optimized directions
MAP	Maximum aposteriori
LASSO	Least absolute shrinkage and selection operator
BPD	Basis pursuit denoising
SM	Sharpness measure
DPA	Dominant phase angle
HF	High frequency
SR	Super-resolution
SISR	Single-image super-resolution
LR	Low resolution
HR	High resolution
MR	Middle resolution
LF	Low frequency
BP-JDL	Beta process joint dictionary learning
MSE	Mean-squared error
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index

Chapter 1

INTRODUCTION

1.1 Sparse Signal Representation

Sparse signal representation has received a lot of attention as a successful representation framework in many signal and image processing areas. Sparse representation using over-complete dictionaries has been employed in image denoising [1], super-resolution [2], compression [3], pattern recognition [4] and inpainting [5] over the last decades. In [6], Mallat and Zhang suggested that over-complete bases have the ability to represent a wider range of signals. Early research by Olshausen and Fieldt [7, 8] pointed out that a dictionary obtained by sparse coding can take the properties of the respective signal fields. Furthermore, they investigated sparse representation of image patches using a sparse linear combination of elements from an appropriately chosen over-complete dictionary.

Image patches can be represented with fixed linear basis functions such as Fourier basis functions, wavelets, curvelets, contourlets and the discreet cosine transform. With such basis functions, the representation coefficients can be easily calculated as the inner product of a signal and the basis functions. It is possible to enforce sparsity on this representation by thresholding the representation coefficients. However, it is noted that this way of sparse representation is not well-suited for a large set of signals [9]. A remedy is to use a redundant dictionary composed of prototype signals.

Redundancy increases the quality of a learned dictionary in representation and contains more prototype signal structures which gives a better signal approximation.

1.2 Problem Formulation

A dictionary is a matrix whose columns are derived from example signals. Dictionary atoms are initialized from a set of randomly selected signals and updated in such a way that they preserve the loyalty in representing training data and keep the representation of data sparse. Signal fitting capability is the great advantage of a learned dictionary and this is based on the fact that the atoms in a learned dictionary are conveyed from natural signal examples.

Sparse signal representation over learned dictionaries is based on the assumption that a vector signal $x \in R^n$ can be approximated as a linear combination of a few atoms in a dictionary $D \in R^{n \times K}$, where n is the dimension of the signal space and K is the dictionary atoms. This approximation can be written as $x \approx Dw$, where w is the sparse coding vector which is mainly composed of zero elements. The problem of determining w, given x and D is referred to as sparse approximation and can be formulated as

$$\arg\min_{w} \|x - Dw\|_2 \quad subject \quad to \quad \|w\|_0 < s \qquad (1.1)$$

Where S is sparsity, and the $\| \ \|_2$ and $\| \ \|_0$ operators denote the vector Euclidean norm and the number of non-zero elements in a vector, respectively.

Sparse approximation is essentially a vector selection problem. This process is shown to be computationally expensive and is known as a non-deterministic polynomial-time (NP)-hard. This computationally-complex problem can be approximately solved with zero-norm minimization via the pursuit methods such as the matching pursuit (MP) [10] and the orthogonal matching pursuit (OMP) [11] algorithms. The zero norm minimization can be replaced with a 1-norm minimization. This replacement significantly reduces the computational complexity of sparse approximation, as done in the basis pursuit (BP) [12] and the focal under-determined system solver (FOCUSS) [13] algorithms.

Learned dictionaries are customarily obtained by training over a set of example signals. This is referred to as the dictionary learning (DL) process. This process is concerned with learning dictionaries that can faithfully and sparsely represent data vectors in a given training set. Therefore, a dictionary is shown to effectively represent data over which it is trained. A most important property of sparse representation is its capability in taking the inherent signal features. Furthermore, in super-resolution (SR) these features would be invariant quantity that can be used to derive relative information about the un-known high-resolution image from the low-resolution image.

It is well-known that the variability of signals within a class is less than the variability of signals in general. Therefore, to improve the representation quality of learned dictionaries, recent research has considered clustering signals into several clusters and learning compact cluster dictionaries. This leads to a set of class dictionaries. Several works are based on employing a certain classifier to separate training signals based on a specific measure into multiple clusters or more precisely, the classification criterion applied to the problem is different. A dictionary is learned for each cluster. The same classifier is then used to classify a test signal to belong to a certain cluster. Then, the sparse representation of the signal is carried out over the cluster dictionary. The multiple dictionary setting allows for better representation at reduced computational complexity. In this thesis, the problem of multiple dictionary learning by choosing suitable clustering criteria is addressed and studied over the problem of single-image super-resolution (SISR).

1.3 Thesis Contributions

The major objective in this thesis is to devise approaches for performing DL and sparse approximation over clusters. This requires using certain signal classifiers that can effectively separate image patches based on specific criteria. Such classifiers are used to cluster training data into several clusters. Then, a dictionary is learned for each cluster over its own data. In essence, cluster data share a common feature. Therefore, the signal variability within a cluster is much less than the signal variability in the signal space. This forms the basic motivation to allow for designing compact cluster dictionaries. Learning compact cluster dictionaries meets two objectives. First, enhancing the representation quality since each dictionary is concerned with a specific signal class. Second is reducing the computational complexity of DL and sparse representation despite the usage of multiple dictionaries. In this thesis, the proposed clustered sparse representation and dictionary learning approaches are applied to the problem of SISR. This is done along the line of multiple dictionary setting. In this work the magnitude and phase of gradient operator is used as a scale-invariance criterion for classifying patches and selecting the model in the reconstruction.

Here is a brief listing of the main contributions made in this thesis:

1. Defining a sharpness measure (SM) based on the magnitude of the patch gradient operator. This measure is used to characterize spatial intensity variations of image patches and shown to be approximately invariant to the patch resolution. Therefore, it is used to separate image patches based on how sharp they are. SM is used as a classifier in the clustering stage and as a model selection criterion in the reconstruction stage. We propose a single-image super-resolution algorithm based on this sparse coding paradigm.
2. In the next contribution, we make use of the phase of the gradient operator as an approximately scale-invariance measure for classifying patches. A dominant phase angle (DPA) measure is defined based on a majority selection of the angles in the phase matrix of the patch gradient operator. This classifier is used for the clustering and model selection purposes. This idea is used in the single-image super-resolution problem.
3. The next contribution considers making use of the SM and DPA measures together for the purposes of clustering and model selection. A first clustering stage is performed using SM. Then, DPA is used as a secondary classifier to further cluster patches in each SM cluster based on their directionality. Again, this sparse coding model is tested over the single image super-resolution problem.
4. Naturally, SM is used to classify patches based on their relative sharpness. It is noted that clusters of low SM values contains patches of insignificant high frequency (HF) components. Besides, the DPA measure can separate patches of non-dominant directional structure. It is also noted that such patches have insignificant HF components. In view of these observations the fourth

contribution in this thesis considers applying the simple bicubic interpolation to super-resolve patches in these clusters. In other words, the computationally expensive sparse representation based super-resolution framework is exclusively applied to patches of significant HF components which form a minority of image patches. Therefore, this contribution serves in substantially reducing the computational complexity without sacrificing the performance.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 1 presents an introduction to dictionary learning and sparse representation and addresses the main contributions in this thesis. In Chapter 2, a brief literature review is made concerning benchmark approaches to sparse representation and DL approaching the multiple dictionary setting. Besides, super-resolution via spares representation over learned coupled dictionaries is also briefly revised. Chapter 3 introduces a super-resolution algorithm based on sparse representation over multiple cluster dictionaries. Clustering and model selection are carried out using SM. Another super-resolution algorithm based on clustering using DPA is outlined in Chapter 4. The two measures are used together to perform clustering and model selection for the purpose of super-resolution in Chapter 5. Then, Chapter 6 considers reducing the computational complexities of the proposed algorithms by making use of bicubic interpolation to super-resolve patches of insignificant HF components. In Chapter 7, conclusions and possible future works are presented.

Chapter 2

LIERATURE REVIEW

2.1 Introduction

Sparse representation assumes that a signal admits being represented as a linear combination of a few elements drawn from a dictionary. The dictionary learning process and the way it is used to achieve a sparse representation are two major processes in this field. This chapter introduces the main concepts of sparse coding and dictionary learning, summarizing the leading approaches for these processes. Besides, it reviews the approach of image super-resolution via sparse representation as a typical application of this representation.

2.2 Sparse Approximation

Sparsity has been shown as a useful property of a representation. A trivial approach to sparsity is to threshold the representation coefficients subject to a certain set of basis functions obtained with an inner product operation. Despite its simplicity, such an approach is shown to offer a poor representation quality. A better alternative is to perform a basis search process over the columns of a given dictionary. Given a vector signal $x \in R^n$ and a dictionary $D \in R^{n \times K}$, this search process is referred to as the sparse approximation process. The aim of this process is to find a sparse representation coefficient vector $w \in R^K$ that results in a loyal and sparse approximation to the original signal x. In view of the loyalty and sparsity requirements, the sparse approximation process can be mathematically posed as follows:

$$\arg\min_{w} \|w\|_0 \quad subject \quad to \quad \|x - Dw\|_2 \le \varepsilon \tag{2.1}$$

The $\| \ \|_0$ and $\| \ \|_2$ operators denote the number of non-zero elements and the Euclidean norm, respectively. The approximation is sparse when most of the coefficients of w are zero subject to a certain level of representation error tolerance ε [14]. Sparse approximation requires the availability of a dictionary D. Generally, dictionaries can be obtained in many ways, for example by scaling and translation of some basis functions like Gabor and wavelet frames or via training over example signals as will be discussed in the following sections. The advantage of employing a dictionary has been well acknowledged [15], especially when one learns redundant (over-complete) dictionaries, i.e., by setting $K > n$.

A more commonly adopted version of sparse approximation formulation is as follows.

$$\arg\min_{w} \|x - Dw\|_2 \quad subject \quad to \quad \|w\|_0 < s \tag{2.2}$$

where S is the number of non-zero elements. Only a few elements in w are allowed to be non-zero. In the following, some major approaches to sparse approximation section are addressed.

2.3 Sparse Approximation Approaches

If a dictionary is an orthogonal basis, the representation coefficients can be obtained through a simple inner product operation. Generally, the formulation in (2.2) is used for the representation purpose and the calculation of l_0 norm is a sparse approximation approach and known to be NP-hard [16]. In the literature, sparse approximation approaches are categorized into greedy algorithms and convex relaxation approaches. In greedy algorithms, the l_0 norm is minimized in an iterative manner while successively approximating the signal. The matching pursuit (MP) [17, 18], basic matching pursuit (BMP) [19], orthogonal matching pursuit (OMP) [16, 11] and order recursive matching pursuit (ORMP) algorithms [20] are examples of this category. On the other hand, convex relation approaches [21] are based on relaxing the l_0 norm minimization to minimizing the l_1 norm, giving a significant reduction to the computational complexity of the process. The focal underdetermined system solver (FOCUSS) [13, 22] and the method of frames [23] with thresholding are in this category.

2.3.1 Greedy Algorithms

Greedy sparse approximation algorithms [24] are based on providing approximate solutions to the sparse approximation problem by minimizing the l_0 norm. This norm is minimized efficiently in an iterative manner. In each iteration, a portion of the signal is represented with one atom drawn from the dictionary. The selected atom is the one that most correlates to the given signal portion. This process is continued until a stopping criterion is met. It is noted that the MP algorithm by Mallat and Zhang in [6] forms the foundation for greedy algorithms based on which many other variants and extensions were developed.

2.3.1.1 Matching Pursuit (MP)

The MP family is one of the well-known algorithms first proposed by Mallat and Zhang in [6]. This algorithm has been employed to obtain an efficient approximate solution to (2.2). For the representation of a signal x, a new atom d_i is selected from a dictionary $D=[d_1,...,d_K]$ in every iteration i. Let us denote the set of selected atoms from the first selected atom till the i-th selected one by S_i, as follows.

$$S_i = [d_1,...,d_i] \tag{2.3}$$

For solving the optimization problem in (2.2), MP iteratively calculates the following

$$\underset{d_i,w_i}{\arg\min} \left| x - [d_{i-1}|d_i] w_i \right|, \tag{2.4}$$

where $w_i = [w_1,...,w_i]$ are the corresponding coefficients for each atom d_i. The approximation to each component of x, $\hat{x}_i$ will be

$$\hat{x}_i = d_i w_i, \tag{2.5}$$

First, the MP algorithm defines a residual signal r and initializes it with the signal x as $r_0 = x$. At each iteration i, the current residual r_i is approximated by picking an atom in D that most correlates to this residual. This is done by calculating the inner product between the residual and each of the dictionary atoms as

$$w_i = d_i^T r_{i-1}, \tag{2.6}$$

Then, the atom of the largest absolute inner product is selected as follows.

$$\underset{d_i}{\arg\max} \left| d_i^T r_{i-1} \right|. \tag{2.7}$$

After atom selection, the residual is updated as follows.

$$r_i = x - \hat{x}_{i-1}. \tag{2.8}$$

In the next iteration, the same process is repeated with the newly updated residual. Iterations go until certain representation sparsity or error is met. Algorithm 1 outlines the main steps of the MP algorithm.

Algorithm 1 Matching Pursuit Algorithm (MP)

```
INPUT: $x$ is training set, $D$ is dictionary, $S$ sparsity, $\varepsilon$.
OUTPUT: $w$.
Initialization: $r \leftarrow 1$.

while $i \le S$ or $\|r\|_2 \le \varepsilon$ do
        //Atom Selection
        $w = D^T r$
        $\arg\max_{w} |w_i|$
        //Residual update
        $r \leftarrow r - w_i d_i$
        $i \leftarrow i + 1$
end while
```

2.3.1.2 Orthogonal Matching Pursuit (OMP)

As an better extension to the MP algorithm, the OMP algorithm is proposed and the same atom selection is adopted [6, 25]. Similar to MP, OMP iterates to find the best atom to represent each residual r_i. However, it differs in the way of atom selection and residual update. In OMP, the atom that maximizes the projection of r_i onto its column space is selected. This is formulated as follows.

$$\arg\max_{d_i} \left| d_i^T r_{i-1} \right|, \tag{2.9}$$

At every iteration, all the coefficients obtained for the previously selected atoms are optimized by OMP. After each iteration, the representation coefficients are calculated via least squares with respect to the chosen atoms. This is done as follows.

$$w_i = S_i^+ x \tag{2.10}$$

where S_i^+ is the Moore-Penrose pseudo-inverse of S_i. The next step is to update the residual for the next OMP iteration. To this end, the new residual is calculated by projecting the previously calculated residual onto the complement of the column space spanned by the set of selected atoms. Consequently, this means that OMP selects an atom only once. This means a better approximation in the sense that the degree of freedom of the selection process is reduced with iteration. Practically, the impact of this property is that OMP has a better approximation quality compared to MP, despite its increased computational complexity overhead. For a given signal x and dictionary D with K columns, OMP can be outlined as in the following steps [26]. It starts by setting $r_0 = x$ and $i = 0$:

I. Select the next dictionary atom by solving for $\arg\max_{d_i} \left| d_i^T r_{i-1} \right|$.

II. Update the approximation $d_i = \arg\min_{x_i} \|x - x_i\|_2^2$, where $d_i = [d_1, \ldots, d_i]$.

III. Update the residual $r = x - x_i$.

2.3.1.3 Other Greedy Algorithms

Many other algorithms offering improvement in both representation fidelity and computational complexity have been proposed along the line of the greedy algorithm family [20, 6, 25, 28, 29, 30]. The aim of these methods is to more efficiently guarantee convergence. In the signal processing literature, their setting goes by the name of MP and OMP and their several other variants and extensions.

2.3.2 Convex Relaxation Algorithms

It has been shown that minimizing the l_1 norm motivates sparsity. In view of this result, convex relaxation methods are based on relaxing the l_0 norm to minimizing the l_1 norm. This replacement has a great advantage in reducing the computational complexity of sparse approximation to be more tractable. Moreover, this reduction allows for solving the sparse approximation problem using standard optimization approaches [27].

2.3.2.1 Basis Pursuit and the Least Absolute Shrinkage and Selection Operator

The Basis pursuit (BP) algorithm considers the optimization problem in (2.2) and replaces the l_0 norm minimization with l_1 norm minimization, as follows [12].

$$\arg\min_{w} \|w\|_1 \quad subject \quad to \quad x = Dw, \qquad (2.11)$$

Generally, there are some methods to find the solution to the BP problem. Under the right conditions, these solutions can lead to a sparse solution or even the sparsest one. This is due to the fact that the l_1 norm is only concerned with values of entry not the quantity. Another similar way to BP is the least absolute shrinkage and selection operator (LASSO) algorithm proposed in [31] which is referred to as basis pursuit denoising (BPD). In LASSO, the l_1 norm is minimized like BP with some restrictions on its value. This is formulated as follows.

$$\arg\min_{w} \|x - Dw\|_2 \quad subject \quad to \quad \|w\|_1 < S, \qquad (2.12)$$

Where the sparsity is controlled by the parameter S. Finding approximation rather than just representation is allowed in LASSO and like BP, reaching the sparsest solution under the right conditions is guaranteed. In this work, the LASSO is used for solving the sparse representation problem.

2.3.2.2 Focal Underdetermined System Solver (FOCUSS)

The focal underdetermined system solver (FOCUSS) is an approximation algorithm to find a solution for (2.2) by considering minimizing the $l_p(p<1)$ norm instead of the l_0 norm [13, 22]. To obtain an exact solution, this method requires solving for the following.

$$\arg\min_{w} \|w\|_p^p \quad \textit{subject} \quad \textit{to} \quad x = Dw, \tag{2.13}$$

This algorithm is presented as a general estimation tool usable across different applications by combining desirable characteristics of both classical optimization and learning-based algorithms.

2.4 Dictionary Learning in Single Feature Space

The sparse signal representation is based on the availability of a dictionary D. The columns of the dictionary can give an approximation to a given signal x. A dictionary can be derived by defining as an analytic function or learning over a set of training signals. The standard algebraic basis functions such as Gabor and wavelet frames, Fourier basis functions, etc are in the first category. Sparse coding over such dictionaries is obtained by performing a simple inner product between the dictionary and training signal. These dictionaries are not fit to a large set of signals [14]. Learning dictionary over a set of training signals are in the second category. It has been shown that the learned dictionaries are more adaptive to structures of signal and have a better capability to signal-fitting [14].

The process of learning or training a dictionary based on some available training data such that it is well adapted to its purpose is known as dictionary learning (DL) [14]. It can be generalized that the learned dictionary should possess two characteristics. First is the loyalty of representing the data over which it is trained, and second is the sparsity of such a representation.

Given a set of training signals denoted by $X = [x_1, x_2, ..., x_M] \in R^{n \times M}$, the DL problem is to learn a dictionary $D = [d_1, d_2, ..., d_K] \in R^{n \times K}$ which represents the training examples with coefficients $W \in R^{K \times M}$. Using the cost function $f(.)$, the DL problem can be formulated as the following minimization problem.

$$\underset{D,W}{\arg\min}\ f(D,W) = \underset{D,W}{\arg\min} \|X - DW\|_F^2 , \tag{2.14}$$

There is imposed sparseness criterion on W. Accordingly, the DL process can be conducted as a successive alternation between two stages which leads to optimal solution. The first stage is a sparse approximation stage where D is unaltered. The second one is a dictionary update stage with the calculated sparse approximation coefficients. To solve the second stage of the DL process (updating the dictionary while sparse coding coefficients are fixed), several approaches have been proposed. There are other different methods of finding the sparse coding coefficients and updating the dictionary. The earliest dictionary training attempt was proposed by Olshausen and Fieldt in 32. They estimated an optimal dictionary using maximumlikelihood estimation. A Gaussian or Laplace prior on the sparse representation coefficients is assumed during optimal dictionary estimation. The steepest descent method is employed for updating the sparse coefficients and the dictionary. In recent years, several other DL algorithms have been proposed [14, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. Here is a brief revision of some benchmark DL approaches.

2.4.1 The Method of Optimized Directions (MOD)

Engan et al.in [44, 45] proposed the method of optimal directions (MOD) as a frame design technique for the use with vector selection algorithms such as MP. In this setting, Engan et al. formulated the dictionary update problem ($\arg\min_{D} \|X - Dw\|_F^2$) as a least squares (LS) problem of solving for a under-determined set of variables with a given set of equations. A LS solution to this problem can be obtained algebraically using the pseudo-inverse solution.

Algorithm 2 MOD Dictionary Learning Algorithm

```
INPUT: $X \in R^{n \times M}$ is training set, $D^0$ is dictionary, $S$ sparsity, $Num$ is number of iterations.
OUTPUT: $D$ and $w$.
Initialization: $D \leftarrow D^0$ and $i \leftarrow 1$.
while $i \le num$ do
        for $j = 1$ to $M$ do
            set $W_j \leftarrow \arg\min_{w_j} \|w_j\|_0 \quad subject \quad to \quad Dw_j = X_j$ // Sparse Coding
        end for
        $D \leftarrow XW^+$ // updating the dictionary
        $D \leftarrow XE$ // normalizing the dictionary
        $i \leftarrow i + 1$
end while
```

Similar to other DL algorithms, the MOD algorithm alternates between this prescribed dictionary update stage and a sparse approximation stage. MOD is shown to give a locally optimal solution to the DL problem. Algorithm 2 explains the main steps of the MOD algorithm.

2.4.2 Recursive Least Squares Dictionary Learning (RLS-DLA)

In the recursive least squares dictionary learning algorithm (RLS-DLA) [14], a dictionary is continuously updated as each new training vector is processed. A forgetting factor is introduced and adjusted in an appropriate way that makes the algorithm less dependent on the initial status. This improves both the convergence properties of RLS-DLA as well as the representation ability of the resulting dictionary. One of the advantage of RLS-DLA is that it leads to a dictionary that can be expected to be general for a given employed signal class, and not solely to the particular (small) training set used in its training [14]. Algorithm 3 outlines the main steps of the RLS-DLA algorithm.

2.4.3 K-SVD Dictionary Learning

The K-SVD algorithm proposed by Aharon et al. in [15] uses a singular value decomposition approach for creating a dictionary for sparse representation which generalizes the K-means clustering algorithm. The K-SVD algorithm updates a dictionary based on minimizing the objective function $C(D,W) = \|X - DW\|_F^T$. It can be re-written as follows,

$$C(D,W) = \left\| X - \sum_{k=1}^{K} D_k W_k \right\|_F^T = \left\| X - \sum_{k' \neq k} D'_k W'_k - D_k W_k \right\|_F^2 \tag{2.15}$$

Algorithm 3 Recursive least squares dictionary learning algorithm

INPUT: $x \in R^n$, D_0 (an initial dictionary), C_0 (an initial C matrix, possibly the identity matrix), $0 \le \lambda \le 1$ (forgetting factor).
OUTPUT: D (learned dictionary)

```
1. Get the new training vector $x_i$
2. Find $w_i$, typically by using $D_{i-1}$ and a vector selection algorithm.
3. Find the representation error $r_i = D_{i-1} w_i$.
4. Apply $\lambda_i$ by setting $C^*_{i-1} = \lambda_i^{-1} C_{i-1}$.
5. Calculate the vector $u = C^*_{i-1} w_i$ and if step 9 is done $v = D^T_{i-1} r$.
6. Calculate the scalar $\alpha = 1/(1 + w_i^T u)$.
7. Update the dictionary $D_i = D_{i-1} + \alpha r u^T$.
8. Update the $C$-matrix for the next iteration, $C_i = C^*_{i-1} - \alpha u u^T$.
9. If needed update the matrix $D_i D_i^T$.
10. If wanted normalized the dictionary.
Return $D$
```

Where the formula (2.15) is sum of rank-one matrices. E_k is a partial residual matrix and can be defined as $E_k = X - \sum_{k' \neq k} D'_k W'_k$. In order to minimize C, a dictionary atom D_k and the sparse approximation coefficients W_k can be updated jointly.This is easily achieved by calculating the best rank-one approximation to E_k. In essence, this rank-one approximation comes as a result of the outer product between the updated dictionary atom and the updated sparse approximation coefficient. Thus, K-SVD algorithm updates the dictionary D one atom at a time. This method is summarized as follows.

I. For each atom D_k in a dictionary, specify the locations of the set of training vectors that use the atom in their sparse approximation coefficient vectors W, labeled as (Λ_k).

II. Calculate a so-called partial residual matrix restricting to have the active set of training signal that use that particular atom as its columns.

III. Using the solution of the best rank-one approximation of the matrix E_k, update the atom D_k and the coefficients $W_{\Lambda_k}^{k}$. SVD (singular value decomposition) can be used to directly obtain that solution.

Hence, during the dictionary update step the support of the sparse approximation coefficients should not be modified, E_k and its rank-one approximation are restricted to the corresponding columns of the signals which employ the k-th atom in their sparse approximation. This is, the indices corresponding to the non-zero elements of the vector W_k. Algorithm 4 summarizes the main steps of the K-SVD algorithm.

2.4.4 Online Dictionary Learning (ODL)

In order to minimize the cost function of (2.2) under some constraints, most dictionary learning algorithms have accessed the whole training set at each iteration. However, they cannot handle very large training sets effectively. Accordingly, Mairal et al. in [9, 40] proposed a new online dictionary learning algorithm (ODL), based on stochastic approximations that process one element or a small subset (batch) of the training set at a time. It is noted that the authors in [9, 40] gave a proof of convergence of the ODL algorithm.

Algorithm 4 K-SVD Dictionary learning algorithm

```
INPUT: $X \in R^{n\times M}$ is training set, $D^0$ is initial dictionary, $S$ is sparsity, $Num$ is number of iteration.
OUTPUT: $D$, $W$.
Initialization: $D \leftarrow D^0$, $i \leftarrow 1$.
while $i \le Num$ do
        for $k = 1$ to $K$ do
          set $\Lambda_k \leftarrow i \subseteq 1,2,m \quad subject \quad to \quad W_{k,i} \neq 0$
          set $E_k \leftarrow [X - \sum_{k' \neq k} D'_k W'_k ]\Lambda_k$
          $[U, \sum \ ,V] \leftarrow SVD(E_k)$
          $D_k \leftarrow u_1$
          $W_{\Lambda_k} \leftarrow \sigma_{1,1} v_1^T$
          end for
          $D \leftarrow DE$ // Normalizing the dictionary
          $i \leftarrow i + 1$
end while
```

Considering a training set composed of i.i.d (independent, identically distributed) samples of a distribution $p(x)$ where each training vector is drawn individually, ODL calculates an updated dictionary by minimizing the objective function in (2.16). Using the steps outlined in Algorithm 5, ODL updates each column of the dictionary sequentially using the procedure presented in Algorithm 6.

$$\hat{f}_t(D) \cong \frac{1}{n} \sum_{i=1}^{t} \frac{1}{2} \|x_i - Dw_i\|_2^2 + \lambda \|w_i\|_1, \tag{2.16}$$

2.5 Dictionary Learning in Coupled Feature Spaces

There is coupled feature space in many signal processing problems for example the high-resolution (HR) and low-resolution (LR) patch space in patch-based image SR [46]. Intuitively, a single sparse coding model is used for learning dictionaries in

Algorithm 5 Online Dictionary Learning

```
INPUT: $x \in R^n \approx p(x)$, $\lambda \in R$ , $D_0 \in R^{n \times K}$ and, $Num$( number of iteration).
OUTPUT: $D_T$ (learned dictionary).
Initialization: $A_0 \leftarrow 0$, $B_0 \leftarrow 0$ (reset a past information).
for $t = 1$ to $Num$ do
Draw $x_t$ from $p(x)$
Calculate $w_t \cong \arg\min_{w \in R^K} \frac{1}{2}\|x_t - D_{t-1}w\|_2^2 + \lambda\|w\|_1$ I
$A_t \leftarrow A_{t-1} + w_t w_t^T$ and $B_t \leftarrow B_{t-1} + x_t w_t^T$
Compute $D_t$ using Algorithm 5, with $D_{t-1}$ as warm restart, so that
$D_t \cong \arg\min_{D \in C} \frac{1}{T}\sum_{i=1}^{t}\frac{1}{2}\|x_i - Dw_i\|_2^2 + \lambda\|w\|_1 = \arg\min_{D \in C} \frac{1}{T}(\frac{1}{2}Tr(D^T DA_t) - Tr(D^T B_t))$ II
end for
return D
```

Algorithm 6 Dictionary update

```
INPUT: $D = [d_1,...,d_K] \in R^{n \times K}$ (input dictionary)
$A = [a_1,...,a_K] \in R^{K \times K} = \sum_{i=1}^{t} w_i w_i^T$
$B = [b_1,...,b_K] \in R^{n \times K} = \sum_{i=1}^{t} x_i w_i^T$
repeat
Initialization: $D \leftarrow D^0$, $i \leftarrow 1$.
for $j = 1$ to $K$ do
  Update the $j$-th column to optimize for II
  $u_j \leftarrow \frac{1}{A_{jj}}(b_j - Da_j) + d_j$
  $d_j = \frac{1}{\arg\max_j(\|u_j\|_2,1)} u_j$
end for
until convergence
Return D (updated dictionary)
```

coupled feature space [2]. These dictionaries couldn't be able to capture the complex, spatial-variant and non-linear relationship between the two feature spaces.

To solve this problem several algorithms have been proposed. A two-step learning problem is proposed by Zeyde et al.[47]. In this method one dictionary is learned by K-SVD [15] and the other one is obtained via least square. The dictionaries are learned individually but the same sparse coding coefficients are used for the two feature spaces. A semi-coupled training model is proposed by Wang et al.[48] to solve the problem where a mapping matrix is used to capture the relationship of the sparse representations between spaces. The learned dictionaries can better minimize the error in both spaces than those learned in concatenated spaces, but the corresponding relationship of dictionaries in two feature spaces are not captured during the learning progress. Yang et al.in [49] proposed the bilevel optimization solution to moves one of the optimization problem to the regularization term of the other problem instead of solving two optimization problem in two feature spaces in [50]. The same sparse coding is used for both feature spaces. A beta process joint dictionary learning (BP-JDL) is provided by He et al. in [46] to customize the problem of learning dictionaries in coupled feature spaces. Furthermore, this method adds more consistent and accurate mapping between the two feature spaces. In this method the sparse representations can be decomposed to values and dictionary atom indicators but they have different priors in two feature spaces. Therefore, sparse representation that corresponds to the same dictionary atoms with the same sparsity is learned and has different values in coupled feature spaces. In this work the method proposed by Yang et al.in [50] is employed to learned the coupled dictionaries.

2.5.1 Sparse Representation over Multiple Dictionaries

It is well-known that the success of sparse representation came as a result of employing redundant learned dictionaries. The representation power of learned dictionaries depends on its redundancy. Generally speaking, more redundancy means more possible atoms to be used for a signal approximation. However, redundancy cannot be arbitrarily increased because of two concerns. First concern is the fact that it increases the computational complexity and the second concern is the associated instabilities and degradation in the sparse approximation process [27]. In view of these observations, recent research has considered a setting where multiple class dictionaries are used instead of a highly redundant one. This multiple dictionary setting is based on the fact that each signal class has certain properties in common to all its signals. This allows for designing compact class dictionaries. The advantage of this setting lies in allowing for high representation quality at reduced redundancy levels. Aside from the computational complexity and stability concerns, this setting allows for reducing the degree of freedom for sparse approximation as compact dictionaries are used.

Example works along the multiple dictionary setting include the work of Dong et al. [51] who used K-means clustering to divide the training data into a number of clusters, and learned compact cluster dictionaries. They then adaptively select the most relevant cluster dictionary for a given signal. Feng et al. [52] split the signal space into subspaces using K-space clustering and extract shared bases in these subspaces to form a dictionary. It is assumed that a testing patch belong to any of the designed subspaces. Another example is the work of Yu et al. [53] where they used a family of orthogonal learned basis functions of different characteristics. Each basis is concerned with a specific signal structure. They applied a corresponding structural sparse model selection over the structural dictionary. Another more recent work by Yang et al. [54] considers using information in the gradient operator to cluster signals into geometric clusters. They have learned geometric dictionaries over data in these clusters.

The work conducted in this thesis comes along the line of multiple dictionary setting. The information from the magnitude and phase of the gradient operator is used as criteria for a signal classification. Such classifiers are used for clustering patches in training stage and model selection in the reconstruction stage. This work considers super-resolution as practical application.

2.6 Single Image Super-Resolution

Single image super-resolution (SR) is an ill-posed inverse problem of obtaining a HR image from a LR one. It is customary to model the relationship between a HR image I_H and its LR counterpart I_L by an assumed blurring and downsampling operation, as described in (2.17).

$$I_L \approx \Psi I_H , \tag{2.17}$$

Where Ψ is the blurring and downsampling operator. In this context, estimating I_H is referred to as SR. Several LR images are required to reconstruct a HR one in conventional SR methods. Recently, sparse representation is successfully employed in SISR approaches to enhance the quality of image reconstruction. Several approaches to SR exist in the literature. These can be classified into three main categories [55].

I. **Interpolation based methods:** The first category includes interpolation-based methods [55, 56, 57] which aim at estimating the unknown pixel values by interpolation. They are shown to blur the high frequency contents in the HR estimate. However, interpolation by exploiting natural image features is shown to perform better, particularly in preserving the edges. Nevertheless, such techniques have a limited capability in handling the visual complexity of natural images, especially for fine textures and smooth shades.

II. **Reconstruction based methods:** The second category is reconstruction-based methods [58, 59, 60, 61]. The key idea behind these methods is to apply a reconstruction constraint on the estimated HR image. This constraint enforces similarity between a blurred and downsampled version of the HR image and the LR input image. Still, such methods are shown to produce jaggy or ringing artifacts around edges because of the lack of regularization.

III. **Learning based methods:** The third category is the learning-based methods [62, 63, 64, 65, 66, 67, 68, 69]. These methods use a training stage and a testing stage. They are based on utilizing the correspondence between the LR and HR image patches as a natural image prior. This is carried out by assuming a similarity between training and testing sets of signals. One of the most successful learning approaches is the sparse representation-based approach.

2.6.1 Single-Image Super-Resolution via Sparse Representation

Dictionary learning techniques have been used by many methods in SISR to capture the concurrent prior between the LR and HR patches [2, 49, 50, 64, 65]. In all these methods two dictionaries in both LR and HR feature spaces are learned and HR patch is normally recovered from the HR dictionary and the sparse coding coefficients calculated from LR patch and LR dictionary. The sparse representation framework for super-resolution proposed by Yang et al.[2, 50] is based on reconstructing HR image patches from their counterparts in the LR image. This reconstruction is based on employing two constraints. First, a reconstruction constraint, which enforces the SR outcome to belong to the solution space of (2.17). Second, is a sparsity prior which is based on assuming that the sparse coding coefficients of LR image patches are identical to those of their counterparts in the HR image. For this purpose, a large database of HR and LR image patches corresponding to the same scene is used in [2, 50]. This approach allows for superior results. Still, it requires a long execution time. Then, Yang et al.employed a pair of over-complete coupled dictionaries in both the LR and HR resolution levels for the purpose of sparse coding [50]. These two dictionaries are learned over a set of HR and LR patch pairs in such a way that imposes the equality of the sparse coding coefficients of HR and LR patches. In this thesis, Yang's method has been used for obtaining LR and HR dictionaries and sparse coding coefficients. Furthermore, the SR algorithms in this thesis come along Yang's method as explained in the next section.

Recently, several methods went beyond the invariance assumption for improving the stability of the recovery [46, 48, 70]. He et al. [46], proposed Bayesian method using a beta process to learn the over-complete dictionaries. Due to unique property of beta process learned dictionaries are more consistent and have accurate mapping between two feature spaces. In [48], Wang et al. used linear mapping between the LR and HR sparse coding coefficients combined with l_1 sparsity to learning. Peleg et al. suggested statistical prediction model based on sparse representation of LR and HR image patches in [70] which allows the method to avoid any invariance assumption. They used MMSE to predict the HR patches and interpretation of a feedforward neural network to decrease a complexity of their model.

2.6.2 The First Approach to Single-Image Super-Resolution via Sparse Representation

Suppose that a HR image (I_H) is divided into patches. Then, these patches are reshaped into one-dimensional (1-D) vector signals where they are combined column-wise to form an array of vector patches x_H. The patch array x_H can be sparsely represented over a HR dictionary D_H as follows

$$x_H \approx D_H w_H, \tag{2.18}$$

where w_H denotes the array of sparse representation coefficient vectors of x_H. The representation coefficients can be obtained by vector selection algorithms such as OMP [11] or LASSO [31]. Let x_L denote an array composed of reshaped patches of the corresponding LR image of the same scene (I_L). Similarly, x_L can be sparsely represented over a LR dictionary D_L as follows

$$x_L \approx D_L w_L, \tag{2.19}$$

where w_L is the array of sparse representation coefficient vectors of x_L. The same blurring and downsampling operator Ψ shown in (2.17) can be used to relate x_L and x_H ($x_L \approx \Psi x_H$).

If D_L and D_H are learned in a coupled manner, it can be further assumed that Ψ relates the atoms of the two dictionaries, i.e., $D_L \approx \Psi D_H$. In view of the above assumptions, one may write

$$x_L \approx \Psi x_H \approx \Psi D_H w_H \approx D_L w_H \tag{2.20}$$

In view of (2.20), it is concluded that $w_H \approx w_L$.

The above ideas lay the foundation for the HR patch reconstruction stage. Let x_H^i and x_L^i denotes the i-th patch in x_H and x_L, respectively. To this end, x_H^i can be reconstructed with the availability of D_H and the sparse coding coefficient of x_L^i over D_L, denoted by w_L^i, as follows

$$x_H^i \approx D_H w_L^i \tag{2.21}$$

It is advantageous to allow a certain overlap between patches to contribute to better local consistency in the reconstructed HR image patches [50]. Then, each LR patch is used to reconstruct its HR counterpart in the HR image according to (2.21). Reconstructed HR patches are then reshaped into the two-dimensional (2-D) form and merged together to form a HR image estimate.

Chapter 3

IMAGE SUPER-RESOLUTION VIASPARSE REPRESENTATION OVER MULTIPLE LEARNED DICTIONARIES BASED ON EDGE SHARPNESS

3.1 Introduction

Most SR algorithms fail to reconstruct the salient image features like edges, corners and texture. However, representation of these features which contain high gradient magnitudes is crucial for visual improvement. An important discriminating property for image patches is the sharpness measure (SM) which is defined via the gradient operator [71]. For taking advantage of this important fact, designing a set of structured coupled LR and HR dictionary pairs is proposed [72, 73]. For this purpose, training data is clustered into a number of clusters based on SM and a pair of coupled LR and HR dictionaries is trained over the training data in each clusters. In the reconstruction stage, the SM value of each LR patch is used to identify its cluster. Then, for reconstructing the HR patch, the dictionary pair of each identified cluster is employed. This classification serves for designing dictionary pairs that are well suited to represent image features with various sharpness levels.

3.2 Approximate Scale-Invariance of the Image Patch Sharpness Measure

SM can be defined via the magnitude of the gradient operator as a numerical measure that can quantify spatial intensity variations of image patches. The SM value of each image patch is defined [74] as

$$SM = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sqrt{\left|G_i^h\right| + \left|G_j^v\right|} \tag{3.1}$$

where N_1, N_2 denote the patch horizontal and vertical dimensions and G^h, G^v denote its horizontal and vertical gradients respectively. In this context SM can be effectively employed as a criterion to classify edges, corners and texture in an image based on how sharp they are.

Sun et al. in [55], defined the gradient profile prior, and studied its behavior with respect to image scale. They reported that the edge sharpness of natural images follows a certain distribution which is independent of image resolution and applied this finding as a prior to the problem of SISR. In this work, SM defined via the gradient operator is used as an approximately scale-invariant quantity for a pair of LR and HR patches coming from the same scene. For image patches that contain strong edges, corners and texture, the invariance of SM is strong.

To investigate the impact of scale on SM, the following experiment is conducted on each of the images shown in Fig. 3.1.The following experiments are carried out to investigate the effect of scale on SM. A HR image is divided into non-overlapping 6x6 patches. Each HR patch is filtered by a bicubic kernel and downsampled by a scale factor of 2 to obtain its 3x3 LR counterpart. Figure 3.2 shows the histograms of the SM values for patches of several HR images (top) and their LR counterparts (bottom). In view of Fig. 3.2, it can be seen that the two histograms in both resolution levels are similar. For the Text Image 1 and ppt3 images, a significant number of sharp patch pairs exist, where for the Barbara and Building Image 1 images, there are few sharp patch pairs.

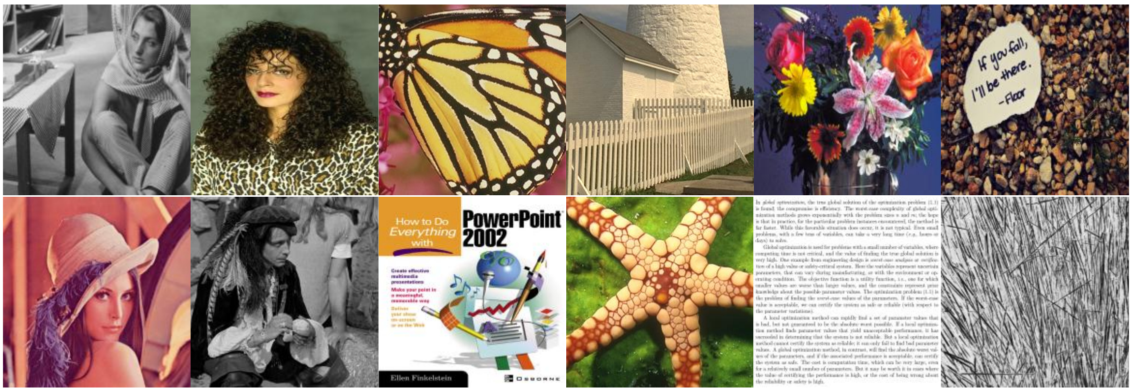


Figure 3.1: Test images from left to right and top to bottom: Barbara, BSDS 198054, Butterfly, Fence, Flowers, Input 6, Lena, Man, ppt3, Starfish, Text Image 1 and Texture.

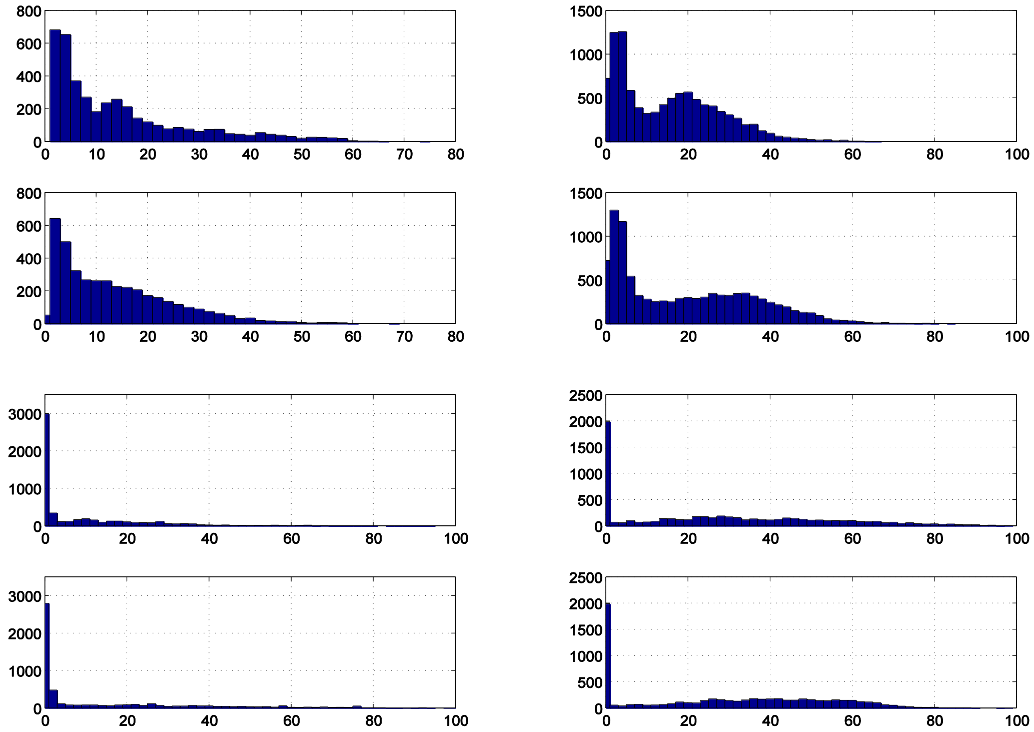


Figure 3.2: Histogram of SM values for HR patches (top), and LR patches (bottom) for images from left to right and top to bottom, Barbara, Building Image 1, ppt3, and Text Image 1.

The statistical distributions of the SM values of LR and HR patches suggest that the SM criterion is approximately scale-invariant.

To further investigate the impact of scale on SM, a similar experiment is conducted on the image set shown in Fig.3.1. Again, each image is divided into non-overlapping 6x6 patches. 3x3 LR patches are obtained as specified earlier. Seven clusters denoted by C_1 through C_7 are defined to correspond to SM intervals of [0, 4], [4, 8], [8, 12], [12, 16], [16, 20], [20, 24] and [24, 255], respectively. The bounds of these intervals are empirically selected such that the last interval contains high SM values and the remaining SM range is uniformly split into the first six intervals.

For each case, the SM values of all LR and HR patches of an image are calculated. Each patch is classified into one of the clusters based on its SM value. The total number of HR patches classified into a given cluster is counted. Then, the number of their LR counterparts that are correctly classified into the same cluster is counted. The SM invariance ratio is defined as the ratio between the number of correctly classified LR patches and the total number of HR patches in a cluster. For each test image, Table 3.1 shows the total number of HR patches classified into each cluster (top) and the corresponding SM invariance ratio (bottom). Considering the Butterfly image, as an example, Table 3.1 indicates that there are 558 HR patches classified into C_1 based on their SM values. 86.0 % of their LR counterparts are correctly classified into the same cluster. In other words, SM for the HR and LR patches is approximately scale-invariant for C_1 in both cases.

In view of Table 3.1, one notices that SM is strongly scale-invariant particularly for the first (unsharp) and the last (sharpest) clusters, rather than the intermediate clusters. The average scale-invariance ratio is 89.6 % in C_1 and 74 % in C_7.

Table 3.1: Number of HR patches in each cluster (top) and the percentage of the corresponding LR patches correctly classified into the same cluster (bottom) via the SM criterion. The largest number of patches in a cluster with the corresponding percentage is in bold face.

Image	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Barbara	**1934**	1406	759	714	556	401	1626
	80.9	45.0	20.9	19.0	14.2	8.5	18.6
BSDS 198054	1039	367	651	557	368	235	**1157**
	96.6	60.5	46.9	34.1	28.3	15.3	**81.0**
Butterfly	**558**	213	146	142	140	164	486
	86.0	47.9	24.7	13.4	7.9	6.1	89.1
Fence	387	211	224	196	169	188	**474**
	94.6	50.7	30.4	25.5	17.8	13.8	**46.0**
Flowers	**1420**	1144	946	630	324	231	429
	87.0	41.3	28.9	17.6	13.9	10.4	70.6
Input6	1842	1454	1723	1776	1659	1348	**2124**
	90.7	51.2	37.6	25.8	19.7	11.8	**62.4**
Lena	**3145**	2145	797	488	364	224	233
	90.7	49.5	21.5	14.8	11.5	10.7	70.4
Man	1237	**2210**	1357	828	641	425	698
	88.0	**54.1**	34.8	29.1	18.3	13.9	43.0
Starfish	310	**504**	278	186	169	163	239
	69.4	**43.3**	24.5	8.1	4.7	8.0	61.1
TextImage1	4965	134	164	213	202	201	**5671**
	85.3	45.5	37.8	18.3	20.3	32.8	**78.1**
Texture	16	290	313	334	383	422	**9691**
	62.5	55.5	39.3	29.0	12.3	13.0	**84.3**
ppt3	**6574**	340	340	393	371	308	1464
	95.5	37.4	17.1	5.6	10.8	2.9	84.8
Average	89.6	49.0	31.7	22.5	16.6	11.9	74.0

However, SM is moderately scale-invariant for the intermediate clusters C_2 through C_6. One can also observe in Table 3.1 that the SM scale-invariance degrades with increasing number of clusters. This observation is valid for almost all images considered in Table 3.1. For sharp images, one observes that most of the patches fall in last cluster for which SM is strongly scale-invariant. Considering Text Image 1 as an example of sharp images, 78.1 % of patches in C_7 have their LR counterparts with SM values falling into the same cluster. Thus, one can predict with good accuracy what cluster a HR patch belongs to, given its LR counterparts.

The above conclusions imply that SM can be effectively used as a criterion for selecting the model (dictionary pair), especially for the last cluster that corresponds to the sharpest patches in an image. Image patches with high sharpness values are those which contain edges, corners and texture. In other words, they are patches of high frequencies which are the most difficult image regions to reconstruct. This observation forms the basis for potential improvement in HR image reconstruction, depending on the availability of cluster dictionary pairs that can effectively represent signals in their respective clusters.

3.3 Clustering and Sparse Model Selection with Patch Sharpness Measure

The proposed algorithm is composed of two stages of training and reconstruction. In the first stage, a set of dictionary pairs is prepared and in the second stage the best dictionary pair is selected to sparsely reconstruct HR patches from the corresponding LR patches. For the training stage, a set of HR images is required. A LR version of each HR image is obtained by filtering it with a bicubic kernel and then downsampling it by a scale factor of 2. Each LR image is then interpolated by a scale factor of 2 to the dimensions of the corresponding HR image and they are said to be in the middle resolution (MR) level. As done in [50], for each input LR patch x_L the sparse representation coefficients are obtained with respect to D_L and the corresponding HR dictionary D_H will be combined to these coefficients to obtain the output HR patch x_H. The sparse representation problem can be formulated as follows.

$$\underset{w}{\arg\min} = \|FD_L W - Fx_L\|_2^2 + \lambda\|w\|_1 \tag{3.2}$$

Where F is a feature extraction operator and the sparsity of solution is balanced by the parameter λ. As shown in Chapter 2, this l_1 linear regularization is known as LASSO. F is used to ensure that the computed coefficients fit the most relevant part of the LR image and have a more accurate prediction for the HR image patch reconstruction. Since the HF components of the HR image are important, the HF contents of the LR image are important for predicting the HF content that has been lost [50]. Typically high-pass filter is chosen for extracting features. In this work, as done in [50], the first-order and second-order derivatives are used as the features for LR patches. Four 1-D filters are used to extract the derivatives. These filters are as follows.

$$f_1 = [-1,0,1], \quad f_2 = f_1^T, \quad f_3 = [1,0,-2,0,1], \quad f_4 = f_3^T. \tag{3.3}$$

where T denotes transpose. By applying these four 1-D filters (feature extraction operator) four gradient maps are obtained and four patches are extracted from these four patches. Then these four patched are concatenated to become the feature vector. Based on [50], it is better to obtain features from MR image. LR and HR patches corresponding to the same spatial location are handled as pairs. Each LR patch is then classified into a specific cluster based on its SM value. The HR patch in each pair is placed into the same cluster. LR and HR patches of each cluster are used to train for a pair of coupled LR and HR dictionaries, respectively. For this purpose, the method proposed in [50] is used. Algorithm 7 outlines the main steps of the training phase.

Algorithm 7 The Proposed Cluster DL Algorithm

INPUT: HR training image set, number of clusters
OUTPUT: A set of dictionary pairs.

```
Divide each HR image into patches and subtract the mean value of each patch.
Reshape patches into vectors and combine them column-wise to form a HR training array.
Blur (bicubic kernel) and downsample each HR image to generate a LR image.
Divide each LR image into patches.
Upsample each LR image to the MR level.
Apply feature extraction filters on each MR image.
Divide the extracted features into patches and reshape them into column vectors.
Combine the features column-wise to form the LR training array.
for Each patch in the LR training array,
    Calculate the SM value of the corresponding patch in the LR image, and find the cluster number.
    Add the MR patch to the LR training set of this cluster.
    Add the corresponding HR patch to the HR training set of this cluster.
end for
For each cluster, learn a pair of coupled dictionaries.
```

In the reconstruction stage, first a LR image is upsampled using bicubic interpolation to the MR level. Then, by applying feature extraction filters features are extracted and divided into patches which are reshaped into the vector form. To assure local consistency between the reconstructed patches, a certain patch overlap (for a patch size of 5×5, 4-patch overlap) is allowed [50]. The SM value of each LR patch is calculated and which cluster it belongs to is identified. Employing the dictionary pair of the identified cluster, the sparse representation coefficient vector of the corresponding MR patch over the cluster LR dictionary is calculated. Then by right-multiplying the cluster dictionary with the sparse representation coefficients of the MR patch the HR patch is reconstructed and for all LR patches the same procedure is repeated. Finally, the reconstructed HR patches are reshaped into the 2-D form and merged to constitute the HR image estimate. In this merging, each pixel value is obtained from the average of its values in the reconstructed patches that contain it. Algorithm 8 outlines a summary of the proposed reconstruction procedure.

Algorithm 8 The Proposed Single-Image Super-resolution Algorithm.

INPUT: A LR test image, cluster dictionary pairs.
OUTPUT: A HR image estimate

```
Divide the LR image into overlapping patches.
Upsample the LR image to the required resolution level (MR).
Apply feature extraction filters on the MR image.
Divide the extracted features into overlapping patches and reshape them into vectors.
for Each LR patch do
    Calculate the SM of the LR patch.
    Determine the cluster this patch belongs to.
    Sparsely code the features of the corresponding MR patch over the cluster LR dictionary.
    Reconstruct the corresponding HR patch by right-multiplying the HR dictionary of the same cluster with the sparse codes of the MR features.
end for
Merge overlapping patches to obtain a HR image estimate.
```

It is noted that the computational complexity of image reconstruction in the proposed algorithm is similar to that of the algorithm of Yang et al. [50]. Despite the use of multiple dictionary pairs, only one pair is selected for each image patch. It is well-known that the most computationally complex stage in the reconstruction process is the sparse coding stage. Since the proposed algorithm uses dictionaries of the same size, the sparse coding computational complexity is the same as the case of using a single dictionary pair. Overall, the dictionary pair selection process (model selection) requires calculating the SM value of each LR patch. This calculation adds a bit more complexity for the proposed algorithm as compared to the algorithm of Yang et al. [50].

3.4 Experimental Validation

In this section, the performance of the proposed algorithm is examined and compared to three leading super-resolution algorithms proposed by Zeyde et al. [47], Yang et al. [50] and He et al. [46]. These algorithms used in the comparison are different in nature. In order to have fair comparisons, care has been taken to ensure that the parameters used in the training and testing stages are as close to each other for all algorithms as possible. If a parameter is unique to a specific algorithm, the value suggested by the authors is used. Image super-resolution results for a scale factor of 2 are presented. The proposed algorithm can easily be modified for other scale factors. However, this requires studying the scale-invariance property of SM with the increased scale factor. Accordingly, the SM intervals can be set. This invariance seems to be the only limiting factor for extending the proposed setting to work with larger scale factors.

A LR patch size of 5×5, with a 4-pixel patch overlap is used in the proposed algorithm and the algorithm of Yang et al. [50] and a patch size of 3×3 with a 2-pixel patch overlap as default values provided by the authors is used in the algorithm of Zeyde et al. [47]. A patch size of 7×7 with a 6-pixel overlap is used in the algorithm of He et al. [46] as suggested by authors.

A LR image is obtained by applying a bicubic filtering operation on its HR counterpart, and then downsampling the filtered image by 2 in both dimensions. Test images include some well-known benchmark natural images which were used in [70, 75]. Several other images have also been selected from different datasets [76, 77, 78] because of their rich high frequency contents. All test images are shown in Fig. 3.1.

Dictionaries of the proposed algorithm are learned in a coupled manner as specified in [50]. Dictionary training for the proposed algorithm is done over the 1000-image Flickr dataset [79], along with several typical text images as a source of high sharpness patches. Some of images in Flickr dataset are chosen and shown in Fig. 3.3 and added text images are shown in Fig 3.4.

Figure 3.3: Selected images from Flickr dataset.

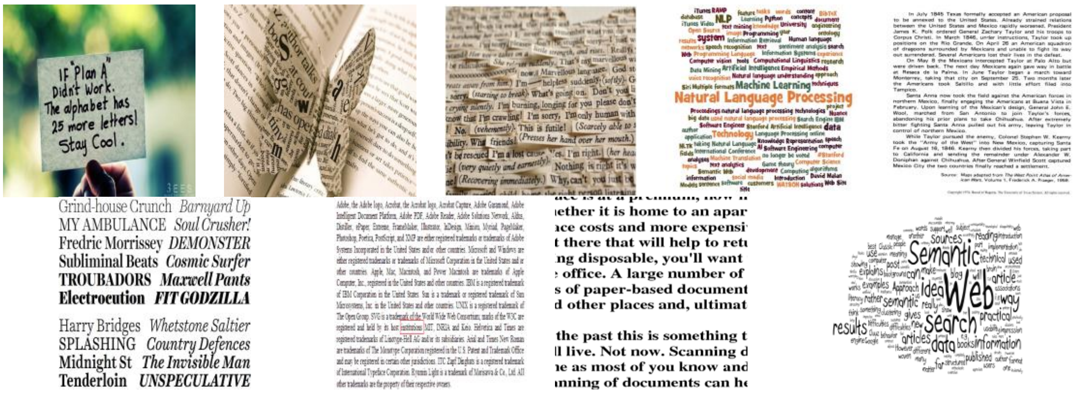

If "Plan A" Didn't Work. The alphabet has 25 more letters! Stay Cool.
Natural Language Processing
Grind-house Crunch Barnyard Up
MY AMBULANCE Soul Crusher!
Fredric Morrissey DEMONSTER
Subliminal Beats Cosmic Surfer
TROUBADORS Maxwell Pants
Electrocution FIT GODZILLA
Harry Bridges Whetstone Saltier
SPLASHING Country Defences
Midnight St The Invisible Man
Tenderloin UNSPECULATIVE
Semantic Web

Figure 3.4: Additional text images added to the Flickr dataset for the training of the proposed algorithm and the algorithm of Yang et al.

The clustering of LR and HR training data is carried out as outlined in Section 3.2. We then randomly selected 40000 pairs of LR and HR training patches for each cluster.

A single dictionary pair with 1000 atoms is learned for the algorithm of Yang et al. [50]. For the learning, 40,000 pairs of LR and HR patches are randomly selected from the same training set used by the proposed algorithm. The algorithm of Zeyde et al. [47] also uses a pair of 1000-atom dictionaries, with the training image dataset provided by the authors [47]. A LR dictionary is learned by K-SVD [15] with 40 iterations and sparsity $S=3$. Then, the coupled HR dictionary is calculated as specified in [47]. For a subjective comparison, we have added a final back-projection stage to Zeyde et al.'s [47] algorithm as the other three algorithms employ it. We used the default design parameters and training image dataset for the algorithm of He et al.as specified in [46] and a pair of 771-atom coupled dictionaries is trained.

Test images include gray-scale and color images. A LR gray-scale image is input to each of the super-resolution algorithms to reconstruct its HR counterpart. However, a

LR color image is first transformed to the luminance and chrominance color space and only the luminance component is input to the super-resolution algorithm to reconstruct the luminance component of the corresponding HR image. As customarily done in most super-resolution algorithms, the two chrominance components are reconstructed by bicubic interpolation. To obtain a full-color HR image, the three components are used. In this experiment, PSNR is used as a quantitative measure of quality. For gray-scale images, PSNR is calculated between the original and reconstructed images. For color images, on the other hand, PSNR is calculated with the luminance color components of the original image and the reconstructed image, in accordance with the common practice in the literature. PSNR is defined as follows,

$$PSNR(y,\hat{y}) = 10\log_{10}\frac{255^2}{MSE(y,\hat{y})} \qquad (3.4)$$

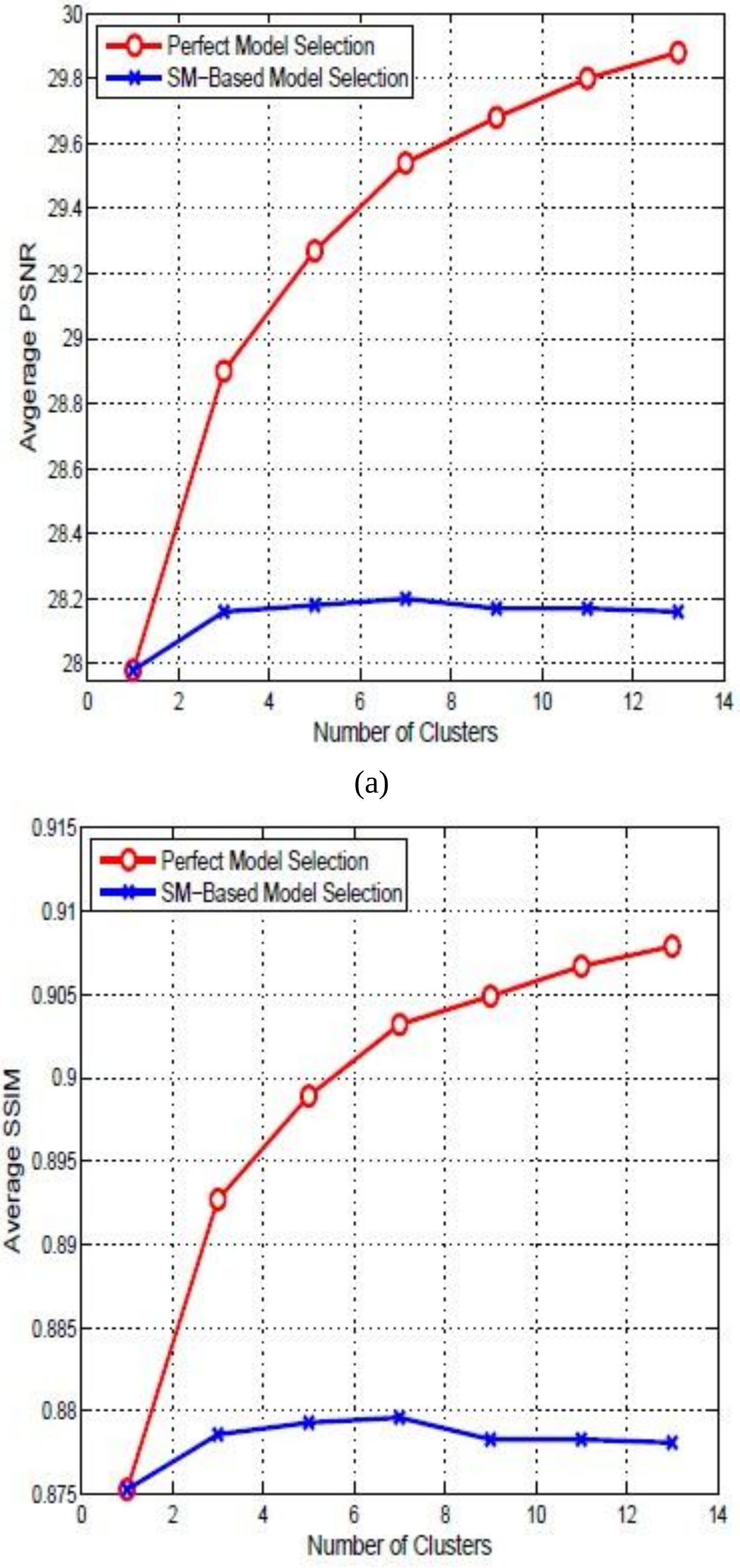
where y is the true image and $\hat{y}$ is its estimate and both are 8-bit gray-scale $M \times N$ images. $MSE(y,\hat{y})$ is the mean-square error between y and $\hat{y}$, which is defined as follows,

$$MS(y,\hat{y}) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(y_{ij} - \hat{y}_{ij})^2 \qquad (3.5)$$

Also, SSIM [80] is used as a perceptual quality metric, which is believed to be more compatible with human perception than PSNR. Similar to most super-resolution algorithms, we calculate SSIM for color images as the average SSIM value of the luminance and two chrominance components of the image.

An important issue in the proposed algorithm is the number of clusters to employ. There is no specific rule of thumb to decide on the optimal number of clusters. We have empirically examined the impact of the number of clusters on the performance of the proposed algorithm. For this purpose, the following experiment which studies the effect of the total number of clusters on the performance is designed. Using the SM of the LR patches, we first classified the training data into a total of 3, 5, 7, 9, 11 and 13 clusters. For each case, we learned a dictionary pair for every cluster. The case which treats the training data as a single cluster (Yang et al [50]) and learns a single coupled dictionary pair is also studied. We then used the learned dictionary pairs for each case and reconstructed each of the 12 test images (shown in Fig. 3.1). For each case, SM is used as a model selection criterion. Average PSNR and SSIM values are recorded for each case. We then repeated the same experiment with perfect model selection. Given a LR patch and the HR ground-truth in perfect model selection, we first reconstruct the HR patch with all the cluster dictionary pairs and then pick the cluster dictionary pair that minimizes the MSE between the ground-truth HR patch and its reconstructions. Fig. 3.5 shows the average PSNR and SSIM values with respect to the total number of clusters for the two scenarios: perfect model selection and SM-based model selection.

Several observations can be made from the results presented in Fig. 3.5. First, it indicates that using a set of cluster dictionary pairs is better than using a single dictionary pair. Second, perfect model selection plots suggest that using more clusters means significantly better performance. This result further suggests that the designed cluster dictionary pairs are selective and best fit their respective cluster signals. Third, when SM is applied as a model selection criterion, the performance of the proposed algorithm increases with increasing clusters.



(a)

(b)

Figure 3.5: Performance of the proposed algorithm with perfect model selection and SM as a model selection criterion. (a) Average PSNR versus number of clusters. (b) Average SSIM versus number of clusters.

The average PSNR and SSIM performances peak at seven clusters and then degrade slightly. This behavior is due to the trade-off between the number of clusters and the accuracy of SM as a model selection criterion. It can be generalized that with more clusters the dictionary pair corresponding to each cluster tends to be more discriminative and better represent signals of the respective cluster. However, using more clusters makes the model selection task less accurate. Plots in Fig. 3.5 suggest that with better model selection criteria, significant improvements in both PSNR and SSIM are possible.

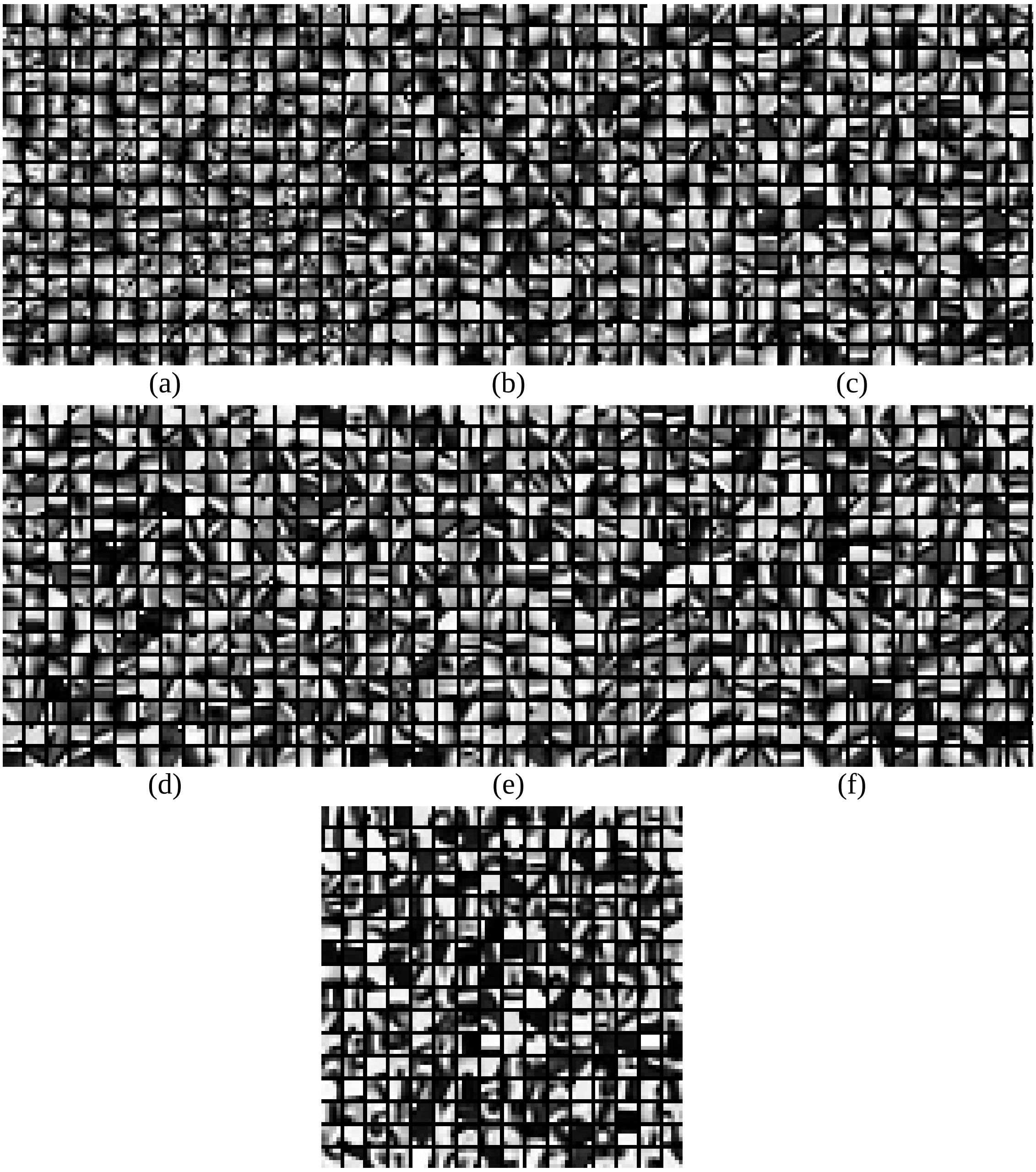
The number of clusters is set to seven and to further investigate the representation power of the designed cluster dictionary pairs another experiment is carried out. The training data of each cluster is used for this purpose. The HR patch in each cluster is reconstructed with every cluster dictionary pair based on its LR counterpart. The MSE between the true HR patch and each of it's reconstructions is obtained. For comparison, a HR patch in each cluster is also reconstructed with the single dictionary pair of Yang et al.'s [50] algorithm. Table 3.2 shows the results of this experiment. It is obvious that data in each cluster is best represented with the dictionary pair of that particular cluster. Considering C_1, it is observed that the dictionary pairs corresponding to other clusters and the single dictionary pair of Yang et al.[50] do in fact adequately represent data in this cluster. This is due to the fact that C_1 contains smooth patches. For this reason, one does not in fact need sophisticated algorithms for the unsharp cluster C_1. A simple algorithm such as bicubic interpolation suffices in satisfactorily reconstructing HR patches in this cluster. However, data in C_7 can not be satisfactorily represented by dictionary pairs belonging to other clusters. Also, the single dictionary pair of Yang et al.[50] fails to represent data in C_7. Table 3.2 clearly indicates that error levels for sharper clusters are significantly higher than unsharp clusters. Therefore, reconstructing very sharp features is a challenging task. One must pay special attention to designing dictionaries that can better address the problem of reconstructing such features.

Table 3.2: Reconstruction quality (MSE) of HR patches in all clusters based on their LR counterparts using the dictionary pair in each cluster. SD denotes the case of a single cluster with a single dictionary pair (as used by Yang et al [50]).

Data in/ Dictionaries in	C_1	C_2	C_3	C_4	C_5	C_6	C_7	SD
C_1	**1.90**	2.15	2.19	2.23	2.26	2.30	2.46	2.28
C_2	67.82	**57.29**	60.18	60.76	61.47	61.72	67.62	67.01
C_3	169.17	149.00	**139.50**	145.32	147.62	147.19	159.77	162.97
C_4	306.12	267.42	261.02	**245.69**	257.26	256.93	275.65	286.23
C_5	488.04	425.97	416.72	406.30	**385.45**	401.07	425.64	450.20
C_6	677.01	583.89	569.63	560.01	556.25	**515.34**	568.51	613.64
C_7	1868.97	1587.96	1531.46	1449.82	1399.02	1374.98	**1074.08**	1562.71

In the light of the above experiments, we choose to use a total of seven clusters which turns out to be a good compromise between the representation power of the dictionary pairs and the accuracy of SM as a model selection criterion.

Figure 3.6 shows example reshaped atoms of the 7 HR cluster dictionaries designed with the proposed algorithm. It can be clearly seen that the learned dictionary atoms inherit the sharpness nature of their respective clusters. The sharpness of the atoms increases from D_1 until D_7.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 3.6: Example reshaped atoms of HR dictionaries from (a) Cluster C_1 (unsharp cluster) up to (g) Cluster C_7 (the sharpest cluster).

The PSNR (top) and SSIM (bottom) values of super-resolution reconstructions obtained with the aforementioned algorithms and setups with respect to the original ground-truth image are listed in Table 3.3. It is noted that we have conducted the simulations with source codes provided by the authors of [50], [47] and [46]. The proposed algorithm clearly out-performs the algorithm of Zeyde et al. [47] with an average PSNR improvement of 0.47 dB. The proposed algorithm performs better than the algorithm of Yang et al. [50] with an average PSNR improvement of 0.17 dB. This improvement validates the added-benefit of employing multiple dictionary pairs instead of a single general dictionary pair.

In view of Table 3.3, one notices that the success of the proposed algorithm is particularly valid for images with sharp features such as Text image 1 and the butterfly and ppt3 images. Considering Text Image 1, as an example, it can be seen in Table 3.1 that the majority of patches are located in clusters C_1 (85.3 %) and C_7 (78.1 %). For those two clusters, SM is strongly scale-invariant. This means that SM performs well as a model selection criterion. Even though SM is moderately or weekly invariant in the middle clusters (C_2 through C_6), a small percentage of patches is located in these clusters. Similar observations can be made for other test images. In general, the SM of highly detailed images spreads over a wide range. Patches with high SM values tend to be approximately invariant with respect to scale. The proposed algorithm exploits this fact and thus has a clear advantage in representing sharp edges and corners.

Table 3.3: PSNR (dB) and SSIM comparisons of bicubic interpolation, the algorithms of Zeyde et al. [47], Yang et al. [50], He et al. [46] and the proposed algorithm, respectively.

Image	Bicubic	Zeyde et al.	Yang et al.	He et al.	Proposed
Barbara	25.35	**25.89**	25.86	25.84	25.87
	0.7930	**0.8374**	0.8357	0.8372	0.8361
Berk. 198054	24.75	26.56	26.85	26.98	**27.05**
	0.8267	0.8771	0.8816	**0.8839**	0.8830
Butterfly	27.46	30.59	31.26	31.44	**31.68**
	0.8985	0.9384	0.9457	0.9463	**0.9480**
Fence	25.05	26.22	26.34	26.22	**26.41**
	0.7449	0.8008	0.8037	0.8045	**0.8046**
Flowers	30.42	32.39	32.76	**32.98**	32.85
	0.8828	0.8984	0.9005	**0.9018**	0.9005
Input6	28.08	29.93	30.15	30.24	**30.31**
	0.8523	0.8957	0.8976	0.8989	**0.8991**
Lena	34.71	35.68	36.36	**36.58**	36.37
	0.8507	0.8625	0.8631	**0.8647**	0.8634
Man	29.26	30.44	30.68	30.80	**30.76**
	0.8315	0.8692	0.8713	**0.8739**	0.8719
ppt3	26.85	29.30	29.68	29.79	**30.04**
	0.9372	0.9572	0.9604	0.9621	**0.9634**
Starfish	30.23	31.99	32.66	**33.28**	32.82
	0.8923	0.9244	0.9279	**0.9325**	0.9298
TextImage1	17.52	18.47	18.58	18.54	**18.85**
	0.7246	0.7893	0.7974	0.7953	**0.8100**
Texture	20.64	22.60	22.55	**22.78**	22.67
	0.8272	0.8949	0.8939	**0.8997**	0.8963
Average	26.69	28.34	28.64	28.79	**28.81**
	0.8385	0.8788	0.8816	0.8834	**0.8838**

It can be seen in Table 3.3 that the proposed algorithm is competitive with the state-of-the-art algorithm of He et al. [46], with a slight average PSNR improvement of 0.02 dB. The performances of the two algorithms are similar for images with limited spread of SM values. For images with wide spread of SM values, however, the

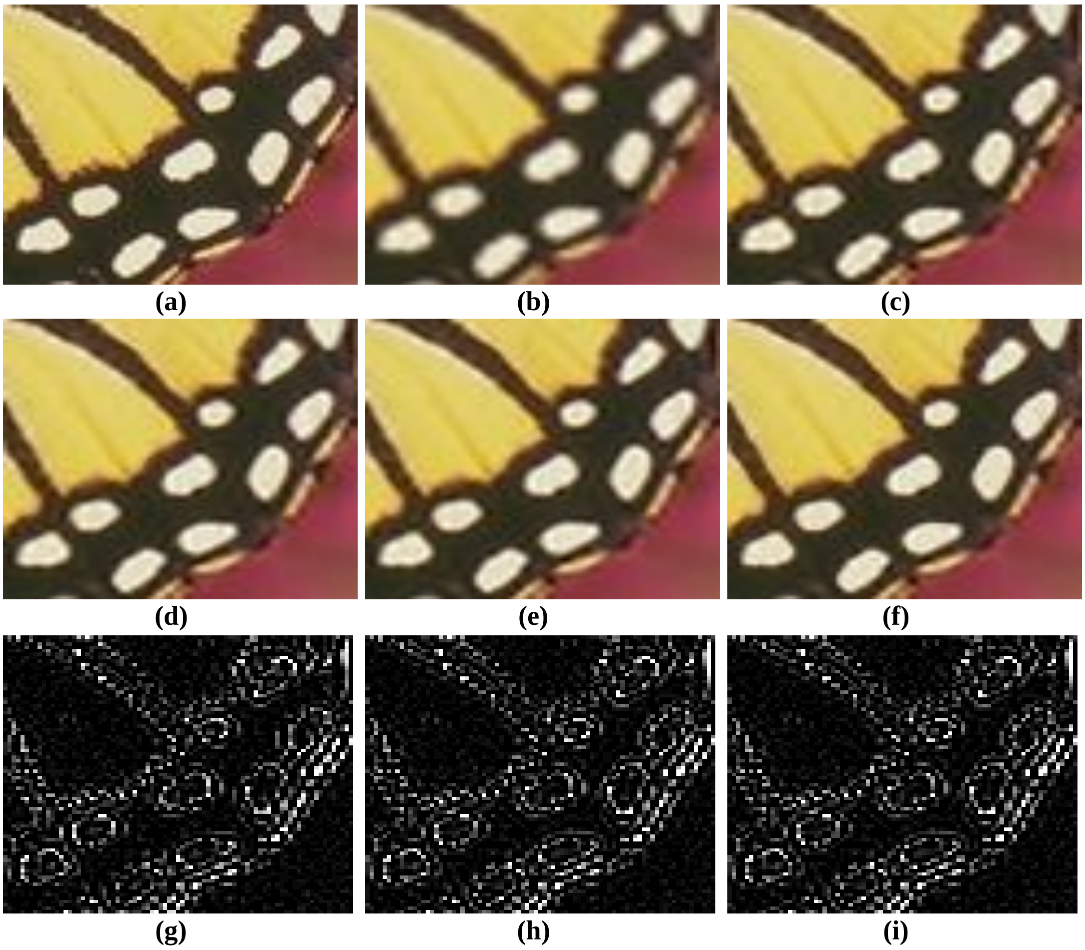

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3.7: Visual comparison of the Butterfly image: (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Zeyde et al. [47], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

performance of the proposed algorithm is significantly better. SSIM simulations also validate the above observations and conclusions.

Figure 3.7 compares a portion of the original Butterfly image to its reconstructions obtained with bicubic interpolation, Zeyde et al.[47], Yang et al.[50], He et al.[46] and the proposed algorithm. One can observe that the proposed algorithm is better able to reconstruct edges in this image as compared to the other algorithms (best viewed on a high-definition computer monitor). This is especially seen in

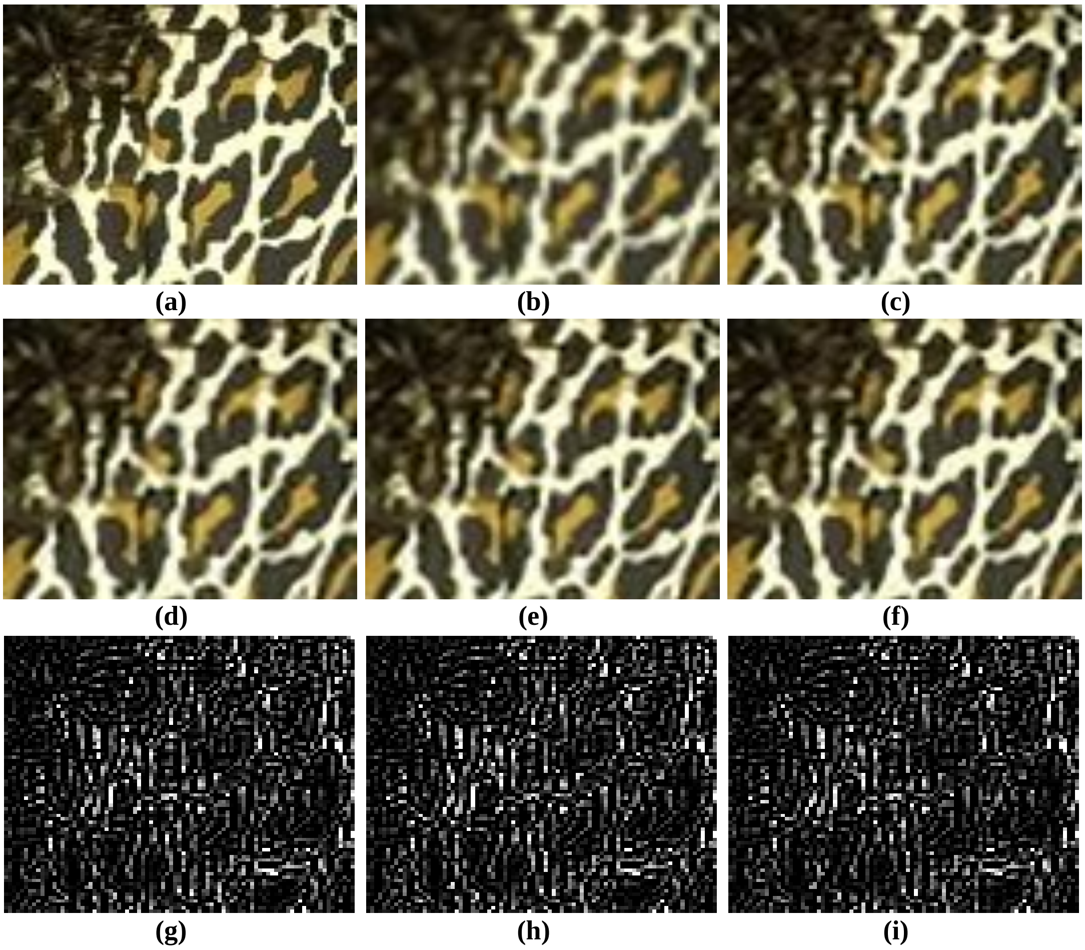


Figure 3.8: Visual comparison of the BSDS 198054 image (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Zeyde et al. [47], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

reconstructing the curvy line around the butterfly's wing. The circular shapes on the butterfly's wings are reconstructed better than those of the other three algorithms. Fig. 3.8 (g), (h), (i) show respectively the difference between the original scene and its reconstructions from Yang et al. [50], He et al. [46] and proposed algorithm. Clearly the proposed algorithm has least amount of artifacts.

Figure 3.8 conducts the same comparison with the BSDS 198054 image from the Berkeley Segmentation Dataset (BSDS). It can be seen that the edges in the proposed

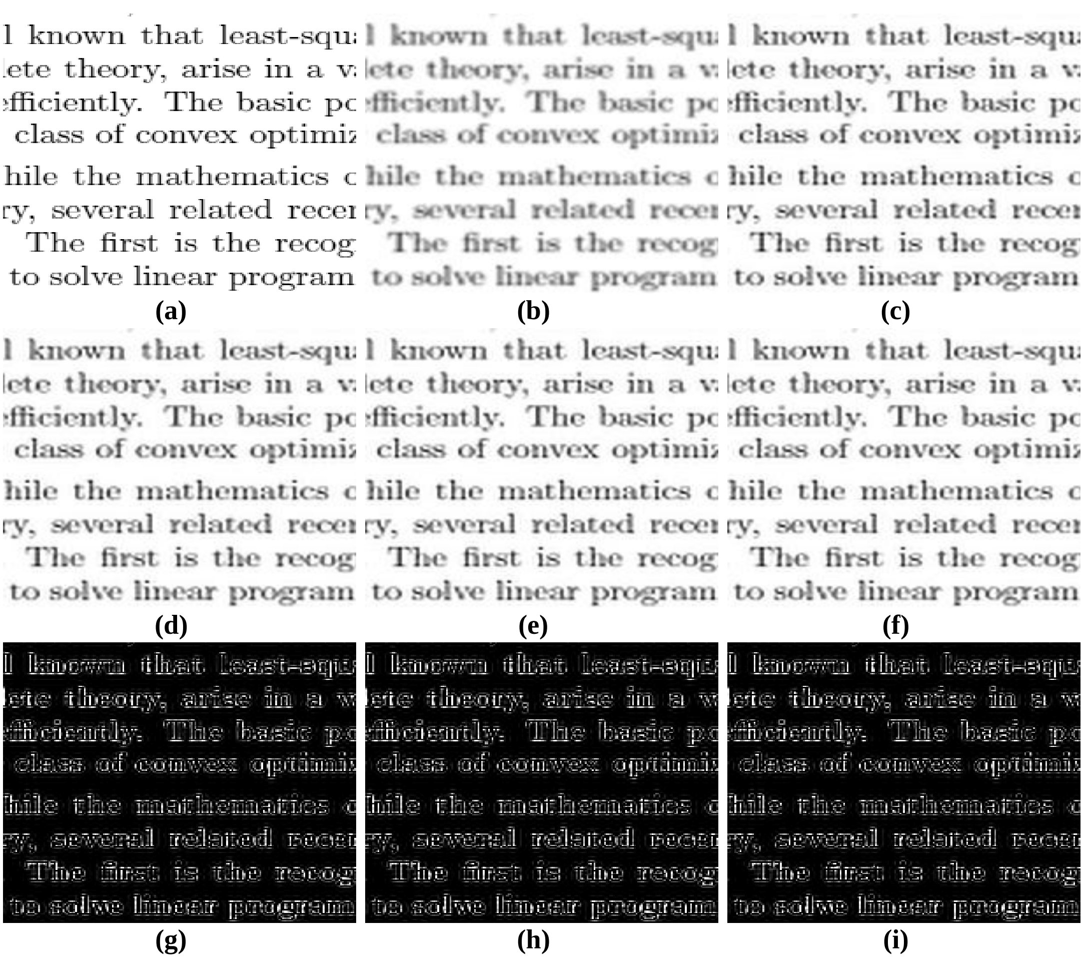


Figure 3.9: Visual comparison of the Text image 1 (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Zeyde et al. [47], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

algorithm's reconstruction are sharper than the edges in the reconstructions of the other algorithms. Fig. 3.8 (g), (h), (i) show respectively the difference images. It is obvious that proposed algorithm produces less artifacts around sharp features.

As a final visual comparison, Fig. 3.9 shows reconstructions of Text Image 1. Similar to the cases in the last two figures, one can see that the difference image for the proposed algorithm is closer to zero when compared to these of the other two algorithms. This clearly shows that the proposed algorithm produces a better reconstruction.

Chapter 4

SINGLE IMAGE SUPER-RESOLUTION VIA SPARSE REPRESENTATION OVER DIRECTIONALITY STRUCTURED DICTIONARIES BASED ON THE PATCH GRADIENT PHASE ANGLE

4.1 Introduction

The work conducted in Chapter 3 considers the magnitude of the gradient operator to quantify patch intensity variations. The phase of the gradient operator carries directional patch information. Intuitively, this information can be quantified as a directional measure that quantifies patch directionality. In view of this, we define a dominant phase angle (DPA) measure based on the majority of the angles in the phase matrix of this operator [81]. Employing this measure as a classifier, one can cluster training patches based on their directionality and obtain directional cluster dictionaries. The applicability of such a measure for the purpose of super resolution requires studying the impact of resolution (scale) on the DPA values. The following sections include an empirical study on the scale-invariance of this measure. Then, we introduce a super-resolution algorithm based on selective sparse representation over DPA cluster dictionaries, where the same measure is used for model selection. Experiments conducted on several images show a competitive performance of the proposed algorithm compared with the state-of-art algorithms. Besides, the designed dictionaries are shown to inherit the intended directional structure of their respective clusters.

4.2 The Proposed Super-resolution Algorithm

In this section, the DPA as an approximately scale-invariant patch measure is introduced and its behavior with respect to scale is empirically studied. Then, the training and testing stages of the proposed algorithm are detailed.

4.2.1 Approximate Scale-Invariance of the Dominant Gradient Phase Angle Measure

The phase is generally more informative than the magnitude and thinking about exploiting information in the phase of the gradient operator is promising. The phase matrix of the gradient operator is defined [74] as follows

$$\Phi = \arctan(\frac{G^v}{G^h}) \tag{4.1}$$

Where, G^h and G^v are horizontal and vertical gradients, respectively. Intuitively, the directional structure of an image patch can be characterized by phase values. The DPA in the phase matrix can be defined so that it describes the directionality. Thus, quantizing the phase matrix and establishing a histogram for the quantized angles is a requirement for this. If a certain angle value is repeated more frequently than half of the number of elements in the phase matrix, the patch can be considered a directional patch where the directionality of this patch is characterized by that dominant angle otherwise, it is considered as a non-directional patch. In this work, angles are quantized into values of 0°, 45°, 90° and 135°.

The following experiment is conducted to investigate the impact of scale on DPA on the images shown in Fig. 3.1. Each image is divided into non-overlapping 6x6 patches and filtered with a bicubic filter and downsampled by 2 in both dimensions to obtain a LR version. Corresponding to DPA values of 0°, 45°, 90° and 135°, clusters C_1 through C_4 are defined respectively, while C_5 is defined for patches that do not have a specific directional nature. Then the DPA value of each HR and LR patch is calculated according to (4.1) and is used to cluster it into one of the clusters. The ratio between the number of LR patches correctly classified into a certain cluster and the total number of HR patches in that cluster is defined as the DPA scale-invariance. For the images shown in Fig. 3.1, DPA scale-invariance ratios are listed in Table 4.1. From Table 4.1 it can be seen that the DPA scale-invariance ratios are generally greater than 50 %. For C_5, the DPA scale-invariance values are high and compared to the cases of C_2 and C_4, these DPA invariance values are high in C_1 and C_3.

4.2.2 Clustering and Sparse Model Selection with the Dominant Gradient Phase Angle Measure

Training image patch pairs are classified into the aforementioned five clusters in this part. Then, the dictionary learning method of Yang et al. [50] is employed to train for coupled LR and HR cluster dictionary pairs. Features are extracted at the MR level from the LR image patches and used to train the LR dictionary. The DPA of each LR patch at the MR level is employed to classify it into a certain cluster. Then, the corresponding HR patch and the extracted MR features are inserted to the HR and LR training sets of that cluster, respectively. The main steps of the training stage are outlined in Algorithm 9. The intention in this work is designing cluster dictionaries that contain the intended directional structures of their respective clusters.

Table 4.1: Number of HR patches in a cluster (top) and cluster scale-invariance ratio (bottom). The listings of the cluster containing the largest number of image patches are in bold face.

Image	DPA Clusters				
	C^0	C^{45}	C^{90}	C^{135}	C^{nd}
Barbara	1819	580	466	508	**3852**
	45.52	43.28	87.34	52.36	**72.04**
BSDS 198054	555	126	130	163	**3266**
	71.89	58.73	61.54	55.21	**82.70**
Butterfly	126	224	269	47	**1098**
	80.95	74.55	92.94	72.34	**66.03**
Fence	624	49	252	12	**827**
	68.91	65.31	59.52	66.67	**76.54**
Flowers	453	249	412	185	**3681**
	72.41	64.26	71.12	68.65	**79.63**
Input6	529	404	951	441	**9379**
	80.15	71.29	75.39	73.02	**72.95**
Lena	894	171	157	474	5529
	85.91	69.59	73.25	82.07	78.15
Man	871	236	550	292	**5276**
	72.10	68.64	75.45	75.34	**76.12**
Starfish	101	119	228	149	**1167**
	66.34	73.95	78.51	79.87	**77.21**
TextImage1	288	1	8	3	**11036**
	12.85	0.00	12.50	0.00	**76.11**
Texture	**6602**	1319	74	97	**3144**
	82.70	76.19	82.43	69.07	**56.04**
ppt3	636	346	814	299	**7497**
	78.77	62.14	86.98	61.87	**86.27**
Average	74	67	78	68	76

In the reconstruction stage, the MR image is then divided into overlapping patches. Then, the DPA of each MR patch is calculated and used to identify the cluster this patch belongs to. Afterwards, the sparse coding coefficients of the features extracted from the MR patch over the cluster LR dictionary are calculated and the

Algorithm 9 The Proposed Dictionary Learning Stage.

INPUT: HR Training image set.
OUTPUT: A set of directionally structured cluster dictionary pairs.

```
Obtain a LR image for each HR one by blurring and downsampling.
Upsample each LR image to obtain a MR image.
Divide the HR and MR images into patches.
Extract features from the MR image by filtering.
Divide feature images into column patches.
Combine HR patches to form a HR training set and MR features to form a LR training set.
for Each patch in the LR training set,,
    Calculate the DPA of the corresponding patch in the MR image, and identify the cluster number.
    Set the MR features and the HR patch to the LR and HR training sets of this cluster.
end for
Learn a pair of coupled dictionaries for each cluster.
```

corresponding HR patch is then reconstructed by imposing these coefficients on the HR dictionary of the same cluster. At the end, the overplaying HR patches are reshaped and merged to form a HR image estimate. Algorithm 10 summarized the proposed reconstruction algorithm.

As long as the proposed algorithm relies on designing cluster dictionaries, high redundancy is not demanded. In his work, 600-atom dictionaries are designed as a good compromise between computational complexity and representation quality. To meet these requirements, this value is empirically determined. The most computationally expensive stage in the sparse reconstruction framework is the sparse coding stage which relies on vector selection [82].

Therefore, five compact dictionaries are used and expected to substantially reduce the sparse coding computational complexity and thus reduce the overall SR computational complexity, as compared to the case of using a single highly redundant dictionary. However, the DPA value of each MR patch should be calculated in the proposed algorithm. It can be seen that the proposed algorithm’s computation complexity is comparable to that of the algorithm of Yang et al.[50] where a 1000-atom dictionary pair is used**.**

Algorithm 10 The Proposed Reconstruction Stage.

```
INPUT: A LR image, cluster dictionary pairs.
OUTPUT: A HR image estimate
Upsample the LR image to the required resolution level (MR).
Extract feature images from the MR image.
Extract patches from the feature images and group them column-wise
Divide the MR image into overlapping patches.
for Each MR patch
    Calculate DPA of the MR patch.
    Determine the cluster this patch belongs to.
    Calculate the sparse coding coefficients of the corresponding features over the LR dictionary.
    Reconstruct a HR patch as the product of the HR dictionary and the calculated coefficients.
end for
Obtain a HR image estimate by merging overlapping HR patches.
```

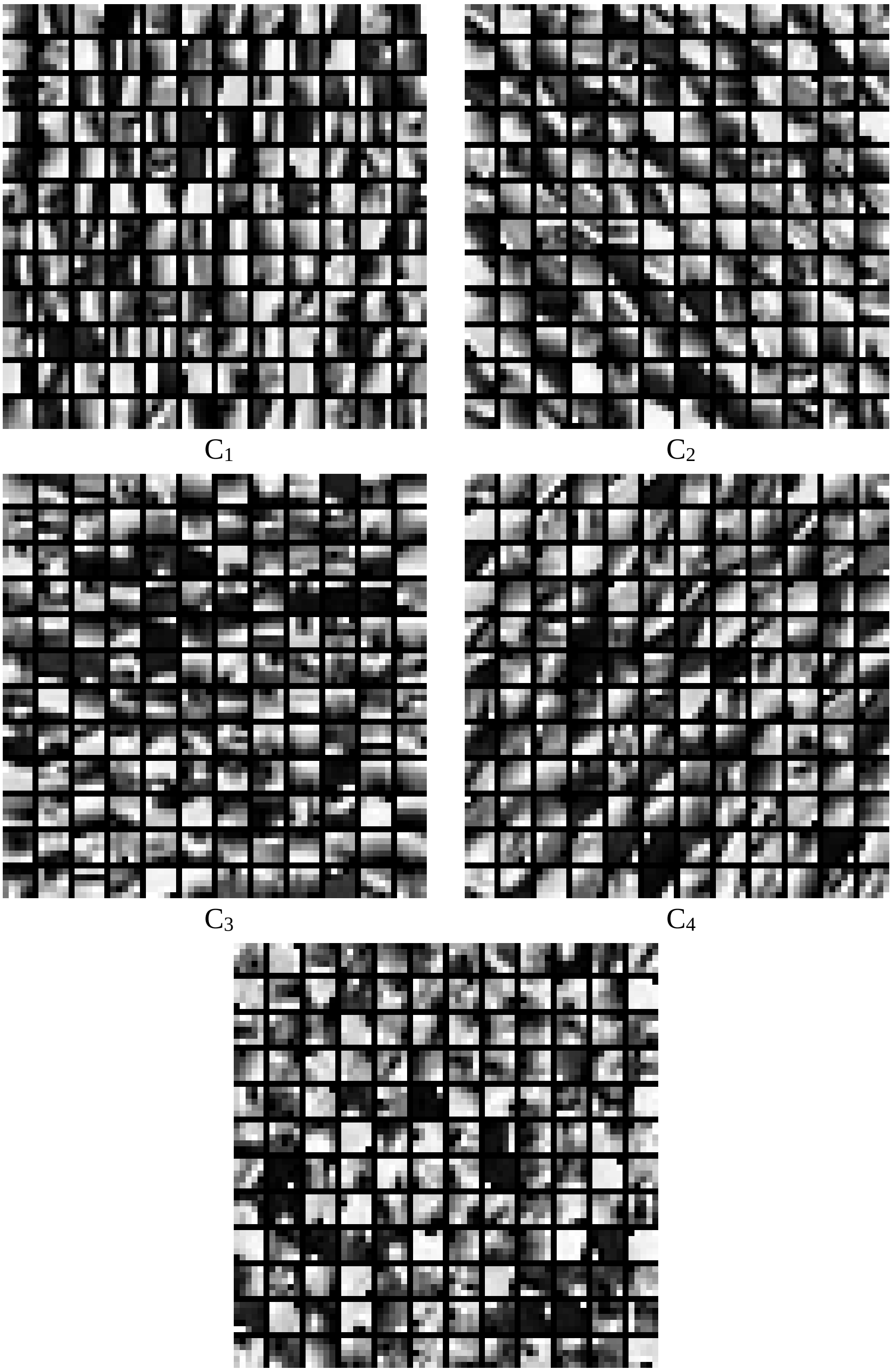
4.3 Experimental Results

The performance of the proposed algorithm is examined and compared to bicubic interpolation, the algorithm of Yang et al. [50] which uses a single dictionary pair and the state-of-the-art algorithms of Peleg et al. [70] and He et al. [46]. The PSNR and SSIM [80] measures are employed for comparisons. In accordance with the common practice in the literature, PSNR and SSIM are calculated exactly the same way as done in Section 3.4.

The proposed algorithm uses a 5×5 patch size, and five cluster dictionary pairs. As done in Chapter 3, the Flickr image dataset [79] is used as a training set. Image patches are classified into the five clusters based on DPA. For each cluster 40,000 patch pairs are randomly selected and used to train for a cluster dictionary pair. The same parameters with the same training data set are used to train for a 1000-atom dictionary for the algorithm of Yang et al. [50]. The default values suggested by the authors are used in the algorithms of Peleg et al. [70] and He et al. [46].

Example reshaped atoms in the HR cluster dictionaries of the proposed algorithm are shown in Fig. 4.1. The atoms of these dictionaries clearly inherit the directional structure of their respective clusters. Atoms of the first four dictionaries are generally perpendicular to the directions of 0°, 45°, 90° and 135° from left to right and top to bottom respectively and atoms in the fifth dictionary have a chaotic directional in accordance with the non-directional nature of C_5.

PSNR and SSIM values of the test images shown in Fig. 3.1 are listed in Table 4.2. The performance of the proposed algorithm is shown with dictionaries of 600, 800 and 1000 atoms, denoted by P_{600}, P_{800} and, P_{1000} respectively. The proposed algorithm outperforms the algorithms of Yang et al. [50] and Peleg et al.[70] with average PSNR improvements of 0.14 dB and 0.13 dB, respectively with 600-atom dictionaries. Meanwhile, it is competitive with the algorithm of He et al. [46]. The



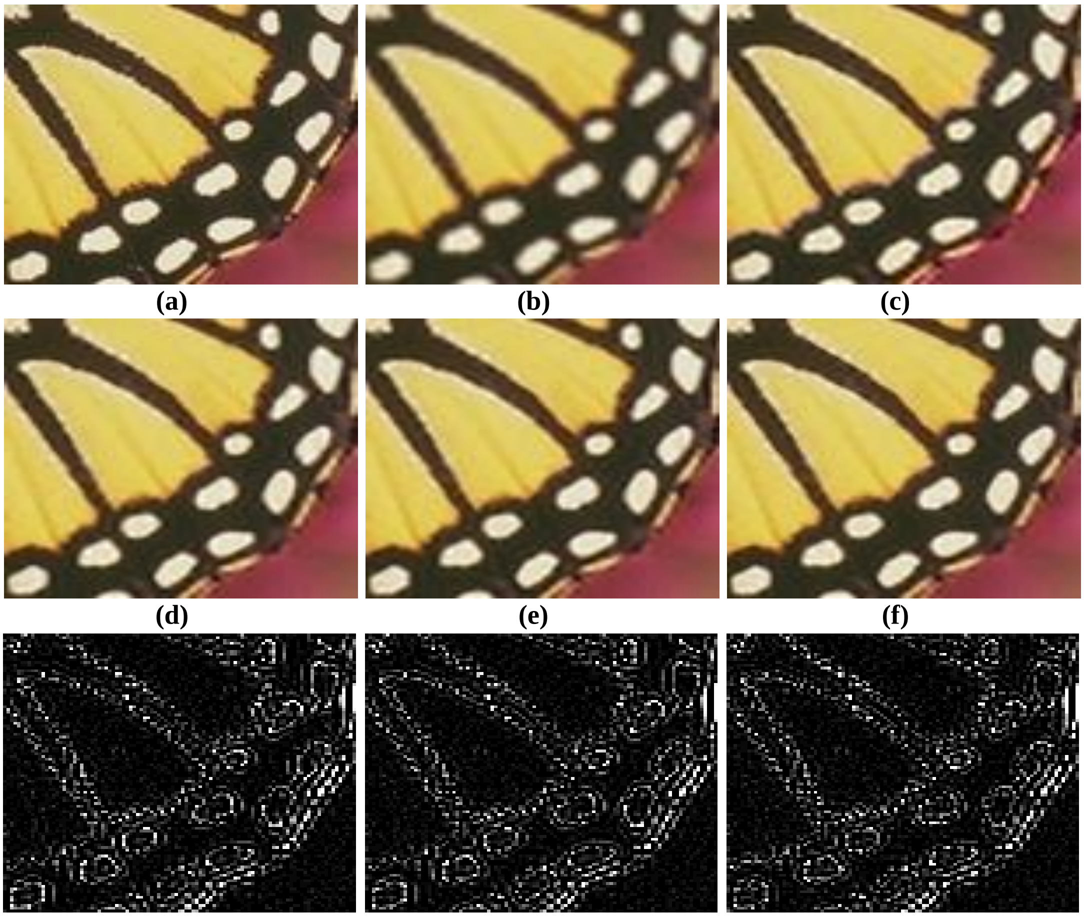
C_1 C_2 C_3 C_4 C_5

Figure 4.1: Example reshaped atoms of HR dictionaries in C_1 through C_5.

Table 4.2: PSNR (dB) and SSIM of bicubic interpolation, the algorithms of Peleg et al. [70], Yang et al. [50], He et al. [46] and the proposed algorithm.

Image	Bicubic	Peleg et al.	Yang et al.	He et al.	$\boldsymbol{P}_{1000}$	$\boldsymbol{P}_{800}$	$\boldsymbol{P}_{600}$
Barbara	25.35	25.76	25.86	25.84	**25.86**	25.85	25.85
	0.7930	0.8359	0.8357	**0.8372**	0.8345	0.8352	0.8347
BSDS 198054	24.75	26.76	26.85	26.98	27.10	**27.11**	27.08
	0.8267	0.876	0.8816	0.8839	**0.8840**	0.8839	0.8834
Butterfly	27.46	30.96	31.26	31.44	**31.83**	31.78	31.73
	0.8985	0.9227	0.9457	0.9463	0.9486	**0.9487**	0.9482
Fence	25.05	26.17	26.34	26.22	**26.37**	26.36	26.36
	0.7449	0.7967	0.8037	0.8045	**0.8053**	0.8048	0.8047
Flowers	30.42	32.57	32.76	**32.98**	32.95	32.92	32.91
	0.8828	0.8522	0.9005	**0.9018**	0.9010	0.9010	0.9008
Input6	28.08	30.03	30.15	30.24	30.32	**30.33**	30.29
	0.8523	0.8737	0.8976	0.8989	0.8991	**0.8991**	0.8988
Lena	34.71	**36.59**	36.36	36.58	36.37	36.37	36.36
	0.8507	0.8387	0.8631	**0.8647**	0.8631	0.8630	0.8630
Man	29.26	30.67	30.68	**30.80**	30.77	30.77	30.76
	0.8315	0.8729	0.8713	**0.8739**	0.8721	0.8723	0.8720
ppt3	26.85	29.71	29.68	29.79	**30.08**	29.96	29.89
	0.9372	0.9494	0.9604	0.9621	**0.9639**	0.9631	0.9623
Starfish	30.23	32.92	32.66	**33.28**	32.78	32.77	32.71
	0.8923	0.9165	0.9279	**0.9325**	0.9290	0.9287	0.9282
TextImage1	17.52	18.73	18.58	18.54	**18.79**	18.79	18.78
	0.7246	0.8118	0.7974	0.7953	**0.8073**	0.8068	0.8050
Texture	20.64	**22.97**	22.55	22.78	22.67	22.72	22.70
	0.8272	**0.9032**	0.8939	0.8997	0.8965	0.8976	0.8973
Average	26.69	28.65	28.64	28.79	**28.82**	28.81	28.78
	0.8385	0.8708	0.8816	0.8834	**0.8837**	0.8837	0.8832

same result is concluded in terms of SSIM. It is clear that using more redundancy in the dictionaries of the proposed algorithm will not significantly improve the



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4.2: Visual comparison of the Butterfly image. (a) Original and reconstructions with : (b) Bicubic interpolation, (c) Peleg et al. [70] (d) Yang et al. [50], (e) He et al. [46] and (d) The proposed algorithm. The last row shows the difference between the original and reconstructions of :(g) Yang et al., (h) He et al. and (i) The proposed algorithm.

performance. Furthermore, the proposed algorithm is particularly superior in images which contain rich edges.

In Fig. 4.2 the proposed algorithm is compared with bicubic interpolation and the algorithms of Peleg et al. [70], Yang et al. [50] and He et al. [46] for Butterfly image. Figure 4.2 (g), (h) and (i) show the differences between the ground-truth scene and its reconstructions from Yang et al., He et al. and the proposed algorithm, respectively. The result of bicubic interpolation is over-smooth and the reconstructions of Peleg et al. and Yang et al. have sharper edges. He et al.'s algorithm has further sharper edges. However, the reconstruction in the proposed algorithm is the best to approximate the ground-truth scene with less artifacts. It can clearly be seen in the edges and details of the Butterfly's wing. The difference image of the proposed algorithm is the darkest one.

Table 4.3: PSNR and SSIM comparisons for 5 clusters and 9 clusters.

Image	P_{600} -5Clusters	P_{600} -9 Clusters
Barbara	25.85 0.8347	**25.90** **0.8352**
BSDS 198054	**27.08** **0.8834**	26.90 0.8811
Butterfly	**31.73** **0.9482**	31.46 0.9467
Fence	**26.36** **0.8047**	26.35 0.8029
Flowers	**32.91** **0.9008**	32.76 0.8993
Input6	**30.29** **0.8988**	30.20 0.8988
Lena	**36.36** **0.863**	36.32 0.8625
Man	**30.76** **0.872**	30.73 0.8715
ppt3	29.89 0.9623	**30.19** **0.9627**
Starfish	32.71 0.9282	**32.73** **0.9274**
TextImage1	**18.78** **0.805**	18.74 0.8037
Texture	**22.7** **0.8973**	22.69 0.8960
Average	**28.78** **0.8832**	28.75 0.8823

4.3.1 The Performance of The Proposed Algorithm with More DPA clusters

In this experiment, we investigate the performance of the proposed algorithm with increasing the number of DPA clusters. In this setting, we quantize the angles into, 0, 22.5, 45, 67.5, 90, 112.5, 135, 167.5. This means having eight directional clusters with the aforementioned orientations with the ninth cluster being non-directional.

Simulation results are shown in Table 4.3. In view of this table, it can be seen that increasing the number of clusters will not improve the performance. This is because the accuracy of DPA in model selection degrades with increased cluster numbers.

Chapter 5

IMAGE SUPER-RESOLUTION VIA SPARSE REPRESENTATION OVER MULTIPLE LEARNED DICTIONARIES BASED ON EDGE SHARPNESS AND GRADIENT PHASE ANGLE

5.1 Introduction

Based on the success of the SM and DPA classifiers presented in Chapters 3 and 4, it seems promising to combine the two measured together. This can be done by first clustering training data based on SM. Then, using DPA as a secondary classifier to further classify data in each SM cluster. A pair of coupled dictionaries is learned over the cluster training patches for each cluster. Correspondingly, the same two criteria (SM and DPA) are together employed as a sparse model selection mechanism in the reconstruction stage [83].

Experimental results conducted on natural images validate a competitive performance of the proposed algorithm as compared to the state-of-the-art super-resolution algorithms. This result is quantitatively validated in terms of the PSNR and SSIM quality measures. Visual comparison results come in line with quantitative results.

5.2 The Proposed Super-Resolution Algorithm

The image gradient operator is believed to be crucial to the perception and analysis of natural images [84, 85, 86]. In this section, first SM of image patches is employed as a criterion to classify image patches into three main clusters based on their sharpness. Then, based on directional structure of the patches, DPA is employed to further cluster them into several sub-clusters. To select the most relevant cluster dictionary pair for each LR patch during the reconstruction stage, these two measures are used in the same manner.

5.2.1 Approximate Scale-Invariance of the Image Patch Sharpness and Dominant Gradient Phase Angle Measures

In this work, three SM clusters C_1, C_2 and C_3 are employed for SM intervals of [0, 10], [10, 20] and [20, 255] respectively. The bounds of two cluster intervals are set to uniformly divide the remaining SM range and the lower bound of the third internal in such a way that it contains very sharp patches. Based on this, patches are classified as un-sharp, moderately sharp and very sharp. More clusters can be defined for finer classification. However, this will deteriorate the ability of SM in correctly estimating which cluster a given patch belongs to, as noted in chapter 3. We then employ the DPA measure (specified in Chapter 4) as a secondary classifier.

To investigate the impact of scale on SM and DPA the following experiment is conducted on each of the test images shown in Fig. 3.1. Each image is divided into non-overlapping 6×6 patches. A LR image is obtained by applying a bicubic filter on the HR image and downsampling it with a scale factor of 2 in both dimensions. The LR image is then upsampled to the MR level via bicubic interpolation. Each MR image is also divided into non-overlapping 6×6 patches. First, three clusters are defined according to SM intervals of [0, 10], [10, 20] and [20, 255].

These are denoted by C_1, C_2 and C_3, respectively. The bounds of these intervals are empirically selected. The lower bound of the SM interval of C_3 is made high in order to capture very sharp features. Then, the remaining interval is uniformly split into two intervals. Afterwards, SM values of all HR and MR patches of an image are calculated. Each patch is classified into a cluster based on its SM value. HR patches classified into each cluster are counted and the number of their MR counterparts which are correctly classified into the same cluster is also counted. The SM invariance is defined as the ratio between the number of MR patches correctly classified into a certain cluster and the total number of HR patches classified into the same cluster.

Similar to SM, five clusters are defined via the DPA. These are denoted by C^0, C^{45} ,C^{90}, C^{135} and C^{nd} respectively. The first four are directional clusters whereas C^{nd} is non-directional. The directionality of the cluster is indicated with the superscript. For each patch, angles in the gradient phase matrix are quantized to 0°, 45°, 90° and 135°. Similar to the first part of the experiment, each of the MR and HR patches is classified into one of the DPA clusters. The DPA is calculated as specified in chapter 4. The DPA scale-invariance is defined as the ratio between the number of MR patches correctly classified into a certain DPA cluster and the total number of HR patches in that cluster. SM and DPA invariance ratios are listed in the left and right sides of Table 5.1, respectively.

In view of Table 5.1, it is clear that SM is, in general, strongly scale-invariant for the first (unsharp) and the last (sharpest) clusters. This is particularly true for the case of images rich with texture, edges and corners. Another observation is that the unsharp cluster C_1 contains significantly more patches than other two clusters.

Table 5.1: Number of HR patches in each cluster (top) and the cluster scale invariance ratio (bottom). Clustering is done by SM for C_1 through C_3 and DPA for C^0 through C^{nd}. The largest number of patches in a cluster with the corresponding percentage is in bold face.

Image	SM Cluster			DPA Clusters				
	C_1	C_2	C_3	C^0	C^{45}	C^{90}	C^{135}	C^{nd}
Barbara	**3686**	1583	1956	1819	580	466	508	**3852**
	99.84	39.67	12.88	45.52	43.28	87.34	52.36	**72.04**
BSDS 198054	**1686**	1228	1326	555	126	130	163	**3266**
	99.41	15.80	66.52	71.89	58.73	61.54	55.21	**82.70**
Butterfly	**833**	329	602	126	224	269	47	**1098**
	98.32	72.34	76.58	80.95	74.55	92.94	72.34	**66.03**
Fence	**698**	448	618	624	49	252	12	**827**
	99.71	21.65	28.64	68.91	65.31	59.52	66.67	**76.54**
Flowers	**3064**	1257	659	453	249	412	185	**3681**
	99.61	39.46	47.04	72.41	64.26	71.12	68.65	**79.63**
Input6	4075	**4215**	3414	529	404	951	441	**9379**
	99.48	**41.02**	36.70	80.15	71.29	75.39	73.02	**72.95**
Lena	**5752**	1114	359	894	171	157	474	**5529**
	99.90	52.33	30.36	85.91	69.59	73.25	82.07	**78.15**
Man	**4181**	1989	1055	871	236	550	292	**5276**
	99.71	29.36	23.98	72.10	68.64	75.45	75.34	**76.12**
ppt3	**7027**	950	1615	636	346	814	299	**7497**
	98.72	68.32	74.30	78.77	62.14	86.98	61.87	**86.27**
Starfish	**967**	432	365	101	119	228	149	**1167**
	99.28	62.96	31.51	66.34	73.95	78.51	79.87	**77.21**
TextImage1	5188	490	**5658**	288	1	8	3	**11036**
	49.67	17.35	**34.54**	12.85	0.00	12.50	0.00	**76.11**
Texture	474	850	**9912**	**6602**	1319	74	97	3144
	81.65	62.35	**78.97**	**82.70**	76.19	82.43	69.07	56.04
Average	92.35	40.88	53.72	73.88	66.97	78.31	68.43	76.19

The fact that many patches are classified into the unsharp cluster suggests a possible usage of a simple interpolation technique for super-resolving these patches. This is because patches of low SM values can have high frequencies. Still, these frequencies are negligibly small in magnitude. Interpolation is therefore an effective technique to handle such patches. About DPA invariance, it is noted that the invariance ratio for all the images in all clusters is generally higher than 50 %, especially for the clusters containing the majority of patches.

To further investigate the invariance of SM and DPA, a similar experiment is conducted while using the two measures for clustering. Clustering is carried out in two levels. In the first level, image patches are clustered into C_1, C_2 and C_3 based on SM values. In the second level, patches of each of the three SM clusters C_1, C_2 and C_3 are further clustered based on their DPA value into the five aforementioned DPA clusters (C^0 through C^{nd}). In total, 15 clusters are obtained. These are denoted by C_1^0 though C_3^{nd}, where the subscript denotes the SM cluster and the superscript denotes the DPA cluster. Fig. 5.1 illustrates this two-level clustering process. Then, the scale-invariance of SM and DPA together is calculated as the number of MR patches correctly classified in a specific cluster and the total number of HR patches in that cluster. Scale-invariance ratios for the images set shown in Fig. 3.1 are listed in Table 5.2.

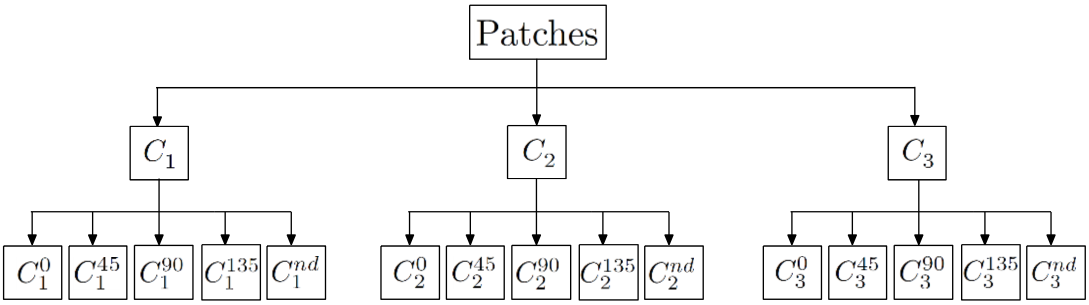


Figure 5.1: The proposed 2-level clustering scheme.

Table 5.2: Number of HR patches in each cluster (top) and the percentage of the corresponding LR patches correctly classified into the same cluster (bottom) by applying SM then DPA. The largest number of patches in a cluster with the corresponding percentage is in bold face.

Image	C_1					C_2					C_3				
	$C_1{}^0$	$C_1{}^{45}$	$C_1{}^{90}$	$C_1{}^{135}$	$C_1{}^{nd}$	$C_2{}^0$	$C_2{}^{45}$	$C_2{}^{90}$	$C_2{}^{135}$	$C_2{}^{nd}$	$C_3{}^0$	$C_3{}^{45}$	$C_3{}^{90}$	$C_3{}^{135}$	$C_3{}^{nd}$
Barbara	421	175	252	167	**2671**	528	214	150	128	563	870	191	64	213	618
	61.28	72.00	88.89	77.84	**76.19**	40.91	21.96	61.33	59.38	13.50	4.60	1.57	57.81	5.16	19.74
BSDS 198054	52	20	55	18	**1541**	146	34	36	35	977	357	72	39	110	748
	63.46	40.00	54.55	33.33	**90.07**	23.29	14.71	19.44	20.00	9.31	62.75	54.17	35.90	47.27	38.64
Butterfly	29	57	54	10	**683**	21	48	76	11	173	76	119	139	26	242
	62.07	68.42	83.33	60.00	**73.94**	33.33	60.42	65.79	54.55	36.42	81.58	68.07	76.26	73.08	31.82
Fence	21	16	144	5	**512**	108	18	77	4	241	495	15	31	3	74
	14.29	75.00	52.08	60.00	**79.88**	37.96	5.56	19.48	25.00	8.30	31.72	6.67	12.90	0.00	9.46
Flowers	228	100	168	92	**2476**	135	84	143	55	840	90	65	101	38	365
	69.30	62.00	69.64	70.65	**83.20**	48.15	32.14	30.77	21.82	24.29	45.56	47.69	45.54	65.79	22.47
Input6	72	81	221	79	**3622**	174	132	314	153	3442	283	191	416	209	2315
	65.28	56.79	61.99	53.16	**83.49**	54.02	45.45	46.50	58.17	20.28	53.71	46.07	44.95	52.63	12.83
Lena	489	74	100	193	**4896**	298	65	49	195	507	107	32	8	86	126
	82.21	59.46	68.00	75.13	**80.60**	58.39	55.38	57.14	63.08	18.15	24.30	46.88	50.00	48.84	7.94
Man	316	99	236	115	**3415**	344	91	219	101	1234	211	46	95	76	627
	62.97	64.65	70.76	67.83	**79.03**	33.14	32.97	44.75	48.51	10.86	34.12	43.48	27.37	27.63	7.50
ppt3	88	213	244	151	**6331**	169	66	238	47	430	379	67	332	101	736
	38.64	49.77	63.52	43.71	**91.31**	73.37	75.76	60.08	55.32	34.42	60.69	67.16	74.40	61.39	41.85
Starfish	28	45	108	66	**720**	34	42	73	44	239	39	32	47	39	208
	53.57	68.89	81.48	74.24	**75.97**	55.88	64.29	61.64	79.55	38.91	41.03	43.75	25.53	41.03	15.87
TextImage1	0	0	0	0	5188	0	0	0	0	490	288	1	8	3	**5358**
	N.A.	N.A.	N.A.	N.A.	47.96	N.A.	N.A.	N.A.	N.A.	12.65	11.46	0.00	12.50	0.00	**18.51**
Texture	48	12	2	0	412	272	59	0	4	515	**6282**	1248	72	93	2217
	47.92	41.67	0.00	N.A.	52.18	44.49	35.59	N.A.	0.00	25.83	**68.48**	74.12	80.56	70.97	41.00
Average	66.41	60.87	69.82	65.85	77.30	45.27	39.04	48.58	54.57	18.80	56.51	60.70	54.88	42.53	23.27

In view of Table 5.2, it can be seen that for images with sharp details such as texture and edges, SM and DPA in the clusters that contain the majority of patches are strongly scale-invariant. However, it is moderate for the other clusters. For the case of regular natural images such the Lena the overall scale-invariance is not as strong.

The above findings point out that each of SM and DPA independent of each other approximately is scale-invariant. Accordingly, it is promising to combine the two quantities as an approximately scale-invariant measure for the purposes of classification and sparse model selection.

5.2.2 Clustering and Sparse Model Selection with the Patch Sharpness Measure and Dominant Gradient Phase Angle

The proposed algorithm is composed of two stages. The first one is the training stage, where a set of dictionary pairs is prepared. The second stage is the reconstruction stage where the best dictionary pair is selected to sparsely reconstruct HR patches from the corresponding LR patches.

A HR image set is required for the training stage. A LR version of each HR image is obtained by filtering it with a bicubic kernel and then downsampling it by a scale factor of 2 in the two dimensions. Each LR image is then interpolated by a scale factor of 2 to the dimensions of the corresponding HR image known as the MR level. Then, features are obtained by applying feature extraction filters on the MR images, as done in [50]. Dictionary learning and sparse coding of the LR patches is done with these features. This is shown to be more advantageous than dealing with LR patches directly [2, 50].

LR and HR patches corresponding to the same spatial location are handled as pairs. Each MR patch is then classified into a specific cluster based on its SM and DPA value. The HR patch in the pair is placed into the same cluster. The mean value of each HR patch is subtracted to allow for better dictionary learning. LR and HR patches of each cluster are used to train for a pair of coupled LR and HR cluster dictionaries, respectively. For this purpose, the method proposed in [50] is used. Algorithm 11 outlines the main steps of the training stage. In this setting, the purpose is to design cluster dictionaries that correspond to different sharpness natures, and have the desired directional structures.

Algorithm 11 The Proposed Cluster DL Algorithm.

INPUT: HR training image set.
OUTPUT: A set of directional cluster dictionary pairs with varying sharpness.

```
Divide each HR image into patches and combine them column-wise to form a HR training array.
Blur and downsample each HR image to the LR and divide it into patches.
Upsample each LR image to the MR level.
Apply feature extraction filters on each MR image, reshape and column-stack them.
For Each patch in the LR training array,
    Calculate SM and DPA of the MR patch, and find the cluster number.
    Add the MR patch to the LR training set of this cluster.
    Add the corresponding HR patch to the HR training set of this cluster.
end for
For each cluster, learn a pair of coupled dictionaries.
```

In the reconstruction stage, a LR image is first upsampled using bicubic interpolation to the MR level. Features are extracted by applying feature extraction filters and then reshaped into the vector form. A certain patch overlap (for a patch size of 5x5, 4 patch overlap) is allowed to assure local consistency between the reconstructed patches [50]. The SM and DPA values of each MR patch are calculated and the cluster that the MR patch belongs to is identified. Using the dictionary pair of the identified cluster, first the sparse representation coefficient vector of the corresponding MR feature vector over the cluster LR dictionary is calculated. Then the HR patch is reconstructed by right-multiplying the cluster HR dictionary with the sparse representation coefficient vector. The same process is repeated for all MR patches. Eventually, the reconstructed HR patches are reshaped into the 2-D form and merged to constitute the HR image estimate. A summary of the proposed reconstruction algorithm is outlined in Algorithm 12.

Algorithm 12 The Proposed Single-Image Super-resolution Algorithm.

INPUT: A LR test image, cluster dictionary pairs.
OUTPUT: A HR image estimate

```
Divide the LR image into overlapping patches.
Upsample the LR image to the required resolution level (MR).
Apply feature extraction filters on the MR image.
Divide the extracted features into overlapping patches and reshape them into vectors.
for Each MR patch
    Calculate SM and DPA of the MR patch.
    Determine the cluster this patch belongs to.
    Sparsely code the features of the MR patch over the cluster LR dictionary.
    Reconstruct the corresponding HR patch by right-multiplying the HR dictionary of the same cluster with the sparse codes of the MR features.
end for
Merge overlapping patches to obtain a HR image estimate.
```

5.2.3 Computational Complexity of the Proposed Algorithm

The proposed algorithm separates the training and testing patches into 3 main clusters based on their SM values. Patches of the first cluster have small SM values. They may correspond to high frequencies but with very small magnitudes. Therefore, it is expected that a simple interpolation technique will suffice to super-resolve these patches. As will be shown in the next section, bicubic interpolation seems as a good technique for that purpose. It has been revealed in Table 5.1 that a relatively large percentage of patches of a given image are located in this cluster. Employing bicubic interpolation to handle such patches is thus promising in significantly saving the computational cost as compared to the case of employing a dictionary pair for all clusters.

The cluster dictionary pairs of the proposed algorithm are learned off-line. During the testing stage, the proposed algorithm calculates the SM and DPA values of each patch. Due to the use of multiple pairs of dictionaries, one can use dictionaries with smaller number of atoms compared to the algorithms that employ a single dictionary pair such as Yang et al. [50]. Thus, the computational complexity of the sparse coding stage can be reduced.

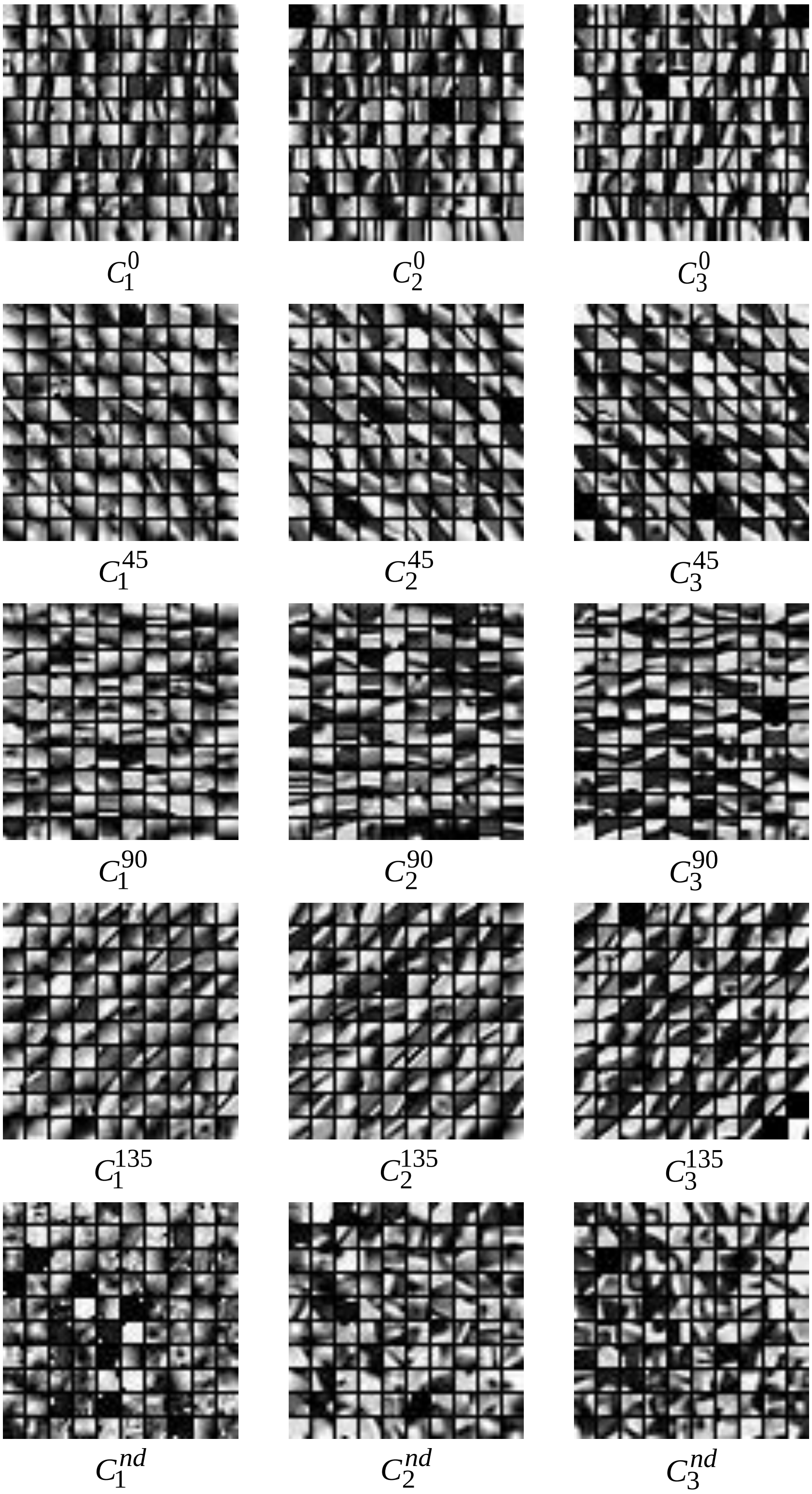
5.3 Experimental Validation

In this section, the performance of the proposed algorithm is examined and compared to several super-resolution algorithms. These include the basic sparse representation-based SR algorithm of Yang et al. [50] which employs one dictionary pair. Besides, we also include the SR algorithms of Peleg et al. [70] and He et al. [46] as state-of-the-art techniques. These algorithms are different in nature. In order to have fair comparisons, care has been taken to ensure that the parameters used in the training and testing stages are as close to each other for all algorithms as possible. If a parameter is unique to a specific algorithm, the value suggested by the authors is used. Image SR results for a scale factor of 2 are presented. However, the proposed algorithm can easily be modified for other scale factors.

Dictionaries of the proposed algorithm are learned as specified in Algorithm 11. Example reshaped atoms of these dictionaries are shown in Fig. 5.3. Dictionary training for the proposed algorithm is done over the 1000-image Flickr dataset [79], along with several typical text images (Fig 3.4) as done in section 3.4. These text images are added to the training set to be sure about the availability of enough patches with relatively high SM values. The clustering of LR and HR training patches is carried out in terms of the SM and DPA of each LR patch, as outlined in Section 5.2. We then randomly selected 40,000 pairs of LR and HR training patches for each cluster. 600-atom dictionary pairs are designed for the proposed algorithm.

As for the example atoms shown in Fig. 5.2, it can be clearly observed that the designed dictionaries inherit the sharpness nature and directionality of their respective clusters. It is notable that the atoms of the dictionaries in C_3 (shown in the last column of Fig. 5.2) are sharper than the corresponding ones in C_2 (shown in the second column) and those in C_1 (shown in the first column). One can clearly see that the first four sub-figures in the first column of Fig. 5.2 have atoms perpendicular to the 0°, 45°, 90° and 135° orientations, respectively. The same observation can be made about the corresponding sub-figures in the second and the third columns, even though they vary in sharpness. Moreover, the dictionaries in C_1^{nd}, C_2^{nd} and C_3^{nd} exhibit a chaotic directional nature. This points out that the designed dictionaries are directionally structured in the specified orientations.

For the algorithm of Yang et al.[50], a single dictionary pair with 1000 atoms is learned. For the learning, 40,000 pairs of LR and HR patches are randomly selected



C_1^0 C_2^0 C_3^0

C_1^{45} C_2^{45} C_3^{45}

C_1^{90} C_2^{90} C_3^{90}

C_1^{135} C_2^{135} C_3^{135}

C_1^{nd} C_2^{nd} C_3^{nd}

Figure 5.2: Reshaped example atoms of the 15 HR cluster dictionaries.

from the same training set used by the proposed algorithm. We used the default design parameters and training image datasets for the algorithms of Peleg et al.[70] and He et al.[46] as specified by the authors.

Similar to the experiment presented in Section 3.4, an experiment is carried out to examine the discrimination of the designed cluster dictionary pairs. For this purpose, patch pairs of the same training set are classified into the aforementioned 15 clusters of the proposed algorithm as illustrated earlier. For each cluster its training data is used as the testing patch pairs for this experiment. The HR patch in each cluster is reconstructed with every cluster dictionary pair based on its LR counterpart. HR patch reconstruction is done similar to the super-resolution problem. This means that the sparse coding coefficients of the MR patch are calculated. Then, a HR patch estimate is obtained by multiplying these coefficients with the HR dictionary. The mean value of the MR patch is added to this multiplicand to form an estimate of the HR patch. The MSE between the ground-truth HR patch and each of its 15 reconstructions is recorded. For comparison, a HR patch in each cluster is also reconstructed with the single dictionary pair of Yang et al.'s [50] algorithm and bicubic interpolation. Results are listed in Table 5.3.

In view of Table 5.3, it is clear that the dictionary pair of each cluster is the best to represent patches of that cluster as compared to the other cluster dictionary pairs, the single dictionary pair of Yang et al. [50] and bicubic interpolation. However, data in C_1^{nd} is an exception; bicubic interpolation is slightly better in representing them than the dictionary pair of this clusters. This is due to the fact that patches of this cluster

Table 5.3: MSE reconstruction quality for the patches in the 15 clusters with the dictionary pair of each cluster, the single dictionary pair of Yang et al. (SD) and with bicubic interpolation.

Data in / Dict.s	C_1					C_2					C_3					SD	Bicubic
	${C_1}^0$	${C_1}^{45}$	${C_1}^{90}$	${C_1}^{135}$	${C_1}^{nd}$	${C_2}^0$	${C_2}^{45}$	${C_2}^{90}$	${C_2}^{135}$	${C_2}^{nd}$	${C_3}^0$	${C_3}^{45}$	${C_3}^{90}$	${C_3}^{135}$	${C_3}^{nd}$		
${C_1}^0$	**70.87**	83.70	87.45	82.68	84.38	73.77	84.96	86.88	83.14	76.76	77.42	84.88	90.98	85.28	77.86	80.16	75.50
${C_1}^{45}$	42.67	**33.95**	42.66	46.08	42.32	43.71	35.54	42.80	48.45	38.87	46.84	38.88	47.23	51.19	41.28	40.89	37.09
${C_1}^{90}$	53.12	49.04	**42.16**	48.68	48.97	54.35	50.11	43.83	49.49	45.16	55.94	52.26	46.24	51.66	46.74	48.81	43.67
${C_1}^{135}$	46.23	50.25	45.54	**36.53**	46.52	46.29	51.98	46.11	38.39	41.41	50.73	55.02	50.67	41.22	44.01	43.66	40.44
${C_1}^{nd}$	18.45	18.25	18.07	18.56	16.19	18.60	18.31	18.10	18.68	17.05	19.77	19.68	20.08	19.59	18.78	18.50	**14.10**
${C_2}^0$	294.91	357.10	360.18	348.95	348.24	**273.24**	349.12	352.24	340.46	301.14	284.21	341.37	363.30	342.47	295.02	319.54	322.75
${C_2}^{45}$	244.56	189.00	241.04	267.02	241.97	238.61	**174.87**	232.78	273.36	197.22	247.26	186.62	249.62	281.14	202.75	214.17	226.96
${C_2}^{90}$	401.01	375.84	330.03	375.56	370.40	402.52	379.46	**300.58**	374.25	327.49	401.48	382.15	315.81	380.23	330.78	356.02	356.68
${C_2}^{135}$	244.08	270.39	244.61	189.59	242.92	234.20	271.63	235.03	**175.64**	197.54	249.78	283.57	248.34	186.83	204.56	215.05	227.32
${C_2}^{nd}$	455.99	459.41	454.35	457.50	455.56	445.35	456.67	440.76	449.47	**389.45**	456.52	462.21	462.67	457.80	413.17	442.64	455.15
${C_3}^0$	772.92	1056.32	983.82	995.49	956.85	704.33	1006.21	986.82	943.57	797.27	**567.06**	954.01	1000.68	937.18	675.07	808.46	930.51
${C_3}^{45}$	715.85	484.46	686.26	783.87	699.11	670.10	426.21	653.14	812.45	489.63	677.12	**372.02**	670.83	816.85	464.28	535.95	664.77
${C_3}^{90}$	1051.33	993.12	798.80	988.96	941.99	1053.50	999.48	727.89	975.10	783.75	1041.49	983.84	**609.26**	974.84	750.02	853.35	947.18
${C_3}^{135}$	698.56	782.26	702.75	478.65	691.11	650.64	789.91	649.97	423.39	502.11	674.45	829.74	673.11	**371.36**	470.26	539.85	663.32
${C_3}^{nd}$	1355.99	1388.08	1354.35	1377.77	1376.38	1310.95	1370.73	1303.05	1343.98	1167.29	1274.32	1336.74	1290.64	1323.30	**1050.34**	1301.60	1387.14

are unsharp and non-directional. This means that they have few high frequency components. Therefore, the designed cluster dictionaries are well-suited to represent patches of their own clusters. It can also be concluded that patches in sub-clusters of C_1 (C_1^0 through C_1^{nd}) may still be effectively represented with dictionaries of the other sub-clusters, the single dictionary pair of Yang et al.[50] or bicubic interpolation without a significant MSE difference. However, error levels for sharper clusters C_2 and C_3 are higher. Thus, dictionaries in the sub-clusters of C_2 and C_3 are clearly more discriminative than the ones in C_1. This result suggests the possibility of employing bicubic interpolation to super-resolve patches in C_1. Given that there are a large percentage of patches in C_1 as seen in Table 5.1, this employment will significantly reduce the SR computational complexity without sacrificing the representation quality. In this setting, the proposed algorithm uses the sparse representation reconstruction only for patches of C_2 and C_3.

Table 5.4 lists the PSNR (top) and SSIM (bottom) values of super-resolution reconstructions obtained with the aforementioned algorithms and setups with respect to the original ground-truth image. It is noted that we have conducted the simulations with source codes provided by the authors of [50], [46] and [70]. Two cases are shown for the proposed algorithm. In the first case, all the cluster dictionary pairs are used. This is denoted by (Prop.600). In the second case, any LR patch classified into C_1 is super-resolved with bicubic interpolation. The second case is denoted by (Prop.600+bic). The proposed algorithm in both cases performs better than the algorithm of Yang et al.[50]. The proposed algorithm has average PSNR improvements of 0.42 dB and 0.35 dB over the algorithm of Yang et al.[50] for the first and the second cases, respectively. These improvements validate the added-benefit of using multiple cluster dictionary pairs instead of a single general dictionary pair.

Table 5.4: PSNR (dB) and SSIM comparisons of bicubic interpolation, the algorithms of Peleg et al. [70], Yang et al. [50] and He et al. [46] and the proposed algorithm, respectively.

Image	Bicubic	Yang et al.	Peleg et al.	He et al.	Prop. 600	Prop. 600+bic
Barbara	25.35	25.86	25.76	25.84	25.86	**25.87**
	0.7930	0.8357	0.8359	**0.8372**	0.8353	0.8350
BSDS 198054	24.75	26.85	26.76	26.98	**27.19**	27.16
	0.8267	0.8816	0.8760	**0.8839**	0.8835	0.8810
Butterfly	27.46	31.26	30.96	31.44	**32.07**	31.93
	0.8985	0.9457	0.9227	0.9463	**0.9511**	0.9478
Fence	25.05	26.34	26.17	26.22	**26.46**	26.41
	0.7449	0.8037	0.7967	0.8045	**0.8069**	0.8031
Flowers	30.42	32.76	32.57	**32.98**	32.97	32.85
	0.8828	0.9005	0.8522	**0.9018**	0.9000	0.8980
Input6	28.08	30.15	30.03	30.24	**30.38**	30.35
	0.8523	0.8976	0.8737	0.8989	**0.9001**	0.8986
Lena	34.71	36.36	**36.59**	36.58	36.44	36.39
	0.8507	0.8631	0.8387	**0.8647**	0.8633	0.8630
Man	29.26	30.68	30.67	30.80	**30.85**	30.79
	0.8315	0.8713	0.8729	**0.8739**	0.8727	0.8702
ppt3	26.85	29.68	29.71	29.79	**30.51**	30.44
	0.9372	0.9604	0.9494	0.9621	**0.9646**	0.9629
Starfish	30.23	32.66	32.92	**33.28**	33.05	32.99
	0.8923	0.9279	0.9165	**0.9325**	0.9302	0.9292
TextImage1	17.52	18.58	18.73	18.54	**18.89**	18.87
	0.7246	0.7974	0.8118	0.7953	**0.8124**	0.8096
Texture	20.64	22.55	22.97	22.78	**23.05**	23.05
	0.8272	0.8939	0.9032	0.8997	**0.9040**	0.9039
Average	26.69	28.64	28.65	28.79	**28.98**	28.93
	0.8385	0.8816	0.8708	0.8834	**0.8853**	0.8835

In view of Table 5.4, one notices that the success of the proposed algorithm is particularly valid for images with sharp features such as text images, the Butterfly and the ppt3 images. Considering the Butterfly image, as an example, it can be seen in Table 5.1 that the majority of patches are located in C_1 and C_3 for which SM is highly scale-invariant. Besides, a high percentage of patches is located in C^{nd} for which DPA is strongly scale-invariant. In view of Table 5.2, the largest percentage of patches is located in cluster C_1^{nd} for which the scale-invariance of the two measures is strong. It can be seen generally that the scale-invariance of SM and DPA reported in Tables 5.1 and 5.2 is reflected to the PSNR and SSIM performances listed in Table 5.4.

It can be seen in Table 5.4 that the proposed algorithm is competitive with the state-of-the-art algorithms of Peleg et al. [70] and He et al. [46]. The proposed algorithm has average PSNR improvements of 0.33 dB and 0.19 dB over the algorithms of Peleg et al. [70] and He et al. [46], respectively. For the case of employing bicubic interpolation for the LR patches of C_1, the average improvements are 0.28 dB and 0.14 dB, respectively. SSIM simulations validate the above observations and conclusions.

Figure 5.3 compares a portion of the original Butterfly image to its reconstructions obtained with bicubic interpolation, Peleg et al. [70], Yang et al. [50], He et al. [46] and the proposed algorithm. Amongst all reconstruction methods, the proposed algorithm's reconstruction is the best to approximate the ground-truth scene. The proposed algorithm is better able to reconstruct edges. This is particularly seen by comparing the curvy line along the butterfly's wing and the patterns on the wing. Fig. 5.3 (g), (h) and (i) show respectively the difference between the original scene and its reconstructions from Yang et al. [50], He et al. [46] and proposed algorithm. Clearly, the proposed algorithm has least amount of artifacts.

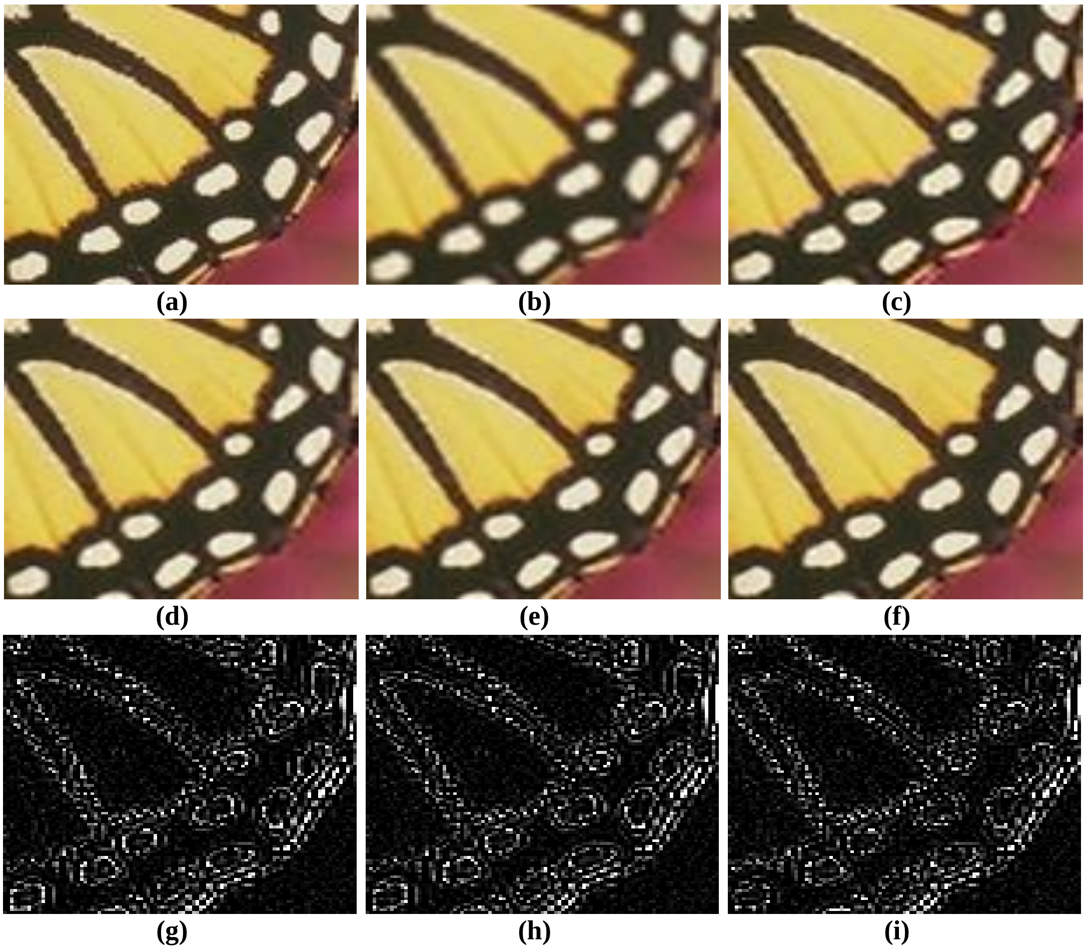
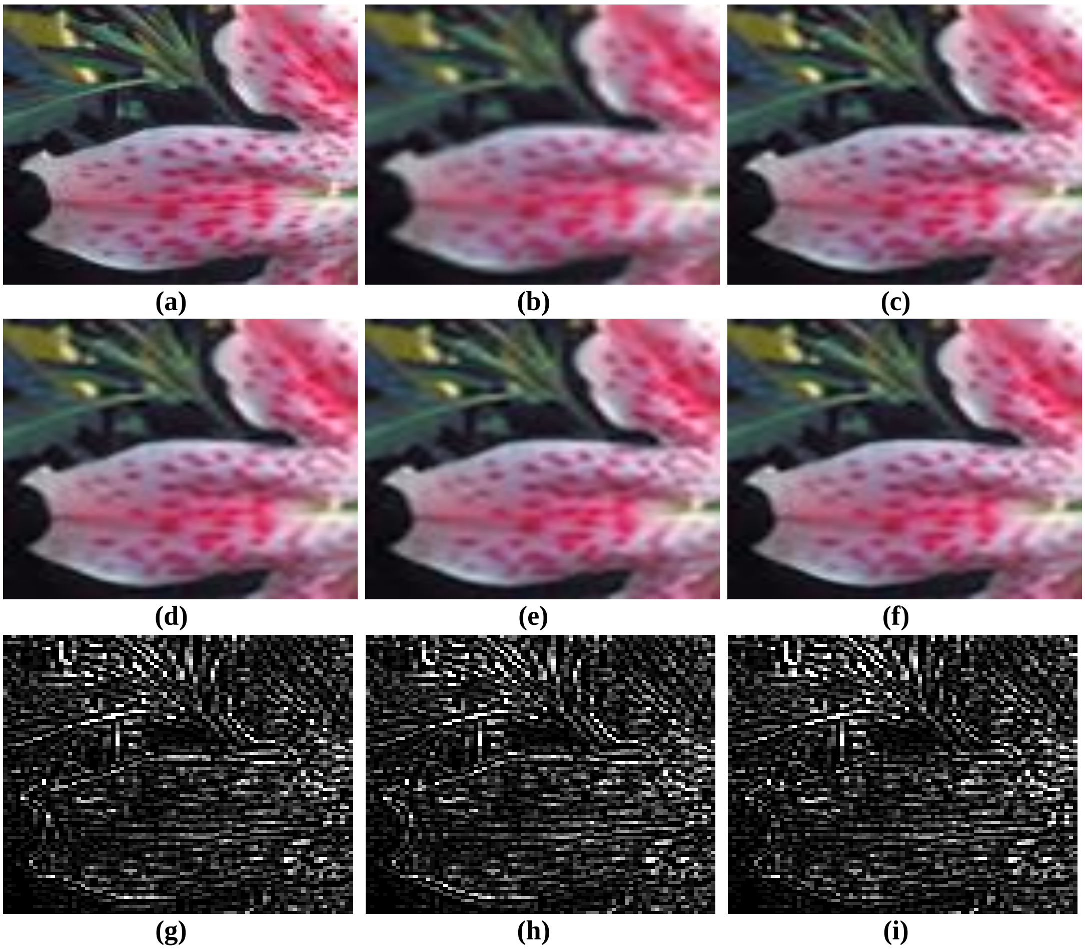


Figure 5.3: Visual comparison of the Butterfly image: (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Peleg et al. [70], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

Figure 5.4 conducts the same comparison with the Flowers image. Similar to the case of Fig. 5.3, the proposed algorithm is better able to conserve image details such as


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.4: Visual comparison of the Flower image: (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Peleg et al. [70], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

curvy edges. This is clearly seen in the boundaries of the flower's leave. Fig. 5.4 (g), (h), (i) show respectively the difference images. It is obvious that proposed algorithm produces less artifacts around sharp features.

As a final visual comparison, Fig. 5.5 shows reconstructions of the ppt3 image. One can make the same conclusions made in last two figures. It is obvious that the sharp boundaries of the letters and the hollow areas in the interior of the text are better reconstructed with the proposed algorithm. The other algorithms generate artifacts in

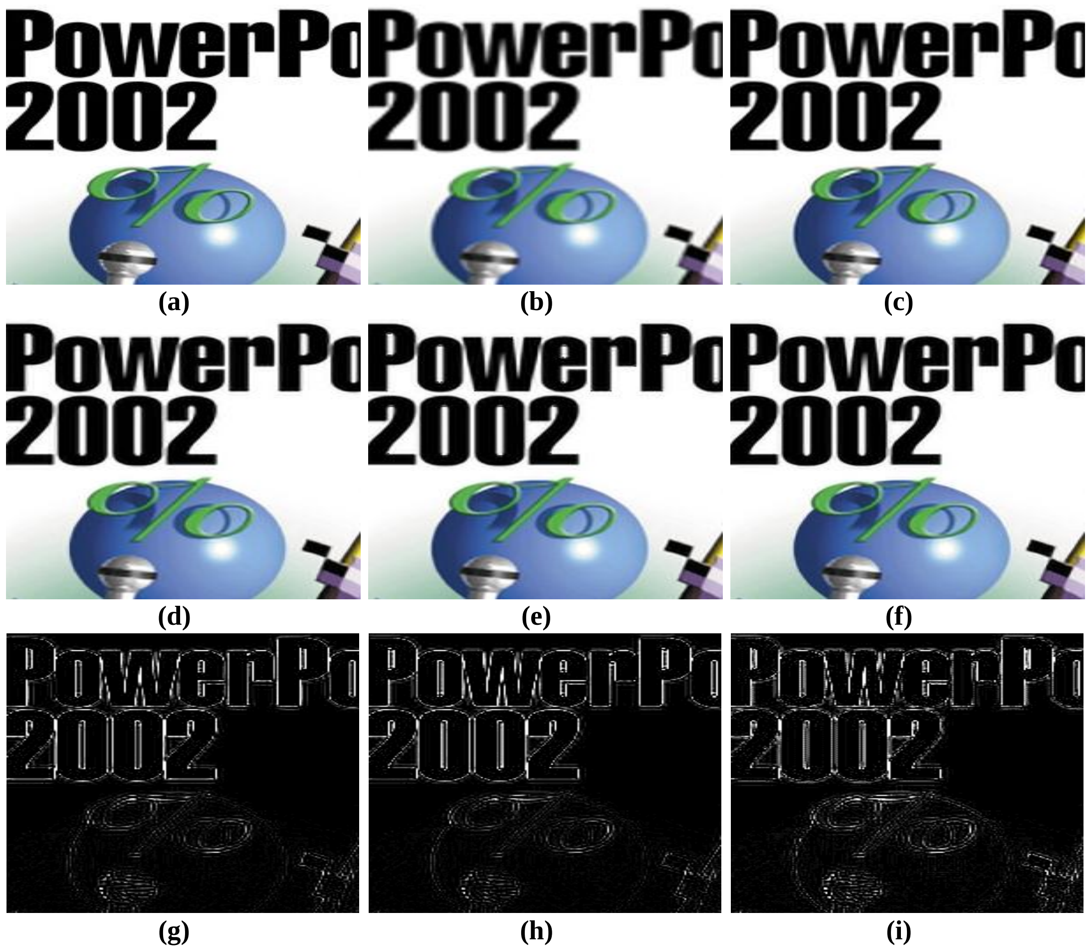


Figure 5.5: Visual comparison of the ppt3 image: (a) Original, and reconstructions of (b) Bicubic interpolation, (c) Peleg et al. [70], (d) Yang et al. [50], (e) He et al. [46] and (f) the proposed algorithm. The last row shows the difference between the original image and reconstructions of:(g) Yang et al., (h) He et al. and (i) The proposed algorithm, respectively.

these regions. This observation can be verified by comparing the difference images between the ground-truth image and different reconstructions (Fig. 5.5 (g), (h), (i)).

To inspect the representation capability of the designed cluster dictionaries, the following experiment is conducted on the images shown in Fig. 3.1. Four scenarios of SR are applied and the average PSNR and SSIM values of the reconstructed images are recorded. In the first scenario, SM is only used as a classification and model selection criterion. Three main clusters corresponding to SM intervals of [0, 10], [10, 20] and [20, 255] are considered. A 1000-atom dictionary pair is designed for each cluster. In the second scenario, the same operation is repeated with the same dictionaries by applying perfect model selection. In this context, perfect model selection is carried out by super-resolving each LR patch with each of the three cluster dictionary pairs. Then the super-resolved patch that is closest to the ground-

Table 5.5: PSNR and SSIM values for the case of using 3 SM clusters, 3 SM clusters with perfect model selection, 15 clusters defined by SM and DPA and 15 clusters defined by SM and DPA with perfect model selection.

Image	S_1	S_2	S_3	S_4
Barbara	25.82 0.8345	26.22 0.8493	25.86 0.8353	26.65 0.8619
BSDS 198054	27.16 0.8829	27.93 0.9001	27.19 0.8835	28.45 0.9089
Butterfly	31.89 0.9483	32.75 0.9561	32.07 0.9511	33.39 0.9607
Fence	26.35 0.8026	27.12 0.8265	26.46 0.8069	27.62 0.8405
Flowers	32.92 0.8997	33.80 0.9099	32.97 0.9000	34.43 0.9135
Input6	30.36 0.8991	31.13 0.9131	30.38 0.9001	31.44 0.9174
Lena	36.30 0.8630	37.17 0.8714	36.44 0.8633	37.50 0.8739
Man	30.77 0.8710	31.43 0.8874	30.85 0.8727	31.88 0.8958
ppt3	30.18 0.9628	30.99 0.9679	30.51 0.9646	32.08 0.9709
Starfish	32.79 0.9292	33.77 0.9407	33.05 0.9302	34.45 0.9453
TextImage1	18.92 0.8136	19.41 0.8292	18.89 0.8124	19.92 0.8476
Texture	22.70 0.8967	23.74 0.9177	23.05 0.9040	24.75 0.9342
Average	28.85 0.8836	29.62 0.8974	28.98 0.8853	30.21 0.9059

truth HR patch in the MSE sense is selected. In the third scenario, SM and DPA are together used to design 15 cluster dictionary pairs. Each cluster dictionary has 600 atoms. SM and DPA are then used together as a model selection criterion according to the proposed algorithm. The same process is repeated in the fourth scenario but with applying perfect model selection. Average PSNR and SSIM values for the four scenarios are listed in Table 5.5 and denoted by S_1, S_2, S_3 and S_4, respectively.

In view of Table 5.5, it can be concluded that using 15 cluster dictionaries each of 600 atoms defined by SM and DPA is better than using 3 SM cluster dictionaries of 1000 atoms. This is the case for SM and DPA bases section and perfect model selection.

Chapter 6

COMPUTATIONAL COMPLEXITY REDUCTION BY USING BICUBIC INTERPOLATION

6.1 Introduction

This chapter presents attempts to reduce the computational complexity levels of the proposed SR algorithms. The key idea in doing so is to employ bicubic interpolation on patches of insignificant HF components. In Chapters 3, 4 and 5 it has been shown that a high percentage of patches are located in clusters of low SM values and the non-directional clusters, respectively. Therefore, a simple reconstruction technique such as bicubic interpolation will efficiently reconstruct these patches. This means that the expensive sparse representation-based reconstruction will be reserved for patches of significantly HF components which form a minority of the overall patches. This will significantly reduce the SR computational complexity without sacrificing the reconstruction quality, as will be explained in the following sections.

6.2 Bicubic Interpolation for Clusters of Low SM Values

In this experiment, we analyze the applicability of bicubic interpolation in super-resolving patches of low SM clusters in the SM-based SR algorithm presented in Chapter 3 [87]. It is concluded in chapter 3 (Table 3.2) that clusters of low SM values have poor dictionary discrimination. This observation also motivates bicubic interpolation for such clusters. Setting the number of clusters to seven, it is interesting to investigate the performance of bicubic interpolation in super-resolving patches of the images in Fig. 3.1 in different clusters. For this purpose, six scenarios are considered and labeled as S_1 through S_6. In the first scenario S_1, all LR patches in C_1 are super-resolved with bicubic interpolation, while others in the remaining clusters are super-resolved using their respective cluster dictionary pairs. In the second scenario S_2, patches in C_1 and C_2 are super-resolved with bicubic interpolation, and so on, till the scenario S_6 where only patches in C_7 are super-resolved with the sparse framework, whereas patches in the other clusters are super-resolved with bicubic interpolation. This also includes the case of using the sparse reconstruction with seven clusters, as proposed in [72]. For a set of test images, PSNR (top) and SSIM (bottom) values of the aforementioned scenarios are listed in Table 6.1. Besides, PSNR and SSIM values of the SR algorithms of Zeyde et al. [47], Yang et al. [50] and He et al. [46] are also shown in Tanle 3.3 before.

In view of Table 6.1, it is evident that the scenario S_4 represents a good compromise between the performance and the computational complexity. We opt to adopt this scenario in this work. S_4 is 0.08 dB on average less than the algorithm in [72] (Prop.7). Beside, S_4 is superior to the algorithm of Zeyde et al. [47] and Yang et al. [46] with average PSNR differences of 0.39 dB and 0.09 dB, respectively. Also, S_4 is competitive with the algorithm of He et al. [46]. SSIM values come inline with the PSNR performances.

To analyze the reduction in computational complexity for each scenario, Table 6.2 shows the percentage of LR patches that are super-resolved with bicubic interpolation, to the total number of patches. When applying S_4, 74.8 % of patches are super-resolved with bicubic interpolation. This means a significant reduction in the computational cost of the algorithm presented in [87].

Table 6.1: PSNR (dB) and SSIM comparisons.

Image	Prop.7	S1	S2	S3	S4	S5	S6
Barbara	**25.87** **0.8361**	25.82 0.8353	25.82 0.8346	25.85 0.8342	25.85 0.8337	25.85 0.8335	25.84 0.8330
BSDS 198054	27.05 0.883	**27.23** **0.8844**	27.22 0.8827	27.18 0.8809	27.14 0.8795	27.10 0.8784	27.02 0.8773
Butterfly	31.68 0.948	**31.81** **0.9478**	31.73 0.9458	31.65 0.9443	31.51 0.9426	31.33 0.9402	31.06 0.9371
Fence	**26.41** **0.8046**	26.37 0.8046	26.34 0.8026	26.30 0.8001	26.28 0.7986	26.24 0.7968	26.22 0.7957
Flowers	32.85 **0.9005**	**32.97** 0.9002	32.91 0.8990	32.81 0.8979	32.70 0.8972	32.55 0.8965	32.39 0.8959
Input6	30.31 **0.8991**	30.36 0.8991	**30.34** 0.8982	30.30 0.8971	30.22 0.8957	30.12 0.8943	30.01 0.8927
Lena	**36.37** **0.8634**	36.35 0.8632	36.32 0.8629	36.26 0.8626	36.18 0.8623	36.13 0.8621	36.07 0.8619
Man	30.76 0.8719	**30.81** **0.8721**	30.78 0.8706	30.71 0.8691	30.62 0.8678	30.52 0.8663	30.40 0.8651
ppt3	30.04 0.9634	**30.16** **0.9636**	30.13 0.9628	30.10 0.9615	30.02 0.9595	29.92 0.9576	29.83 0.9564
Starfish	32.82 0.9298	32.87 **0.9299**	**32.85** 0.9292	32.76 0.9277	32.57 0.9255	32.31 0.9231	32.06 0.9210
TextImage1	18.85 0.81	**18.92** **0.8136**	18.91 0.8124	18.91 0.8106	18.88 0.8068	18.84 0.8028	18.78 0.7973
Texture	22.67 0.8963	22.77 0.8983	22.77 0.8982	**22.78** **0.8981**	22.77 0.8977	22.74 0.8967	22.70 0.8953
Average	28.81 0.8838	28.87 **0.8843**	**28.84** 0.8833	28.80 0.8820	28.73 0.8806	28.64 0.8790	28.53 0.8774

Table 6.2: Percentage of LR patches located in the low-sharpness clusters to the total number of image patches.

Image	C1	C1-2	C1-3	C1-4	C1-5	C1-6
Barbara	43.8	67.2	82.4	90.4	95	97.7
BSDS 198054	32.7	55.9	68.5	74.3	78.2	81.8
Butterfly	36.2	49	57.8	64.9	71.9	78.4
Fence	38.9	58.9	70.8	79.1	86.1	91.5
Flowers	39.5	66.3	81	88.3	92.4	95.1
Input6	23.4	46.4	65.4	78.1	86.6	92.1
Lena	64.7	81.2	89.8	94.9	97.6	98.9
Man	40.8	69.6	83.1	90.7	95.2	97.6
ppt3	66.4	72.2	76.9	81.9	86.4	89.4
Starfish	29.6	55.4	69.8	81.6	90.4	96.1
TextImage1	39.2	44.7	49.8	55.4	61.9	69.8
Texture	1.4	5.2	10.4	17.6	27.1	38.3
Average	38.1	56.0	67.1	74.8	80.7	85.6

6.3 Bicubic Interpolation for Low SM and Non-Directional Clusters

In this section, we attempt at reducing the computational complexity of the SM-DPA SR algorithm presented in Chapter 5 [83]. The key idea for this reduction is to use bicubic interpolation to super-resolve patches of insignificant HF components, as mentioned in the previous section. This computational complexity can be further reduced by using bicubic interpolation to super-resolve un-sharp patches [88]. Table 5.2 shows the scale-invariance ratios of Fig. 3.1. From Table 5.2 one can notice that the SM cluster C_1 contains un-sharp patches with insignificant high frequency components. This suggests the possibility of using bicubic interpolation to super-resolve such patches. Following the same logic, patches in C_2^{nd} are non-directional and can therefore be effectively super-resolved with bicubic interpolation. In contrast, C_3^{nd} is located in the sharpest SM cluster C_3, and its patches have significantly high frequency components. So, bicubic interpolation will not effectively reconstruct patches in C_3^{nd} .

In view of Table 5.2, C_1 contains a high percentage of image patches. Besides, the scale-invariance of SM and DPA in C_2^{nd} is particularly weaker than in the other clusters. In view of Table 5.3, it is concluded that the dictionaries in C_1 do not possess good discrimination. This observation is also valid for the cluster C_2^{nd} . The lack of dictionary discrimination in these clusters also motivates the applicability of bicubic interpolation in super-resolving their patches. Therefore, applying bicubic interpolation to patches in C_1 and C_2^{nd} , while the sparse representation-based SR is only applied to clusters C_2^0 though C_2^{135} and C_3 is expected to substantially reduce the SR computational complexity without sacrificing the reconstruction quality. This scenario is also studied and denoted by (P_{bic}) in Table 6.3. In this table, P_{bic} is compared with the proposed algorithm of Chapter 5 (denoted by P) and the SR algorithms of He et al. [46], Peleg et al. [70] and Yang et al. [50]. With this scenario, the proposed algorithm is still generally superior to the other algorithms with PSNR improvements of 0.16 dB, 0.16 dB and 0.29 dB over the algorithms of He et al. [46], Peleg et al. [70] and Yang et al. [50], respectively. SSIM values come inline with the PSNR performances.

Table 6.3: PSNR and SSIM comparisons. The best PSNR and SSIM results are in bold.

Image	Bicubic	Yang et al.	Peleg et al.	He et al.	Prop.600	Prop.600+Bic
Barbara	25.35	25.86	25.76	25.84	25.86	**25.87**
	0.793	0.8357	0.8359	**0.8372**	0.8353	0.8350
BSDS 198054	24.75	26.85	26.76	26.98	**27.19**	27.16
	0.827	0.8816	0.876	**0.8839**	0.8835	0.8810
Butterfly	27.46	31.26	30.96	31.44	**32.07**	31.93
	0.899	0.9457	0.9227	0.9463	**0.9511**	0.9478
Fence	25.05	26.34	26.17	26.22	**26.46**	26.41
	0.745	0.8037	0.7967	0.8045	**0.8069**	0.8031
Flowers	30.42	32.76	32.57	**32.98**	32.97	32.85
	0.883	0.9005	0.8522	**0.9018**	0.9000	0.8980
Input6	28.08	30.15	30.03	30.24	**30.38**	30.35
	0.852	0.8976	0.8737	0.8989	**0.9001**	0.8986
Lena	34.71	36.36	**36.59**	36.58	36.44	36.39
	0.851	0.8631	0.8387	**0.8647**	0.8633	0.8630
Man	29.26	30.68	30.67	30.8	**30.85**	30.79
	0.832	0.8713	0.8729	**0.8739**	0.8727	0.8702
ppt3	26.85	29.68	29.71	29.79	**30.51**	30.44
	0.937	0.9604	0.9494	0.9621	**0.9646**	0.9629
Starfish	30.23	32.66	32.92	**33.28**	33.05	32.99
	0.892	0.9279	0.9165	**0.9325**	0.9302	0.9292
TextImage1	17.52	18.58	18.73	18.54	**18.89**	18.87
	0.725	0.7974	0.8118	0.7953	**0.8124**	0.8096
Texture	20.64	22.55	22.97	22.78	**23.05**	23.05
	0.827	0.8939	0.9032	0.8997	**0.9040**	0.9039
Average	26.69	28.64	28.65	28.79	**28.98**	28.93
	0.839	0.8816	0.8708	0.8834	**0.8853**	0.8835

To analyze the reduction in computational complexity in scenario P_{bic} compared to scenario P, Table 6.4 shows the percentage of patches for which the sparse representation-based SR is applied out of the total number of patches for each image, as denoted by ($Perc$). On average, only 26.93 % of patches are super-resolved with the sparse representation framework while the others are handled with bicubic interpolation. Table 6.4 also shows the super-resolution execution times for each image with scenarios P and P_{bic}. This is denoted by (Run Time). Simulations are conducted on an Intel Core i3-3220 3.30 GHz PC under Matlab R2013a environment. The average execution time for scenario P is 451.06 seconds, while it is 233.60 seconds for scenario P_{bic}.

Table 6.4: Percentage of patches super-resolved with sparse representation in scenario P_{bic}, and SR run times of P and P_{bic}.

Image	Perc	P Run Time	P_{bic} Run Time
Barbara	15.3	569.4	216.2
BSDS 198054	17.1	297	116.5
Butterfly	41.3	133.2	72.1
Fence	25.3	119.6	61.8
Flowers	12.4	343.8	135.4
Input6	22.9	779.7	385.3
Lena	10.3	531.7	193.4
Man	13.4	491.1	193.7
ppt3	21.7	542	300.5
Starfish	18.8	113.5	51.9
TextImage1	40.4	692.4	444
Texture	84.3	799.3	632.4
Average	26.93	451.06	233.60

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis aimed at improving the quality of sparse representation over a set of compact class dictionaries. This required the usage of certain classifiers to perform clustering of training and testing signals. A classifier is used to describe a certain signal class. In other words, each signal class should have a common property shared by its signals. The findings of this thesis are applied to the problem of single-image super-resolution. Therefore, another requirement imposed upon these classifiers is the scale-invariance. This means that a decision made about clustering a patch into a certain cluster should be as dependent of scale as possible. In this work, the magnitude and phase of the gradient operator are empirically shown to fit for these purposes. When applied to the super-resolution problem, the proposed sparse coding paradigms are shown to produce the state-of-the-art performance. In conclusion, the following ideas can be made through the work conducted in this thesis.

- The proposed sharpness measure (SM) and dominant phase angle (DPA) classifier are used to separate image patches based on their spatial intensity variations, and directionality, respectively. Besides, they are shown to be approximately scale-invariant. This finding motivates their usage for clustering and model selection for the purpose of super-resolution.

- Super-resolution via sparse representation over multiple dictionaries trained over SM and DPA clusters is shown to produce results that are competitive with the state state-of-the-art super-resolution algorithms. This result is validated through experiments in terms of qualitative and quantitative comparisons. In terms of computational complexity, these two algorithms are comparable to the case for using a single, highly redundant dictionary.
- Following on the above observation, a super-resolution algorithm is proposed via sparse representation over multiple dictionaries learned over clusters obtained by employing the two criteria: SM and DPA. This sparse coding paradigm uses many dictionaries. However, each dictionary is compact. Therefore, its computational complexity is still comparable to that of standard sparse coding.
- A desirable character of the proposed classifiers is their capability in separating image patches of insignificant low frequency components. This character can be effectively exploited to reduce the computational complexity. This is done by allocating bicubic interpolation for super-resolving such patches. Given the simplicity of bicubic interpolation and the fact that a large percentage of patches lies in the clusters of insignificant high frequency components, one can substantially reduce the computational complexity of patch reconstruction. In other words, the computationally demanding super-resolution via sparse representation framework is exclusively applied to patches of significant high frequency components, i.e., to a limited minority of image patches. This reduction in computational complexity is experimentally verified in terms of simulation run time.

7.2 Future Work

To this end, the following is an account for possible future extensions that can be made on the work conducted in this thesis.

- Perfect model selection curves provided in Chapter 3 and 5 suggest that there is still a room for improvement. However, this requires improvement in the prediction quality in the reconstruction stage. Further work can be done along this direction. Moreover, one may consider performing Naive Bayes or support vector machine (SVM) clustering.
- The proposed super-resolution algorithms can be extended using better coupled dictionary learning algorithms. With the availability of such algorithms in the future, the same setting can be adopted.
- In view of the conclusions made in Chapter 6, one can work towards developing a super-resolution algorithm that is based on bicubic interpolation along with the sparse representation framework which is applied in rare cases where bicubic interpolation is expected to fail.
- It seems interesting to optimize the SM intervals in a more systematic manner. Further work can be made to formulate the interval bounds in an optimization problem where the objective function is to jointly maximize scale-invariance and to improve the reconstruction quality.
- This work considered the usage of five DPA clusters. This number is shown to be a good compromise between reconstruction quality and scale-invariance. However, further work may still be done to find the

optimal number of DPA clusters and the way of quantizing the angles in the clusters.

REFERENCES

[1] Elad, M., & Aharon, M. (2006). Image denoising via learned dictionaries and sparse representation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Vol. 1, pp. 895-900). IEEE.

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

[3] Skretting, K., & Engan, K. (2011). Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. *In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (pp. 1517-1520). IEEE.

[4] Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S., & Yan, S. (2010). Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6), 1031-1044.

[5] Mairal, J., Elad, M., & Sapiro, G. (2008). Sparse learned representations for image restoration. In Proc. of the 4th World Conf. of the Int. Assoc. for Statistical Computing (IASC).

[6] Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12), 3397-3415.

[7] Olshausen, B. A., & Field, D. J. (1996). Natural image statistics and efficient coding*. *Network: computation in neural systems*, 7(2), 333-339.

[8] Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1?. *Vision research*, 37(23), 3311-3325.

[9] Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11, 19-60.

[10] Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12), 3397-3415.

[11] Pati, Y. C., Rezaiifar, R., & Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *In Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on* (pp. 40-44). IEEE.

[12] Chen, S. S., Donoho, D. L., & Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1), 33-61.

[13] Gorodnitsky, I. F., & Rao, B. D. (1997). Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3), 600-616.

[14] Skretting, K., & Engan, K. (2010). Recursive least squares dictionary learning algorithm. Signal Processing, *IEEE Transactions on*, 58(4), 2121-2130.

[15] Aharon, M., Elad, M., & Bruckstein, A. M. K-SVD (2005): An algorithm for designing of overcomplete dictionaries for sparse representation Technion—*Israel Inst. of Technology*. Tech. Ref.

[16] Davis, G. (1994). Adaptive nonlinear approximations *(Doctoral dissertation, Courant Institute of Mathematical Sciences New York)*.

[17] Cotter, S. F., Rao, B. D., Kreutz-Delgado, K., & Adler, J. (1999). Forward sequential algorithms for best basis selection. *IEE Proceedings-Vision, Image and Signal Processing*, 146(5), 235-244.

[18] Skretting, K., & Husøy, J. H. (2008). Partial search vector selection for sparse signal representation.

[19] Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. Signal Processing, *IEEE Transactions on*, 41(12), 3397-3415.

[20] Gharavi-Alkhansari, M., & Huang, T. S. (1998). A fast orthogonal matching pursuit algorithm. In Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 *IEEE International Conference on* (Vol. 3, pp. 1389-1392). IEEE.

[21] Chen, S. S., Donoho, D. L., & Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1), 33-61.

[22] Engan, K., Rao, B. D., & Kreutz-Delgado, K. (2000). Regularized FOCUSS for subset selection in noise. *In Proc. Nordic Signal Processi. Symp* (pp. 247-50).

[23] Daubechies, I. (1992). Ten lectures on wavelets (Vol. 61, pp. 198-202). *Philadelphia: Society for industrial and applied mathematics*.

[24] Tropp, J. (2004). Greed is good: Algorithmic results for sparse approximation. Information Theory, *IEEE Transactions on*, 50(10), 2231-2242.

[25] Davis, G., Mallat, S., & Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive approximation*, 13(1), 57-98.

[26] Aharon, M. (2006). Overcomplete dictionaries for sparse representation of signals *(Doctoral dissertation, Technion-Israel Institute of Technology, Faculty of Computer Science).*

[27] Bruckstein, A. M., Donoho, D. L., & Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1), 34-81.

[28] Chen, S., Billings, S. A., & Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5), 1873-1896.

[29] Couvreur, C., & Bresler, Y. (2000). On the optimality of the backward greedy algorithm for the subset selection problem. *SIAM Journal on Matrix Analysis and Applications*, 21(3), 797-808.

[30] Gilbert, A. C., Muthukrishnan, S., & Strauss, M. J. (2003). Approximation of functions over redundant dictionaries using coherence. *In Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 243-252). Society for Industrial and Applied Mathematics.

[31] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

[32] Olshausen, B. A., & Field, D. J. (1996). Natural image statistics and efficient coding*. *Network: computation in neural systems,* 7(2), 333-339.

[33] Engan, K., Rao, B., & Kreutz-Delgado, K. Frame design using FOCUSS with method of optimized directions (MOD). *In Proceedings of the Nordic Signal Processing Symposium* (pp. 65-69).

[34] Cotter, S. F., & Rao, B. D. (2002). Application of total least squares (TLS) to the design of sparse signal representation dictionaries. *In Signals, Systems and*

Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on (Vol. 1, pp. 963-966). IEEE.

[35] Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T. W., & Sejnowski, T. J. (2003). Dictionary learning algorithms for sparse representation. Neural computation, 15(2), 349-396.

[36] Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. Signal Processing, *IEEE Transactions on*, 54(11), 4311-4322.

[37] Engan, K., Skretting, K., & Husøy, J. H. (2007). Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation. *Digital Signal Processing*, 17(1), 32-49.

[38] Mailhé, B., Lesage, S., Gribonval, R., Bimbot, F., & Vandergheynst, P. (2008). Shift-invariant dictionary learning for sparse representations: extending K-SVD. *In Signal Processing Conference, 2008 16th European* (pp. 1-5). IEEE.

[39] Yaghoobi, M., Blumensath, T., & Davies, M. E. (2009). Dictionary learning for sparse approximations with the majorization method. *Signal Processing, IEEE Transactions on*, 57(6), 2178-2191.

[40] Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2009). Online dictionary learning for sparse coding. *In Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 689-696). ACM.

[41] Skretting, K. (2002). Sparse signal representation using overlapping frames.

[42] Aase, S. O., Husoy, J. H., Skretting, K., & Engan, K. (2001). Optimized signal expansions for sparse representation. *Signal Processing, IEEE Transactions on*, 49(5), 1087-1096.

[43] Skretting, K., Engan, K., Husoy, J. H., & Aase, S. O. (2001). Sparse representation of images using overlapping frames. *In proceedings of the scandinavian conference on image analysis* (pp. 613-620).

[44] Engan, K., Aase, S. O., & Hakon Husoy, J. (1999). Method of optimal directions for frame design. In Acoustics, Speech, and Signal Processing, 1999. Proceedings., *1999 IEEE International Conference on* (Vol. 5, pp. 2443-2446). IEEE.

[45] Engan, K., Aase, S. O., & Husoy, J. H. (1998). Designing frames for matching pursuit algorithms. *In Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on* (Vol. 3, pp. 1817-1820). IEEE.

[46] He, L., Qi, H., & Zaretzki, R. (2013). Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. *In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on* (pp. 345-352). IEEE.

[47] Zeyde, R., Elad, M., & Protter, M. (2012). On single image scale-up using sparse-representations. *In Curves and Surfaces* (pp. 711-730). Springer Berlin Heidelberg.

[48] Wang, S., Zhang, L., Liang, Y., & Pan, Q. (2012). Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. *In Computer Vision and Pattern Recognition (CVPR),* 2012 IEEE Conference on (pp. 2216-2223). IEEE.

[49] Yang, J., Wang, Z., Lin, Z., Shu, X., & Huang, T. (2012). Bilevel sparse coding for coupled feature spaces. *In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 2360-2367). IEEE.

[50] Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11), 2861-2873.

[51] Dong, W., Zhang, L., Shi, G., & Wu, X. (2011). Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *Image Processing, IEEE Transactions on*, 20(7), 1838-1857.

[52] Feng, J., Song, L., Yang, X., & Zhang, W. (2011, September). Learning dictionary via subspace segmentation for sparse representation. *In Image Processing (ICIP), 2011 18th IEEE International Conference on* (pp. 1245-1248). IEEE.

[53] Yu, G., Sapiro, G., & Mallat, S. (2010). Image modeling and enhancement via structured sparse model selection. *In Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 1641-1644). IEEE.

[54] Yang, S., Wang, M., Chen, Y., & Sun, Y. (2012). Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding. *Image Processing, IEEE Transactions on*, 21(9), 4016-4028.

[55] Sun, J., Sun, J., Xu, Z., & Shum, H. Y. (2008, June). Image super-resolution using gradient profile prior. *In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.

[56] Hou, H. S., & Andrews, H. (1978). Cubic splines for image interpolation and digital filtering. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(6), 508-517.

[57] Dai, S., Han, M., Xu, W., Wu, Y., & Gong, Y. (2007). Soft edge smoothness prior for alpha channel super resolution. *In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (pp. 1-8). IEEE.

[58] Irani, M., & Peleg, S. (1993). Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4), 324-335.

[59] Baker, S., & Kanade, T. (2002). Limits on super-resolution and how to break them. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9), 1167-1183.

[60] Lin, Z., & Shum, H. Y. (2004). Fundamental limits of reconstruction-based superresolution algorithms under local translation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1), 83-97.

[61] Ben-Ezra, M., Lin, Z., & Wilburn, B. (2007). Penrose pixels super-resolution in the detector layout domain. *In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (pp. 1-8). IEEE.

[62] Freeman, W. T., Pasztor, E. C., & Carmichael, O. T. (2000). Learning low-level vision. *International journal of computer vision*, 40(1), 25-47.

[63] Freeman, W. T., Jones, T. R., & Pasztor, E. C. (2002). Example-based super-resolution. Computer Graphics and Applications, IEEE, 22(2), 56-65.

[64] Sun, J., Zheng, N. N., Tao, H., & Shum, H. Y. (2003). Image hallucination with primal sketch priors. *In Computer Vision and Pattern Recognition, 2003.*

Proceedings. 2003 IEEE Computer Society Conference on (Vol. 2, pp. II-729). IEEE.

[65] Chang, H., Yeung, D. Y., & Xiong, Y. (2004). Super-resolution through neighbor embedding. *In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (Vol. 1, pp. I-I). IEEE.

[66] Tappen, M. F., Russell, B. C., & Freeman, W. T. (2003). Exploiting the sparse derivative prior for super-resolution and image demosaicing. *In IEEE Workshop on Statistical and Computational Theories of Vision.*

[67] Fattal, R. (2007). Image upsampling via imposed edge statistics. *In ACM Transactions on Graphics (TOG)* (Vol. 26, No. 3, p. 95). ACM.

[68] Liu, C., Shum, H. Y., & Freeman, W. T. (2007). Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1), 115-134.

[69] Wang, Q., Tang, X., & Shum, H. (2005). Patch based blind image super resolution. *In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Vol. 1, pp. 709-716). IEEE.

[70] Peleg, T., & Elad, M. (2014). A statistical prediction model based on sparse representations for single image super-resolution. *Image Processing, IEEE Transactions on*, 23(6), 2569-2582.

[71] Kumar, J., Chen, F., & Doermann, D. (2012). Sharpness estimation for document and scene images. *In Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 3292-3295). IEEE.

[72] Yeganli, F., Nazzal, M., Unal, M., & Ozkaramanli, H. (2015). Image super-resolution via sparse representation over multiple learned dictionaries based on edge sharpness. *Signal, Image and Video Processing*, 1-8.

[73] Yeganli, F., Nazzal, M., Unal, M., & Ozkaramanli, H. (2014). Image Super-Resolution via Sparse Representation over Coupled Dictionary Learning Based on Patch Sharpness. *In Proceedings of the 2014 European Modelling Symposium* (pp. 203-208). IEEE Computer Society.

[74] Gonzalez, R. C. (2009). *Digital image processing*. Pearson Education India.

[75] Elad, M. (2010). Sparse and redundant representations: from theory to applications in signal and image processing, *Springer*.

[76] Michael Elad's Personal Website
http://www.cs.technion.ac.il/~elad/software
date visited: 1-Jan-2012

[77] Kodak Lossless True Color Image Suite:
http://r0k.us/graphics/kodak/
date visited: 20-Jan-2014

[78] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (Vol. 2, pp. 416-423). IEEE.

[79] Flickr Image Data Set
http://see.xidian.edu.cn/faculty/wsdong/wsdong
Date visited: 1-March-2015

[80] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4), 600-612.

[81] Nazzal, M., Yeganli, F., & Ozkaramanli, H. (2014). Single Image Super-Resolution via Sparse Representation over Directionally Structured Dictionaries Based on the Patch Gradient Phase Angle. *In Proceedings of the 2014 European Modelling Symposium* (pp. 209-214). IEEE Computer Society.

[82] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1), 259-268.

[83] Yeganli, F., Nazzal, M., & Ozkaramanli, H. (2015). Image super-resolution via sparse representation over multiple learned dictionaries based on edge sharpness and gradient phase angle. *Signal, Image and Video Processing,* 9(1), 285-293.

[84] Lin, L., Liu, X., Peng, S., Chao, H., Wang, Y., & Jiang, B. (2012). Object categorization with sketch representation and generalized samples. *Pattern Recognition*, 45(10), 3648-3660.

[85] Peissigô½, J. J., Young, M. E., Wasserman, E. A., & Biederman Á, I. (2005). The role of edges in object recognition by pigeons. *Perception*, 34, 1353-1374.

[86] Zitnick, C. L., & Parikh, D. (2012). The role of image understanding in contour detection. *In Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on (pp. 622-629). IEEE.

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

[88] Yeganli, F., Nazzal, M., & Ozkaramanli, H. (2015). Super-Resolution Using Multiple Structured Dictionaries Based on the Gradient Operator. *European Signal Processing Conference. (EUSIPCO). IEEE*.