1-1-2012

Medical Image Segmentation and Classification Based on Sparse Representation and Dictionary Learnng Algorithms

Mohammadali Julazadeh
Ryerson University

MEDICAL IMAGE SEGMENTATION AND CLASSIFICATION BASED ON SPARSE REPRESENTATION AND DICTIONARY LEARNING ALGORITHMS

by

Mohammadali Julazadeh
BASc, Tehran Azad University, Tehran, Iran, 2008

A thesis
Presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Applied Science in the program of Electrical and Computer Engineering

Toronto, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Ryerson University requires the signatures of all persons using or photocopying this thesis.

Please sign below, and give address and date.

MEDICAL IMAGE SEGMENTATION AND CLASSIFICATION BASED ON SPARSE REPRESENTATION AND DICTIONARY LEARNING ALGORITHMS

Master of Applied Science

2012

Mohammadali Julazadeh

Electrical and Computer Engineering

Ryerson University

Abstract

In this thesis a novel classification approach based on sparse representation framework is proposed. The method finds the minimum Euclidian distance between an input patch (pattern) and atoms (templates) of a learned-base dictionary for different classes to perform the classification task. A mathematical approach is developed to map the sparse representation vector to Euclidian distances. We show that the highest coefficient of the sparse vector is not necessarily a suitable indicator for classification. The proposed algorithm is compared with the conventional Sparse Representation Classification (SRC) framework as well as non-sparse based methods to evaluate its performance. Taking advantage of the introduced classification framework, we then propose a novel fully automated method for the purpose of segmenting different organs in medical images of the human body. Our results demonstrated an acceptable accuracy rate for both classification and the segmentation frameworks. To our knowledge, no other method utilizes sparse representation and dictionary learning techniques in order to segment medical images.

Acknowledgements

I would like to extend my sincere appreciation to Professor Javad Alirezaie, my MASc. supervisor in Department of Electrical and Computer Engineering, Ryerson University for helping me to research in the field of my interest, image and signal processing, and for his advices through my research and studies.

I would like to express my appreciation to my co-supervisor, Dr. Paul Babyn in the Department of Medical imaging at university of Saskatchewan and Saskatoon Health region for his support and comments on my papers and my research trend.

I would like to thank my beloved parents, Farzaneh and AliAkbar as well as my two sisters, Maryam and Leila for their encouragement and supports during my research. I also thank my friends and co-researchers at Ryerson University to help me better achieve my purpose.

Table of Contents

List of figures

List of tables

List of Acronyms

BOMP: Batch Orthogonal Matching Pursuit

BP: Basis Pursuit

CAD: Computer Aided Diagnosis

DCT: Discrete Cosine Transform

DL: Dictionary Learning

DK-SVD: Discriminative KSVD

FDDL: Fisher Discriminative Dictionary Learning

FOCUSS: Focal Under-determined System Solver

K-SVD: Generalized K-Means, Sigular Value Decomposition

MP: Matching Pursuit

MOD: Method of Optimal Directions

OMP: Orthogonal MAtching Pursuit

St-OMP: Stagewise Orthogonal Matching Pursuit

SRC: Sparse Representation Classification

SEC: Sparse Euclidian Classification

SSF: Separable Surrogate Functions

PDIP: Primal Dual Interior Point algorithm

Chapter One: Introduction

1.1 Preface

Recent advances in the field of medical imaging and the emergence of new techniques in acquiring images using Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) technologies have enabled these different imaging modalities to be widely used in the diagnosis and quantifying different diseases. As a matter of fact the expanding volume of CT and MRI studies and their image data has illuminated the need for Computer Aided Diagnosis (CAD) schemes to assist the radiologists. The first step in most CAD systems is to segment the acquired image. Image segmentation is basically subdividing an image into components such as lines or regions based on some local similarities in the divisions. Segmenting an image is often a very essential part of computer aided surgery prior to the surgery itself. In order to plan a surgery or diagnose a disease it is often necessary for the medical experts to obtain a patient specific 3-D model of the organ. These 3-D models of the organs are generated by segmenting the desired organ from a set of medical images which are usually acquired from different modalities such as CT and MRI [1]. In various medical applications such as cardiology and radiography it is necessary to generate a four dimensional model of the organ of interest [2]. This four dimensional model may be used to describe the temporal change in the organ position and the shape of the organ which is very critical in follow up studies for the medical personnel. An example of the use of four dimensional models is in the radiotherapy treatment of the thorax and/or upper abdomen [2]. Organ volumetry is another significant application of segmenting organs in medical images [3]. Organ volumetry which is required in 3-D medical datasets requires an accurate segmentation of the desired organ. Organ segmentation is one of the most challenging steps prior to 3-D rendering of the organs.

Although the need for medical image segmentation is inevitable, there are still many limitations and difficulties in this field. One of the most significant problems is that there is no generic method that can perform the task of segmentation for all medical imaging modalities. If an image is proven to be useful for a specific organ in a specific modality it is not guaranteed that the same method performs well for another organ or a different modality. This dilemma is mainly due to the large variations of medical images characteristics, human anatomy and pathology and different techniques that are being used to acquire such images [4]. Knowing the value of segmenting images in the biomedical field, one cannot deny the fact that for a segmentation technique to be suitable for the medical personnel it has to require minimum human interface and it should be automatic. Manual segmentation of organs is not desired for medical datasets because of the large volume of the data that is assessed for medical application. This need for an automatic segmentation algorithm is another significant challenge in CAD related studies. Although many algorithms have been developed to automatically segment medical images, due to the complexity of these images most proposed methods require human interaction during segmentation [3]. In this thesis we are proposing a novel approach to segment different organs in the human body in an entirely automatic approach using the new emerging field of dictionary learning and sparse representation of signals. Unlike most methods that are limited to a specific modality and a specific organ our method is generic and can be utilized within different modalities for a variety of organs. Acceptable results have been gained both in CT and MRI modalities with minimum human interaction required.

In recent years utilizing methods for sparsely representing a signal/image over a given dictionary has gained considerable attention by scholars around the world. Applications of signal sparse representation varies from compression [5] to denoising [6], restoration [7] and many more. In this thesis we have expanded this growing area of research into a new level by introducing a new approach for segmenting different organs in MRI and CT images utilizing sparse representation techniques.

1.2 Computer Aided Diagnosis (CAD)

While improvements in computer technology have had a remarkable impact on the medical imaging field, the interpretation of medical images is still best performed by medical personnel. During the past decade the use of computers in image interpretation and medical diagnostics have grown tremendously [8]. So far CAD has been described as “a second pair of eyes for the radiologists” [8] and there have been some cases that computers have outperformed human eye’s observations.

In CAD related studies scholars are mainly focusing on segmentation and feature extraction approaches and techniques in pattern recognition and classification frameworks. These applications can be related to any organ in the body from chest, to brain and in any modality from ultrasound to x-ray, CT and MRI. The focus of this thesis is mainly on segmenting different human organs in images acquired from CT scans and MRI modalities.

1.3 Image Segmentation

The operation of subdividing an image into components such as points, lines or regions based on local similarities, pixel's intensities, frequency components and etc is called image segmentation. Image segmentation is primarily essential to processes such as CAD, quantitative analysis, visualization, registration and many more.

Generally, image segmentation algorithms are based on two intensity value properties, either on discontinuity or similarity [9]. For segmentation based on discontinuity the approach is to obtain partitions of the image based on abrupt changes in intensity such as edges. In segmentation based on intensity similarity however, partitioning the image is based on some predefined similarities. Thresholding, region growing, region splitting and merging are a few examples of segmentation based on similarity.

There are different classifications to image segmentation techniques in the literature; here we briefly present some of the important image segmentation techniques:

Thresholding Methods [10]**:**

These methods are considered to be the simplest methods in segmenting images. In these approaches usually an image is turned into a binary image utilizing a defined threshold. Using the edges in the acquired binary image the image can be segmented.

Clustering based methods [9]**:**

In these approaches the segmentation procedure is carried out through a classification method such as K-means type of approaches. Having classified the image into different clusters, each pixel is assigned to a cluster centriod. By assigning the pixels to different clusters (usually in an iterative manner) the image will be segmented according to the desired cluster centroid.

Histogram based methods [11]**:**

These segmentation techniques are considered to be among the most efficient segmentation methods in the literature. In these methods the histogram of the image is computed for all pixels and based on the peaks and the valleys in the histogram diagram the image pixels will be categorized. Repeating the process of categorizing the pixels based on the peaks and the valleys will result in segmentation of the desired area in the image.

Based on a survey study published by Srinivasan *et. al* [11]on segmentation techniques for target recognition purposes, image segmentation techniques are categorized as:

1. Edge Detection Methods
2. Tree/Graph Based Methods
3. Region Splitting Methods
4. Region Growing Methods
5. Model Based segmentation
6. Neural Network Based segmentation
7. Graph Partitioning Methods
8. Watershed Transformation
9. Multiscale segmentation
10. Probablistic and Bayesian approaches

In this thesis we introduced a new automated segmentation framework which is based on the new concept of sparse representation algorithms. The proposed method can be categorized to be a

“Clustering based” method since it first classifies the image based on sparse techniques and following that the boundaries of the clustered area is segmented accurately.

1.4 Sparse Representation of Signals

The problem of sparse representation of signals is basically estimating a sparse multi-dimensional vector with respect to a linear system of equations; given a high dimensional observed data and a matrix known as the dictionary. If we consider $y = D\alpha$ as the linear system of equations in which D is an $m \times n$, ($m \ll n$) matrix with $y \in \mathbb{R}^m$ and $\alpha \in \mathbb{R}^n$, D is called the dictionary which is usually over complete and the elements of it are called atoms, the problem is to estimate the vector (or signal) y subject to the constraint that it is sparse. It is important to note that the difference between a complete and an over complete dictionary lies in the fact that an over complete dictionary contains more amount of data since its atoms are not just orthogonal to each other but can be in any directions with respect to each other. Because of the nature of sparsity, α contains only a few non-zero elements which imply that y is decomposable as a linear combination of only a few numbers of atoms inside D. The sparse representation problem is characterized as:

$$\min_{\alpha \in \mathbb{R}^n} \|\alpha\|_0 \text{ such that } y = D\alpha \tag{1.1}$$

In which $\|\alpha\|_0$ is a pseudo-norm l_0 indicating the number of on-zero coefficients of the vector α . However solving such a problem is NP-Hard [12], and with a change of l_0 into l_1 norm the solution to the problem (1.1) would be in the convex form of:

$$\min_{\alpha \in \mathbb{R}^n} \|\alpha\|_1 \text{ such that } \|D\alpha - y\|_2 \le \xi \tag{1.2}$$

Where ξ determines the termination criterion. This problem and its derivations can be solved using different numerical algorithms which will be discussed in the following chapters.

1.5 Our contributions

In this thesis we are introducing a novel automated image segmentation approach to segment different organs in biomedical images. Our method is proposed based on a novel and original classification framework which we develop in conjunction with the new emerging concept of signals sparse representation and dictionary learning. We learn a particular dictionary for a specific organ, and based on the derived clustering algorithm, we first classify and later segment the desired organ. Furthermore we have proven that our method is a generic method and is not limited to a specific organ or even a specific modality. The method has been tested on standard databases such as Brodatz textures in order to evaluate its performance and for comparison with other techniques. Since the method is based on dictionary learning, one can basically learn a specific dictionary for a particular purpose and modality and takes advantage of our classification framework to cluster and later segment the image in applications other than medical images. To our best knowledge no other technique utilizes sparse based techniques in order to segment medical images. In this thesis we are aiming to introduce the advantages of sparse representation and dictionary learning algorithms in the field of medical image processing.

This thesis is organized in six chapters as follow:

First a general overview of sparse representation and approximation is presented in chapter 2. The original problem of sparse approximation and the mathematical approaches and algorithms to solve this problem are defined in this chapter. Chapter 3 is a comprehensive section on dictionaries and dictionary learning approaches. Since the main focus of this thesis is on classification, discriminative dictionary learning algorithms are presented in a separate sub-section in chapter 3. Chapter 4 focuses on the new emerging area of using sparse based techniques and dictionary learning for the applications of pattern recognition and computer vision. The famous Sparse Representation Classification (SRC) framework is presented in this chapter. The chapter ends with a short section on the contribution that we had to the field of medical image processing using SRC based methods. Chapter 5 is a comprehended section on the contribution that we had to the field of sparse representation and classification. We present a novel classification framework with a methodology that has never been investigated before in this field. This methodology is based on defining a transform between sparse features and Euclidian distances for the purpose of clustering. Following this method we present our automated segmentation algorithm for CT and MRI images. To the knowledge of the author this is the first study focusing on segmenting medical images by means of sparse representation and dictionary learning topic. The thesis ends with a short discussion and overview of the future works in chapter 6.

Chapter Two: Sparse Representation

In this chapter, we first explain the concept of the sparse representation and following that numerical and mathematical algorithms to solve the sparse approximation problem are presented in details.

In order to explain the concept of Sparse Representation, first the term “Sparse” should be clarified. In linear algebra the term sparse refers to a measurable property of vectors. Sparsity is not an indicator of the size of the vector, but it concerns the number of non-zero coefficients in the vector. One of the main advantages of sparsity is the simplicity of calculation that this property brings in vector calculations, as a matter of fact multiplication of a matrix by a sparse vector takes less computational time compared to a non-sparse vector, also sparse vectors can be saved on a computer by only having the position and value of its non-zero coefficients, hence utilizes less memory.

One indicator to measure the sparsity of a vector is the l_0 norm. The l_0 norm which is demonstrated as $\|.\|_0$ refers to the number of non-zero entries in a vector. For example l_0 norm of the vector $[1,0,0,6,0,0,0]$ is 2.

l_1 And l_2 norms indicated as $\|.\|_1$ and $\|.\|_2$ respectively are also indicators of sparsity. The l_1 norm is the sum of the absolute value of the coefficients in a vector and l_2 is the Euclidian length of a vector. The l_1 norm of a vector x can be indicated as:

$$\sum_{i=1}^{n} |x_i| \tag{2.1}$$

And the l_2 norm of this vector is defined as:

$$\sqrt{(\sum_{i=1}^{n} |x_i|^2)} \tag{2.2}$$

2.1 Sparse Approximation

Given a vector $x \in R^n$ and a matrix $D \in R^{n+m}$ the problem of sparse approximation can be characterized as finding the vector $\alpha \in R^n$ such that $x = D\alpha$. In which D is usually an over complete ($m \ll n$) matrix known as the dictionary. This problem is visually demonstrated in Figure 2-1.

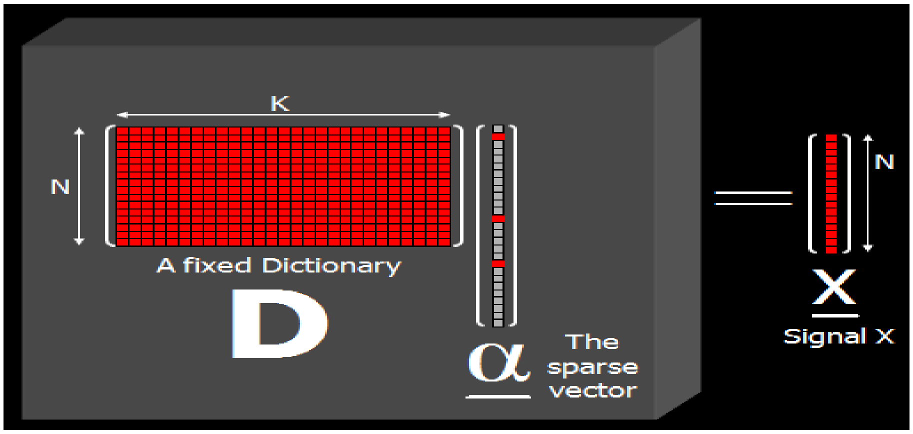


Figure 2-1 A visual demonstration of the sparse representation problem

The fact that D is an over complete matrix implies that this problem does not have a unique solution, in this problem if a sparse vector α can be found, it is called the sparse representation of x. This is due to the fact that α is the vector that can be used to reproduce or better to say, "represent" x. With the above mentioned explanations the problem of sparse approximation can now be characterized as:

$$\| D\alpha - x \| < \xi \tag{2.3}$$

With ξ (*epsilon*) is the reconstruction error. If a sparse vector α can be found as the solution to (2.3), it can be stated that α is a sparse approximation of x. In this problem α is no longer exactly reproducing x, but it produces approximations of x (because of ξ) and this brings more flexibility for the choice of x, also there will be a wider range of matrices for D. Below is an example of sparse representation and sparse approximation using a randomly generated matrix and vector:

1) $$\begin{bmatrix} 0.9593 & 0.2575 & 0.2435 & 0.2511 & 0.8308 \\ 0.5472 & 0.8407 & 0.9293 & 0.6160 & 0.5853 \\ 0.1386 & 0.2543 & 0.3500 & 0.4733 & 0.5497 \\ 0.1493 & 0.8143 & 0.1966 & 0.3517 & 0.9172 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0.7537 \\ 0.1285 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2254 \\ 0.7531 \\ 0.2366 \\ 0.6390 \end{bmatrix}$$

2) $$\begin{bmatrix} 0.9593 & 0.2575 & 0.2435 & 0.2511 & 0.8308 \\ 0.5472 & 0.8407 & 0.9293 & 0.6160 & 0.5853 \\ 0.1386 & 0.2543 & 0.3500 & 0.4733 & 0.5497 \\ 0.1493 & 0.8143 & 0.1966 & 0.3517 & 0.9172 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0.7537 \\ 0 \\ 0 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.2254 \\ 0.7531 \\ 0.2366 \\ 0.6390 \end{bmatrix}$$

Equation (1) is an example of sparse representation while equation (2) is an example of sparse approximation of a vector. It can be seen that compare to the length of the original signal which is 1.0403 the amount of error in approximation is relatively small (0.1338).

2.1.1 Uniqueness of sparsest approximation:

The problem of sparse approximation shown in (1.1) can also be written in the form of:

$$\min_{x} \|x\|_0 \text{ subject to } x = D\alpha \tag{2.4}$$

The main goal of sparse approximation should be referred to this equation ((2.4)) hereafter.

This equation has two major shortcomings:

1) The equality requirement of $x = D\alpha$ is too strict and representing the vector x by a few elements of D can sometimes be unlikely. If some deviations are allowed it would have better results (approximation).
2) The sparsity measure is too sensitive to very small entries in α, a more appropriate measure would adopt a more forgiving approach towards such small entries.

Based on the definition that was given in (2.4), generally sparse approximations are not unique. However there are some conditions that under which the sparse approximation can be unique. Some of these conditions are: Uniqueness via the Spark, Uniqueness via the Mutual-Coherence and Uniqueness via the Babel Function [12]. Below the Uniqueness via Spark condition is explained in detail, just to prove its uniqueness:

Spark of a matrix which is a term that was first coined by Donoho and Elad [12] in 2003 is defined as the size of the smallest set of linearly dependent vectors of the matrix. This definition should not be considered the same with the Rank of a matrix which is the largest number of columns of the matrix which are linearly independent.

Uniqueness via Spark condition states that a sparse representation α of x over the matrix (Dictionary) D is unique if:

$$\|\alpha\|_0 < \frac{Spark(D)}{2} \tag{2.5 }$$

Proof: assuming that α_1 and α_2 are sparse representations of x where $\alpha_1 \neq \alpha_2$ and $\|\alpha_1\|_0, \|\alpha_2\|_0 < \frac{Spark(D)}{2}$ it can be stated that:

$$\begin{aligned} & D\alpha_1 = D\alpha_2 = x \\ & \Rightarrow D\alpha_1 - D\alpha_2 = 0 \\ & \Rightarrow D(\alpha_1 - \alpha_2) = 0 \\ & \Rightarrow \|\alpha_1 - \alpha_2\| < Spark(D) \end{aligned}$$

This contradicts the minimality of Spark (D).

2.2 Importance of Sparse Approximation:

Finding a sparse approximation is more than just an abstract mathematical problem. Sparse approximations have a wide range of practical applications. Vectors are often used to represent large amounts of data which can be difficult to store or transmit. By using a sparse approximation of the data the amount of space needed to store the vector would be reduced to a fraction of what was originally needed. Sparse approximations can also be used to analyze data by showing how column vectors in a given basis come together to produce the data. Many areas of science and technology have been greatly promoted by taking advantage of sparse approximation techniques. The very useful nature of sparse approximations have prompted much interest and research in recent years and there is no doubt that sparse approximations will continue to be of great interest in the coming years also.

In recent years sparse approximation techniques have influenced the image and signal processing community in numerous ways such as denoising [6], image compression [5], feature extraction [13] and many more [12]. One of the fields that sparse approximation and dictionary learning technique have not yet deeply investigated is the case of image classification and segmentation. In our work we took one step further towards image classification and segmentation by taking advantage of sparse approximation and dictionary learning techniques.

In the following section, numerical algorithms and approaches for solving the problem of sparse approximation in (2.3) and/or (2.4) will be presented in details.

2.3 Numerical Methods and Algorithms for Sparse Coding

Sparse Coding or Atom Decomposition is the process of computing the representation coefficients α based on the given signal y and the dictionary D. This process which in fact is no different than solving the equation (2.3) or (2.4) is generally carried out through a "pursuit algorithm", which finds an approximate solution for the original problem in any of (2.3)or (2.4). The approximate solution is acceptable because the exact solution in determining the spars representation is proved to be a NP-hard problem [14]. In this section we briefly discuss several such pursuit algorithms, and their prospects for success. Matching Pursuit (MP) [15] and Orthogonal Matching Pursuit (OMP) [16], [17], [18] are among the very well known pursuit greedy algorithms that solve the sparse approximation problem. These methods are considered to be simple because in them the greedy algorithm selects the dictionary atoms sequentially and by computing the inner product between the signal and the dictionary atoms arranges some least squares solvers or projections.

Another well-known pursuit approach in sparse coding is the Basis Pursuit (BP) [19] algorithm. By suggesting the l_1 solution instead of l_0, BP introduces a convex method to solve the sparse approximation problem. The Focal Under-determined System Solver (FOCUSS) [20] is a very similar approach; instead, it uses the l_p norm with $p<1$ as a replacement to the l_0 norm. In FOCUSS for $p<1$ the similarity to the true sparsity measure is better, but the overall problem becomes non-convex, giving rise to local minima that may be misleading when searching for a solution. Lagrange multipliers are used to convert the constraint into a penalty term, and an iterative method is derived based on the idea of iterated reweighed least-squares that handles the l_p norm as a l_2 weighted norm.

In all the algorithms there will be a stopping criteria which based on application can be the reconstruction error ξ (*epsilon*) or the desired number of non-zero coefficients for representation.

In this chapter greedy and convex sparse representation algorithms are presented. The chapter concludes with a short discussion about the stopping criterion.

2.3.1 Greedy Methods:

If the dictionary is orthogonal it is possible to solve the sparse representation problem in (2.3) or (2.4) by choosing the atoms with the largest inner product value between the dictionary atoms and the target signal. A conventional approach to perform such procedure is to find the atoms which have the maximum correlation with the signal, and then subtract the calculated contribution from the signal and repeat this procedure in an iterative process.

In this section first Matching Pursuit (MP) [15] which is a straight forward representation method is presented. Following that Orthogonal Matching Pursuit (OMP) [16] which adds a least-square minimization process to MP in order to improve its performance is presented. OMP is considered to be the most admired leading sparse representation method in the literature. After OMP, batch-OMP and stage-wise OMP are also presented.

The section is concluded with a brief discussion on the history of greedy methods.

2.3.1.1 Matching Pursuit

Introduced in 1993 by Mallat *et. al,* [15] Matching Pursuit (MP) is a method capable of decomposing a signal into a liner expansion of waveforms that describe the time-frequency properties of the signal.MP is a greedy method which iteratively decomposes the signal into its representing waveforms (atoms). Considering a fixed dictionary D and a stopping criterion the Matching Pursuit algorithm [15] operates as follows:

Purpose: solving the problem of $\min_x \|x\|_0$ subject to $x = D\alpha$.

Inputs: original signal x , the dictionary D and the stopping criterion ξ

Initialization: set $k = 0$ and initialize:

- Initial solution $\alpha^0 = 0$
- Initial residual $r^0 = x - D\alpha^0 = x$
- Initial solution support $S^0 = Support\{\alpha^0\} = 0$

Main loop: $k = k + 1$ and perform:

- Sweep: Compute the errors $\varepsilon(j) = \min_{z_j} \| a_j z_j - r^{k-1} \|_2^2$ for all j using the optimal choice $z_j^* = a_j^T r^{k-1} / \| a_j \|_2^2$.
- Update Support: find a minimizer, j_0 of $\varepsilon(j) : \forall 1 \le j \le m, \varepsilon(j_0) \le \varepsilon(j),$ and update $S^k = S^{k-1} \bigcup \{j_0\}$.
- Update Provisional Solution: set $\alpha^k = \alpha^{k-1}$ and update $\alpha^k(j_0) = \alpha^k(j_0) + z_j^*$
- Update Residual: Compute $r^k = x - D\alpha^k = r^{k-1} - z_{j_0}^* a_{j_0}$.
- Stopping regulation: if $\| r^k \|_2 < \varepsilon_0$ stop! Else: apply one more iteration.

Output: The approximated α^k vector after k iteration.

Table 2.1 Matching Pursuit algorithm for sparse approximation

The sweep step of the main loop indicates the greedy selection of the atoms of the dictionary that have the most correlation with the residual part of the signal. An important fact about MP is that in this algorithm, one index from the dictionary might be chosen several times when the dictionary is not orthogonal. This repetition occurs because the inner product between an atom and the residual does not account for the contributions of other atoms to the residual. In the update stage the current coefficient vector gets updated to account for the effect of atom S^k. Following that a new residual is computed by subtracting a component which is in the direction of the atom D_{S^k}.

2.3.1.2 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit or simply the OMP [16] [17] [18] is a greedy algorithm similar to the Matching Pursuit. It adds a least-squares minimization to the MP method to obtain the best approximation over the dictionary atoms that have already been selected. This fact considerably improves the performance of OMP compare to MP. At each step OMP picks the dictionary atoms that have the maximal projection onto the residual signal; note that the dictionary elements should be normalized in this process. Following the selection of atoms the sparse representation coefficients are found by means of least-squares with respect to the atoms that are chosen so far.

Given a signal $x \in R^n$ and a dictionary $D \in R^{n+m}$ with normalized atoms as its columns the algorithm starts by initializing $k = 0$ and the residual $r^0 = x$ and performing the following procedure:

In the sweep stage, the algorithm computes the reconstruction error values in the following form:

$$\varepsilon(j) = \min_{z_j} \| a_j z_j - r^{k-1} \|_2^2 = \left\| \frac{a_j^T r^{k-1}}{\| a_j \|_2^2} a_j - r^{k-1} \right\|_2^2 \qquad \textbf{(2.6)}$$

Which indicates that finding the smallest error for representation is in fact no different than finding the largest absolute value of the inner product between the residual r^{k-1} and the normalized atoms of the dictionary D.

In order to update the provisional solution stage, the algorithm minimizes the term $\left\| D\alpha - x \right\|_2^2$ with respect to α such that S^k is its support. Considering D_{S^k} as parts of the dictionary which contains atoms of D that belong to this support the problem is to minimize the term $\left\| D_{S^k} \alpha_{S^k} - x \right\|_2^2$ where α_{S^k} is the non-zero part of the vector α . The solution to this problem is obtained by zeroing the first derivative of the following quadratic equation:

$$D_{S^k}^T(D_{S^k}\alpha_{S^k} - x) = -D_{S^k}^T r^k = 0 \qquad \textbf{(2.7)}$$

This relation implies that the columns (atoms) in D that are parts of the support S^k are orthogonal to the residual r^k and hence guarantees that in the following iterations, the already chosen atoms (columns) will not be chosen again for the support; this is the main reason for calling the method “Orthogonal” Matching Pursuit.

The algorithm will iteratively run until convergence is reached, that is when it has selected a pre-defined number of atoms or when the reconstruction error (ξ) is smaller than a given initial value. As a matter of fact the stopping criterion is based on the norm of the residual or/and the maximal inner product computed in the following atom selection stage. Table 2.2 demonstrates an overview of the OMP algorithm.

Purpose: solving the problem of $\min_x \|x\|_0$ subject to $x = D\alpha$.

Inputs: original signal x , the dictionary D and the stopping criterion ξ

Initialization: set $k = 0$ and initialize:

- Initial solution $\alpha^0 = 0$
- Initial residual $r^0 = x - D\alpha^0 = x$
- Initial solution support $S^0 = Support\{\alpha^0\} = 0$

Main loop: $k = k + 1$ and perform:

- Sweep: Compute the errors $\varepsilon(j) = \min_{z_j} \| a_j z_j - r^{k-1} \|_2^2$ for all j using the optimal choice $z_j^* = a_j^T r^{k-1} / \| a_j \|_2^2$.
- Update Support: find a minimizer, j_0 of $\varepsilon(j) : \forall j \notin S^{k-1}, \varepsilon(j_0) \le \varepsilon(j),$ and update $S^k = S^{k-1} \bigcup \{j_0\}$.
- Update Provisional Solution: Compute x^k ,the minimizer of $\| D\alpha - x \|_2^2$ subject to $Support\{x\} = S^k$.
- Update Residual: Compute $r^k = x - D\alpha^k$.

- Stopping regulation: if$\| r^k \|_2 < \varepsilon_0$ stop! *Else*: apply one more iteration.

Output: The approximated α^k vector after k iteration.

Table 2.2 Orthogonal Matching Pursuit pseudo algorithm for sparse approximation

A persuasive method, OMP, unlike many methods has the advantage of being able to sparsely represent a signal with an a-priori fixed number of non-zero elements, which makes it a desired approach in dictionary learning methods [21]. The fact that OMP is capable of representing a signal with an a-priori knowledge of non-zero elements is what we use later in our novel sparse classification approach.

Since OMP is a successful method in signal sparse representation, different variations of it has been developed. Based on a categorization introduced by M.Aharon [22]these variants are:

- skipping the least-squares and using the inner product itself as a coefficient;
- applying least-squares per every candidate atom, rather than just using inner-products at the selection stage;
- projecting all non-selected atoms onto the space spanned by the selected atoms before each new atom selection;
- doing a faster and less precise search where instead of searching for the maximal inner product, a nearly maximal one is selected, thereby speeding up the search;

The following two sections introduce two of OMP variants known as Batch OMP [23], [24] and Stagewise OMP [25].

2.3.1.3 Batch Orthogonal Matching Pursuit:

One of the conventional algorithms that reduces the computational complexity compared to OMP is Batch Orthogonal Matching Pursuit (B-OMP) [23], [24]. This method reduces the computational time needed for calculating the inner product (correlation) between the residual vector and atoms of the dictionary. The algorithm assigns a predefined matrix $G = D^T D$ in the memory to eliminate the unnecessary computations.

Considering the linear equation $x = D\alpha$ for the signal x, dictionary D and the sparse representation vector α, it can be stated that:

$$\begin{aligned} &\bar{\alpha}' = D^T x \\ &\bar{\alpha} = D^T r = D^T(x - D_I(D_I^T D_I)^{-1} D_I{}^T x) = \bar{\alpha}' - G_I(G_{I,I})^{-1}\bar{\alpha}_I' \end{aligned} \tag{2.8}$$

Equation (2.8) indicates that the relation (correlation) between the residual vector and the dictionary atoms is calculated through the matrix G and is independent of having the residual vector.

Just like OMP the stopping criterion can be imposed either based on the number of non-zero coefficients or the reconstruction error which basically is the norm of the residual. If the stopping criterion is chosen to be based on the norm of the residual, it can be calculated based on the value of error in the current and the previous iteration:

$$r_k = \bar{x} - D\bar{\gamma}_k = x - D\bar{\gamma}_k + D\bar{\gamma}_{k-1} - D\bar{\gamma}_{k-1} = r_{k-1} + D\left(\bar{\gamma}_{k-1} - \bar{\gamma}_k\right) \tag{2.9}$$

The orthogonality property of the OMP implies that the residual is perpendicular to the signal approximation ($(\bar{r}_k)^T D\gamma_k = 0$) [24] ; Hence the residual norm can be obtained as:

$$\begin{aligned} &\|\bar{r}_k\|_2^2 = (\bar{r}_k)^T \bar{r}_k = (\bar{r}_k)^T(\bar{r}_{k-1} + D\left(\bar{\gamma}_{k-1} - \bar{\gamma}_k\right)) = \bar{r}_k^T r_{k-1} + \bar{r}_k^T D\bar{\gamma}_{k-1} \\ &= (\bar{r}_{k-1} + D\left(\bar{\gamma}_{k-1} - \bar{\gamma}_k\right))^T \bar{r}_{k-1} + (\bar{r}_k)^T D\gamma_{k-1} \\ &= \|\bar{r}_{k-1}\|_2^2 - (\bar{r}_{k-1})^T D\bar{\gamma}_k + (\bar{r}_k)^T D\bar{\gamma}_{k-1} \\ &= \|\bar{r}_{k-1}\|_2^2 - (\bar{x} - D\bar{\gamma}_{k-1})^T D\bar{\gamma}_k + (\bar{x} - D\bar{\gamma}_k)^T D\bar{\gamma}_{k-1} \\ &= \|\bar{r}_{k-1}\|_2^2 - \bar{x}^T D\bar{\gamma}_k + \bar{x}^T D\bar{\gamma}_k \\ &= \|\bar{r}_{k-1}\|_2^2 - (\bar{r}_k + D\bar{\gamma}_k)^T D\bar{\gamma}_k + (\bar{r}_{k-1} + D\bar{\gamma}_{k-1})^T D\bar{\gamma}_{k-1} \\ &= \|\bar{r}_{k-1}\|_2^2 - (\bar{\gamma}_k)^T D^T D\bar{\gamma}_k + (\bar{\gamma}_{k-1})^T D^T D\bar{\gamma}_{k-1} \\ &= \|\bar{r}_{k-1}\|_2^2 - (\bar{\gamma}_k)^T G\bar{\gamma}_k + (\bar{\gamma}_{k-1})^T G\bar{\gamma}_{k-1} \end{aligned} \tag{2.10}$$

The last equation of the above formulation ($\|\bar{r}_{k-1}\|_2^2 - (\bar{\gamma}_k)^T G\bar{\gamma}_k + (\bar{\gamma}_{k-1})^T G\bar{\gamma}_{k-1}$) is nothing but the norm of the residual or basically the error which is computed in each iteration and makes the error update stage faster. The pseudo algorithm of B-OMP is demonstrated in Table 2.3.

```
Purpose: solving the sparse approximation problem in an efficient order.
Inputs: original signal $x$, the dictionary $D$ and the stopping criterion $\xi$, $\bar{x}' = D^T \bar{\alpha}$,
Initialization: set $k = 0$ and initialize:
    • $G = D^T D$
    • $\bar{x}' = D^T \bar{\alpha}$
    • $I = (), L = [1], \varepsilon^0 = \bar{x}^T \bar{x}$, $\bar{\gamma} = \bar{0}$, $\bar{\alpha} = \bar{\alpha}'$, $\delta_0 = 0$,
Main loop:
    • $j := \max(\bar{\alpha})$
    • If $n > 1$
        o $\bar{w} := solve \quad L\bar{w} = G_{I,j}$
        o $L = \begin{bmatrix} L & 0 \\ \bar{w}^T & \sqrt{1 - \bar{w}^T \bar{w}} \end{bmatrix}$
    • Update $I = I \cup j$
    • $\bar{y} := solve \quad L\bar{y} = \bar{\alpha}_I$
    • $\bar{\gamma} := solve \quad L^T \bar{\gamma}_I = \bar{y}$
    • $\bar{\beta} = G_I \bar{\gamma}_I$
    • $\bar{\alpha} = \bar{\alpha}' - \bar{\beta}$
    • $\delta_n = \gamma_I^T \bar{\beta}_I$
    • $\varepsilon_n = \varepsilon_{n-1} - \delta_n + \delta_{n-1}$
    • $n = n + 1$
If $\varepsilon^{n-1} > \varepsilon^n$ then go to step 2
Output: sparse representation vector $\alpha_k$ after $k$ iteration
```

Table 2.3 Batch Orthogonal Matching Pursuit algorithm for fast sparse approximation

2.3.1.4 Stagewise Orthogonal Matching Pursuit:

Stagewise Orthogonal Matching Pursuit (St-OMP) [25] is another efficient technique based on OMP which provides a sparse solution in the case of underdetermined sparse representation problems. Inspired by Orthogonal Matching Pursuit and the Least-Angle Regression (LARS), StOMP is especially tailored for the cases in which dictionary D is random (such as in compressed sensing problems).

The goal of StOMP is to reduce the reconstruction error in a stagewise trend by approximating the solution of $x = D\bar{\alpha}_0$ in which $\bar{\alpha}_0$ is the sparsest (best) solution. The algorithm extracts multiple atoms in each stage (stage-wise) whereas conventional OMP finds only one atom per iteration and hence the number of iteration and the computational time in StOMP is significantly reduced compared to OMP. Similar to OMP, the process of selecting atoms is performed through a Matching filter ($x = D^T \alpha$) and the reconstruction accuracy is computed by subtracting the reconstructed signal in each stage from the original signal ($z = \bar{x} - x_0$).

If we assume that the dictionary $D \in R^{n \times N}$ is randomly chosen with its columns being independent from one another, then the elements of the vector z will approximately have a Gaussian distribution with a standard deviation of σ as follows:

$$\sigma = \frac{\|x_0\|_2}{\sqrt{n}} \tag{2.11}$$

Just like OMP the St-OMP initiates the residual with the input signal and in each stage the correlation between the residual $\bar{r}_s$ and dictionary atoms are calculated (Match Filtering). A hard threshold is applied to the output of the Match filter to select the atoms with maximum correlation. The index of selected atoms will be saved and after wards the input signal is projected to these selected atoms:

$$(x_s)_{I_s} = (D_{I_s}^T D_{I_s})^{-1} D_{I_s}^T y \tag{2.12}$$

The result of the algorithm is the sparse representation vector (α_s) as well as an updated residual ($\bar{r}_s = x - D\alpha_s$) gained from the final stage, note that s indicates the number of the stages.

Like other methods the algorithm runs for a predefined number of stages or until a desired reconstruction error is achieved. The threshold in each iteration is computed based on considering the noise to be Gaussian with a standard deviation σ . The block diagram of St-OMP is proposed by Donoho et al [25]as follows:

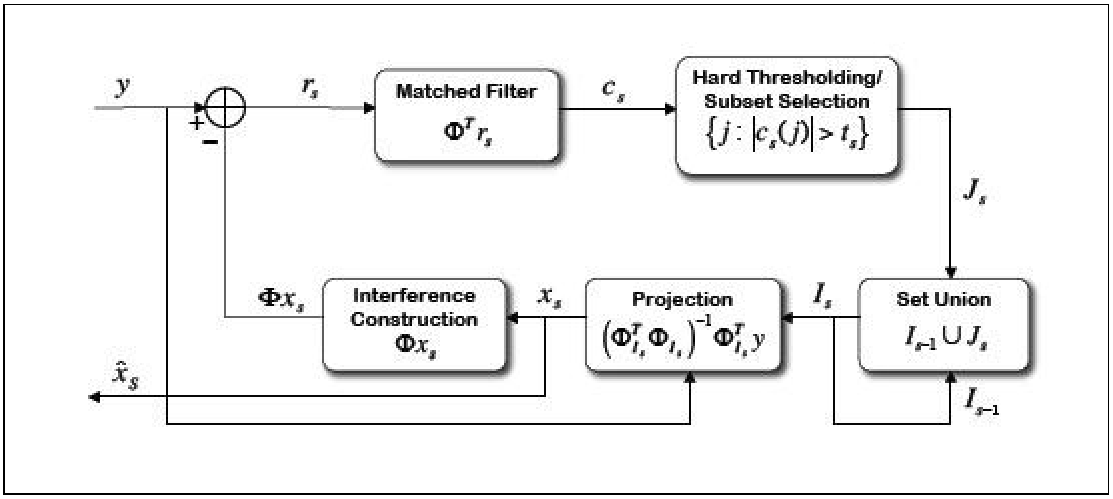


Figure 2-2 Block diagram presentation of St-OMP method proposed by Donoho et al, image is acquired from [25].

2.3.1.5 History of greedy methods for sparse approximation:

Greedy algorithms for sparse approximation first came to existence in the statistics literature in the 1950s to solve subset selection methods. The first greedy technique is called Forward Selection which is an intricate version of MP. Details on preliminary greedy algorithms are provided in [26].

Greedy sparse representation methods as we know them today came to existence in the late 1980s and early 1990s. MP was originally invented in 1981 [27] but it was introduced to the signal processing community by Mallat and Zhang [15] in 1993.

OMP was developed independently by many researchers around the globe; the earliest references cited goes back to 1989 in the paper of Chen, Billings and Luo [16]. It was first introduced to the signal processing community around 1993 by Pati, Zhang and Mallat [17] [28].

2.3.2 Convex Relaxation Methods:

The presence of l_0 norm makes the sparse approximation problem a combinatorial problem. A conventional approach in solving combinatorial problems is to replace the original problem with a relaxed version of the original problem which can be solved more efficiently. In this section the Basis Pursuit algorithm [29] which is known as the most famous convex relaxation approach in sparse representation is presented. For other convex relaxation techniques such as “Error-Constrained Approximation” and “Subset Selection” the readers are advised to study [30] and references therein.

2.3.2.1 Basis Pursuit:

Among the sparse representation techniques, Basis Pursuit [29] is considered more as an optimization technique rather than a direct algorithm. Basis Pursuit (BP) is a method to decompose signals by means of over-complete dictionaries; Note that an over-complete dictionary is basically a concatenation of multiple dictionaries, these dictionaries can be image based (constructed directly from image patches) or they can be composed of basic wave forms such as Wavelets, DCT, Curvelets, Ridgelet and many more. The purpose of creating such dictionaries is to combine the characteristics of different dictionaries. A detailed explanation about dictionaries will be presented in chapter 3.

In order to explain how BP works consider an input signal (image) x and an over-complete dictionary D with d_i as its atoms, the representation equation can be defined as:

$$x = \sum_{j=1}^{m} \alpha_{i_j} d_{i_j} + r^{(m)} \qquad \textbf{(2.13)}$$

In which $r^{(m)}$ is the residual. Unlike many other sparse representation methods, BP uses a convex optimization problem which minimizes the l_1 nom of the coefficients in the sparse representation vector [34]. Substituting l_0 norm with l_1 norm leads to a non-linear optimization problem which results in a higher sparsity. Another feature that emerges in the BP method is the ability to solve the optimization problem globally and therefore it sturdily finds the global optimum representation. This is a feature that a method like MP which is based on l_0 norm cannot perform. The sparse representation problem in this case can be rewritten as:

$$\min \|\alpha\|_1 \text{ subject to } x = D\alpha \tag{2.14}$$

The solution to the optimization problem in (2.14) lies within an algorithm known as Primal Dual Interior Algorithm [31] introduced by Florine *et. al,* discussed briefly below.

2.3.2.1.1 Primal Dual Interior Point Algorithm for linear programming:

The Primal Dual Interior Point (PDIP) described in [31] is a linear programming method aiming to solve the following optimization problem:

$$x = \arg\min_{x} c^T x \text{ subject to } Ax = b\ x \geq 0, \tag{2.15}$$

In which $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and A is a matrix with the size of $m \times n$. There is a dual problem associated with the primal problem:

$$\arg\max_{y,s} b^T y \text{ subject to } A^T y + s = c, s \geq 0 \tag{2.16}$$

Where $y \in R^m$ and $s \in R^n$ is called the dual slack [31]. Introduced by Florian *et. al* [31] $c^T x - b^T y$ is called the duality gap which acts as the termination criterion in the linear programming. The equation in (2.16) can be rewritten as:

$$\begin{aligned}&A^T y + s = c\\&Ax = b\\&XSe = 0\\&(x,s) \geq 0\end{aligned} \tag{2.17}$$

Where $e \in R^n$, $[e_i] = 1,...,n$ and X and S are diagonal matrices defined as:

$$X = \begin{pmatrix} [x]_1 & 0 & \cdots & 0 \\ 0 & [x]_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & [x]_n \end{pmatrix}, \quad S = \begin{pmatrix} [s]_1 & 0 & \cdots & 0 \\ 0 & [s]_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & [s]_n \end{pmatrix} \tag{2.18}$$

In order to solve the system of equations in (2.18) the equations in (2.17) can be rewritten in the following format:

$$F(x,y,s) = \begin{bmatrix} A^T + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0, \ (x,s) \geq 0 \tag{2.19}$$

Aiming to solve equation (2.19) iteratively by considering $(\delta_x, \delta_y, \delta_s)$ as the difference between the results in each iteration the Jacobean of F can be defined as:

$$F'(x,y,s) \begin{bmatrix} \delta x \\ \delta y \\ \delta s \end{bmatrix} = -F(x,y,s) \tag{2.20}$$

Based on (2.20) and (2.19) the final formulation can be written as:

$$\begin{bmatrix} 0_{n\times m} & A^T_{n\times m} & 0_{n\times n} \\ A_{m\times n} & 0_{m\times m} & 0_{m\times n} \\ S_{n\times n} & 0_{n\times m} & X_{n\times n} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta s \end{bmatrix} = \begin{bmatrix} 0_{n\times 1} \\ 0_{m\times 1} \\ -XSe \end{bmatrix} \tag{2.21}$$

The value of (x,y,s) in the $K+1^{th}$ iteration which is the value of (x,y,s) in K^{th} iteration plus $(\delta_x, \delta_y, \delta_s)$ will be iteratively modified so that the amount of duality gap reaches a value less than the error. The following table demonstrates the pseudo algorithm of (PDIP):

```
Purpose: Determining $(x^0, y^0, s^0)$ strictly feasible by solving the primal dual interior point
Inputs: $x, y$ and $s$
Initialize: set $k = 0$
Repeat:
    • Set
        $\tau^k \in [0,1]$ and $\gamma^k = \frac{x^T s}{n}$
Solve system of equation in

(2.21) to obtain $(\delta x^k, \delta y^k, \delta s^k)$
    • Set
        $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha^k (\delta x^k, \delta y^k, \delta s^k)$

Choosing $\alpha^k$ so that $(x^{k+1}, s^{k+1}) > 0.$
    • $K = K + 1$
Until: Convergence
```

Table 2.4 General Primal Dual Interior Point algorithm [31]

2.3.2.1.2 Solving the Basis Pursuit optimization problem:

Having discussed the Primal Dual Interior Point (PDIP) algorithm to solve the l_1 norm problem, we can take a similar approach to solve the Basis Pursuit (BP) problem in (2.14).

BP finds a sparse solution with n non-zero coefficient corresponding to n columns of D. In order to solve the problem in BP by taking advantage of PDIP the patches in the signal are split into two positive sub patches such that $s = u - v$ and variables $x = [u \quad v]^T$ and $A = [D \quad -D]$ are defined. The l_1 norm of the signal's patch can be represented as:

$$\|s\|_1 = \|u\|_1 + \|v\|_1 = \sum u_i + \sum v_i = c^T x \qquad \textbf{(2.22)}$$

The above formulation is not very different from the problem demonstrated in PDIP by Florien *et. al* [31], and a similar approach can be utilized to solve this problem. The following table demonstrates the pseudo algorithm of the Basis Pursuit method:

```
Purpose: solving the $l_1$ optimization problem in BP
Initialize: $x, y$ and $s$
    • Solve: Solving equation (3.16) to find $[\Delta x \quad \Delta y \quad \Delta s]$
Update: $x, y$ and $s$
    • Check convergence criteria, if not converged go to step 2
End
```

Table 2.5 Basis Pursuit optimization problem

2.3.3 Stopping Criterion:

So far a few greedy sparse representation algorithms have been introduced; in all of these methods there is one fact which is universal and that is the algorithms stopping criterion. In general stopping criterion can be classified into three major categories of EXACT, SPARSE And ERROR, each of which is useful for a specific sparse approximation problem which can be summarized as follows:

- **EXACT:** This stops the algorithm after the norm of the residual r^k equals zero. In cases in which recovering a sparse input signal is desired, this criterion is usually used.
- **ERROR:** This may halt the algorithm when the norm of the residual r^k declines below a specific defined threshold. In dealing with error constrained problems this criterion is more appropriate.
- **SPARSE:** This criterion will stop the algorithm whenever a predefined number of atoms have been selected from the dictionary. Using this criterion will result in a sparse vector with the exact length of the predefined number of atoms.

Chapter Three: Dictionary Learning

3.1 Quest for a proper dictionary

A fundamental question in the field of sparse representation of signals and its deployment to different applications is the selection of the proper dictionary to represent the data over. How can a proper dictionary be wisely chosen to perform well on the input signal so it can have the optimum efficiency for that specific purpose? This chapter of this thesis is dedicated to explain the nature of different dictionaries, and some of the most significant dictionary learning methods. First a brief discussion on the difference between fixed and learned dictionaries is presented following that some of the important fixed dictionaries and the mathematical approaches behind them is offered. After fixed dictionaries the concept of dictionary learning in the form of two very important and famous dictionary learning methods is discussed, finally the chapter ends with a section on discriminative dictionaries.

3.2 Fixed or Learned Dictionaries?

The quest for a proper dictionary so that it can be used for a specific application can reach the point that one considers pre-constructed or fixed dictionaries. Such dictionaries are the discrete Cosine Transform (DCT), Wavelets, Contourlets, Curvelets and many more. Some of these proposed dictionaries (also known as transforms) are accompanied by a detailed theoretical analysis establishing the sparsity of the representation coefficients for such simplified content of signals. This is usually done in terms of representation of signal using the best non-zero coefficients from the transform. Whether in discrete or continuous domain, one can alternatively utilize a tunable selection of a fixed dictionary. As an example wavelets and wavelet packets can be mentioned which suggest an acceptable control over time-frequency subdivision, tuned to optimize performance over a specified instance of signals.

While pre-constructed (fixed) dictionaries can typically lead to a fast transform, they are limited in sparsifying the signals which they are meant for. Furthermore in most of the cases fixed dictionaries are restricted only to a specific class of signals (or images) and cannot be used for an arbitrary class of new input signals. Although this family of dictionaries are suitable to address some of the sparse representation applications, their limitations lead to a new idea of learning the dictionaries based on the application that they are meant for.

The above mentioned learning procedure works by building a training dataset of signals similar to the signals that are supposed to be used based on the desired application, and constructs an empirically learned dictionary in which its atoms (elements) are generated based on an underlying empirical data rather than a theoretical (or mathematical) model like fixed dictionaries. This learned dictionary can then be used as a fixed dictionary itself in the specific application that it is meant for. Having mentioned the advantages of learning the dictionaries it should be mentioned that learning a dictionary itself is a computationally expensive procedure compared to using pre-constructed dictionaries.

3.3 Fixed dictionaries

In this section some of the famous fixed dictionaries are presented.

3.3.1 Time-Frequency Dictionaries

Fourier transform is the most well-known method in extracting a signal's frequency characteristics. Fourier transform is capable of representing a signal as its frequency domain components.

Since by definition Fourier transform represents the signal in the form of exponentials (sine and cosine) and knowing the fact that sinusoidal functions are pair wise orthogonal, all signals can be represented as a linear combination of these orthogonal coefficients. The representation coefficients themselves are obtained using the inner product of the given signal with the Fourier basis:

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{i2\pi ft} df \tag{3.1}$$

Where $X(f)$ represents the frequency representation of the signal x .

3.3.2 Discrete Cosine Transform

Discrete cosine transform (DCT), first introduced by Desai. *et. al* [32]and Watson. *et. al* [33] to the image processing community expresses a sequence of finite data points in terms of a summation of cosine functions oscillating at different frequencies. The use of Cosine instead of Sine function can be decisive and useful in specific applications such as compression in which fewer functions are needed to approximate a signal. A typical signal can be decomposed with DCT as:

$$X(m)=\frac{1}{\sqrt{2}}G_x(0)+\sum_{k=1}^{M-1}G_x(k)\cos\frac{(2m+1)k\pi}{2M},\ m=1,2,...,M-1 \tag{3.2}$$

In which $G_x(k)$ stands for the k^{th} coefficient.

By definition DCT is not much different from Fourier transform, as a matter of fact DCT is similar to Discrete Fourier Transform (DFT) but only with real parts and no imaginary element. Expanding equation (3.2) from one dimensional into a two dimensional space the DCT transform for 2D signals or images can be expressed as follows:

$$X(m,n)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}G_{xy}(k,l)\cos\frac{(2k+1)u\pi}{2n}\cos\frac{(2y+1)v\pi}{2m},\ u=1,....,N-1,\ v=1,2,...,M-1 \tag{3.3}$$

Figure 3-1 demonstrates a two dimensional DCT based dictionary with 8×8 size patches.

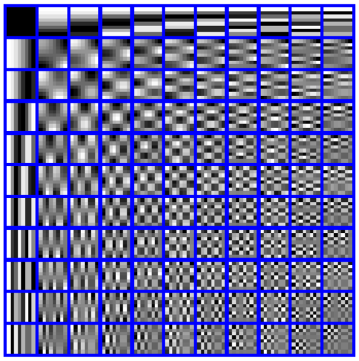


Figure 3-1 A DCT dictionary with 8*8 size patches

3.3.3 Wavelet and Wavelet Transform

A wavelet is a wave-like oscillation with an amplitude that starts out at zero, increases, and then decreases back to zero. In wavelet transforms, a given signal of finite energy is projected on a continuous family of frequency bands or similar subspaces. The Wavelet transform offers a resizable structure for atoms. This structure of different frequencies is based on each atom size so it can address the problem of having different structure sizes. As a matter of fact, the flexible time-frequency windows (wave forms) in the wavelet transform provides a non-uniform frequency bandwidth in which the frequency resolution is higher at lower frequencies and vice versa. These variable size windows or the so called wavelets are generated by scaling and shifting the basic wavelet $\Psi(t)$ [34] as follows:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi(\frac{t-b}{a}) \tag{3.4}$$

Where a is a positive value that defines the scale, b is a real number that defines the shift and the pair (a,b) defines a point in the right half-plane. The actual wavelet transform is defined by calculating the inner product of the above basis function in the following integral:

$$F(a,b) = (\frac{1}{\sqrt{a}}) \int_{-\infty}^{\infty} f(t) \psi^{*} (\frac{t-b}{a}) dt \tag{3.5}$$

It is proven [34] that if $a = 2^{-m}$ and $b = 2^{-m} k$ the original signal $f(t)$ can be recovered using the wavelet series as:

$$f(t) = \sum_{m,k \in Z} < f, \psi_{mk}(t) > \psi_{mk}(t) \tag{3.6}$$

In which $<.,.>$ is indicator of the inner product operator and wavelet basis functions, $\{\psi_{mk}\}$, are supposed to be orthonormal.

The wavelet transform is a perfect tool in representing 1D signals like audio. It is sensitive to high-frequency changes while it detects low-frequency terms in the signal. However in higher dimensions like images (2D) it fails to track directional information. As a matter of fact it performs very well in representing horizontal and vertical data while resulting in a non-convincing representation for other directions. Moreover, the wavelet transform is sensitive to discontinuities in the edge points. However it fails to accurately represent smoothness along the contours in images. The following figure is a demonstration of a wavelet transform based dictionary.

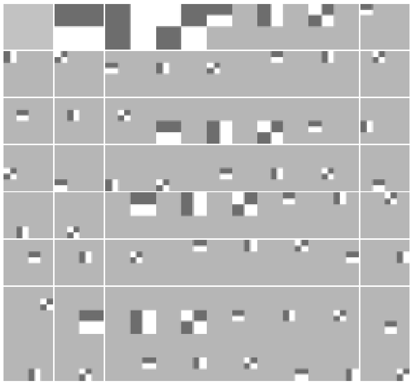


Figure 3-2 A Har base wavelet dictionary

3.4 Learning the Dictionary

This section describes the learning methodologies and justifications required in order to construct a proper dictionary so that the data can be represented over. Assuming that a training data set is given, if this dataset is generated by a fixed but unknown source or model, the question would be: can this training dataset allow us to be able to recognize the original generating model and specifically the dictionary? This is a problem that was first looked into by Field and Olshausen [35] in 1996 when they were trying to establish a similarity between atoms of a dictionary and the population of simple cells in the visual cortex. Working on simple cells they considered the possibility of learning a dictionary that can model the evolutionary process which led to the presented collection of simple cells; and to a good extent they were able to do so. Later works by other scholars around the world such as Elad [6], Aharon [22], Engan [12] and many more completed this work and resulted in novel dictionary learning methods, however this topic is still open and there is lot of room to grow in the field of dictionary learning. Here we present “Method of Optimal Directions” (MOD) and K-SVD algorithm as two of the most well known dictionary learning algorithms that are being used by researchers in recent years.

Having the problem of sparse representation in (1.1), dictionary learning can be addressed as follows:

The aim is to find a proper estimation of the dictionary D by assuming that the stopping criterion ξ (model deviation) is known in the following optimization problem:

$$\min_{D,x_i} \sum_{i=1}^{M} \|x_i\|_0 \text{ subject to } \|x_i - D\alpha_i\|_2 \leq \xi \text{ ,} 1 \leq i \leq M \quad \textbf{(3.7)}$$

Equation (3.7) describes any given signal x_i as the spars representation x_i over an unknown dictionary D and aims to find not only the representation but also the unknown dictionary. Clearly if a solution to this equation can be found that can meet the restrictions on the number of non-zero coefficients a candidate feasible model for the dictionary is also found. Just like the problem of sparse representation, dictionary learning can also be presented in the form of constraining the sparsity and obtain the best fit for it:

$$\min_{D,x} \sum_{i=1}^{M} \|x_i - D\alpha_i\|_2^2 \text{ subject to } \|x_i\|_0 \leq N_0 \text{ ,} 1 \leq i \leq M \quad \textbf{(3.8)}$$

In which N is the number of non-zero coefficients for representation. Similar to the problem of sparse representation the uniqueness of a gained dictionary can be derived by using the concept of Spark described in chapter 1. Aharon et al [22] showed that for the case of $\xi = 0$, if there exists a dictionary D_0 and a sufficiently diverse database of samples that all of which are representable using at most $k < sparkD_0 / 2$ number of atoms, the derived dictionary is unique to achieve the desired sparsity (N) for all the elements in the training dataset.

3.4.1 Method of Optimal Directions (MOD):

Method of Optimal Directions (MOD) is an algorithm which was first developed by Engan *et. al* [36] to learn dictionaries for sparse representation. The core idea in this method is to look at the problem in equation (3.7) or (3.8) as a nested minimization problem consisting of an inner minimization of the number of non-zero coefficients in sparse representation vector α over a fixed dictionary D and an outer minimization problem over the dictionary D itself. The nested minimization approach in the problem can result in an alternating minimization approach in solving the problem; developing an iterative algorithm at the k^{th} stage the algorithm uses the dictionary D_{k-1} from the previous stage and solves M instances of the l_0^{ξ} norm, one for each dataset entry x_i by using the dictionary D_{k-1}. This will result in the matrix (or vector) α_k and then the algorithm can be solved for D_k by Least-Squares:

$$\begin{aligned} D_k &= \arg\min_D \|X - D\alpha_k\|_F^2 \\ &= X\alpha_k^T(\alpha_k\alpha_k^T)^{-1} \\ &= X\alpha_k^+ \end{aligned} \tag{3.9}$$

The purpose of the Frobenious norm ($\|.\|_F$) is to evaluate the error. The columns of the gained dictionary can be re-scaled and k will increment until the convergence criterion is met. The block diagram of the proposed method is demonstrated in Table 3.1.

Purpose: To train the dictionary D so it can sparsely well represent the data x_i by means of approximating the solution to (3.7) or (3.8).

Initialization: set $k = 0$ and:

- **Initialize Dictionary** D: build an initial $D_0 \in R^{n\times m}$, usually by using random entries.
- **Normalization**: normalize the columns of D_0.

Main loop: increment k by 1 and:

- **Sparse Coding Stage:** use any of the pursuit algorithms to approximate the solution of:

  $\hat{\alpha}_i = \arg\min_\alpha \|x_i - D_{k-1}\alpha\|_2^2$ subject to $\|\alpha\|_0 \le k_0$

  Obtaining the sparse representations $\hat{\alpha}_i$ to form the matrix α_k

MOD dictionary update stage: updating formula:

$$D_k = \arg\min_D \|X - D\alpha_k\|_F^2 = X\alpha_k^T(\alpha_k\alpha_k^T)^{-1}$$

- **Stopping Criterion:** if the amount of change in $\|X - D\alpha_k\|_F^2$ is small enough, stop. Else: apply another iteration.

Output: Desired dictionary D_k

Table 3.1 The MOD dictionary learning algorithm

Figure 3-3 demonstrates the MOD behavior in learning a dictionary on a synthetic data in terms of representation error and number of recovered atoms.

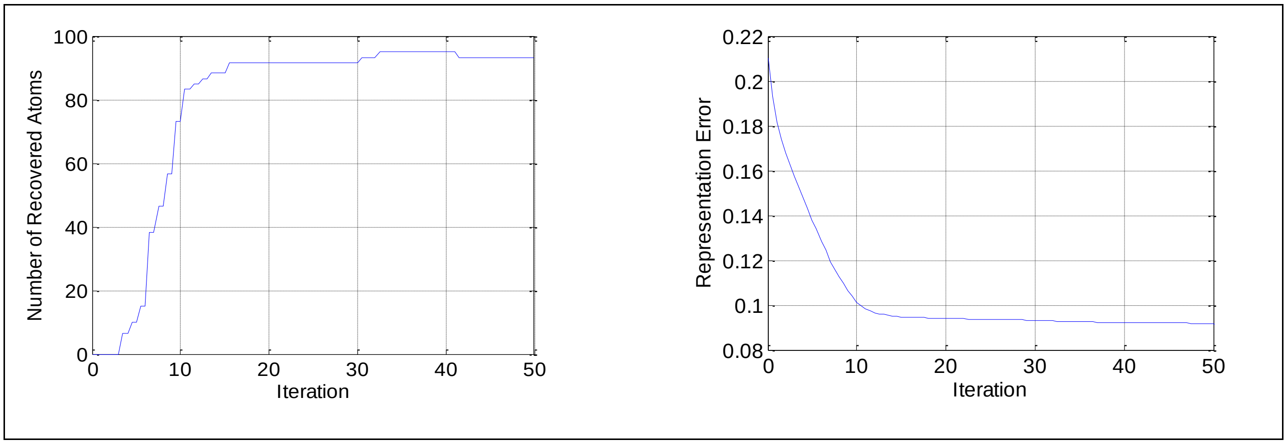


Figure 3-3 The MOD dictionary learning behavior

3.4.2 K-SVD Dictionary Learning

Inspired from K-means clustering technique by M. Aharon and M. Elad [21] K-SVD dictionary learning uses a different approach in updating the atoms (columns) of the dictionary by handling them sequentially. K-SVD update stage works as it keeps all of the atoms of the dictionary fixed except for the $j_0^{'th}$ one, d_{j_0}. This one column will be updated along with the coefficients that multiply it in α. The dependency on d_{j_0} can be gained by rewriting equation (3.9):

$$\begin{aligned} \|X - D\alpha\|_F^2 &= \left\| X - \sum_{j=1}^{m} d_j \alpha_j^T \right\|_F^2 \\ &= \left\| (X - \sum_{j \neq j_0} d_j \alpha_j^T) - d_{j_0} \alpha_{j_0}^T \right\|_F^2 \end{aligned} \tag{3.10}$$

The term in the parentheses is known as a pre-computed error matrix:

$$E_{j_0} = X - \sum_{j \neq j_0} d_j \alpha_j^T \tag{3.11}$$

With X as the data matrix, optimization of equation (3.10) lies in the hands of d_{j_0} and $\alpha_{j_0}^T$, the optimal value of d_{j_0} and $\alpha_{j_0}^T$ is called the "**rank-1"** approximation of the error matrix E_{j_0} which can be gained via the SVD method. However an SVD based approach would usually yields in a dense representation vector $\alpha_{j_0}^T$ which introduces more non-zero entries in the sparse representation vector α. In order to maintain the number of non-zero entries of the sparse vector to a minimum level (acceptable sparse value) a subset of columns of E_{j_0} which are those elements that correspond to the original signal from the dataset should be taken. These components are usually the ones using the elements of the $\alpha_{j_0}^T$ row in the vector α . In this way only the non-zero coefficients in $\alpha_{j_0}^T$ vary and the sparsity is maintained. In order to remove the non-relevant columns a limitation operator P_{j_0} that multiplies E_{j_0} from the right is defined. This P_{j_0} matrix has M (number of overall examples in the dataset) rows and M_{j_0} columns (number of elements that use the $j_0^{'th}$ atom). In order to choose the non-zero entries only, a restriction on the row $\alpha_{j_0}^T$ is defined as: $(\alpha_{j_0}^R)^T = \alpha_{j_0}^T P_{j_0}$. For the intention of updating the atom d_{j_0} and also the corresponding coefficients $x_{j_0}^R$ in the sparse vector, a rank-1 approximation for the sub-matrix $E_{j_0} P_{j_0}$ is applied via SVD algorithm which results in a simultaneous update. An alternate approach in updating the dictionary atoms and elements of the sparse vector can be proposed by fixing the atom d_{j_0} first and updating $\alpha_{j_0}^R$ by a plain Least-Squares problem:

$$\min_{\alpha_{j_0}^R} \left\| E_{j_0} P_{j_0} - d_{j_0} (\alpha_{j_0}^R)^T \right\|_F^2 \Rightarrow \alpha_{j_0}^R = \frac{P_{j_0}^T E_{j_0}^T \alpha_{j_0}}{\left\| \alpha_{j_0} \right\|_2^2} \qquad \textbf{(3.12)}$$

Once the sparse coefficient is updated then the atom d_{j_0} will be updated:

$$\min_{d_{j_0}} \left\| E_{j_0} P_{j_0} - d_{j_0} (\alpha^{R}_{j_0})^{T} \right\|_F^2 \Rightarrow d_{j_0} = \frac{E_{j_0} P_{j_0} \alpha^{R}_{j_0}}{\left\| \alpha^{R}_{j_0} \right\|_2^2} \tag{3.13}$$

Using the above formulas the desired updated dictionary can be achieved after a few iterations. If the above process is considered for the case of sparse representation with only one non-zero element ($N_0 = 1$) the problem is simplified into a K-means clustering task. For the cases of more than one non-zero coefficient the algorithm performs as an SVD operation for each of the K different subsets and thus the name K-SVD.

Table 3.2 demonstrates the pseudo algorithm of the K-SVD algorithm.

```
Purpose: To train the dictionary $D$ so it can sparsely well represent the data $x_i$ by means of
approximating the solution to (3.7) or (3.8).
Initialization: set $k = 0$ and:
    • Initialize Dictionary $D$: build an initial $D_0 \in R^{n \times m}$, usually by using random entries.
    • Normalization: normalize the columns of $D_0$.
Main loop: increment $k$ by 1 and:
    • Sparse Coding Stage: use any of the pursuit algorithms to approximate the solution of:
      $\hat{\alpha}_i = \arg\min_{\alpha} \|x_i - D_{k-1}\alpha\|_2^2$ subject to $\|\alpha\|_0 \le k_0$
      Obtaining the sparse representations $\hat{\alpha}_i$ to form the matrix $\alpha_k$
K-SVD Dictionary-Update stage: Repeating for $j_0 = 1, 2, ..., m$ the updating procedure is as follows:
    • Defining the data samples using the atom $d_{j_0}$
          $\Omega_{j_0} = \{ i \mid 1 \le i \le M, \alpha_k[j_0, i] \ne 0 \}.$
    • Compute the residual matrix
          $E_{j_0} = X - \sum_{j \ne j_0} d_j \alpha_j^T,$
    • Restrict $E_{j_0}$ (pre-computed error) by cho0sing only the columns corresponding to $\Omega_{j_0}$ and
      obtain $E^R_{j_0}$.
    • Apply SVD decomposition $E^R_{j_0} = U \Delta V^T$, update the dictionary atom $d_{j_0} = u_1$, and
      representations by $\alpha^R_{j_0} = \Delta[1,1].v_1$.
```

Stopping criterion: if the amount of change in $\left\| X - D_k \alpha_k \right\|_F^2$ is small enough (defined by user) stop, else; apply another iteration.

Output: learned dictionary D.

Table 3.2 K-SVD dictionary learning Pseudo algorithm

In order to be able to analyze the behavior of K-SVD more clearly, the method is implemented on synthetic data. The following figures illustrate K-SVD's performance in terms of number of recovered atoms and the representation error on the synthetic data.

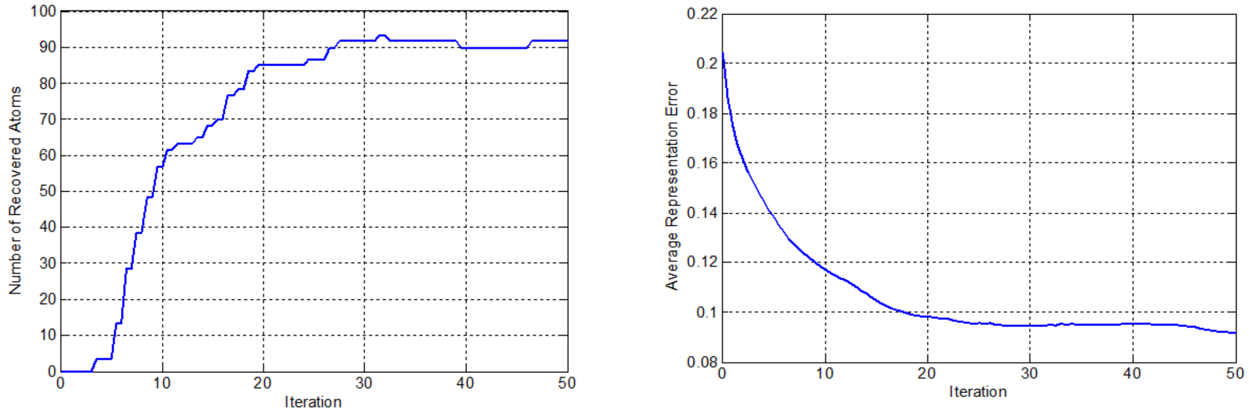


Figure 3-4 K-SVD dictionary learning behavior

The fact that dictionary learning is a generalization of a clustering approach reveals that:

1. Just like K-means neither K-SVD nor MOD do not guarantee to reach a global or even a local minimum of the penalty function in (3.8) since both methods have the potential to get stuck in a steady state solution.
2. Regarding the convergence of the algorithms, since both algorithms use a pursuit sparse representation technique, having a monotonic non increasing penalty value based on the number of iterations is not guaranteed.

In order to have a proper comparison between the above mentioned dictionary learning algorithms (K-SVD and MOD), both of them have been implemented on the same data and the following results have been gained:

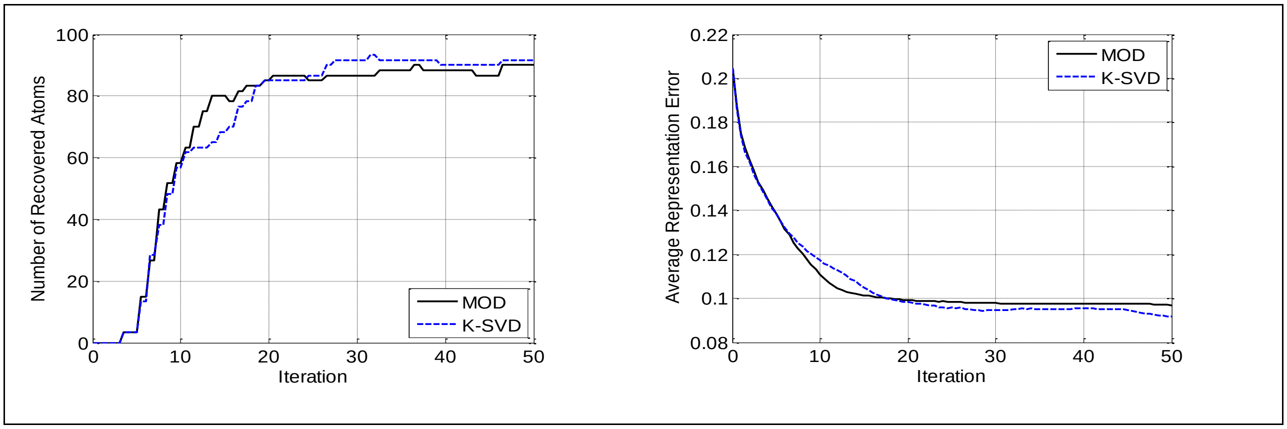


Figure 3-5 K-SVD behavior VS MOD behavior

Figure 3-6 shows the famous Barbara Image, and the MOD and K-SVD learned dictionaries based on this image. The representation error of the image for both dictionaries is also displayed.

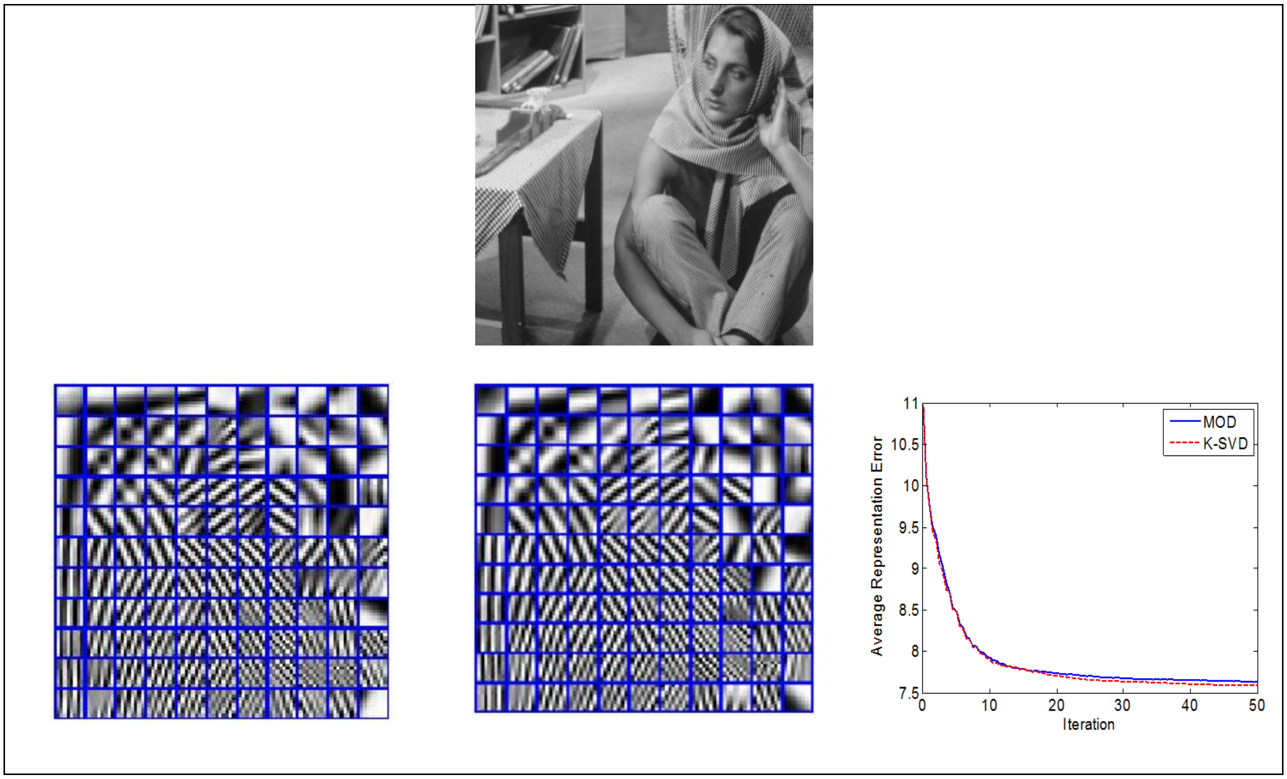


Figure 3-6, Original Barbara Image (Top), MOD Dictionary (bottom left), K-SVD Dictionary (bottom middle) and the representation error graph (bottom right).

3.5 Discriminative Dictionary Learning

In the previous section two famous dictionary learning methods have been demonstrated, however there are other dictionary learning algorithms ([37], [38] and references therein). Most of the dictionary learning algorithms in image processing focus on applications such as image denoising [6], compress sensing [5]and image restoration [7], etc. Most of these dictionary learning techniques are not suitable for the purpose of image classification, this is due to the fact that in the process of learning a dictionary there are no constraints on separating the learned patches based on the specific class that they belong to. In fact dictionary learning creates a new space in which the sparse representation of the data can be performed with more accuracy. However if the learning algorithm does not consider any restrictions on separating the patches based on the class that they belong to, the resulting space is of no use for the purpose of classification. In this section three famous dictionary learning techniques that consider class specific limitations in the process of learning are presented. Using these methods result in a new space in which sparse representation of the signal can be performed based on the fact that each patch belongs to a specific class of the training dataset. This will result in the process of labeling the patches based on the dictionary that they belong to.

3.5.1 Dictionary Learning with Structured Incoherence and Shared Features for Classification and Clustering

The work presented by Ramirez et al in [39] is among the well-known methods in learning a dictionary for classification and clustering (discriminative dictionary learning). In this framework unlike K-SVD type of approaches that finds a set of centroids which best fits the data, this method optimizes for a set of dictionaries, one dictionary for each class. As a matter of fact the method trains a dictionary consisting of various sub-dictionaries, one for each class. An incoherence promoting term is utilized to force the class specific dictionaries (sub-dictionaries) to be as independent as possible. It is a robust method that is suited to handle large data sets in both supervised and unsupervised type of approaches. The algorithm uses l_1 regularization in its sparse coding stage.

Given K clusters, the method learns K dictionaries to represent the data, and then associates each signal to the dictionary for which the "best" sparse decomposition is obtained. In other words, the test data (signal or image) is represented "only" with the atoms of the sub-dictionary that it is assigned for that specific class. The first block to build such a clustering technique is based on the following equation:

$$\min_{D_i, C_i} \sum_{i=1}^{K} \sum_{x_j \in C_i} R(x_i, D_i) \qquad \textbf{(3.14)}$$

Where $D_i = \left[d_1, d_2, ..., d_{K_i} \right] \in \mathbb{R}^{n \times K_i}$ is the class C_i dictionary with K_i atoms, and R as the function measuring how well the sparse representation of signal x_i over the dictionary D_i is performed. In order to make the sub-dictionaries as independent as possible a new term "Q"is added to the above block:

$$\min_{D_i, C_i} \sum_{i=1}^{K} \sum_{x_j \in C_i} R(x_i, D_i) + \eta \sum_{i \neq j} Q(D_i, D_j) \qquad \textbf{(3.15)}$$

This new term will work as an energy formulation optimizing the dictionaries to properly represent the corresponding class strongly (R) while at the same time makes a weak representative for other classes (Q). The proposed Q function for this case is equal to:

$$Q(D_i, D_j) = \left\| D_i^T D_j \right\|_F^2 \qquad \textbf{(3.16)}$$

Before describing the nature of the constraint R we recall that the original sparse approximation problem introduced at the beginning of this thesis as: $\min_a \|x - Da\|_2^2 + \lambda \|a\|_1$ with λ acting as the tradeoff parameter between reconstruction error and sparsity. On the other hand creating a reconstructive dictionary to represent the data that has been discussed in details in the previous section results from: $\min_{\{a_i\}i=1,2,\dots,m} \sum_{i=1}^{m} \|x_i - Da_i\|_2^2 + \lambda \|a_i\|_1$ (the term λ sometimes gets neglected). For the purpose of discriminative dictionary learning and to have a good within class representation and between class discrimination the proposed constraint for R is defined as:

$$R(x,D) = \|x - Da\|_2^2$$

or **(3.17)**

$$\hat{R}(x,D) = \min_a \|x - Da\|_2^2 + \lambda \|a\|_1$$

Imposing the defined Q and R in the original discriminative dictionary learning problem in (3.15) will result in the following formulation to learn a dictionary with big between class and a small within class scatter:

$$\min_{\{D_i, A_i\}_{i=1,\dots K}} \sum_{i=1}^{K} \left\{ \|X_i - D_i A_i\|_2^2 + \lambda \sum_{j=1}^{m_i} \|a_i^j\|_1 \right\} + \eta \sum_{i \neq j} \|D_i^T D_j\|_F^2 \qquad \textbf{(3.18)}$$

Experimental results using this approach on different datasets shows an accuracy of 85% to 95% for different datasets:

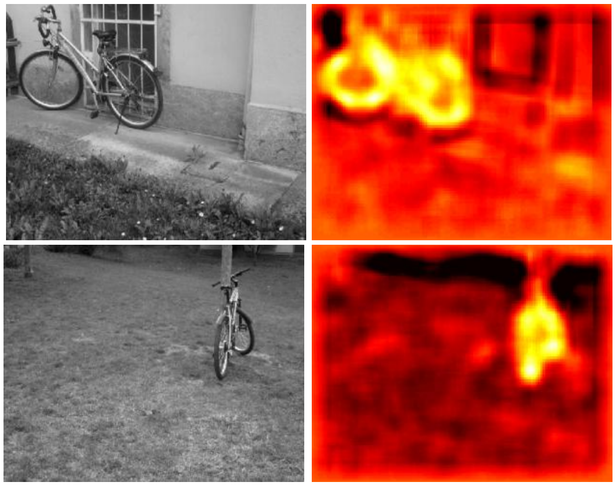


Figure 3-7 Bike detection on the Graz dataset using the described method. The detected object is displayed with hot pixels. Image is acquired from [39].

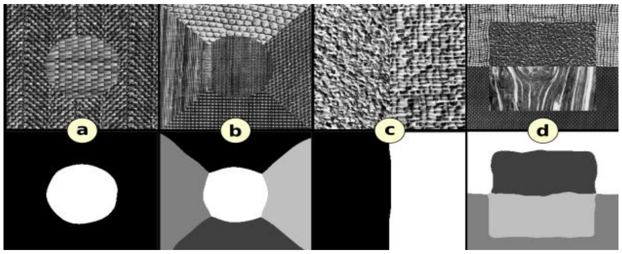


Figure 3-8 Brodatz dataset classification using the described method, classification results of the images on the top is shown in the bottom row. Image belongs to [39].

3.5.2 Discriminative K-SVD

A popular dictionary learning algorithm, K-SVD has no discriminative behavior; in fact it is a proper method to learn reconstructive dictionaries which can be used within image representation, compression and denoising applications. In order to utilize this popular method in a clustering and/or classification frame work it needs to be learned under certain conditions which can impose the discriminative behavior to the trained dictionary. Discriminative K-SVD (DK-SVD) dictionary learning algorithm [40], [41] adds the necessary discrimination criterion to the conventional K-SVD as well as maintaining its reconstructive features in order to perform local image discrimination tasks. This is done by proposing an energy formulation with both sparse reconstruction and class discrimination components getting optimized during the processes of learning and updating.

DK-SVD tries to learn multiple class specific dictionaries which are simultaneously reconstructive for one class and discriminative for the other(s). It uses the reconstruction error of the dictionaries on image patches to achieve a pixelwise classification rate. This learning technique has the advantages of not only learning redundant non-parametric dictionaries, but also the sparse local representations are learned with an explicit discriminative purpose. As in K-SVD and sparse representation pursuit algorithms DK-SVD uses the residual vector as well. The residual vector r is defined as:

$$\begin{aligned}&\alpha^*(x,D) \equiv \underset{\alpha\in\mathbb{R}^k}{\arg\min} \|x - D\alpha\|_2^2 \text{ such that } \|\alpha\|_0 \le L \\ &r(x,D,\alpha) \equiv \|x - D\alpha\|_2^2 \\ &r^*(x,D) \equiv \|x - D\alpha^*(x,D)\|_2^2\end{aligned} \tag{3.19}$$

As mentioned above the algorithm takes advantage of the residual vector just like the K-SVD, but the method increases the discriminative power of $r^*(x, D_i)$ with the goal of dictionary D_i associated to a specific class S_i should be "good" to reconstruct this class but at the same time "bad" for representing other classes. A discriminative term for different classes $(i = 1, 2, ...N)$ is defined as follows:

$$C_i^\lambda(y_1, y_2, ..., y_N) \equiv \log\left(\sum_{j=1}^{N} e^{-\lambda(y_j - y_i)}\right) \tag{3.20}$$

Which is close to zero when y_i is the smallest value among y_j, and provides an asymptotic linear penalty cost $\lambda(y_i - \min_j y_j)$ otherwise. A positive value for λ results in a high penalty cost for each missed classified patch. Having N dictionaries for N distinct classes the question is to solve:

$$\min_{\{D_j\}_{j=1...N}} \sum_{\substack{i=1...N \\ l\in S_i}} C_i^\lambda\left(\left\{R^*(x_l, D_j)\right\}_{j=1}^{N}\right) + \lambda\gamma r^*(x_l, D_i) \tag{3.21}$$

In which $\gamma \ge 0$ is the parameter controlling the trade-off between reconstruction and discrimination (The higher γ is, the more reconstructive is the dictionary). Solving this problem makes the resulting dictionary D_i good for its own class other than any other class.

Like most of the dictionary learning methods DK-SVD consists of two major parts: sparse coding and dictionary update. In DK-SVD the sparse coding step is no different than K-SVD or MOD, that means keeping the dictionary atoms fixed and computing the sparse decomposition of the patches; Dictionary update stage on the other hand is a bit different than other methods. This is the part that makes the algorithm discriminative by updating the dictionary atoms sequentially while letting the corresponding

α coefficients associated with an atom in the sparse coding stage to change as well. Making the update stage similar to that of MOD by converting the optimization problem of (3.21) in the following format:

$$\min_{\{D_j\}_{j=1...N}} \sum_{\substack{i=1...N\\ l\in S_i}} C_i^{\lambda}\left(\left\{R(x_l,D_j,\alpha_{lj})\right\}_{j=1}^{N}\right)+\lambda\gamma r(x_l,D_i,\alpha_{li}) \tag{3.22}$$

With α_{li} being fixed and computed during the previous sparse coding stage. If the patches are classified correctly their cost is rapidly close to zero. It can be shown that performing *truncated Newton* iteration to update the p^{th} dictionary is equivalent to solve:

$$\min_{D^{'}\in\mathbb{R}^{n\times k}} \sum_{i=1}^{N}\sum_{l\in S_i}\omega_l R(x_l,D^{'},\alpha_{lp})$$

in which:

$$\omega_l \equiv \frac{\partial C_i^{\lambda}}{\partial y_p}\left(\left\{R^{*}(x_l,D_j)\right\}_{j=1}^{N}\right)+\lambda\gamma 1_p(i) \tag{3.23}$$

1_p is 1 if $i=p$ and 0 otherwise.

The DK-SVD dictionary learning algorithm is displayed in details in the following table.

Purpose: To train N distinct dictionaries to be able to sparsely represent as well as discriminate between the classes.

Main loop: for $i=1...N$ and $j=1...k$ update d , the j^{th} column of D_i and:

- Select the set of patches that uses d :
$$\omega \leftarrow \left\{l\in 1...M \mid \alpha_{li}[j]\neq 0\right\}. \tag{3.24}$$
- For each patch l in ω, compute the residual of the decomposition of $x_l : r_l = x_l - D_i\alpha_{li}$.
- Compute the weights ω_l (equation***) for $p=1...N$ for all l .
- Compute a new atom $d^{'}\in\mathbb{R}^{n}$ and associate coefficients $\beta\in^{|\omega|}$ that minimizes the residual error on the selected set ω :
$$\min_{\substack{\|d^{'}\|\\ \beta\in^{|\omega|}}} \sum_{\substack{p=1...N\\ l\in S_p\cap\omega}} \omega_l\left\|r_l+\alpha_{li}[j]d-\beta_l d^{'}\right\|_2^2$$
- Update D_i and α using the new atom $d^{'}$, and replace the scalars $\alpha_{li}[j]\neq 0$ from equation

```
        $\omega \leftarrow \{ l \in 1 \ldots M \mid \alpha_{li}[j] \neq 0 \}.$          (3.24)
End
```

Table 3.3 DK-SVD pseudo algorithm [40]

The results of segmenting the standard Brodatz dataset using this method are shown in Figure 3-9.

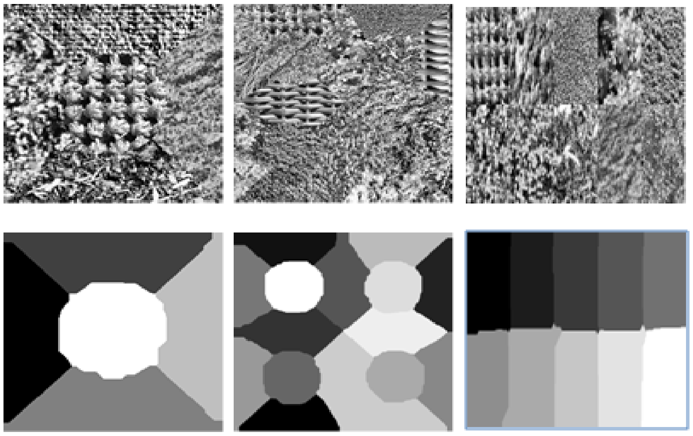


Image	Error Rate
1	1.61
2	16.42
3	4.15
4	3.67
5	4.58
6	9.04
7	8.80
8	2.24
9	2.04
10	0.17

Figure 3-9 Segmentation results of the Brodatz dataset using DK-SVD method. Original images are displayed on the top and the result of the segmentation is displayed on the bottom. The table on the right displays the error rate for 10 of Brodatz mosaics segmentation.

3.5.3 Fisher Discriminative Dictionary Learning

Last but not least among discriminative dictionary learning algorithms is Fisher discriminative dictionary learning (FDDL) introduced by Yang *et.al* [42]in 2011. In this algorithm based on *Fisher Discrimination Criterion* a dictionary with its atoms corresponding to a specific class labels is learned such that sparse coding reconstruction error (error between a reconstructed and original signal/image) can be used as an indicator for pattern classification. Using Fisher's criterion has the advantage of having a small within-class scatter but a big between-class scatter for the coding coefficients. The difference between this method and the two aforementioned algorithms is that DK-SVD and classification using l_1 regularization use the reconstruction error (residual) as the discriminative criterion for classification and they don't impose discriminative information in sparse coding coefficients, but in fisher dictionary learning scheme both the reconstruction error and fisher's discrimination criterion are used as classifiers for dictionary learning. In this method a structured dictionary $D=\left[D_1,D_2,...,D_c\right]$ is learned with D_i being the class-specific sub-dictionary associated with class i and c being the total number of classes. Denote by $A=\left[A_1,A_2,...,A_c\right]$ the set of training samples and $X=\left[X_1,X_2,...,X_c\right]$ the coding coefficient matrix of A over D ($A\approx DX$). The proposed FDDL model is as follows:

$$J_{(D,X)}=\underset{(D,X)}{\arg\min}\left\{r(A,D,X)+\lambda_1\left\|X\right\|_1+\lambda_2 f(x)\right\} \tag{3.25}$$

With $r(A,D,X)$ being the discriminative fidelity term, $\left\|X_1\right\|$ sparsity term, $f(X)$ the coefficients discrimination constraint and λ_1 and λ_2 being scalar parameters. In order to be able to present the FDDL model in details, first we have to clarify the concept of Discriminative fidelity term ($r(A,D,X)$) and the discriminative coefficient term ($f(X)$).

3.5.3.1 Discriminative fidelity term

As we already mentioned above, in this method the dictionary D is a structured dictionary with its sub-dictionaries being class specific. According to such a dictionary the matrix X_i can be written as $X_i=\left[X_i^1,...,X_i^j,...,X_i^c\right]$ in which X_i^j is the coding coefficient of the signal A_i over the sub-dictionary D_j. We indicate the representation of D_k to A_i as $R_k=D_kX_i^k$. The main goal of learning a dictionary is based on the fishers criterion so that it would be able to well represent the signal A_i, $\left(A_i\approx DX_i=D_1X_i^1+...+D_i\,X_i^i+...D_cX_i^c=R_1+...+R_i+...R_c\right)$. And also it is desired that A_i should be represented by D_i and not the D_j ($j\neq i$). This implies that $\left\|A_i-D_iX_i^i\right\|_F^2$ is small, while X_i^j should have as minimum non-zero coefficients as possible such that $\left\|D_jD_i^j\right\|_F^2$ is small. Thus the discriminative fidelity term is defined as:

$$r(A_i, D, X_i) = \left\| A_i - DX_i \right\|_F^2 + \left\| A_i - D_i X_i^i \right\|_F^2 + \sum_{\substack{j=1 \\ j \neq i}}^{c} \left\| D_j X_i^j \right\|_F^2 \qquad \textbf{(3.26)}$$

Figure 3-10 demonstrates the role and importance of each of the components in equation (3.26). Figure 3-10 (a) indicates that although D is ensured to represent A_i well, R_i may deviate much from A_i so that D_i could not well represent A_i. Figure 3-9 (b) illustrates that if the constraint of $\left\| A_i - D_i X_i^i \right\|_F^2$ being small is added better discrimination results will be achieved but the problem is that A_i may also be well represented by other sub-dictionaries (not D_i).Figure 3-10 (c) shows how to overcome this problem by forcing the representation of D_j, $j \neq i$ to A_i to be small and the discrimination fidelity term is proposed completely.

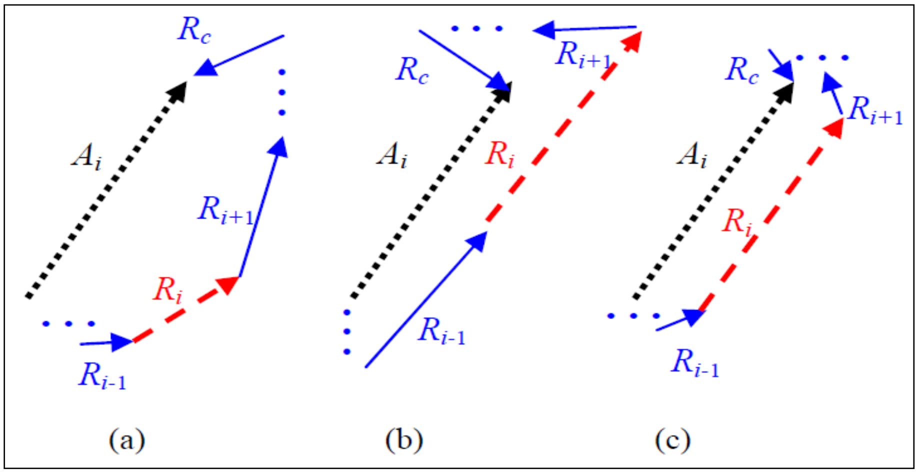


Figure 3-10 FDDL discriminative fidelity term. (a) Only D is required to well represent A_i . (b) Both D and D_i are required to well represent A_i .(C) The discriminative fidelity term in equation(3.26). Figure is acquired from [42].

3.5.3.2 Discriminative Coefficient Term

In order to impose a more discriminative capability to the dictionary for the signal samples in A, we make the coding coefficient of A over D discriminative. This is a fact that can be achieved by minimizing the within-class scatter and maximizing the between-class scatter of X . Within-class and between-class scatters are denoted as S_W and S_B respectively and are defined as:

$$\begin{aligned} S_W(X) &= \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - m_i)(x_k - m_i)^T \\ S_B(X) &= \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T \end{aligned} \tag{3.27}$$

Where m_i and m are the mean vector of X_i and X and n_i being the number of samples in class A_i . The discriminative fidelity term is defined as:

$$f(x) = tr(S_W(X)) - tr(S_B(X)) + \eta \|X\|_F^2 \tag{3.28}$$

With *"tr"* being the trace of the matrices and η is a parameter that will be discussed in the following sub-section.

Having introduced the fidelity and the discriminative coefficient terms the actual FDDL model can be defined by substituting equations (3.26) and (3.27) in equation (3.28):

$$J_{(D,X)} = \arg\min_{(D,X)} \left\{ \sum_{i=1}^{c} r(A_i, D, X_i) + \lambda_1 \|X\|_1 + \lambda_2 \left(tr(S_W(X) - S_B(X)) + \eta \|X\|_F^2 \right) \right\} \tag{3.29}$$

Just like any dictionary learning algorithm the process of learning a dictionary through FDDL is consist of two main fractions: updating X by fixing D and updating D by fixing X both in an iterative procedure. While X is fixed D_i is updated class by class (it updates D_i when all $D_j, j \neq i$ are fixed). In order to solve the FDDL dictionary learning problem in (3.29) this problem is first optimized to the following form:

$$J_{X_i} = \arg\min_{X_i} \left\{ r(A_i, D, X_i) + \lambda_1 \|X_i\|_1 + \lambda_2 f_i(X_i) \right\} \tag{3.30}$$

In which:

$$f_i(X_i) = \|X_i - M_i\|_F^2 - \sum_{k=1}^{c} \|M_k - M\|_F^2 + \eta \|X_i\|_F^2 \tag{3.31}$$

With M_k and M be the mean vector matrices of class k and all classes respectively.

In order to update D class by class, the objective function would be:

$$J_{D_i} = \arg\min_{D_i} \left\{ \left\| A - D_i X^i - \sum_{j=1, j \neq i}^{c} D_j X^j \right\|_F^2 + \left\| A_i - D_i X_i^i \right\|_F^2 + \sum_{j=1, j \neq i}^{c} \left\| D_i X_i^i \right\|_F^2 \right\} \quad \textbf{(3.32)}$$

The FDDL algorithm is demonstrated in Table 3.4.

Purpose: To train c distinct dictionaries to be able to sparsely represent as well as discriminate between the classes.

Initialization: the p_i atoms of each D_i dictionary can be initialized randomly. (it has to be with a unit l_2 norm)

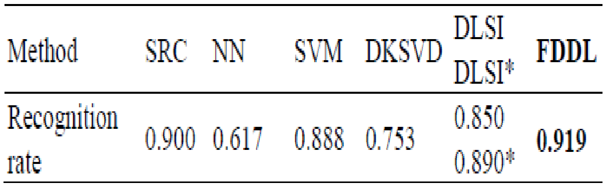
Main loop:

- **Updating the sparse coding coefficients X :**
  Fix the dictionary D, solve for $X_i, i = 1, 2, ..., c$ one by one by solving equation (3.30)
- **Updating the dictionary D :**
  Fix X and update each D_i independently by solving equation (3.32)
- **Stopping criterion:**
  Return to step one (updating X) until the maximum number of iterations are achieved or the value of $J_{(X,D)}$ in adjoining iterations are close enough.

Output: the representation vector X and discriminative dictionary D consists of sub-dictionaries for c classes.

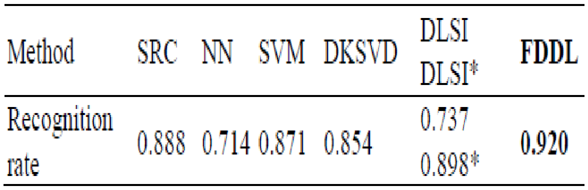
Table 3.4. The Fisher's discriminative dictionary learning pseudo algorithm

FDDL has been implemented on various datasets to evaluate its performance and comparison with other classification methods. The following tables acquired from [42] are represented in order to evaluate FDDL performance.



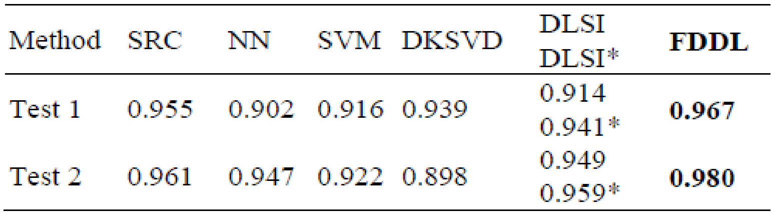
Method	SRC	NN	SVM	DKSVD	DLSI DLSI*	**FDDL**
Recognition rate	0.900	0.617	0.888	0.753	0.850 0.890*	**0.919**

Table 3.5 recognitions rates for various methods on Extended Yale database



Method	SRC	NN	SVM	DKSVD	DLSI DLSI*	**FDDL**
Recognition rate	0.888	0.714	0.871	0.854	0.737 0.898*	**0.920**

Table 3.6 recognitions rates for various methods on AR database



Method	SRC	NN	SVM	DKSVD	DLSI DLSI*	**FDDL**
Test 1	0.955	0.902	0.916	0.939	0.914 0.941*	**0.967**
Test 2	0.961	0.947	0.922	0.898	0.949 0.959*	**0.980**

Table 3.7 recognitions rates for various methods on Multi-Pie database

Chapter Four: Classification and Clustering based on Sparse Representation

During the past few years, classification and clustering has gained a considerable amount of attention by image and signal processing researchers and scholars around the world. Successful applications of sparse representation in image restoration [7], denoising [6] and compressed sensing [5] as well as the fact that most natural signals can compactly be represented by only a few coefficients that carry the most important information in a certain basis or dictionary, proved that sparse coding algorithms can benefit the image processing community in numerous ways. Yet image classification, clustering and even segmentation using sparse based techniques are not yet vastly analyzed. The key factor in classification and clustering based on sparse representation is that different classes of an image should be presented over different class-specific dictionaries so that their patches can be labeled accordingly.

The reason for the emergence of this application (classification) through the field of sparse representation and dictionary learning is that despite the high dimensionality of natural signals, the signals of the same class usually lie within a low-dimensional subspace. Hence if the class specific dictionary is able to cover this subspace, it can result in a proper classification results. Therefore it is correct to say that for every typical signal (input patch) there exists a sparse representation with respect to a proper dictionary. Different works in the field of compress sensing ensure that a sparse signal can be recovered from its projections with a high probability [5], [43], [44]. This fact allows the recovery of the sparse representation vector(s) by decomposing the sample over a dictionary. Having the representation vector, one can extract the semantic information from the recovered sparse representation vector. Applications of sparse representation based classification techniques in the field of pattern recognition and computer vision can be found in: face recognition [45], iris recognition [13], image super-resolution [46] and other applications. In this thesis we have extended this new emerging field of research into the concept of medical image processing.

As discussed before sparse representation techniques have a beneficial effect on classical signal processing problems (compression, representation, etc) with a compact high-fidelity representation. However in pattern recognition problems, the main interest is in the content or semantics of an image, not just the compact representation of it. This computer vision point of view is a new field in sparse representation studies that has grown due to the development of l_1 minimization techniques in the past few years. In most approaches sparsity is usually used as a prior for clustering, in fact sparsity and proper dictionary learning have influenced the emergence of not only new algorithms but also new physical imaging systems [47] [48] [49].

The ability of sparse representations to uncover semantic information derives in part from a simple but important property of the data:

- Although the images (or their features) are naturally very high dimensional, in many applications, images belonging to the same class lie on or near low-dimensional subspaces.
- Such a sparse representation, if computed correctly, might naturally encode semantic information about the image.

Among these newly emerging algorithms, “Robust Face Recognition via Sparse Representation” introduced by Wright *et. al* [45] has gained most of the attention by scholars working in the field of sparse representation based classification. This algorithm resulted in the famous classification scheme based on sparse representation known as: “Sparse Representation Classification” or simply SRC. SRC consists of two phases: Coding and Classification. First a query image is coded over a dictionary with some sparsity constraint; the classification is then performed based on the coding coefficients and the minimal reconstruction error. This use of the minimal reconstruction error for classifying signals/images in a sparse representation framework has opened a new and fast growing field in the theme of pattern recognition and computer vision based on sparse representation. This chapter starts by looking at the first sparse image separation study which is very significant as it motivated further studies in image classification based on sparse representation techniques. Following that the SRC frame work is presented. Some of these new emerging approaches which are based on reconstruction error are then introduced. An example of our contribution to the field of medical image processing using sparse based techniques is presented at the end of the chapter.

4.1 Motivation: Texture Separation

The first work that focused on representing different classes of an image using different dictionaries focused on decomposing an image into its texture and non-texture components or as it is said in the literature: Cartoon and Texture parts [12]. The core idea of separating texture images into cartoon and texture components is to try to represent these different components using separate dictionaries. As it is mentioned before, representing different parts or classes of an image over different dictionaries is the core idea behind classification and clustering using sparse methods.

For image texture separation using sparse representation like most of the signal decomposition techniques it is assumed that the original image is a linear combination of its constructing sub signals: $I = I_c + I_t + n$ with I_c and I_t being the non-texture (cartoon) and texture components respectively and n as the white Gaussian noise with a known standard deviation σ. In all sparse based separation frameworks selecting the proper dictionaries is a key point. This is mainly due to the fact that the separation process is carried out through the representation of different classes over the dictionaries. Starck *et. al* [50]showed that since DCT and Gabor transforms result in oscillatory atoms; these dictionaries are suitable for representing the texture data while Bi-Orthogonal Wavelet Transform, Curvelet Transform, Local Ridgelet Transform are the proper candidate dictionaries for sparsely representing the cartoon components of an image. Considering the image as a linear combination of its components, solving the following equation will result in the corresponding sparse representations of the two classes:

$$\hat{X}_c, \hat{X}_t = \arg\min_{X_c, X_t} \lambda \|X_c\|_1 + \lambda \|X_t\|_1 + \frac{1}{2}\|I - D_c X_c - D_t X_t\|_2^2 \qquad \textbf{(4.1)}$$

In which λ is the constant balancing between fitting the data completely and having the sparsest solution, D_c and D_t are the corresponding dictionaries for each class and X_c and X_t are the sparse vectors for representing each class. Finding the representation of each class, they can be estimated as: $\hat{I}_c = D_c \hat{X}_c$ and $\hat{I}_t = D_t \hat{X}_t$ with $\hat{I}_c$ and $\hat{I}_t$ as the estimation for each class. Imposing such formulation in equation (4.1) will result in:

$$\hat{I}_c, \hat{I}_t = \underset{I_c, I_t}{\arg\min}\, \lambda \left\| T_c I_c \right\|_1 + \lambda \left\| T_t I_t \right\|_1 + \frac{1}{2} \left\| I - I_c - I_t \right\|_2^2 \tag{4.2}$$

Where $T = D^{-1}$. Daubechies *et. al* [51] suggested Separable Surrogate Functions (SSF) algorithm to solve the following optimization problem to find an iterative solution for determining $\hat{I}_c$ and $\hat{I}_t$:

$$\hat{X}_c, \hat{X}_t = \underset{X_c, X_t}{\arg\min}\, \lambda \left\| X_c \right\|_1 + \lambda \left\| X_t \right\|_1 + \frac{1}{2} \left\| I - D_c X_c - D_t X_t \right\|_2^2 \text{subject to } T_c D_c X_c = X_c \text{ and } T_t D_t X_t = X_t \tag{4.3}$$

Using SSF will result in the followed iteration formulation to find$\hat{I}_1$,$\hat{I}_2$:

$$\begin{aligned} I_c^{K+1} &= D_c S_\lambda \left(\tfrac{1}{c} D_c^T \left(I - \hat{I}_c^K - \hat{I}_t^K \right) + T_c \hat{I}_c^K \right) \\ I_t^{K+1} &= D_t S_\lambda \left(\tfrac{1}{c} D_t^T \left(I - \hat{I}_c^K - \hat{I}_t^K \right) + T_t \hat{I}_t^K \right) \end{aligned} \tag{4.4}$$

Where $S_\lambda(r)$ is an element-wise soft thresholding operation on r with a threshold λ and the parameter c should be chosen such that $c > \lambda_{\max}(D_a D_a^T) = \lambda_{\max}(D_c D_c^T + D_t D_t^T)$. For further backgrounds and mathematical justifications readers are advised to refer to [50]and [12]chapter15.

The aforementioned method has been implemented on the famous Barbara image and on a MRI image of the human knee using a DCT dictionary for texture and a Curvelets dictionary for cartoon contents of the images.

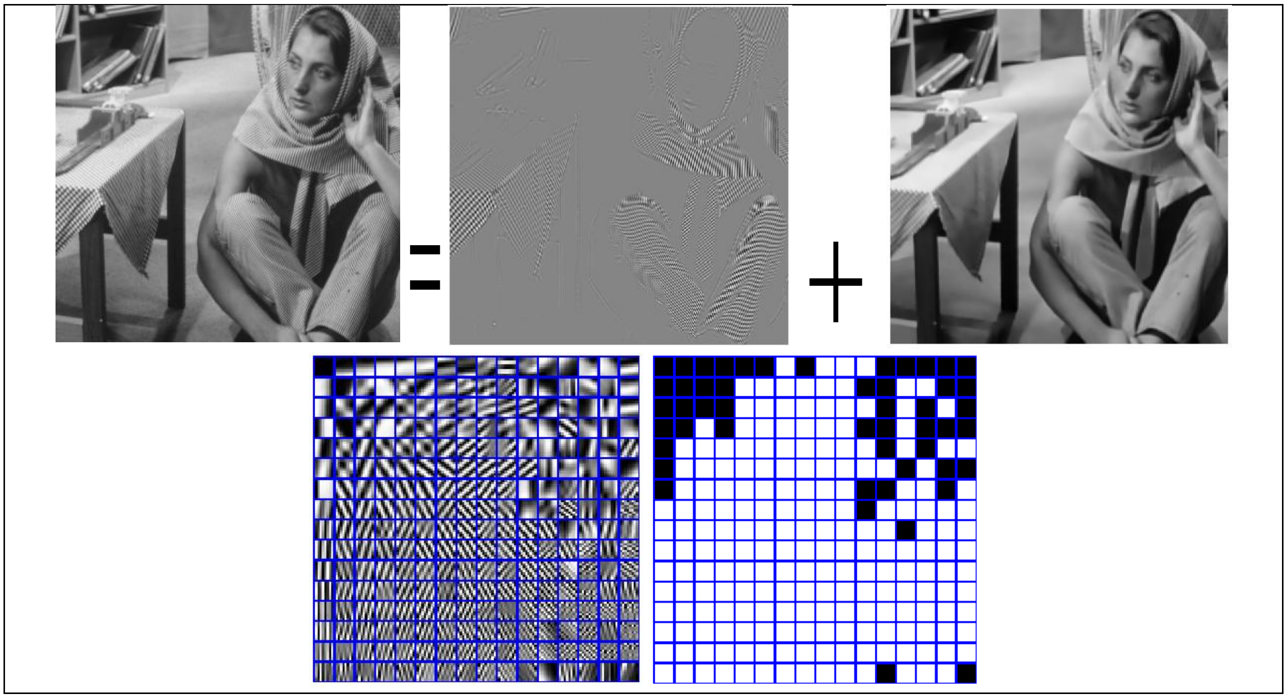


Figure 4-1 Top: shows the original Barbara image (left) with the separated texture (middle) and the cartoon contents (right). Bottom: Both dictionary atoms in the left. Image in the right is a mapping that shows the label of each dictionary atom. Patches with a white label belong to the DCT dictionary, and patches with a black label belong to the Curvelets dictionary.

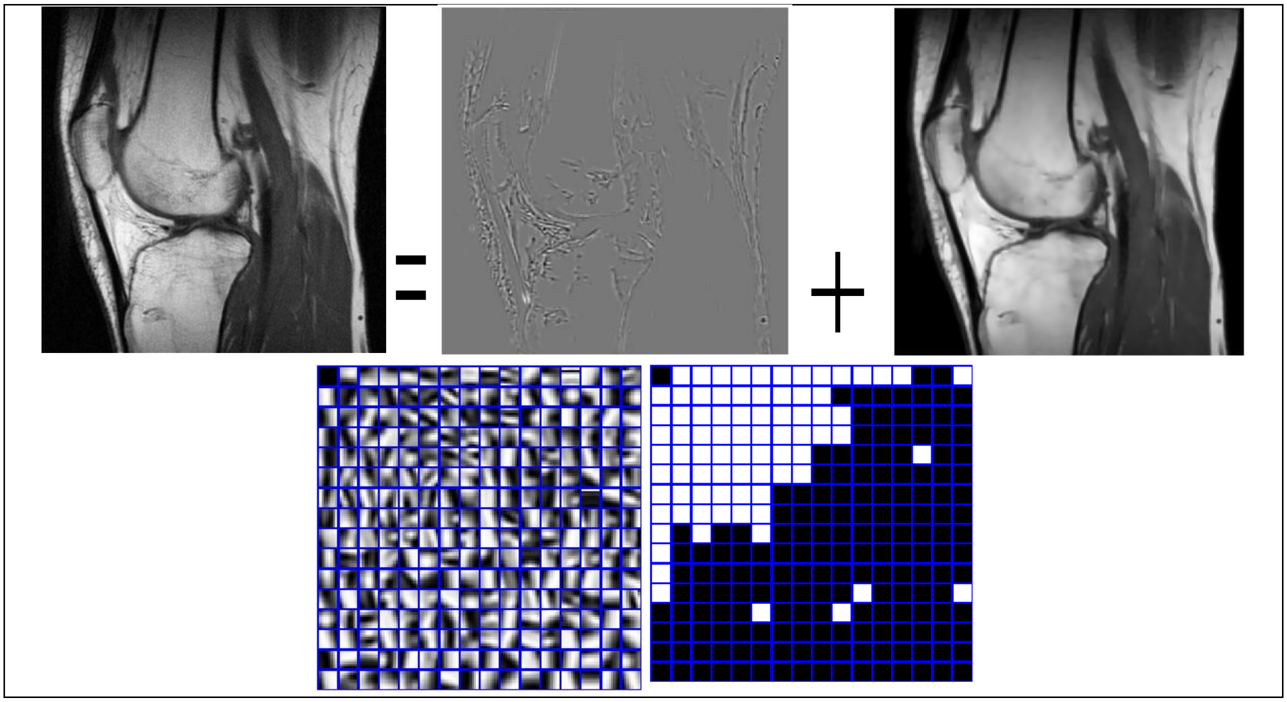


Figure 4-2 same experiment and results as Figure 4-1 on an MRI Image of the human knee.

The discussed frame work for texture detection takes advantage of Separable Surrogate Functions in order to solve the problem in (4.3). It utilizes two class specific learned (learned with K-SVD) dictionaries and represents the data over them to separate the two label of the classes. It is among the first methods that perform an image separation task based on sparse representation and dictionary learning approach.

4.2 Sparse Representation Classification (SRC)

SRC is considered to be one of the leading methods in the field of classification based on sparse representation. One important fact which can be considered as a drawback of this method is that SRC does not use a generic dictionary (whether it is reconstructive or discriminative) to represent the data over, but instead it uses the actual training dataset as a space to represent the data over. If the training data set is large enough the SRC eliminates the critical need of features for the purpose of classification [45]. On the other hand using large training datasets and representing the data over them, not only makes the method slow but also dependent on large training data samples which are not always available. Some methods like the one in [42], have overcome this problem by introducing the use of SRC in parallel with specific discriminative dictionaries such as Fisher discriminative dictionaries. In SRC if the dictionary is an over-complete dictionary of the training samples (sufficient numbers of training samples are available in the dictionary for each class) the algorithm will sparsely represent the test samples of each class as a linear combination of the training samples of the same class. It will be shown that this sparse representation is actually the sparsest representation (in most cases) of the image over this dictionary and the original test image can be recovered by means of l_1 minimization. As a result seeking the sparsest representation is discriminating between different classes. Basically each test sample is individually sparsely presented and adaptively selects the training sample that provides the most compact representation (the use of the reconstruction error). This approach makes the general form of SRC to be somewhat close to a generalization of the Nearest Neighbor (NN) method [52]. NN classifies a test sample based on the best representation in terms of a single training sample whereas SRC uses a similar approach but it considers all possible supports and adaptively chooses the minimum number of atoms (atoms in this case are the training samples) needed to represent each sample.

4.2.1 The Classification Scheme

In most object detection techniques a common approach is to use labeled training data from K different object classes, consequently after giving a new test sample it determines to which class this test data belongs. In this case the training data for a distinct class i is arranged as columns of a matrix $A_i = \left[\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,n_i}\right] \in \mathbb{R}^{m \times n_i}$ for n_i training samples. For the purpose of classification the algorithm will classify object classes using the vector $\lambda \in \mathbb{R}^m$. Basically each vector in A is a training sample for the i^{th} class which lies on a linear subspace. Subspace models are flexible enough to capture much of the variation in real data sets. It is assumed that the training samples from a single class lie on a single subspace [45]. A new object sample $y \in \mathbb{R}^m$ belonging to one the classes can and will be presented as a linear combination of the training samples:

$$y = \alpha_{i,1}\lambda_{i,1} + \alpha_{i,2}\lambda_{i,2} + \ldots + \alpha_{i,n_i}\lambda_{i,n_i} \tag{4.5}$$

For some scalar $\alpha_{i,j} \in \mathbb{R},\ j = 1, 2, \ldots, n_i$. In order to generalize the problem so that any given object can be represented and later classified without knowing to which class it belongs, dictionary matrix D for k distinct classes is defined as:

$$D = \left[A_1, A_2, \ldots, A_k\right] = \left[\lambda_{1,1}, \lambda_{1,2}, \ldots, \lambda_{k,n_k}\right] \tag{4.6}$$

So the linear representation of y can be written in the general form of:

$$y = D\alpha_0 \tag{4.7}$$

In which $\alpha_0 = \left[0,...,0,\alpha_{i,1},\alpha_{i,2},...,\alpha_{i,n},0,...0\right]^T \in \mathbb{R}^n$ is a coefficient vector with all of its entries being zero except those associated with the i^{th} class. The solution to equation (4.7) can be gained either by choosing the minimum l_2-norm solution of:

$$\hat{\alpha}_1 = \arg\min \|\alpha\|_2 \text{ subject to } y = D\alpha \tag{4.8}$$

Or by solving the convex optimization problem of:

$$\hat{\alpha}_1 = \arg\min \|\alpha\|_1 \text{ subject to } \|D\alpha - y\|_2 \leq \xi \tag{4.9}$$

This representation is naturally sparse if the number of object classes k is reasonably large.

For the purpose of classification of a new test sample y one should compute the sparse representation $\hat{\alpha}_1$ over the dictionary D ($D = \left[A_1, A_2, ..., A_k\right]$) via (4.8) or (4.9) and associate the non-zero entries of the estimate $\hat{\alpha}_1$ with the columns of D from a single object class i , ideally the test sample y can be assigned to that specific class. Unfortunately the presence of noise, modeling errors and the similarity of information in specific patches will lead to some non-zero entries associated to multiple classes (miss classified patches). In order to overcome such inaccuracies y is instead classified based on how well the coefficients associated with all training samples of each object reproduce it (y). In order to overcome the aforementioned problems γ_i is defined for each class to choose the best coefficients associated with the class i . By using a class specific coefficient, a given test sample y can then be approximated as:

$$\hat{y}_i = D.\gamma_i(\hat{\alpha}_1) \tag{4.10}$$

y can then be classified based on the approximations that minimize the residual r between y and $\hat{y}_i$ by:

$$\min_i r_i(y) = \|y - D.\gamma_i(\hat{\alpha}_1)\|_2 \tag{4.11}$$

Equation (4.11) will result in a representation of y over D which is not only sparse but also labels y based on its minimum reconstruction error for its specific class. Table 4.1 demonstrates the pseudo algorithm for SRC method

Purpose: to sparsely represent a test sample y, and label it based on its minimum reconstruction error over the dictionary D.

Input: A dictionary consists of as many training samples as possible, a test sample $y \in \mathbb{R}^m$ and ξ as the termination criterion.

Normalization: Normalize columns of D to have a unit l_2-norm.

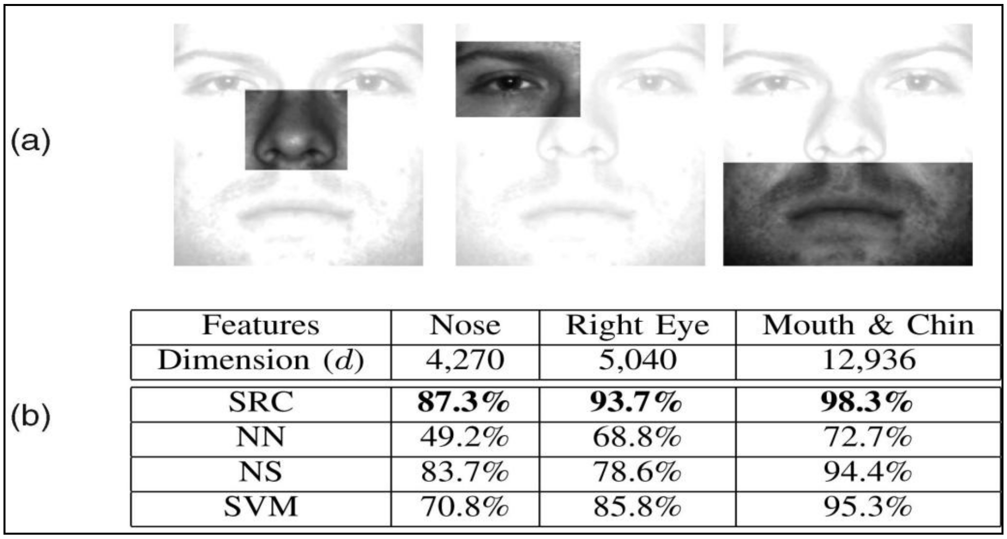
Main loop:

- **Solve :**
  $\hat{\alpha}_1 = \arg\min \|\alpha\|_1$ subject to $\|D\alpha - y\|_2 \le \xi$
- Compute the residual $r_i(y) = \|y - D\gamma_i(\hat{\alpha}_1)\|_2$

Output: $y = \arg\min_i r_i(y)$

Table 4.1 The SRC pseudo algorithm

Figure 4-3 exhibits a comparison between the performance of SRC and some of the well known conventional methods for face recognition applications.



Features	Nose	Right Eye	Mouth & Chin
Dimension (d)	4,270	5,040	12,936
SRC	**87.3%**	**93.7%**	**98.3%**
NN	49.2%	68.8%	72.7%
NS	83.7%	78.6%	94.4%
SVM	70.8%	85.8%	95.3%

Figure 4-3 Face recognition results with partial features. (a) example features, (b) results for different methods on Extended Yale B database [45].

Recently SRC has gained lots of attention by researchers in the field of classification and clustering based on sparse representation. Many variations of this method have been developed using the concept of reconstruction error for sparsely classifying images. In the following section we will shortly discuss the main concept of two of these methods.

4.3 Reconstruction Error Base, Classification

After proposing the sparse representation texture separation method by Starck *et. al* [50]the idea of utilizing dictionary learning and sparse representation for the purpose of classification flourished. After many works by scholars around the world, this idea turned into a practical method with the emergence of SRC framework by Wright *et. al* [45]in 2009. Since then many variations of using reconstruction error as an indicator for classification have been developed. Among these recently developed works the papers by Guha *et. al* [53] and Sivalingam *et. al* [54]are of more interest to us.

Guha *et. al* [53] proposed a sparse classification method for image and video signals and took advantage of the reconstruction error in their method. They describe each training signal by an error vector consisting of the reconstruction errors that a test signal produces with respect to each class specific dictionary. Since there exists a dictionary for each class, it is expected to gain small reconstruction error over that class specific dictionary and simultaneously large error for reconstructing a signal over a dictionary which is learned for other classes. Utilizing the Mahalonobis distance as a measure for clustering based on errors; they classify a test signal according to their reconstruction error.

Another recent method which performs the classification task based on the reconstruction error is presented by Sivalingam *et. al* [54] in 2010. The core idea of this method is based on the proposed method by Mairal *et. al* [40], which was discussed earlier in chapter 3, section 3.5.2 under the topic of Discriminative K-SVD dictionary learning algorithm. Learning the dictionaries through DK-SVD algorithm and representing the data utilizing OMP method, they create the reconstruction error curves. These curves will be used as indicators for classifying different classes in an image.

For a comprehended explanation on these two methods, the authors are referred to [53], [54]and references therein.

4.3.1 Segmenting T_2 MRI images using the reconstruction error

Following the same path that the above mentioned scholars in [53] and [54]passed, by taking advantage of the reconstruction error, we developed a method to classify T_2 MRI images of the brain and segment the lateral ventricle in these images. Utilizing K-SVD dictionary learning we trained two dictionaries for the lateral ventricle and cerebral cortex of the brain. By taking advantage of the introduced SRC frame work, we sparsely represent the data over these dictionaries individually and by calculating the minimum reconstruction error in representing the patches of the image; we label and classify them. We then extract and employ the center of the detected patches as a seed point for the Active Contour Without Edges segmentation framework [55] to fully segment this area. The following block diagram shows the steps taken in this method.

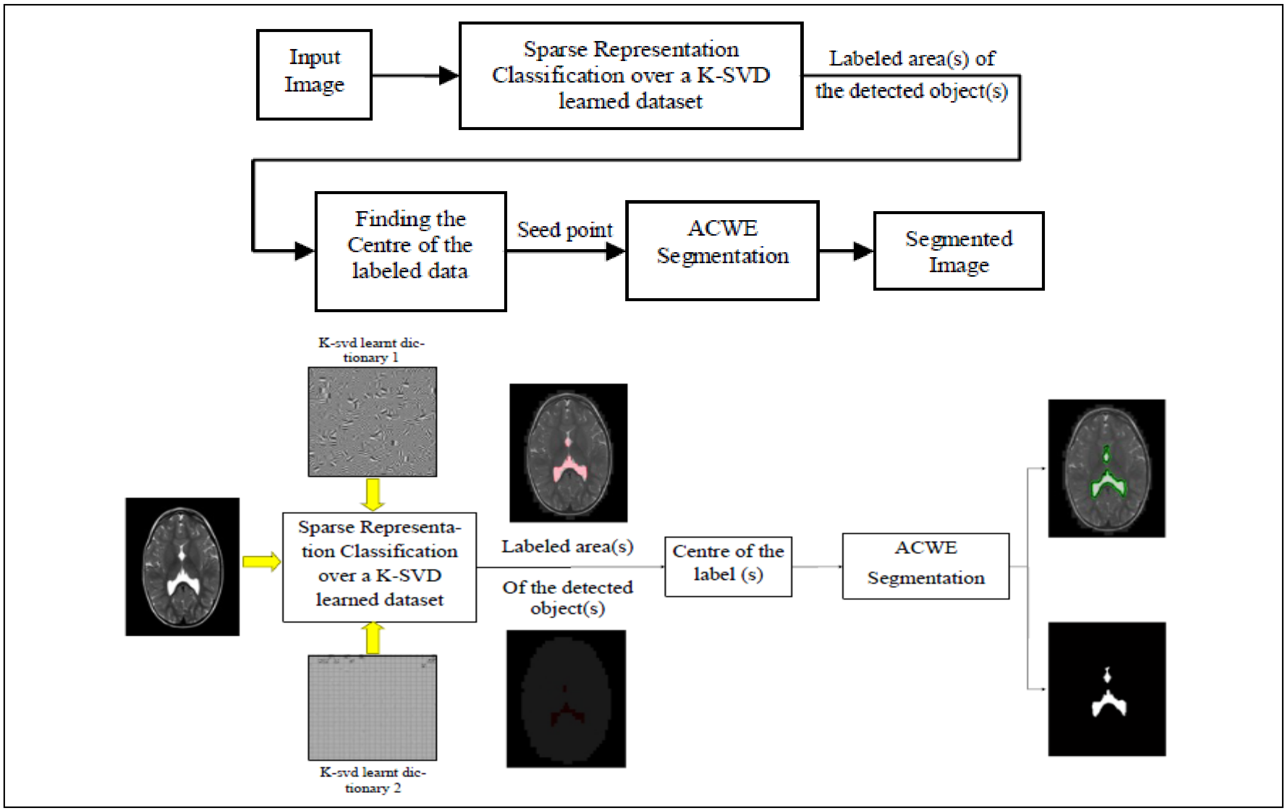

Input Image
Sparse Representation Classification over a K-SVD learned dataset
Labeled area(s) of the detected object(s)
Finding the Centre of the labeled data
Seed point
ACWE Segmentation
Segmented Image
K-svd learnt dictionary 1
Sparse Representation Classification over a K-SVD learned dataset
Labeled area(s)
Of the detected object(s)
Centre of the label (s)
ACWE Segmentation
K-svd learnt dictionary 2

Figure 4-4 The image on the top demonstrates the block diagram of the method and the image in the bottom illustrates the segmentation procedure for an example image.

The implementation of the algorithm on 25 T_2 weighted MRI images showed an average accuracy of 88.7% and an average computational time of 430 seconds for 400×400 size images. The results of this work is published in IEEE 11th international conference on information science, signal processing and their applications.

More details on MRI images and the importance of segmenting the lateral ventricle will be presented in the final chapter when we present our novel segmentation framework.

Chapter Five: Image Segmentation and Sparse Representation Classification Based on Euclidian Distances

In this chapter of the thesis a novel framework for clustering and classification signals and images based on sparse representation and dictionary learning approach and our segmentation framework are presented. We are presenting a technique that can map the sparse representation vector to Euclidian distances. We basically transform the sparse domain into a new space which is based on Euclidian distances. Based on these distances we can perform the classification task with a higher accuracy and much faster than other existing sparse based classification techniques. Like most sparse based classification techniques we are using class specific learned dictionaries to increase the accuracy of the algorithm. Unlike other sparse based methods however, we are not using the reconstruction error for the purpose of classification, instead, we are using the aforementioned transform. This method finds the minimum Euclidian distance between an input patch and the atoms of a learned-base dictionary to perform the classification task. Since in sparse based methods that perform the labeling task based on the minimum reconstruction error, the algorithm has to compute the residual vector for each and every patch of the test signal therefore making these methods computationally expensive. Besides the computational time, in SRC a very large training data set is needed; and as we have mentioned earlier this is the main drawback of the SRC. In our method the reconstruction error calculations are not required, making the method fast and also the problem of requiring a large dataset is solved by means of using learned over-complete dictionaries. In this chapter we present the methodology and justifications behind this novel framework in details. To evaluate our method we implemented the method on the famous Brodatz texture dataset on the same images that other methods such as SRC are tested on. We also compare our classification results with SVM and K-NN as two non-sparse based classification techniques. This framework and its results have been presented to IEEE 26th international conference on visual communication and image processing (VCIP) conference with a special session on sparse representation. It will be available on IEEE xplore digital library in the near future.

5.1 Methodology

5.1.1 OMP and K-SVD

This method takes advantage of the Orthogonal Matching Pursuit (OMP) and K-SVD dictionary learning algorithms presented in sections 2.3.1.2 and 3.4.2 respectively. Having the sparse representation problem of:

$$\arg\min_{\alpha} \left\| y - D\alpha \right\| \text{ subject to } \left\| \alpha \right\|_0 \leq \Gamma \qquad \textbf{(5.1)}$$

For a signal/image y, a dictionary D and a representation vector α with a termination criteria Γ the OMP finds a sparse solution for (5.1) in a recursive manner. It maintains a converging solution for non-orthogonal dictionaries. The OMP attempts to recursively minimize the residual of the reconstructed patch by finding the best matching direction to the residual among all dictionary atoms, per iteration. The main idea behind the OMP algorithm is to solve the following equation in each iteration:

$$\tilde{\alpha}^{(iter)} = \tilde{D}^{\dagger} y \tag{5.2}$$

In which $\tilde{D}^{\dagger} = (\tilde{D}^{T}\tilde{D})^{-1}\tilde{D}^{T}$ is the pseudo-inverse of the dictionary matrix $\tilde{D}$. In order to learn class specific dictionaries for k distinct classes ($D = \{D_1 | ... | D_k\}$), K-SVD algorithm is used. Here we just use OMP and K-SVD algorithms as parts of our classification framework, for more details about these algorithms readers are referred to chapter 2 and chapter 3 of this thesis.

5.1.2 Sparse Euclidean Classification:

The clustering frame work that we call it Sparse Euclidian Classification (SEC) is based on sparsely representing a signal with two non-zero coefficients ($\Gamma = 2$) the method is as follows:

Suppose there are two classes of images (can be two classes in one image) and OMP is used to represent the image patches with two non-zero coefficients α_1 and α_2 (sparsity level $\Gamma = 2$), the vector $\tilde{\alpha}$ corresponding to two atoms (templates) can be defined as:

$$\tilde{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}^{T} = (\tilde{D}^{\dagger} y)^{T} \tag{5.3}$$

In which $\tilde{D} = \left[d_{i1}, d_{i2}\right]$ is the dictionary with two sub-dictionaries learned to have minimum distances to the input pattern for each class. By minimum distance we basically indicate that d_1 and d_2 are learned through K-SVD algorithm specifically for class 1 and class 2 respectively. The pseudo inverse problem of $\tilde{\alpha} = \tilde{D}^{\dagger} y$ can be expanded as:

$$\begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \end{bmatrix} = \tilde{D}^{\dagger} y = \left(\begin{bmatrix} d_{i1}^T \\ d_{i2}^T \end{bmatrix} [d_{i1} \quad d_{i2}] \right)^{-1} \cdot \begin{bmatrix} d_{i1}^T \\ d_{i2}^T \end{bmatrix} y \tag{5.4}$$

And:

$$\begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \end{bmatrix} = \frac{\begin{bmatrix} (d_{i2}^T \cdot d_{i2}) & -(d_{i1}^T \cdot d_{i2}) \\ -(d_{i2}^T \cdot d_{i1}) & (d_{i1}^T \cdot d_{i1}) \end{bmatrix} \cdot \begin{bmatrix} d_{i1}^T \cdot y \\ d_{i2}^T \cdot y \end{bmatrix}}{(d_{i1}^T \cdot d_{i1})(d_{i1}^T \cdot d_{i1}) - (d_{i2}^T \cdot d_{i1})(d_{i1}^T \cdot d_{i2})}$$

$$\tilde{\alpha} = \begin{bmatrix} \alpha_{i1} \\ \alpha_{i2} \end{bmatrix} = \frac{M}{c} \cdot \begin{bmatrix} d_{i1}^T \cdot y \\ d_{i2}^T \cdot y \end{bmatrix} \tag{5.5}$$

The Euclidean distance between a pattern (patch) and two templates (atoms) can be determined as:

$$\overline{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} (y - d_{i1})^T \cdot (y - d_{i1}) \\ (y - d_{i2})^T \cdot (y - d_{i2}) \end{bmatrix}$$

$$= \begin{bmatrix} y^T y \\ y^T y \end{bmatrix} + \begin{bmatrix} d_{i1}^T \cdot d_{i1} \\ d_{i2}^T \cdot d_{i2} \end{bmatrix} - 2 \begin{bmatrix} d_{i1}^T \cdot y \\ d_{i2}^T \cdot y \end{bmatrix} = C_1 + C_2 - 2\tilde{r} \tag{5.6}$$

Replacing equation (5.6) in (5.5) will result in a relation between $\overline{\alpha}$ and $\overline{r}$:

$$\tilde{\alpha} = \frac{M}{c} \left(\frac{1}{2} \left(C_1 + C_2 \right) - \frac{1}{2} \overline{r} \right) \tag{5.7}$$

And also:

$$\overline{r} = C_1 + C_2 - 2cM^{-1} \tilde{\alpha}$$

$$= \begin{bmatrix} y^T y \\ y^T y \end{bmatrix} + \begin{bmatrix} d_{i1}^T \cdot d_{i1} \\ d_{i2}^T \cdot d_{i2} \end{bmatrix} - 2 \begin{bmatrix} (d_{i1}^T \cdot d_{i1}) & (d_{i1}^T \cdot d_{i2}) \\ (d_{i2}^T \cdot d_{i1}) & (d_{i2}^T \cdot d_{i2}) \end{bmatrix} \tilde{\alpha} \tag{5.8}$$

Equation (5.8) demonstrates the minimum distance as a transform of $\tilde{\alpha}$; this equation also proves that $\tilde{\alpha}$ alone is not a proper indicator to classify different input pattern classes. The minimum distance to the pattern for each class is attained by performing the OMP for the corresponding sub-dictionary as:

$$r_{\min}^{p} = \min\begin{bmatrix} d_{i1}^{T}.d_{i1} \\ d_{i2}^{T}.d_{i2} \end{bmatrix} - 2\begin{bmatrix} (d_{i1}^{T}.d_{i1}) & (d_{i1}^{T}.d_{i2}) \\ (d_{i2}^{T}.d_{i1}) & (d_{i2}^{T}.d_{i2}) \end{bmatrix} \tag{5.9}$$

In which p refers to the class label and $r_{\min}^{p}$ is the minimum distance of the corresponding class. The input pattern class is specified by finding the smallest minimum distance of all classes. This $r_{\min}^{p}$ (minimum distance of each input patch with dictionary atoms) acts as the classifier for the method. The pseudo algorithm of the proposed method is demonstrated in Table 5.1.

Purpose: To classify different classes in a given test image/signal using Sparse Euclidian Classification technique.

Input: Test image/signal and the dictionary D consist of k sub-dictionaries separately learned for k distinct classes.

Main loop:

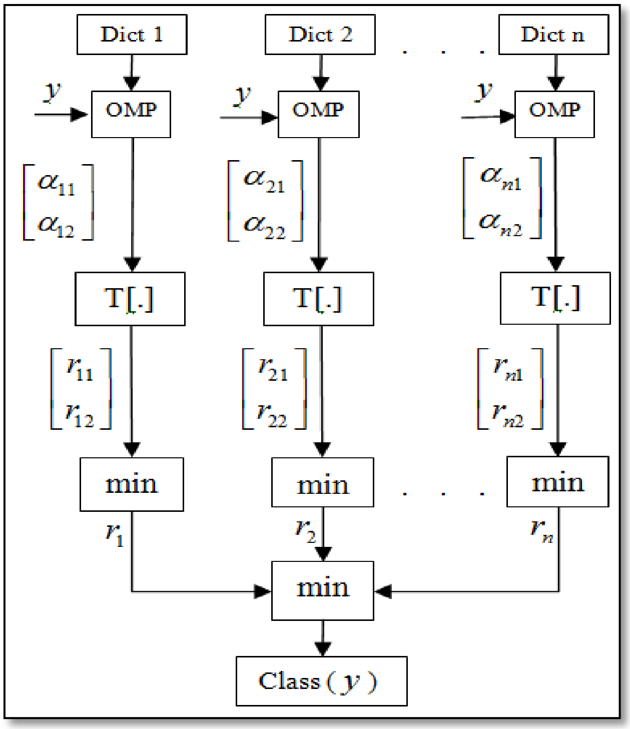
```
• For $k = 1, 2, ..., n$
• $\overline{\alpha_i} = OMP(Dict_i, \overline{y})$
• $\overline{r_i} = 2\overline{\mu} - M.\alpha_i$
• $Class(y_k) = \min(\min(\overline{r_k})), k = 1, 2, ..., n$
```

End.

Output: The labeled image/signal patch according to the label of the class specific dictionary.

Table 5.1 the pseudo algorithm of the proposed SEC method

This method is not limited to two classes only, it can cluster up to $k = 1, 2, ..., n$ classes, the algorithm will simply iterate for as many classes in the image. For a better demonstration of the algorithm the following block diagram is presented:

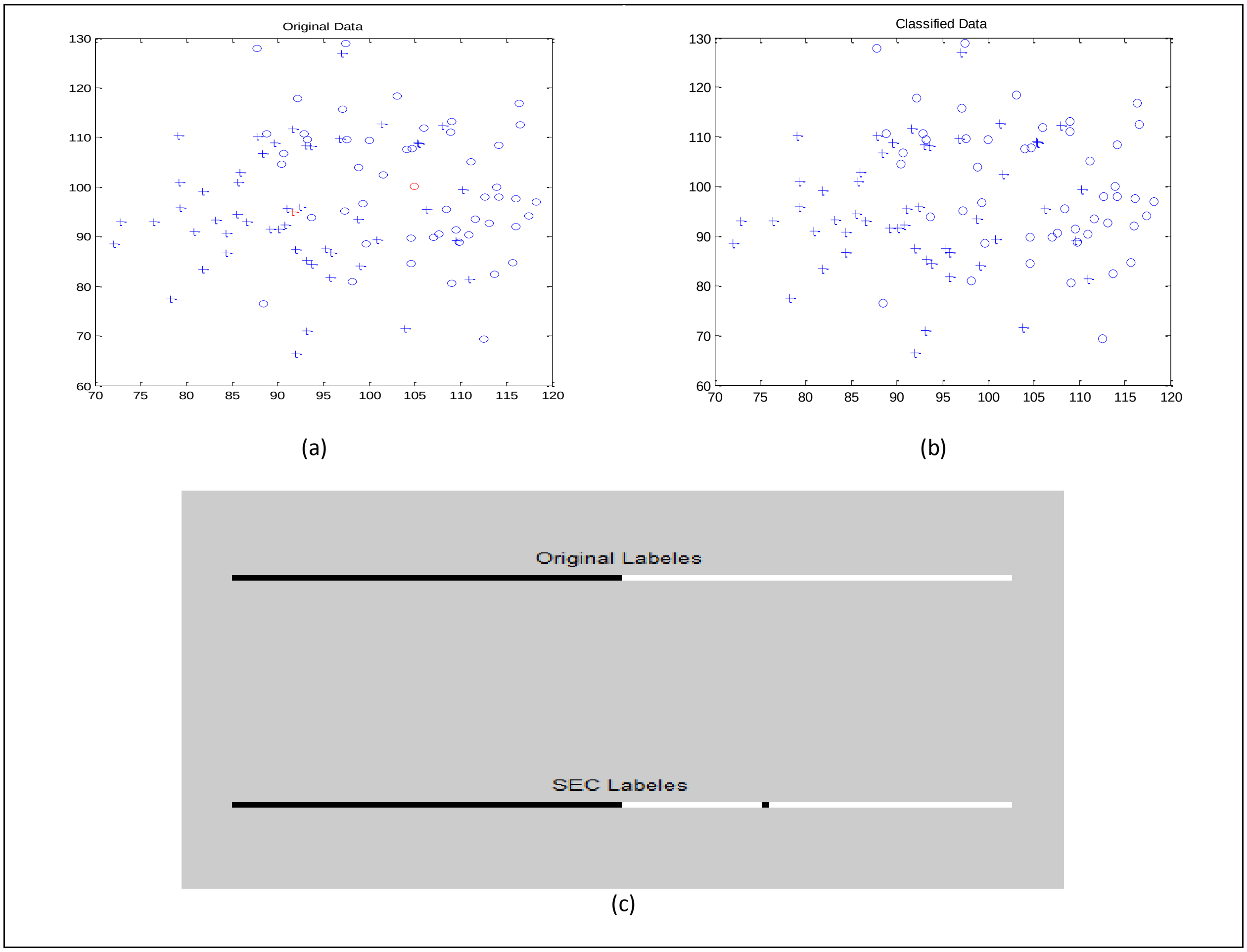


Dict 1
Dict 2
Dict n
y
OMP
$\begin{bmatrix}\alpha_{11}\\ \alpha_{12}\end{bmatrix}$
$\begin{bmatrix}\alpha_{21}\\ \alpha_{22}\end{bmatrix}$
$\begin{bmatrix}\alpha_{n1}\\ \alpha_{n2}\end{bmatrix}$
T[.]
$\begin{bmatrix}r_{11}\\ r_{12}\end{bmatrix}$
$\begin{bmatrix}r_{21}\\ r_{22}\end{bmatrix}$
$\begin{bmatrix}r_{n1}\\ r_{n2}\end{bmatrix}$
min
r_1
r_2
r_n
min
Class (y)

Figure 5-1 a block diagram representation of the proposed method

5.2 performance

The proposed classification technique has been implemented to evaluate its performance. The following figure demonstrates classification results for two randomly projected Gaussian data, one with a mean of 95 and the other with a mean of 100 with a standard deviation of 12. Although the means of both classes are close to each other the result shows only one miss classified sample.



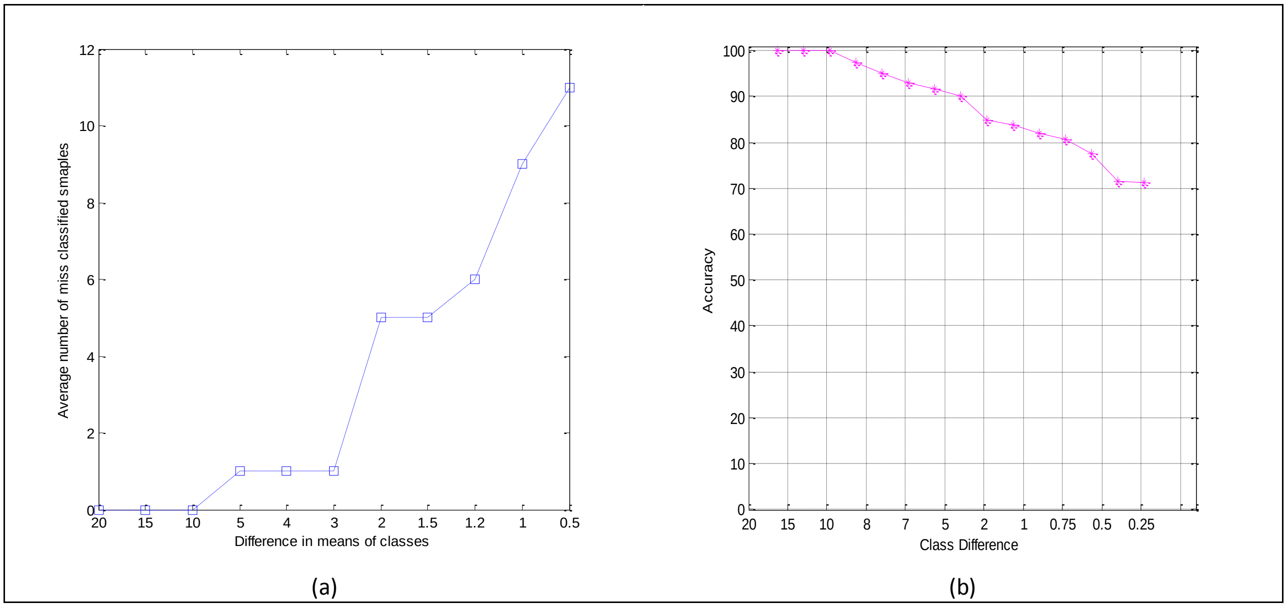
(a)

(b)

(c)

Figure 5-2 (a). Demonstrates the original data with their mean in red. (b) Demonstrates the sparsely represented data. (c) Illustrates the original data labels on top which was assigned by hand, the first fifty samples are from class1 and the rest are from class2, on the bottom are the labels that are resulted from the proposed method. One miss classified sample can be seen.

For randomly generated Gaussian distributions with different means and standard deviations the proposed framework has been tested in different scenarios (different means and variances). For a total number of ten repetitions per each test (since the data is randomly generated) the average error rate for labeling the classified data and accuracy using the proposed framework can be demonstrated as follows:

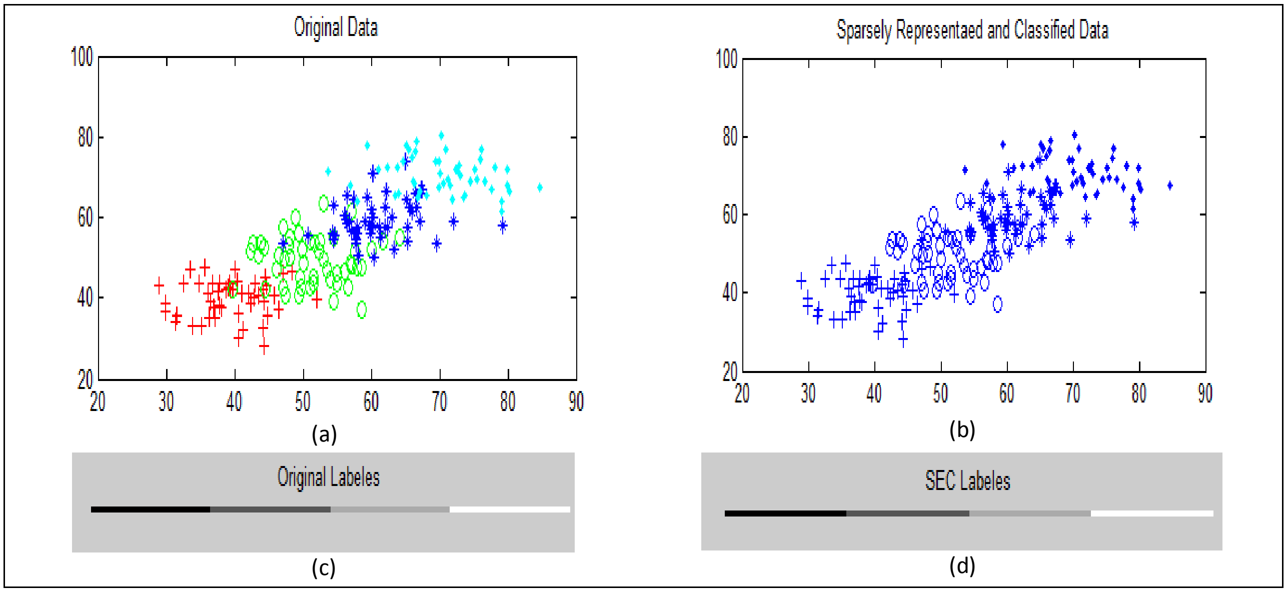


(a)

(b)

Figure 5-3 (a) Number of miss classified samples based on the difference between the means of classes. (b) Average accuracy (percent) versus the difference between classes

It can be seen that even in the case of having classes with a small difference in their means; the classification rate is still acceptable (around 5% when the difference of means is about 2 to1).



(a)

(b)

(c)

(d)

Figure 5-4 (a) Four randomly Gaussian distributed data with means of 40,50,60 and 70 displayed in four different colors, (b) Sparse representation and classification of the original data using SEC method. (c) Labels of the original data, first 50 samples are from class 1 and are displayed with a black label. Similarly dark gray label for class2, light gray label for class 3 and white label for the 50 samples of class 4. (d) Labels that are correctly assigned to the representation of the data using SEC method.

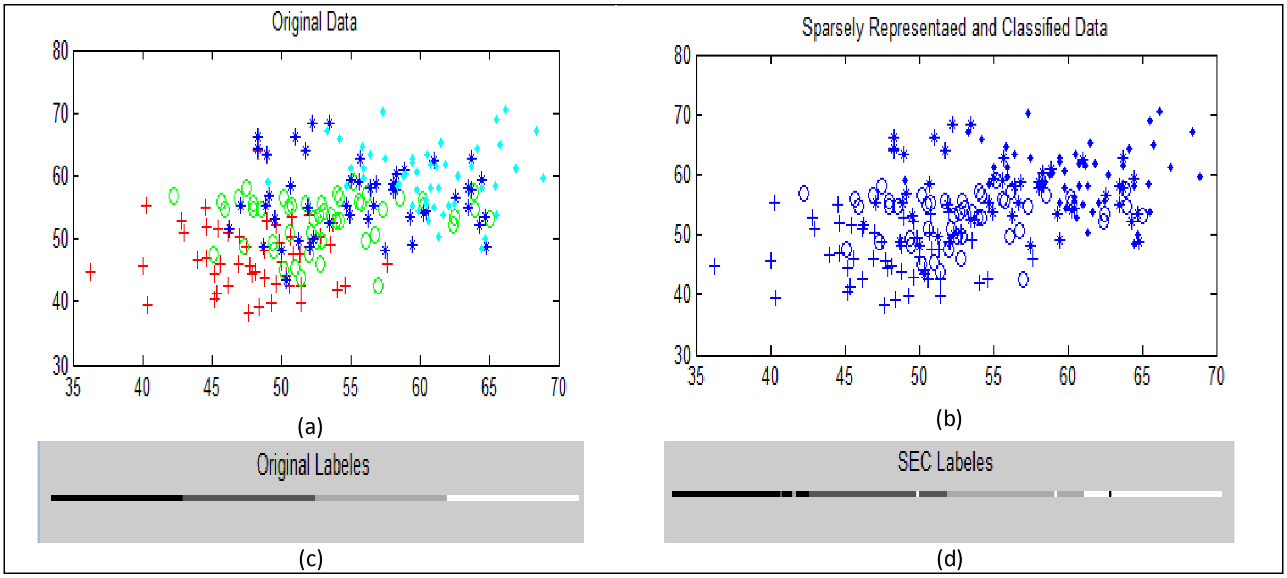


Figure 5-5 Four randomly Gaussian distributed data with means of 48,51,53 and 54 displayed in four different colors, (b) Sparse representation and classification of the original data using SEC method. (c) Labels of the original data. (d) Labels of the represented data assigned by SEC method. 5 misclassified samples among 200 samples can be seen.

In order to make a more comprehended evaluation of the error rate of the method, the method has been compared with the famous Sparse Representation Classification (SRC) method under the same conditions. The comparison result is as follows:

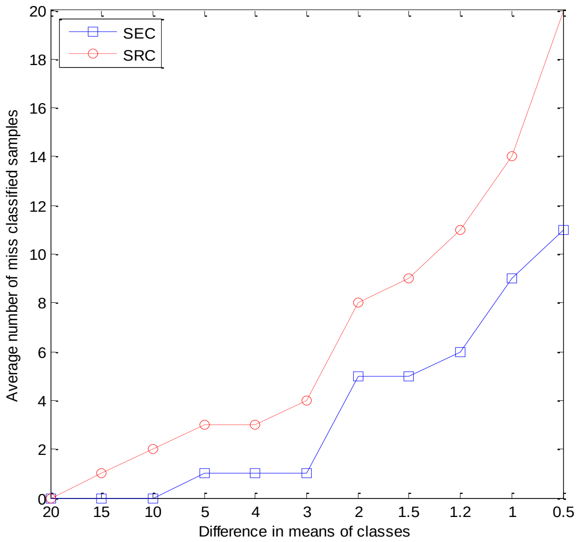


Figure 5-6 classification results comparison between SRC and SEC based on the distance of the means of different classes

The Brodatz dataset is used to compare the accuracy and efficiency of this method with SRC as the best kwon sparse based classification method as well as SVM and K-NN. The class specific sub-dictionaries are learned using K-SVD with 20 iterations, creating a dictionary with the size of $m = 100$ and $n = 400$. Figure 5-7 demonstrates the classification results of the proposed method on the Brodatz textures.

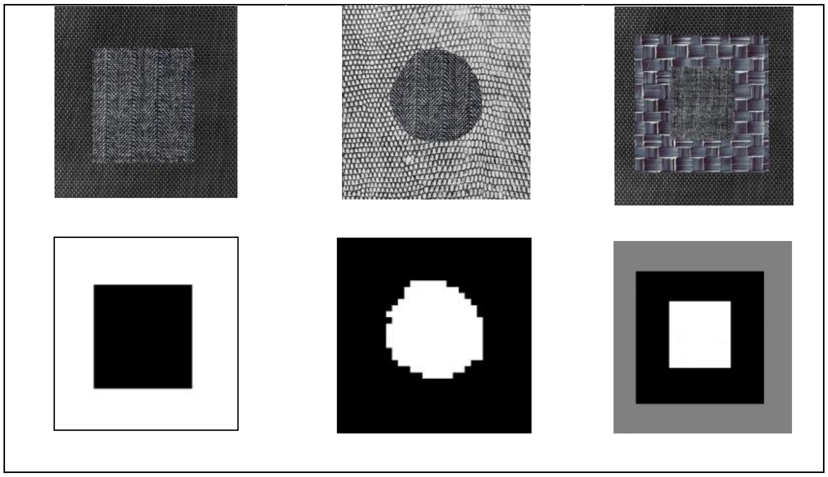


Figure 5-7 classification results of the proposed method on the standard Brodatz database texture images. The top row shows the original image while the bottom row shows the classification results. Class 1 is shown in black, while class 2 is shown in white and Class 3 in gray.

Figure 5-8 demonstrates the evaluated accuracy based on different patch sizes for the proposed method, SRC and classification based on the sparse vector (α). It is important to note that minimum accuracy is obtained in the results when the sparse vector alone is used as an indicator for classification.

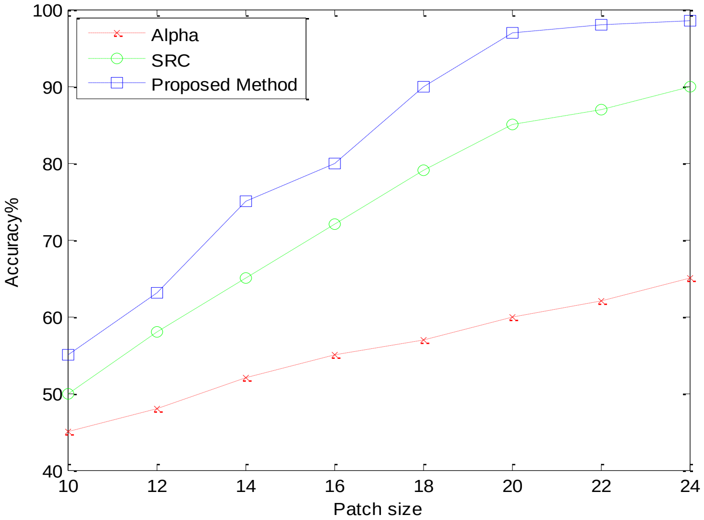


Figure 5-8 Classification rates for three different sparse based methods

In this method we are utilizing learned class specific dictionaries to represent the data over, while in SRC the actual training samples (training images) are being used as the dictionaries to represent the data over and perform the classification. Hence the computational time is reduced in our method compare to SRC.

In this method unlike SRC not only we are not using all the training samples as the space to represent the data over (we utilize class specific dictionaries which have a very smaller size but an acceptable accuracy), but also we do not compute any reconstruction error (or any error vector) for the samples. When the sparse representation of each patch is computed, using the transform in (5.9) we calculate its Euclidian distance from the dictionary atoms in the same iteration (Table 5.1). This will result in a noticeable increase in the accuracy as well as reducing the computational time significantly. Table 5.2 demonstrates a comparison between SRC and the proposed method's computational time and Table 5.3 illustrates the measured accuracy for both methods on the same data.

Patch Size	SRC	Proposed Method
10x10	500	3.1
16x16	290	2.96
20x20	230	2.27
24x24	200	2.1

Table 5.2 Computational time (seconds) for the proposed method and the SRC method

Patch Size	SRC	Proposed Method
10x10	50%	55%
16x16	72%	80%
20x20	85%	97%
24x24	90%	98.5%

Table 5.3 Accuracy measures for proposed method and the SRC method

In order to compare the accuracy of the proposed framework with non-sparse based classification techniques, as well as SRC the error rate for classifying Brodatz dataset textures is compared with K-NN and SVM methods. These results are demonstrated in Table 5.4.

Method	Error Rate (percent)
K-NN	5.2
SVM	1.5
SRC	10
Proposed Method	**1.5**

Table 5.4 Error rate measurements

In this chapter we proposed a sparse representation-base image classification method using a transformation which maps sparse vectors to Euclidian distances. It is a generic method that can be utilized within a variety of image and signal classification as well as other sparse representation applications. The uniqueness of this method lies in the fact that unlike other conventional sparse-base classification methods, we are not using the representation vector or the reconstruction error as an indicator for classification. Conversely we employed a transform domain in which each class sparse features appear to be more distinct. Using the new proposed measure introduces a strong classification approach that eliminates the need for large dictionaries; hence the computational time is significantly reduced compared to other prevailing methods. The proposed framework is suitable to be used as a platform for different image sparse classification algorithms. It can be used in conjunction with other dictionary learning methods such as FDDL or MOD and instead of OMP other sparse representation frameworks can be utilized. A variety of sparse pursuit algorithms such as FOCUSS, Batch OMP, and other methods that utilize the pseudo-inverse operation in order to update the sparse vector can be substituted with OMP. OMP was employed here because it is the leading sparse representation method and is the building block for most other pursuit representation methods. As a result our method can be expanded by most of these methods.

5.3 Organ Boundary Segmentation

After proposing the novel sparse Euclidian classification framework and evaluating its performance on texture images for image classification, in this section by taking advantage of the proposed sparse based classification method, we present algorithms capable of segmenting different body organs in medical images. These algorithms are compatible with both MRI and CT scan imaging modalities. The main advantage of these methods is that they are not limited to one specific organ and a single imaging modality. In fact in the aim is to present a new image classification and segmentation technique to the medical image processing community based on sparse representation. A method that unlike many other conventional techniques is not limited to a specific organ or modality. The main goal in the near future is to be able to learn class specific dictionaries for any organ in any modality and by means of the introduced SEC method, classify and label different organs in the image accurately. There are still lots of room for research and investigation in this field. However our results were quite convincing and have been evaluated and approved by medical experts.

5.3.1 Methodology

The main idea behind this segmentation approach is to use class specific labeled dictionaries and try to sparsely represent the image over the labeled and learned dictionaries; the dictionaries are trained for different classes using either K-SVD or FDDL algorithms introduced in chapter 3. Using the SEC classification framework we label each patch of the test image with respect to its minimum Euclidian distance from the dictionaries. Having labeled every patch in the image, the desired organ is detected according to the labels. With the desired organ detected, its boundaries can be segmented effortlessly utilizing an image processing tool such as edge detection or active contours [55]. Figure 5-9 demonstrates a block diagram of the proposed method.

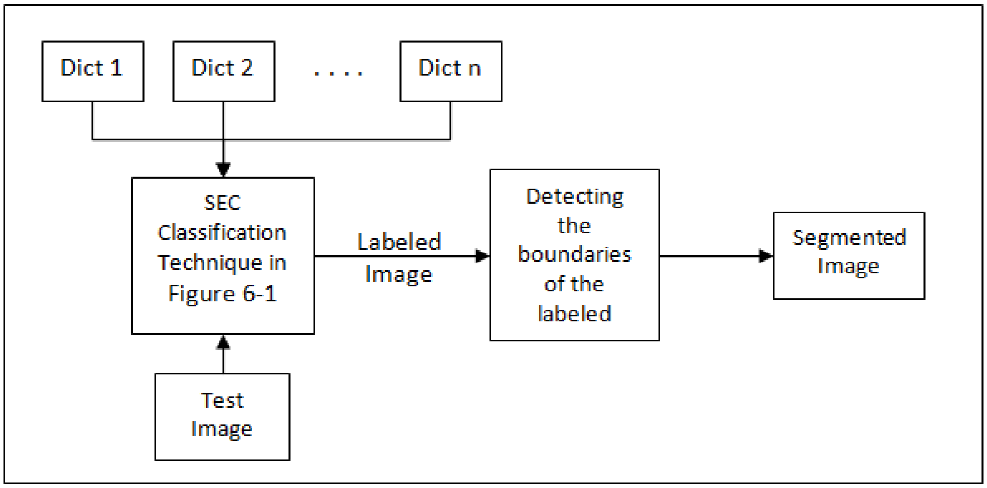

Dict 1
Dict 2
. . . .
Dict n
SEC Classification Technique in Figure 6-1
Labeled Image
Detecting the boundaries of the labeled
Segmented Image
Test Image

Figure 5-9 Block diagram of the proposed organ segmentation method

5.3.2 Segmenting the human heart in MR images

Compared to other imaging modalities (such as ultrasound and magnetic resonance imaging), cardiac magnetic resonance (MR) can provide detailed anatomic information about the heart chambers, large vessels, and coronary arteries [56]. Therefore, MRI is an important imaging modality for diagnosing cardiovascular diseases. Complete segmentation of the heart, is a prerequisite for clinical investigations, providing critical information for quantitative functional analysis for the whole heart [56]. We propose our sparse classification and segmentation technique for segmenting the human heart in MR images (it can be used in CT modality as well). In this method we train two different dictionaries, either through KSVD or FDDL algorithms. We learn one dictionary based on the texture, intensity, and frequency information (when we utilize DCT dictionaries) of the desired organ (heart in this case), and we train another dictionary for the rest of the image. The purpose for training two dictionaries in such a way is that we want to have a good representation of the heart itself, but the representation of rest of the image is of no interest to us. As a result we dedicate one dictionary to the heart and one dictionary for all other classes in the image. Based on experience we usually utilize one DCT and one image patch based dictionary. If both dictionaries are DCT or image patch based, the algorithm will still classify the image but with less accuracy (in some cases only). Both learned and labeled dictionaries along with the test image are fed into the SEC classification framework, the image patches that are representing the heart will be labeled according to the dictionary that is learned for the heart. Having detected the heart patches, the rest of the patches will be eliminated resulting in a binary image of the heart. The boundaries in this image can easily get detected by means of an active contour model automatically. The steps in this procedure are demonstrated in the following figure:

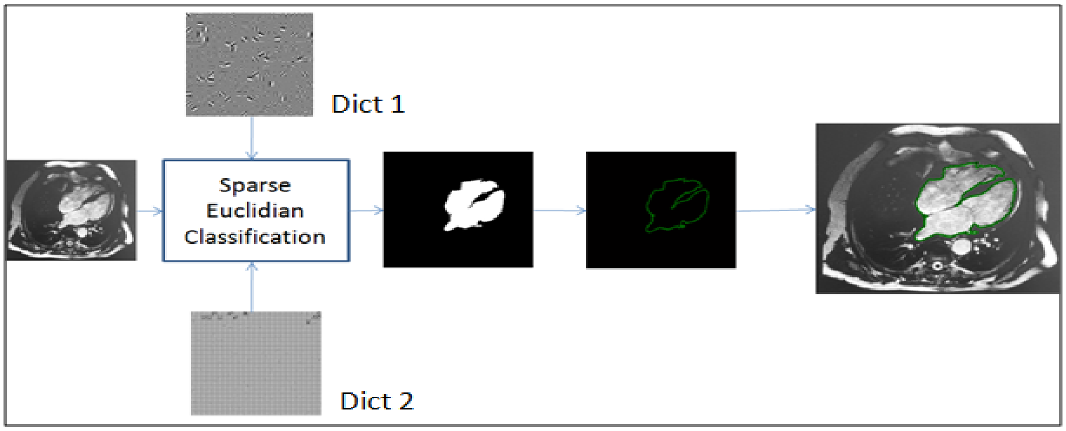


Figure 5-10 An example of segmenting the human heart in a MR image. Dictionary1 is a KSVD learned dictionary with a DCT basis, while dictionary 2 is based on image patches.

Figure 5-11 shows the segmentation results of this method on few of our images.

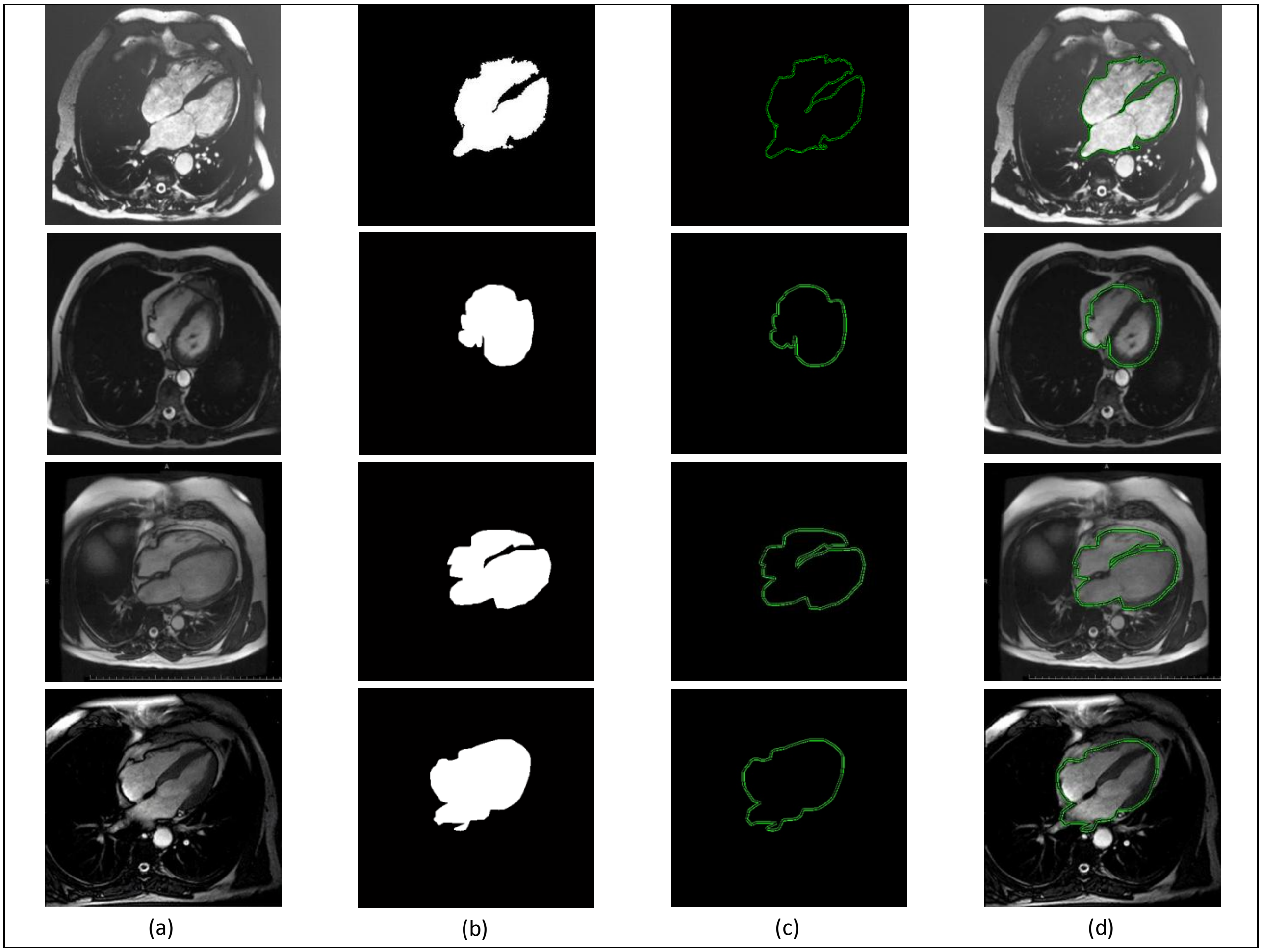


Figure 5-11 (a) Original image, (b) The detected organ, (c) Boundary of the detected organ, (d) final segmented image

We evaluated this method by comparing the results of our work for 30 images with the manual segmentation of the images performed by an expert. For all the Images, Dice coefficient was computed in order to measure the accuracy. An average accuracy of 92.5% for all the images was gained using the proposed method.

5.3.3 Segmentation of the kidney in CT images

Isolating the kidney from its surrounding anatomical structures and organs is a crucial step in many medical diagnosis frameworks that assess the renal functions. Frameworks that are proposed for automatic classification of normal kidneys and acute rejection transplants [57] are among the applications of kidney segmentation. Acute rejection is the immunological response of the human immune system to the foreign transplanted kidney after surgery. It is the most important cause of graft failure after renal transplantation [58]. Unlike invasive monitoring and post renal transplant follow-ups such as biopsy, noninvasive follow-up studies are based on acquiring images from the transplanted kidney. These imaging modalities are usually in CT or sometimes MRI (Dynamic Contrast Enhance-MRI) modalities.

In order to segment the kidney in acquired images for pre or post-surgery studies we propose a similar approach to what we introduced for segmenting the heart in MRI images. In this approach we train one over-complete dictionary based on the patches acquired from the kidneys pixels in CT images (a similar approach can be utilized within MRI); another dictionary is learned based on the rest of the image. Feeding both dictionaries as well as a new test image to the proposed classification framework we can label and detect the patches that are representing the kidney. Since the number of other organs in these images are relatively larger compared to the images acquired from heart MRIs or CT scans, the chance of misclassifying the patches are higher. In order to overcome this problem usually a threshold is utilized to neglect the misclassified patches based on the density of the detected patches in an area. The concentration of the patches is relatively higher in the kidneys areas. The following figure shows the segmentation of the kidneys in acquired CT images.

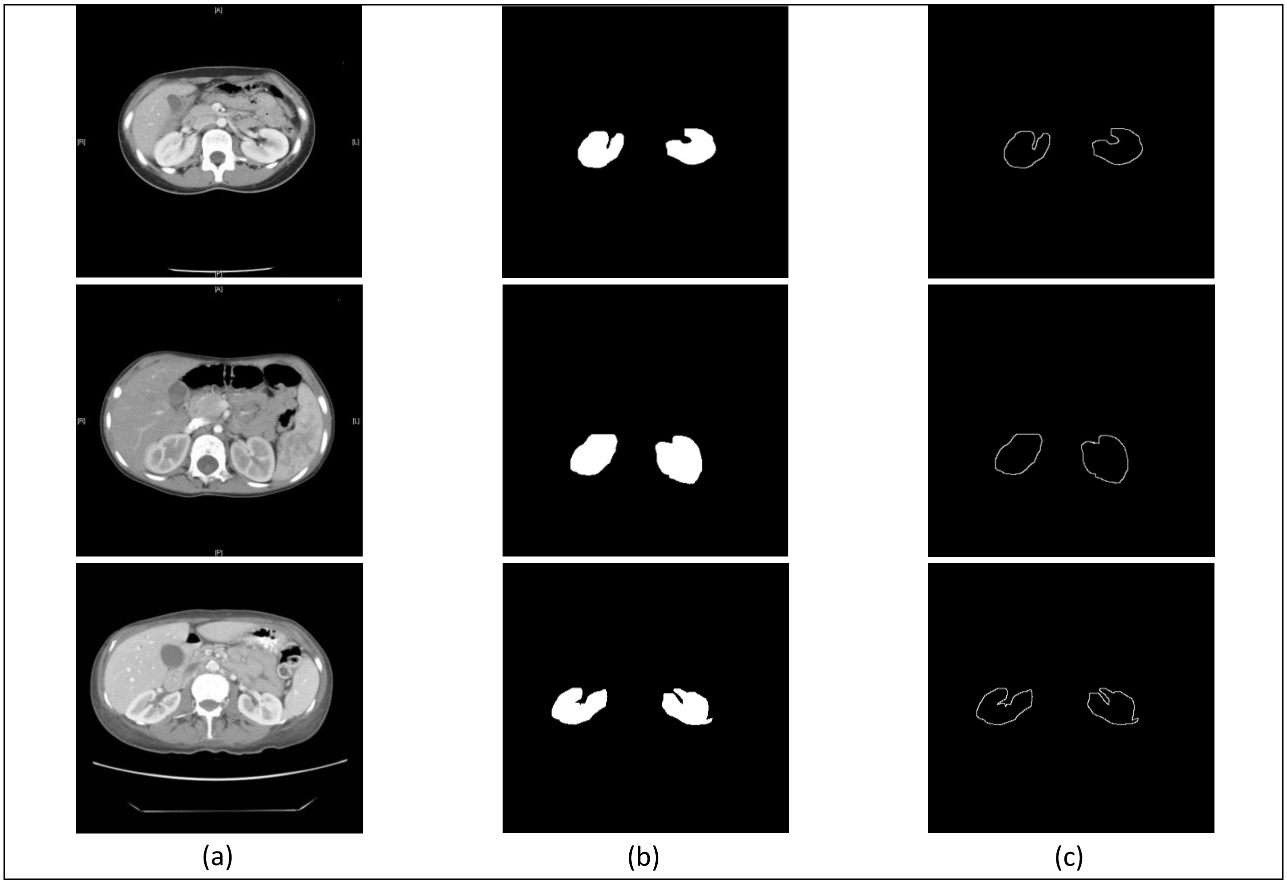


Figure 5-12 (a) Original CT image. (b) Segmented kidneys. (c) Boundaries of the segmented kidneys

Since neither the images nor the algorithm of other kidney segmentation frameworks were available, in order to evaluate the accuracy of the method, all segmentation results were compared with manual segmentation of the images. The acquired dice coefficient calculation for 12 CT images showed an average accuracy of 87%.

5.3.4 Segmentation of ventricular system in MR Images of the Brain

Segmenting the ventricular system of the brain in medical images plays an important role in medical diagnosis. The volume of lateral ventricle increases with age and it is an important indicator of Alzheimer’s, schizophrenia, and depressive disorders. For medical diagnosis both CT and MRI imaging modalities are used in brain studies. Within the MRI itself there are T_1, T_2, Proton Density and Diffusion model imaging modalities. In most diagnostics, especially in the ventricular system studies the T_1 and T_2 modalities are of more concern. T_1 and T_2 are two basic yet famous imaging modalities in MRI. T_1-weighted scans refer to a set of standard scans that represent differences in the spin-lattice (or T_1) relaxation time of various tissues within the body. T_1-weighted sequences have the ability to be acquired rapidly because of the fact that they use short inter-pulse repetition times (T_R). These sequences are often collected before and after infusion of T_1-shortening MRI contrast agents. In the brain T_1-weighted scans provide appreciable contrast between gray and white matter. In the body, T_1 weighted scans work well for differentiating fat from water, with water appearing darker and fat brighter. T_2 -weighted scans are another basic MRI modality. Like the T_1 scan, fat is differentiated from water, but in this case fat shows darker, and water lighter.

In order to segment the ventricular system using the SEC framework we follow the same path that we did before, which is learning class specific dictionaries. The difference is that in MRI images we implement the method on both modalities simultaneously. In each modality we train a dictionary for the lateral ventricle and the cerebral cortex classes in the image (four dictionaries in total). For each modality the image and the dictionaries are fed to the SEC frame work and the intersect of the detected area from each modality would be the final segmentation of the ventricular system. The block diagram of this method is presented below:

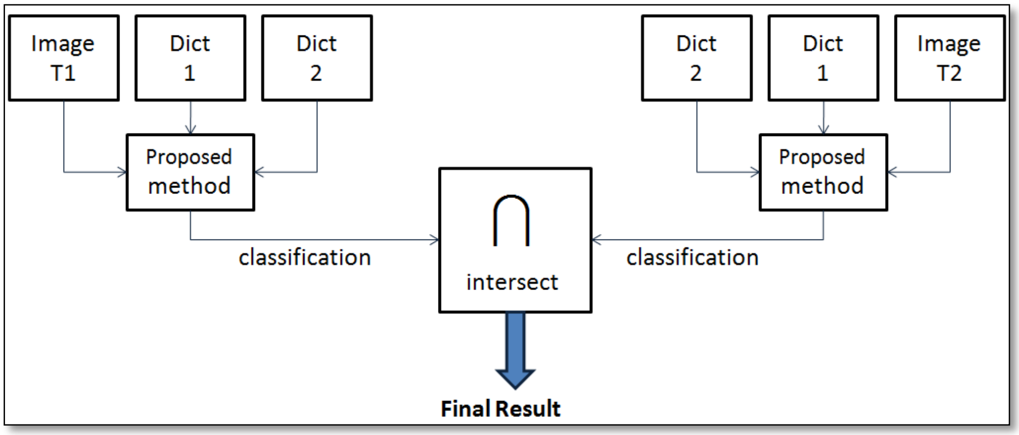


Figure 5-13 the block diagram of the ventricular system segmentation technique

In order to evaluate the performance of the method, a comparison between the results of this method and manual segmentation of the ventricular system in the images has been carried out. Figure 5-14 shows an MRI image in both T_1 and T_2 modalities with their ventricular system segmentation results. And Figure 5-15 demonstrates the manual segmentation process of the brain ventricles in another MRI image.

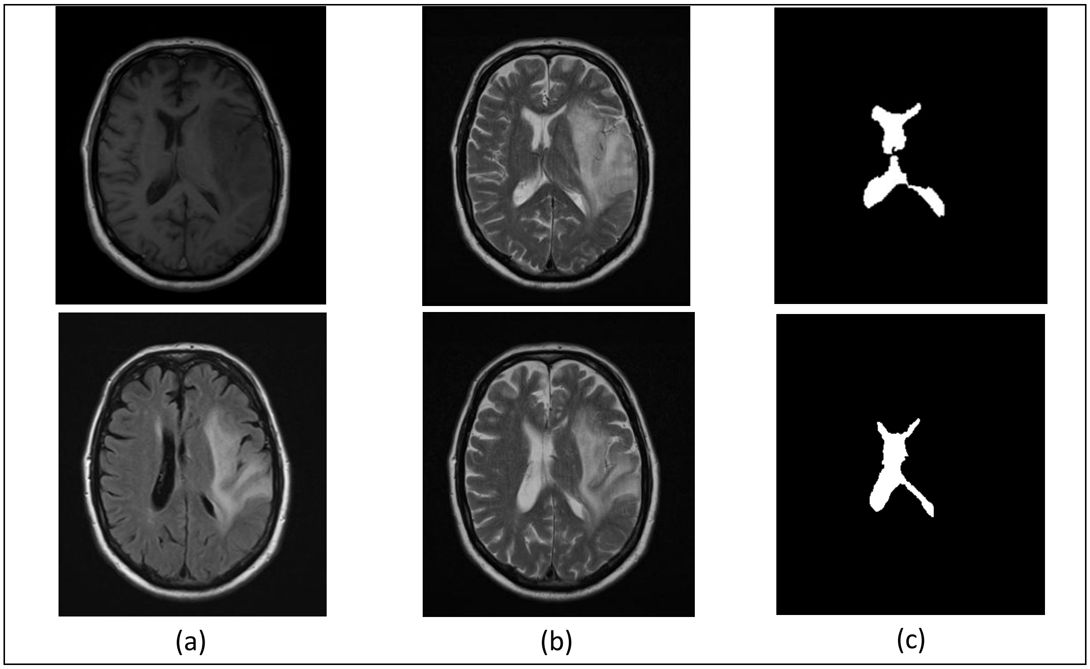


Figure 5-14 ventricular system segmentation in MRI images. (a)T_2 Image. (b) T_1 Image. (c) Final segmented image.

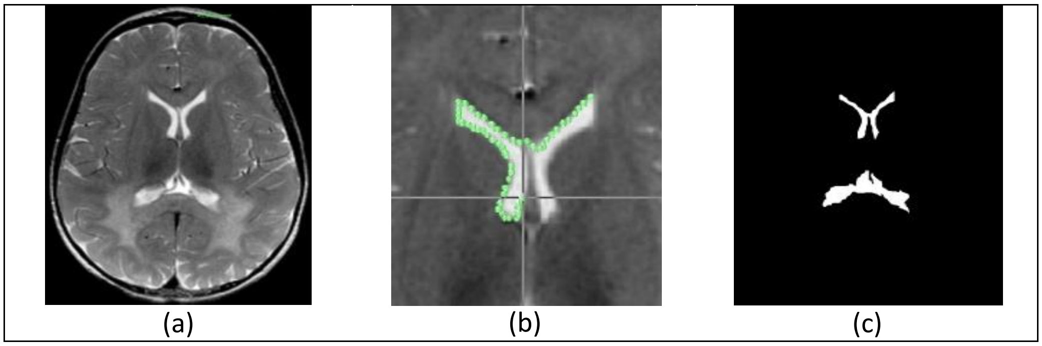


Figure 5-15 Manual segmentation of the lateral ventricle. (a) Original Image. (b) manual selection of boundary points for segmentation. (c) Manually segmented image.

For all of the Images, Dice coefficient was computed in order to measure the accuracy of the method. Table 5.5 demonstrates the Dice coefficients comparisons gained from 10 of the images. An average accuracy of 94.3% for all the images was gained using the proposed method.

Image	1	2	3	4	5	6	7	8	9	10
Dice Coeff	87.5	93.75	91.2	89.3	92.5	86	87.3	95	90	88

Table 5.5 Dice coefficient calculated for ten MRI images

Chapter Six: Conclusion and Discussion

Sparse representations of signals and learning a dictionary to represent the signal over, has gained considerable amount of attention by many scholars focusing on different applications around the world. In this thesis we took a step further exploring the effects of sparse representation and dictionary learning in medical image processing. Sparse representation techniques are considered to be a powerful method in image processing, especially in applications such as denoising and compression. However these techniques have not yet vastly been explored for other application purposes such as classification and segmentation. We presented a novel approach in classification images using sparse based techniques that is capable of clustering different classes in an image with an acceptable accuracy. We then utilized this classification technique and extract a novel fully automated method to segment medical images. To our knowledge we are the first to utilize sparse classification techniques as well as dictionary learning for image segmentation purposes.

6.1 Sparse Euclidian Classification

In this thesis we proposed a sparse representation-base image classification method that we call Sparse Euclidian Classification (SEC). We utilize a transformation which maps sparse vectors to Euclidian distances. Unlike the few conventional sparse based classification techniques we are not using the representation vector or the reconstruction error as an indicator for classification. Conversely we introduced a transform that maps the sparse representation vectors for each patch to Euclidian distances. By using these Euclidian distances each class sparse features appear to be more distinct therefore resulting in strong classification results. Since these Euclidian classification features have strong clustering characteristics the proposed method is independent of very large datasets other methods need for the purpose of classification, hence making the method considerably faster. Comparisons between the results of SEC and the famous SRC method demonstrated higher accuracy and noticeably less computational time for our method. We developed the Sparse Euclidian Classification algorithm based on Orthogonal Matching Pursuit method for sparse representation; however it is possible to utilize other sparse representation techniques within the SEC framework and evaluate its performance.

6.2 Organ Segmentation Framework

In this thesis we proposed a novel fully automated segmentation algorithm to segment different organs in different images using our Sparse Euclidian Classification (SEC) framework. After gaining acceptable classification results utilizing SEC and implementing it on different texture images for evaluation, we utilized this classification framework as the main element in our segmentation technique. In order to segment an organ in an image first a class-specific dictionary is learned through K-SVD algorithm. Having gained the specific dictionary, a test image is sparsely represented and classified over it within the SEC framework. Utilizing SEC, the patches that belong to the specific organ are labeled and classified. Since the information on the edges of the detected organs is of great importance to us, we then detect the edges of the classified organ/patches using active contours or edge detection methods. As an advantage, this approach requires minimum human interaction and is fully automated. The other significant feature that this method has is that since it is based on learning a class-specific dictionary, we can utilize this method for different organs in different modalities. As a matter of fact this method can be used as a platform for other segmentation applications (both medical and non-medical). So far we were able to use this method in MRI and CT images for different organs; the main goal is to be able to introduce a novel method that by learning its specific dictionaries is capable of segmenting any organ within any modality.

6.3 Future Work

Future works of this thesis can be categorized in two main categories:

1. Sparse classification
2. Image segmentation

1. In sparse classification, the goal is to continue the idea of using distances rather than the sparse vector or the reconstruction error as indicators for classification. Use of other distances such as Mahalanobis distance is currently being investigated. Using distances other than the Euclidian, has the

potential to introduce stronger classification results. The mathematical approach in the original SEC might need some minor changes in order to utilize other distances.

2. For the purpose of segmentation we are investigating the use of the proposed method for other organs and other image modalities. Our studies illustrated that promising results in CT and MRI modalities are expected, however in other modalities such as X-ray and ultrasound imaging the method might need more discriminative properties. In order to add more discriminative characteristics to the method the use of a new stronger discriminative dictionary learning algorithms might be highly beneficial.

The ultimate goal is to introduce a method which is capable of segmenting any area with specific characteristics (texture, intensity, frequency info, etc…) in any imaging modality. This can be achieved by sparsely representing images over class-specific learned dictionaries followed by labeling the desired organ patches accordingly.

References:

[1] H. Badakhshannoory and P. Saeedi, "A Model-Based Validation Scheme for Organ Segmentation in CT Scans Volumes," *IEEE Transactions on Biomedical Engineering,* vol. 58, no. 9, pp. 2681-2693, 2011.

[2] J. Ehrhardt, A. Schmidt-Richberg and H. Handels, "A Variational Approach for Combined Segmentation and Estimation of Respiratory Motion in Temporal Image Sequences," in *IEEE 11'th International Conference on Computer vision (ICCV)*, 2007.

[3] O. Gloger, K. D. Tonnies, V. Liebscher, Brend. Kugelmann, R. Laqua and H. Volzke, "Prior Shape Level Set Segmentation on Multistep Generated Probability Maps of MR Datasets for Fully Automatic Kidney Parenchyma Volumetry," *IEEE Transactions on Medical Imaging,* vol. 31, no. 2, pp. 312-325, 2012.

[4] F. Fischer, M. A. Selver, W. Hillen and G. Guzelis, "Integrating Segmentation Methods From Different Tools Into a Visualization Program Using an Object-Based Plug-In Interface," *IEEE Transactions on Information Technology in Biomedicine,* vol. 14, no. 4, pp. 923-934, 2010.

[5] Donoho, D.L., "Compressed sensing," *IEEE Transactions on Information Theory,* vol. 52, pp. 1289-1306, 2006.

[6] Elad, M., "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Transactions on Image Processing,* vol. 15, no. 12, pp. 3736-3745, 2006.

[7] J. Mairal, M. Elad and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing,* vol. 17, pp. 53-69, 2008.

[8] S. Armato, "CAD dissects growing volume of data from lung CT exams," *Diagnostic Imaging,* pp. 6-12, 2003.

[9] R.C. Gonzalez and R.E. Woods, Digital Image Processing, New Jeresy: Prentice Hall Publishing company, 2002.

[10] K.J. Batenburg and J. Sijbers, "Adaptive thresholding of tomograms by projection distance minimization," *Elsevier Journal on Pattern Recognition,* vol. 42, no. 10, pp. 2297-2305, 2009.

[11] G.N. Srinivasan and G. Shobha, "Segmentation Techniques for Target Recognition," *International Journal of Computers and Communications,* vol. 1, no. 3, pp. 313-333, 2007.

[12] M. Elad, Sparse and Redundant Representation, Springer New York Dordrecht Heidelberg London, 2010.

[13] J. K. Pillai, V. M. Patel and R. Chellappa, "Sparsity inspired selection and recognition of iris images," in *IEEE International Conference on Biometrics*, 2009.

[14] G. Davis, S. Mallat and M. Avellaneda, "Adaptive greedy approximations," *Journal of Constructive Approximation,* pp. 57-98, 1997.

[15] Zhang, S. Mallat and Z., "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal processing,* pp. 3397-3415, 1993.

[16] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-

linear system identification," *International Journal of Control,* vol. 50, no. 5, pp. 1873-1896, 1989.

[17] Y.C. Pati, R. Rezaiifar and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Conference Record of The Twenty Seventh Asilomar Conference on Signals, Systems and Computers,* 1993.

[18] Tropp, J.A., "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory,* pp. 2231-2242, 2004.

[19] S.S. Chen, D.L. Donoho and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review,* pp. 129-159, 2001.

[20] Rao, I.F. Gorodnitsky and B.D., "sparse signal reconstruction from limited data using, FOCUSS: A re-weighted norm minimization algorithm," *IEEE Transactions on Signal Processing,* vol. 45, no. 3, pp. 600-616, 1997.

[21] M. Aharon, M. Elad and A. Bruckstein, "KSVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing,* vol. 54, no. 11, pp. 4311-4322, 2006.

[22] M. Aharon, "Overcomlepet Dictionaries for sparse representation of Signals," *PhD Thesis,* 2006.

[23] R. Rubinstein, M. Zibulesky, M. Elad, "Efficient Implementation of the KSVD algorithm Using Batch Orthogonal Matching Pursuit," Technical Report- CS Technion, 2008.

[24] H. Wang, J. Vieira, P. Ferreira, B. jesus and I. Duarte, "Batch Algorithms of matching pursuit and orthogonal matching pursuit with applications to compressed sensing," in *International Conference*

on Information and Automation (ICIA), Aveiro Prtugal, 2009.

[25] D.L. Donoho, Y. Tsaig, I. Drori; Starck, J.L., "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory,* vol. 58, no. 2, pp. 1094-1121, February 2012.

[26] A. Miller, Subset selection in regression, 2nd Edition ed., London: Chapman and Hall, 2002.

[27] Stuetzle, J.H. Friedman and W., "Projection Pursuit Regression," *Journal of the American Statistical Association,* vol. 76, pp. 817-823, 1981.

[28] Zhang, S. Mallat and Z., "Adaptive time-frequency decompositions," *Optical Engineering,* vol. 33, no. 7, pp. 2183-2191, 1994.

[29] Donoho, S. Chen and D., "Basis Pursuit," in *Twenty-Eighth Asilomar Conference on Signal, Systems and Computers*, 1994.

[30] J. Tropp, "Topics in Sparse Approximation," PhD Thesis, University of Texas at Austin , 2004.

[31] A. Florian, A. Potra and S.J Wright, "Interior-point methods," *ELSEVIER,* vol. 124, no. 1, pp. 281-302, 2000.

[32] U. Y. Desai, "DCT and wavelet based representations of arbitrarily shaped image segments," in *IEEE International Coneference on Image Processing*, 1995.

[33] A. B. Watson, "Image Compression Using the Discrete Cosine Transform," *Mathematica Journal,* vol. 4, no. 1, pp. 81-88, 1994.

[34] A. Averbuch, D. Lazar and M. Israeli, "Image compression using wavelet transform and multiresolution decomposition," *IEEE Transactions on Image Processing,* vol. 5, no. 1, pp. 4-15, 1996.

[35] Field, B.A. Olshausen and D.DJ., "Natural image statistics and efficient coding," *Network-Computation in Neural Systems,* vol. 7, no. 2, pp. 333-339, 1996.

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

[37] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online Dictionary Learning For Sparse Coding," in *IEEE 26th International conference on Machine Learning*, Montreal, Canada, 2009.

[38] H. Lee, A. Battle, R. Raina and A.Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2006.

[39] I. Ramirez, P. Sprechmann and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Computer Vision and Pattern Recognition*, 2010.

[40] J. MAiral, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Computer Vision and Pattern Recognition*, 2008.

[41] Li, Q. Zhang and B., "Discriminative K-SVD for dictionary learning in face recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[42] M. Yang, L. Zhang, X. Feng and D. Zhang, "Fisher Discrimination Dictionary Learning for Sparse Representation," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[43] J. Wright, Y. Ma, J. Mairal, G. Sapiro and S. Yan, "Sparse representation for computer vision and pattern recognition," *IEEE Proceedings,* vol. 98, no. 6, pp. 1031-1044, 2010.

[44] E. Candes, J. Romberg and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory,* vol. 52, no. 2, pp. 489-509, 2006.

[45] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Patteren analysis and Machine intelligence ,* vol. 31, no. 2, pp. 210-227, 2009.

[46] J. Yang, J. Wright, T. Huang and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *IEEE conference on Computer Vision and Pattern Recognition*, 2008.

[47] P. K. Baheti and M. A. Neifield, "Feature-specific structured imaging," *Applied Optics,* vol. 45, no. 28, pp. 7382-7391, 2006.

[48] P. K. Baheti and M. A. Neifeld, "Random projections based feature-specific structured imaging," *Optics Express,* vol. 16, no. 3, pp. 1764-1776, 2008.

[49] J. Romberg, "Imaging via Compressive Sampling," *IEEE Signal Processing Magazine,* vol. 25, no. 2, pp. 14-20, 2008.

[50] J.-L Starck, M. Elad and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Transactions on Image Processing,* vol. 14, no. 10, pp. 1570-1582, 2005.

[51] I. Daubechies, M. Defrise and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Pure and Applied Mathematics ,* vol. 57, no. 11, pp. 1413-1457, 2004.

[52] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley & Sons, 2001.

[53] T. Guha and R. Ward, "A SPARSE RECONSTRUCTION BASED ALGORITHM FOR IMAGE AND VIDEO CLASSIFICATION," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[54] R. Sivalingam, G. Somasundaram, V. Morellas, N. Papanikolopouls, O. Lotfallah and Y. Park, "Dictionary learning based object detection and counting in traffic scenes," in *IEEE International Conference on Distributed Smart Cameras*, 2010.

[55] T.F. Chan and L.A. Vese, "Active Contours Without Edges," *IEEE Transactions on Image Processing,* vol. 10, pp. 266-277, 2001.

[56] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering and D. Comaniciu, "Four-Chamber Heart Modeling and Automatic Segmentation for 3-D Cardiac CT Volumes Using Marginal Space Learning and Steerable Features," *IEEE Transactions on Medical Imaging,* vol. 27, no. 11, pp. 1668-1681, 2008.

[57] A. M. Ali, A. A. Farang and A. S. El-Baz, "Graph Cuts Framework for Kidney Segmentation with Prior Shape Constraints," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007.

[58] S. E. Yuksel, A.El-Baz, M. El-ghadr, T. Eldiasty and M. A. Ghoneim, "A Kidney Segmentation Framework for Dynamic Contrast Enhanced Magnetic Resonance Imaging," *Journal of Vibration and Control,* vol. 13, no. 9-10, pp. 1505-1516, 2007.

List of Accepted Publications:

- A. Julazadeh, J. Alirezaie and Paul Babyn, "A novel automated approach for segmenting lateral ventricle in MR images of the brain using sparse representation classification and dictionary learning ", IEEE 11th international conference on information science, signal processing and their applications (ISSPA), Montreal, Quebec, Canada. 2012
- A. Julazadeh, M. Marsousi and Javad Alirezaie, "classification based on sparse representation and Euclidian distance", IEEE 26th international conference on visual communication and image processing (VCIP), San Diego, California, USA. 2012