

DICTIONARY LEARNING FOR EFFICIENT CLASSIFICATION WITH  
1-SPARSE REPRESENTATIONS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EGE ENGIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2018



Approval of the thesis:

**DICTIONARY LEARNING FOR EFFICIENT CLASSIFICATION WITH  
1-SPARSE REPRESENTATIONS**

submitted by **Ege Engin** in partial fulfillment of the requirements for the degree  
of **Master of Science in Electrical and Electronics Engineering Department,**  
**Middle East Technical University** by,

Prof. Dr. Halil Kalipçılar \_\_\_\_\_  
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu \_\_\_\_\_  
Head of Department, **Electrical and Electronics Engineering**

Assist. Prof. Dr. Elif Vural \_\_\_\_\_  
Supervisor, **Electrical and Electronics Engineering**  
**Department, METU**

**Examining Committee Members:**

Assist. Prof. Dr. Emre Özkan \_\_\_\_\_  
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Elif Vural \_\_\_\_\_  
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Sevinç Figen Öktem \_\_\_\_\_  
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Mustafa Mert Ankaralı \_\_\_\_\_  
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Zafer Arıcan \_\_\_\_\_  
Electrical and Electronics Engineering Department,  
Konya Food and Agriculture University

**Date:** \_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: Ege Engin

Signature :

## **ABSTRACT**

### **DICTIONARY LEARNING FOR EFFICIENT CLASSIFICATION WITH 1-SPARSE REPRESENTATIONS**

Engin, Ege

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assist. Prof. Dr. Elif Vural

May 2018, 79 pages

Sparse representations have the goal of expressing a given signal as a linear combination of a small number of signals that capture well its characteristics. Dictionary models allowing sparse representations have proven to be quite useful for the treatment and analysis of data in recent years. In particular, the learning of dictionaries in a manner adapted to the characteristics of each data class in a supervised learning problem and representing the data with the learned dictionaries significantly improve the accuracy of classifiers. However, large dictionary sizes and the complexity of the computation of sparse representations may limit the applicability of these methods especially over platforms with limited storage and computational resources. In this thesis, we study the problem of supervised dictionary learning for fast and efficient classification of test samples. In order to achieve low computational complexity and efficient usage of memory, our method learns analytically represented supervised dictionaries that allow an accurate classification of test samples based on 1-sparse representations. We adopt a representation of dictionary atoms in a two-dimensional analytical basis, where the atoms are learned with respect to an objective involving their distance to the samples

from the same class and different classes, as well as an incoherence term encouraging the variability between dictionary atoms. The performance of the proposed method is evaluated with experiments on different image datasets. The comparison of the method to reference supervised and unsupervised dictionary learning methods suggests that it provides satisfactory classification performance under 1-sparse signal representations.

**Keywords:** Supervised Dictionary Learning, Classification, Sparse Coding, Incoherence, Analytical Dictionaries, 1-Sparse Representations

## ÖZ

### **TEK KATSAYILI SEYREK GÖSTERİMLERLE HIZLI SINIFLANDIRMA İÇİN SÖZLÜK ÖĞRENME**

Engin, Ege

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Dr. Öğr. Üyesi Elif Vural

Mayıs 2018 , 79 sayfa

Seyrek gösterimler, bir sinyalin, kendi özelliklerini taşıyan az sayıda sinyalin lineer kombinasyonu olarak gösterilmesini hedefler. Sözlük modellemesi sayesinde seyrek gösterimlerin verilerin analizinde ve işlenmesinde faydalı olduğu geçtiğimiz yıllarda gözlemlenmiştir. Sınıflandırma problemlerinde denetimli sözlüklerin her sınıftaki verilerin yapısına uygun bir şekilde öğrenilmesi ve daha sonra verilerin bu denetimli sözlükte gösterilmesi sınıflandırıcıların performansını önemli ölçüde arttırmaktadır. Fakat, yüksek sözlük boyutları ve seyrek gösterim hesaplama işleminin karmaşıklığı özellikle sınırlı hafıza ve hesaplama kaynaklarına sahip platformlarda bahsedilen yöntemlerin kullanımını zorlaştırmaktadır. Bu tezde test numunelerinde hızlı ve verimli sınıflandırma hedefleyen bir denetimli sözlük öğrenme yöntemi önerilmiştir. Yöntemimiz düşük hesaplama karmaşıklığı ve hafıza kullanımı elde edebilmek amacıyla seyrekliği 1 olan gösterimlerle test numunelerinin doğru sınıflandırılmasına imkan sağlayan analitik gösterimli denetimli sözlükler öğrenmektedir. Çalışmamızda hem sözlük atomlarının kendi sınıfındaki ve diğer sınıflardaki numunelerle arasındaki uzaklıklarını, hem de atomlar arasında çeşitlilik sağlayan bir uyumsuzluk terimini içeren bir optimizasyon problemi vasıtasıyla analitik

fonksiyonlar türünden ifade edilen atomlar öğrenilmiştir. Önerilen yöntemin performansı farklı veri kümeleri ile ölçüldü. Önerilen yöntemin referans denetimli ve denetimsiz sözlük öğrenmesi yöntemleriyle karşılaştırılması yönetimimizin başarılı sınıflandırma sonuçlarına ulaştığını göstermiştir.

Anahtar Kelimeler: Denetimli Sözlük Öğrenmesi, Sınıflandırma, Seyrek Gösterim, Uyumsuzluk, Analitik Sözlükler, Tek Katsayılı Gösterimler



*This thesis is dedicated to my lovely wife, Melis Engin*

## ACKNOWLEDGMENTS

*Above all, I would like to thank my supervisor Assist. Prof. Elif Vural for her memorable support at each step of this study. Studying with her is a once in a lifetime chance due to her personal characteristic of being patient, tolerant and warm-hearted.*

*I want to express my sincere gratitude to my family for their unfailing support, love and thrust throughout my life.*

*I would like to thank my company ASELSAN Inc. for the possibilities that have provided to me.*

*Finally, I owe a special thank to my wife for being the best of me in all aspects of my life. I will love you always and forever.*

## TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xi
LIST OF TABLES . . . . .	xiv
LIST OF FIGURES . . . . .	xv
LIST OF ABBREVIATIONS . . . . .	xviii
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Thesis Outline . . . . .	4
2 RELATED WORK . . . . .	5
2.1 Sparse Signal Representations . . . . .	5
2.1.1 Matching Pursuit (MP) . . . . .	8
2.1.2 Orthogonal Matching Pursuit (OMP) . . . . .	9
2.1.3 Sparse Signal Representations by $l_1$ Minimization . . . . .	10
2.1.3.1 Basis Pursuit (BP) . . . . .	10
2.1.3.2 Least Absolute Shrinkage and Selection Operator (LASSO) . . . . .	11

2.2	Dictionary Learning . . . . .	11
2.2.1	Unsupervised Dictionary Learning . . . . .	12
2.2.1.1	Method of Optimal Directions (MOD)	13
2.2.1.2	K-means . . . . .	14
2.2.1.3	K-SVD . . . . .	16
2.2.2	Supervised Dictionary Learning . . . . .	17
2.2.2.1	Discriminative K-SVD (D-KSVD) . . . . .	18
2.2.2.2	Label Consistent K-SVD (LC-KSVD)	19
3	SUPERVISED DICTIONARY LEARNING WITH 1-SPARSE REPRESENTATIONS	21
3.1	Transformation Invariant Dictionary Learning (TIDL) . . . . .	22
3.2	Our Method . . . . .	27
4	EXPERIMENTAL RESULTS . . . . .	35
4.1	Data Sets and Experimentation Settings . . . . .	36
4.1.1	MNIST Data Set . . . . .	36
4.1.2	Yale Face Data Set . . . . .	37
4.2	Variation of the Performance with the Algorithm Parameters . . . . .	38
4.2.1	Selection of the Weights of Training Samples . . . . .	38
4.2.2	Effect of the Number of Iterations . . . . .	41
4.2.3	Optimization of $\beta$ . . . . .	43
4.2.3.1	Optimization of $\beta$ in MNIST Data Set	43
4.2.3.2	Optimization of $\beta$ in Yale Data Set . . . . .	46
4.3	Comparison with Other Algorithms . . . . .	50
4.3.1	Results in MNIST Data Set . . . . .	51

4.3.2	Results in Yale Face Data Set . . . . .	57
5	CONCLUSION . . . . .	69
	REFERENCES . . . . .	73

## LIST OF TABLES

### TABLES

Table 4.1	Misclassification Rate (%) of Algorithms . . . . .	41
Table 4.2	Selection of $\beta$ for Different Dictionary Sizes in MNIST Data Set . .	44
Table 4.3	Selection of $\beta$ for Different Dictionary Sizes on 3 Classes in Yale Face Data Set . . . . .	48
Table 4.4	Selection of $\beta$ for Different Dictionary Sizes on 5 Classes in Yale Face Data Set for 5 Classes . . . . .	49
Table 4.5	Misclassification Rate (%) Comparisons of Algorithms in MNIST Data Set . . . . .	53
Table 4.6	Residual Comparisons of Algorithms in MNIST Data Set . . . . .	53
Table 4.7	Misclassification Rates (%) on 3 Classes in Yale Face Data Set . . .	59
Table 4.8	Residual Error on 3 Classes in Yale Face Data Set . . . . .	59
Table 4.9	Misclassification Rates (%) on 5 Classes in Yale Face Data Set . . .	64
Table 4.10	Residual Error on 5 Classes in Yale Face Data Set . . . . .	64
Table 5.1	Computational Complexities of Classification Algorithms . . . . .	70

## LIST OF FIGURES

### FIGURES

Figure 3.1 Hermite 2D Basis Visualizations . . . . .	23
Figure 3.2 Illustration of Initialization Procedure . . . . .	33
Figure 4.1 Mnist Data Set Samples . . . . .	36
Figure 4.2 Yale Face Data Set Samples . . . . .	37
Figure 4.3 Misclassification Rate vs Kernel Scale $\sigma$ for Dictionary Size 10 in MNIST Data Set . . . . .	39
Figure 4.4 Misclassification Rate vs Kernal Scale $\sigma$ for Dictionary Size 1 in MNIST Data Set . . . . .	40
Figure 4.5 Misclassification Rate (%) vs Number of Iterations in MNIST Data Set . . . . .	42
Figure 4.6 Residual vs Number of Iterations in MNIST Data Set . . . . .	42
Figure 4.7 Atoms vs Number of Iterations in MNIST Data Set . . . . .	43
Figure 4.8 Misclassification Rate (%) vs $\beta$ in MNIST Data Set . . . . .	44
Figure 4.9 Residual vs $\beta$ in MNIST Data Set . . . . .	45
Figure 4.10 Misclassification Rate (%) vs $\beta$ in Yale Face Data Set for 3 Classes	46
Figure 4.11 Residual vs $\beta$ in Yale Face Data Set for 3 Classes . . . . .	47
Figure 4.12 Misclassification Rate (%) vs $\beta$ in Yale Face Data Set for 5 Classes	48
Figure 4.13 Residual vs $\beta$ in Yale Face Data Set for 5 Classes . . . . .	49

Figure 4.14 Atoms with the change of $\beta$ in Yale Face Data Set . . . . .	50
Figure 4.15 Misclassification Error (%) Comparisons of Algorithms in MNIST Data Set . . . . .	52
Figure 4.16 Residual Comparisons of Algorithms in MNIST Data Set . . . . .	52
Figure 4.17 Dictionaries Learnt with Proposed Method in MNIST Data Set for 3 Classes . . . . .	54
Figure 4.18 Dictionaries Learnt with TIDL in MNIST Data Set for 3 Classes . .	55
Figure 4.19 Dictionaries Learnt with K-SVD in MNIST Data Set for 3 Classes .	55
Figure 4.20 Dictionaries Learnt with LC-KSVD1 in MNIST Data Set for 3 Classes . . . . .	55
Figure 4.21 Dictionaries Learnt with LC-KSVD2 in MNIST Data Set for 3 Classes . . . . .	56
Figure 4.22 Misclassification Error (%) Comparisons of Algorithms in Yale Face Data Set for 3 Classes . . . . .	57
Figure 4.23 Residual Comparisons of Algorithms in Yale Face Data Set for 3 Classes . . . . .	58
Figure 4.24 Dictionaries Learnt with Our Method in Yale Face Data Set for 3 Classes . . . . .	60
Figure 4.25 Dictionaries Learnt with TIDL in Yale Face Data Set for 3 Classes .	60
Figure 4.26 Dictionaries Learnt with K-SVD in Yale Face Data Set for 3 Classes	60
Figure 4.27 Dictionaries Learnt with LC-KSVD1 in Yale Face Data Set for 3 Classes . . . . .	61
Figure 4.28 Dictionaries Learnt with LC-KSVD2 in Yale Face Data Set for 3 Classes . . . . .	61



Figure 4.29 Misclassification Error (%) Comparisons of Algorithms in Yale Face Data Set for 5 Classes . . . . .	62
Figure 4.30 Residual Comparisons of Algorithms in Yale Face Data Set for 5 Classes . . . . .	63
Figure 4.31 Dictionaries Learnt with Proposed Method in Yale Face Data Set for 5 Classes . . . . .	65
Figure 4.32 Dictionaries Learnt with TIDL in Yale Face Data Set for 5 Classes .	65
Figure 4.33 Dictionaries Learnt with K-SVD in Yale Face Data Set for 5 Classes	66
Figure 4.34 Dictionaries Learnt with LC-KSVD1 in Yale Face Data Set for 5 Classes . . . . .	66
Figure 4.35 Dictionaries Learnt with LC-KSVD2 in Yale Face Data Set for 5 Classes . . . . .	67

## LIST OF ABBREVIATIONS

BP	Basis Pursuit
D-KSVD	Discriminative K-SVD
FOCUSS	FOCAL Underdetermined System Solver
LASSO	Least Absolute Shrinkage and Selection Operator
LC-KSVD	Label Consistent K-SVD
MOD	Method of Optimal Directions
MP	Matching Pursuit
NN	Nearest Neighbor
OMP	Orthogonal Matching Pursuit
TIDL	Transformation Invariant Dictionary Learning
SVD	Singular Value Decomposition
SVM	Support Vector Machine

## CHAPTER 1

### INTRODUCTION

In recent years, dictionary learning has emerged as a new field of machine learning, which has the purpose of learning overcomplete signal models that are well-fit to the characteristics of the type of data at hand.

Nowadays, large volumes of high dimensional information are produced with different kinds of sensors. The rapid and large increase in the upload of high-resolution images on the Internet is only one example related to it. These huge amounts of data have to be processed efficiently with respect to time and memory. In this context, we focus on dictionary learning for the classification problem which is one of the most challenging tasks in machine learning and computer vision. Dictionary learning has broad application areas, some examples of which are as follows:

- Object recognition [4], [49], [64]
- Face recognition [11], [55], [60]
- Video processing [15], [21], [39]
- Medical diagnosis [17], [54], [61]

In dictionary learning, an overcomplete set of signals are to be learned, so that the signals of interest in an application have a sparse representation over the learned dictionary model. Sparse representations have the goal of expressing a given signal as a linear combination of a small number of signals. Leading examples of these types of algorithms are basis pursuit de-noising [9], [10], least absolute shrinkage and selection operator (LASSO) [44], sparse Bayesian learning [50] and FOCal Underdetermined System Solver (FOCUSS) [24]. In sparse representation problems, the signals used

for sparsely representing the input data are called atoms. The collection of atoms used for the sparse decomposition of an input signal is called as a dictionary. To create such a model, either predefined dictionaries are used or a special dictionary can be learned for each specific purpose. Fourier bases [26], discrete cosine bases [42] and wavelets [36] are very common examples of predefined dictionaries. However, Rodriguez and Sapiro [43] mention the odds of learned dictionaries over predefined ones in terms of classification.

Many unsupervised and supervised dictionary learning algorithms have been proposed in the recent years. The purpose of unsupervised dictionary learning is mostly to learn models with high signal approximation capabilities, whereas supervised dictionary learning usually refers to the learning of dictionary models well-adapted to a data classification problem. The leading examples of unsupervised dictionary learning are K-SVD [1] and Method of Optimal Directions (MOD) [20]. Such algorithms aim to create over-complete dictionaries consisting of atoms that capture well the characteristics of the input signals. Unsupervised dictionary learning methods have a wide application area in image processing tasks, such as:

- De-noising [1] , [18] , [63]
- Restoration [32]
- Super-resolution [56]
- Compression [5]

Since training data is unlabeled in unsupervised dictionary learning, these algorithms are usually not as powerful in classification applications as they are in data reconstruction and compression applications. This is because the information of the class labels of training images is typically needed to learn models with high discrimination power. Hence, supervised dictionary learning algorithms are more likely to be used for this aim. For example, Discriminative K-SVD [62] and Label Consistent K-SVD [29] add discriminate power to K-SVD [1] with the help of class labels. Besides, the Fisher criterion is used in supervised methods such as those of Zhou *et. al*[65] and Zhang *et. al* [57]. Moreover, there are many examples of supervised dictionary learning methods [34], [41] which have better classification performance than unsupervised

ones.

The time complexity is an important point for the design of real-time or time critical applications. Moreover, the platform where an algorithm is running may have limited memory resources in some problems. Object detection in images [48] and web page ranking [8] are some examples of dictionary learning applications where computational and time optimization is necessary. In this thesis, we aim to develop a supervised dictionary learning algorithm having good classification ability in a fast and computationally efficient way. We build on the recently proposed Transformation Invariant Dictionary Learning (TIDL) [59] method. Yuzuguler, Vural and Frossard aim to learn supervised dictionaries using an analytical basis in TIDL [59]. The representation of the dictionary atoms over an analytical basis allows the dictionary atoms to be stored with fewer coefficients than storing itself. The TIDL method is based on an objective function that includes an approximative and a discriminative term. For each atom, TIDL method define an index set having a predetermined number of images and calculate the representation error and discrimination power of an atom over the training images within this index set rather than using all of the training images for all atoms.

In this thesis, we aim to increase the discrimination power of TIDL considering the time and memory costs. For this reason, we introduce two main changes. Firstly, we remove the index sets from the learning algorithm and let all atoms be affected by all training images. In this new setting, the effect of each training image is not the same when calculating an atom. Secondly and more importantly, the TIDL algorithm has been observed to suffer from a lack of diversity between the atoms of the same class, which affects the classification performance. In order to increase the diversity between different atoms of the same class, we propose to introduce an incoherence term to the objective of the TIDL method. By the help of this term, the similarity between the dictionary atoms is reduced.

The classification and representation power of different algorithms are compared experimentally with that of the proposed method in the MNIST data set and the Yale Face data set. proposed method is compared to the K-SVD [1], LC-KSVD [29] and TIDL [59] methods as K-SVD [1] is one of the most leading algorithms, LC-KSVD [29] is a popular supervised extension of K-SVD with discrimination term and our

method has some inspirational ideas from TIDL [59].

## **1.1 Thesis Outline**

The purpose of this study is to develop a fast and computationally efficient supervised dictionary learning algorithm with good classification ability. The rest of the thesis is organized as follows.

In Chapter 2, we give a brief overview of sparse representations and dictionary learning algorithms. We also review the literature by presenting some leading examples of sparse representation methods, unsupervised and supervised dictionary learning algorithms.

In Chapter 3, we formulate the supervised dictionary learning problem and present the proposed algorithm in detail.

In Chapter 4, we evaluate our method with experiments on several data sets with respect to the classification and representation performance. Also, the performance changes with respect to the optimization parameters are discussed.

Finally in Chapter 5, the thesis is concluded with a summary of the study and discussions on the experimental findings.

## CHAPTER 2

### RELATED WORK

Dictionary learning is a learning process of an overcomplete set of signals with sparse representations. For this reason, we will give the definition of sparse signal representations and some leading examples of it: matching pursuit [37] and orthogonal matching pursuit [46], basis pursuit denoising [9], [10] and least absolute shrinkage and selection operator (LASSO). Later, two widely used unsupervised dictionary learning algorithms, namely the Method of Optimal Directions (MOD) [20] and K-SVD [1] are discussed with their definitions, application areas and leading examples in detail. We also mention the K-means clustering algorithm because it is relevant to the 1-sparse signal representations studied in this thesis. Lastly, we describe the supervised dictionary learning algorithms and discuss two widely used algorithms of it, namely Discriminative K-SVD [62] and Label Consistent K-SVD [29].

#### 2.1 Sparse Signal Representations

Sparse signal representations are widely used for collecting, compressing and reproducing high-dimensional signals. Some application areas are listed below:

- Face recognition [52]
- Image super-resolution [56]
- Clustering [19]
- Image and video restoration [35]
- Background subtraction [7]

- Image classification [33]

Before discussing sparse signal representations, it is useful to start with the  $l_p$  norm definition and the sparse matrix definition by the help of the  $l_0$  norm.

Let  $c \in \mathbb{R}^n$  be a row vector. To define the  $l_p$  norm of this vector, the following equations are used:

$$l_p \text{ norm, } \|c\|_p := \left( \sum_{i=1}^n |c_i|^p \right)^{\frac{1}{p}} \quad (2.1)$$

Likewise;

$$\begin{aligned} l_0 \text{ norm, } \|c\|_0 &= \#(i|c_i \neq 0) \\ &= \text{number of non-zero element of } c \end{aligned}$$

$$l_1 \text{ norm, } \|c\|_1 = \sum_{i=1}^n |c_i| \quad (2.2)$$

$$l_2 \text{ norm, } \|c\|_2 = \left( \sqrt{\sum_{i=1}^n |c_i|^2} \right)$$

$$l_\infty \text{ norm, } \|c\|_\infty = \max_i |c_i|$$

Please note that the  $l_0$  norm is not a norm because it does not satisfy the homogeneity,  $\|ac\|_0 = |a|\|c\|_0$  in general. Thus, it can be called as *quasi-norm*. The  $l_p$  norm is a norm for  $p > 1$  and *quasi-norm* for  $0 < p < 1$ .

The mathematician James H. Wilkinson [16] created the widely used definition of a sparse matrix as "A sparse matrix is a matrix with enough zeros that it is worth taking advantage of them." Using sparse matrices has the advantage of reducing the complexity of some computations. To clarify, calculations with zero entries take relatively much less time. Also, there are many algorithms helping to save memory on sparse matrix storage because this kind of algorithms do not store the zero elements



but only store the actual values in the matrix.

Let the  $X_{exp} \in \mathbb{R}^{5 \times 1}$  be an example for a sparse vector:

$$X_{exp} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The sparsity of  $X_{exp}$  is calculated as follows:

$$\text{Sparsity of } X_{exp} = \|X_{exp}\|_0 = 1$$

In other words, the sparsity of any vector can be calculated by counting the number of non-zero elements in it. Hence,  $X_{exp}$  is a 1-sparse vector.

Sparse representations have the goal of expressing a given signal as a linear combination of a small number of signals belonging to the data group. For this reason, the following sparse representation problem is often encountered where the vector  $y$  represents the input signal,  $c$  is the coefficient vector to represent  $y$  over the dictionary  $D$ , which is an overcomplete matrix each column of which is called an atom.

$$\min \|c\|_0 : y = Dc \quad (2.3)$$

or more commonly

$$\min \|c\|_0 : \|y - Dc\|_p \leq \delta \quad \text{for some } \delta > 0 \quad (2.4)$$

where the  $l_0$  norm is used as a sparsity constraint and  $p$  is often used as 2, but it could be 1 or  $\infty$  as well. Also, in some cases, the  $l_0$  norm sparsity constraint may be relaxed to higher-order norm constraints such as  $l_1$  and  $l_2$ .

While the calculation of the representation in equation (2.3) poses an NP-hard problem

[38], greedy algorithms can be used to solve these types of problems. Matching pursuit [37] and orthogonal matching pursuit [46] are leading and widely used algorithms to approximately solve this problem. Another approach to solve such problems is to relax the  $l_0$  norm sparsity constraint to  $l_1$  norm so that the problem statement changes as follows:

$$\min \|c\|_1 : y = D c \text{ or } \min \|c\|_1 : \|y - D c\|_p \leq \delta \quad \text{for some } \delta > 0 \quad (2.5)$$

Basis pursuit denoising [9], [10] and least absolute shrinkage and selection operator (LASSO) [44] are well-known examples of convex relaxation algorithms for solving problems as in eq. (2.5). Moreover, there are many different approaches trying to solve sparsity problem such as sparse Bayesian learning [50] and FOCal Underdetermined System Solver (FOCUSS) [24]. A review of the sparse recovery algorithms can be found in [47].

### 2.1.1 Matching Pursuit (MP)

Greedy algorithms aim to heuristically solve an optimization problem in an iterative way, by taking a locally optimal but often globally suboptimal step in each iteration. They are widely used in sparse approximation problems where a small number of non-zero coefficients are desired in the representation of a signal.

Matching pursuit [37] is an iterative greedy algorithm of pursuing the sparse approximation of a signal by selecting a single atom at each iteration.

The MP algorithm includes two steps in each iteration. The first step of each iteration has an aim of finding the atom having the highest correlation with the current residual error. The measure of this correlation is the norm of the orthogonal projection of the residual over the candidate atoms. At the second and the last step of each iteration, the residual error is updated by subtracting from the residual its projection onto the selected atom. The termination of the algorithm is done by halting conditions. The algorithm steps are stated below assuming the atoms are normalized:

## 1. Atom selection step

$$d^j = \arg \max_{\{d_i\}} | \langle r^{j-1}, d_i \rangle | \text{ for } 1 \leq i \leq N \quad (2.6)$$

where  $\langle . \rangle$  denotes the inner-product,  $N$  is the dictionary size,  $r^{j-1}$  is the residual at the  $(j-1)^{th}$  iteration and  $d_i$  shows the  $i^{th}$  column of the dictionary  $D$ . Please note that  $r^0 = y$  and  $c^j = \langle r^{j-1}, d^j \rangle$  is the coefficient of the atom  $d^j$  in the representation of the signal  $y$ .

## 2. Residual update step

$$r^j = r^{j-1} - c^j d^j \quad (2.7)$$

After the  $j^{th}$  operation, the approximation  $y^j$  of  $y$  can be found as:

$$y^j = \sum_{k=1}^j c^k d^k \quad (2.8)$$

The main advantage of the matching pursuit algorithm is its simplicity. However, it has drawbacks related to its convergence rate [37].

### 2.1.2 Orthogonal Matching Pursuit (OMP)

As an improvement over MP, the orthogonal matching pursuit algorithm is suggested by Tropp and Gilbert [46]. It is also a greedy and iterative algorithm. Although using the same procedure with MP, in each iteration OMP projects the input signal to the subspace spanned by all the atoms selected until that iteration. The approximation can be gathered by the following equation:

$$y^j = D^j c^j \text{ where } c^j = (D^j)^\dagger y \quad (2.9)$$

Please note that the *pseudo-inverse* of  $D^j$  is denoted by  $(D^j)^\dagger$ . Moreover,  $D^j = [D^{j-1} \ d^j]$  where  $d^j$  is the same as in eq. (2.6) because the atom selection step remains

the same. Also, multiple solutions can be obtained from the atom selection step at (2.6) if the matrix  $D$  has duplicated columns. For this case, transformations must be performed to make the matrix  $D$  *full-rank*.

Hence, the one and only difference of OMP from MP occurs in the residual update step as follows:

$$r^j = y - D^j c^j \quad (2.10)$$

Orthogonal matching pursuit algorithm introduces orthogonalization. By the help of orthogonality, any selected atom cannot be used again as in MP. Thus, the convergence rate increases although it does not guarantee the convergence [46].

### 2.1.3 Sparse Signal Representations by $l_1$ Minimization

#### 2.1.3.1 Basis Pursuit (BP)

Basis Pursuit published by Chen and Donoho [9] suggests the  $l_1$  norm relaxation on eq. 2.4. By the help of this change, the equation becomes a convex optimization problem and it may become solvable by linear programming algorithms. Thus, it is possible to say that the Basis Pursuit method makes the global optimization of a relaxed version of the original problem. The Basis Pursuit algorithm solves the following sparse representation problem with convex relaxation:

$$\min \|c\|_1 : y = Dc \quad (2.11)$$

For noisy conditions ( $\delta > 0$  at (2.5), Chen and Donoho, in their Basis Pursuit Denoising paper [10], also proposed the following problem, where the  $\lambda$  coefficient adjusts the sparsity of the representation:

$$\arg \min_c \|y - Dc\|_2^2 + \lambda \|c\|_1 \quad (2.12)$$

### 2.1.3.2 Least Absolute Shrinkage and Selection Operator (LASSO)

Tibshirani proposed the method LASSO [44]; i.e, Least Absolute Shrinkage and Selection Operator, to find a sparse solution where the parameter  $s$  adjusts the sparsity.:

$$\arg \min_c \|y - Dc\|_2^2 \quad \text{subject to } \|c\|_1 \leq s \quad (2.13)$$

It can also be represented as in eq (2.12).

The Basis Pursuit and LASSO problems can be handled by interior point methods. The in-crowd algorithm (The In-Crowd Algorithm for Fast Basis Pursuit Denoising [23]) and fixed-point continuation [25] can be leading examples of solution approaches to BP.

## 2.2 Dictionary Learning

In the sparse signal representations discussed in Section (2.1), the approximation of the input  $y$  as  $y = D c$  is covered with the sparsest possible coefficient vector  $c$ , which uses the least number of columns from a given dictionary  $D$ . Some examples of these fixed dictionaries can be the Fourier bases [26], discrete cosine bases [42] or wavelets [36]. Also, these dictionaries can be designed with tuning with respect to input signals. Some examples of these, still fixed, dictionaries are wavelet packets [12], curvelets [27] or bandelets [40]. Although these fixed dictionaries give good results in some applications, the learning of  $D$  instead of using a fixed one may be necessary for most applications. Dictionary learning algorithms have been proposed for this purpose. Given the input signal  $y$  and a sparse approximation problem as in eq. (2.4), the general procedure of the dictionary learning algorithms can be summarized as follows:

After initializing the dictionary (most of the applications use normalized atoms), do the following until reaching the stopping criterion:

- Fix the dictionary  $D$  and compute the sparse coefficient vector  $c$

- Fix the sparse coefficients  $c$  and optimize the dictionary  $D$

The optimization of the atoms in the dictionary can be done sequentially or concurrently, depending on the algorithm. If the optimization is done with sequential updates, each atom  $d_i \forall i$  is updated consecutively after fixing the sparse coefficients. This type of dictionary optimization can be called as an atom-based optimization. If the optimization is done concurrently, the dictionary matrix  $D$  is optimized so that each atom  $d_i \forall i$  is optimized concurrently.

Most of the dictionary learning algorithms use one or more of these stopping criteria:

- There is no significant change in  $D$  and  $c$ .
- Desired approximation level is reached. In other words, the minimization of the residual is fair enough.
- The maximum iteration number is reached.

### 2.2.1 Unsupervised Dictionary Learning

Unsupervised dictionary learning can be described as the computation of a dictionary that gives the sparsest possible approximations of a set of unlabeled training signals. In other words, no classification or grouping of the training signals is involved in the algorithm. Unsupervised dictionary learning algorithms try to create such dictionaries discovering and presenting the interesting and common structures in the training data. Hence, unsupervised dictionary learning algorithms are widely used in application areas like image denoising [1] , [18] , [63], compression [5], restoration [32] and super-resolution[56].

In the following, we give an overview of two widely used unsupervised dictionary learning algorithms, namely the Method of Optimal Directions (MOD) [20] and K-SVD [1]. We also discuss the K-means clustering algorithm due to its relevance to the 1-sparse signal representations studied in this thesis.

### 2.2.1.1 Method of Optimal Directions (MOD)

Engan *et. al.* [20] introduced the Method of Optimal Directions (MOD) in 1999 and it is one of the pioneer examples of dictionary learning methods. Like many dictionary learning algorithms, MOD has alternating two steps to solve the following optimization problem:

$$\arg \min_{D,C} \|Y - D C\|_F^2 \text{ such that } \forall i, \|c_i\|_0 < s \quad (2.14)$$

where  $Y$  denotes the input data set consisting of  $y_i$ s which are the input data samples  $\forall i$ ,  $F$  denotes the Frobenius norm,  $D$  is the dictionary and  $C$  is the sparse coefficient matrix consisting of the columns  $c_i$ s which are the sparse coefficient vectors  $\forall i$ .

#### 1. Sparse Coding Step

$$\min \|C\|_0 \text{ subject to } \|Y - D C\|_F^2 < \epsilon \text{ for small } \epsilon \quad (2.15)$$

This step can be done with fixing  $D$  at eq. (2.14) and calculating sparse coefficients. The sparse coding problem is given in (2.15) and it can be solved by any matching pursuit algorithm like MP (2.1.1) or OMP (2.1.2). The solution of the minimization in eq. (2.15) is given by its Moore-Penrose pseudo-inverse in eq. (2.16).

$$C^\dagger = C^T (C C^T)^{-1} \quad (2.16)$$

The computationally efficient way of calculating Moore-Penrose pseudo-inverse of  $C$  can be done by using singular value decomposition of  $C$  where  $C = U S V^T$  shown as follows:

$$C^\dagger = V S^{-1} U^T \quad (2.17)$$

#### 2. Dictionary Update Step

The dictionary update step can be done by fixing  $C$  and calculating the dictionary in the problem at eq. (2.14). The analytical solution of that problem gives the result as:

$$D = Y C^\dagger \text{ where } C^\dagger \text{ is a Moore-Penrose pseudo inverse} \quad (2.18)$$

If the input is low-dimensional, using MOD to learn a dictionary is efficient due to fast convergence. However, the MOD algorithm is not practical for high dimensional data. Because of the fact that finding the pseudo-inverse of high dimensional matrices is a striving process, MOD is not widely used nowadays with the need for high dimensional processing.

### 2.2.1.2 K-means

K-means [31] is a heuristic and iterative clustering algorithm. Although K-Means is a clustering algorithm, it will be discussed under unsupervised dictionary learning algorithms. It is commonly used in signal processing and data mining applications. It aims to separate the input data set into a pre-defined number of clusters. Let's say there are  $k$  clusters desired. First of all, the algorithm selects a cluster center for each cluster. The initialization step is done by taking all of the input samples and assigning each sample to the cluster corresponding to the nearest centroid. Then, all  $k$  centroids are updated as well as the assignments of the signals to the clusters, by minimizing the following objective function:

$$\min \sum_{j=1}^k \sum_{i=1}^{M_j} \|y_i^j - d_j\|^2 \quad (2.19)$$

where  $y_i^j$  is the data point of index  $i$  assigned to cluster  $j$ ,  $d_j$  is the centroid of the cluster  $j$ ,  $M_j$  is the number of data points belonging to the cluster  $j$  and  $\|y_i^j - d_j\|^2$  is the distance measure. Please note that the eq. (2.3) with 1 sparsity approaches the K-means clustering problem in eq. (2.19). In other words, the K-means algorithm can be called as an extreme sparse representation algorithm, where only one atom is allowed in the signal decomposition.



This algorithm has two iterating steps to minimize its objective function:

### 1. Data Assignment Step

Each data point  $y_i \forall i$  be labeled with the nearest centroid to itself. Let the nearest centroid is labeled with cluster  $j$ . The data assignment step can be formulated as follows:

$$I(y_i) = \arg \min_j \|y_i - d_j\|^2 \quad \forall i \quad (2.20)$$

where  $I(y_i)$  is the cluster index of the data sample  $y_i$ ,  $i$  is the index of the data point and  $d_j \in D$  is the  $j^{\text{th}}$  cluster center as initialized with some initialization rule for the first iteration.

### 2. Centroid Update Step

The new mean of the data points belonging to each cluster gives the new centroid of that cluster as follows:

$$d_j = \frac{1}{M_j} \sum_{i:I(y_i)=j} y_i \quad (2.21)$$

where  $M_j$  is the number of data points belonging to cluster  $j$ .

The algorithm iterates between these two steps and is stopped with respect to a stopping criterion such as:

- There is no change in clusters after some iterations.
- The distances to the cluster centers are sufficiently minimized.
- The maximum number of iterations is reached.

Being a heuristic method, the K-means algorithm does not ensure the convergence. However, initialization with random centroids and allowing a sufficient number of iterations may be helpful for a better outcome.

### 2.2.1.3 K-SVD

Aharon *et al.* [1] proposed the K-SVD algorithm. It can also be called as generalized K-means. To solve the sparse coding problem with the dictionary  $D$ , the coefficient matrix  $C$  and the sparsity  $s$  below, the algorithm suggests two main steps; namely, the sparse approximation and the dictionary update step.

$$\arg \min_{D,C} \|Y - D C\|_F^2 \quad \text{subject to} \quad \forall i, \|c_i\|_0 < s \quad (2.22)$$

#### 1. Sparse Approximation Step

The sparse approximation step is done by fixing  $D$  in the previous equation and proceeding with orthogonal matching pursuit algorithm as discussed in Section (2.1.2). Any pursuit algorithm like Basis Pursuit (BP) [9] or Focal Under-determined System Solver (FOCUSS) [24] can be replaced with OMP.

$$\arg \min_C \|Y - D C\|_F^2 \quad \text{such that} \quad \forall i, \|c_i\|_0 < s \quad (2.23)$$

#### 2. Dictionary Update Step

Dictionary update step is done sequentially by processing each column of input signal  $Y$  using the singular value decomposition (SVD). As in K-means, numerous atoms represent each input signal with various weights. That is the reason why this algorithm is called as generalized K-means. To clarify, the algorithm updates one column of  $D$  at a time by fixing all columns in  $D$  except one,  $d_k$  and update  $d_k$  by optimizing the target function as follows:

- $E_k = Y - \sum_{j \neq k} c_j^j d_j$  where  $c_j^j$  is the  $j^{th}$  row vector in the coefficient matrix  $C$
- Apply SVD decomposition as  $E_k = U \Delta V^T$
- Update:  $d_k = u_1$  and  $x_k = \Delta_{1,1} v_1$  where  $u_1$  and  $v_1$  are the first column vectors of  $U$  and  $V$  matrices, accordingly.  $\Delta_{1,1}$  is the first element of  $\Delta$  matrix.

Although the convergence of K-SVD is not guaranteed, it gives good results in most applications.

## 2.2.2 Supervised Dictionary Learning

Unlike unsupervised dictionary learning, supervised dictionary learning needs a supervision from a user, data scientist or expert. The supervision involves the assignment of a class label to each training signal by the help of its characteristics.

The road-map for supervised dictionary learning for a given classification application is listed below:

1. Obtain a suitable training set for the classification problem at hand.
2. Transform each input signal into a feature vector to include only its representative characteristics
3. Apply the dictionary learning algorithm to create representative dictionaries
4. If the algorithm includes some control parameters, make sure to adjust them for optimized results
5. Test the algorithm with test sets using them as input signals and measure the success rate

Supervised dictionary learning methods often have objective functions including discriminative terms in addition to reconstructive terms; in other words, the objective function ensures different data representations for different classes. The discrimination can be achieved by altering the sparse coding step in dictionary learning algorithms considering the following needs:

- The sparsest possible representation of an input signal and also
- The most different representation of an input signal from signals belonging to other classes

The learning problem underlying supervised dictionary learning algorithms can be generalized with the following formula suggested by I. Tosic and P. Frossard [45]:

$$\arg \min_{D,C} [ \|Y - D C\|_2^2 + \gamma_1 \|C\|_1 + \gamma_2 \Omega(C, D, \theta) ] \quad (2.24)$$

where  $\Omega(C, D, \theta)$  is the discrimination function depending on the coefficient matrix  $C$ , the dictionary  $D$  and optional parameters  $\theta$  of the classification model.

It can be easily said that the algorithm is dependent on the classification function and in most applications it may be non-convex. Still, fixed-point continuation methods can solve the problem efficiently when logistic loss function is used as a classification term [34].

The formula in eq. (2.24) uses the  $l_1$  norm, the convex relaxation as a sparsity constraint. It can also be formulated in terms of the  $l_0$  norm as:

$$\arg \min_{D, C} [ \|Y - D C\|_2^2 + \gamma_1 \|C\|_0 + \gamma_2 \Omega(C, D, \theta) ] \quad (2.25)$$

Rodriguez and Sapiro [43] propose a method of supervised OMP using singular value decomposition at its dictionary update stage. They aim the discrimination of classes and reconstruction of the input signals at the same time on the image data. Zhang and Li propose Discriminative K-SVD for Dictionary Learning in Face Recognition [62] which combines the representation objective of K-SVD with the discrimination of a linear classifier. LC-KSVD [29] is an extended version of D-KSVD due to newly represented discriminative sparse-code error. Moreover, Yankelevsky and Elad [58] add a graph regularization term to the formula of D-KSVD. Additionally, the sparse reconstruction term is moved from the objective function to the constraints of the problem by Zhou *et. al* [65]. The Fisher criterion is another term used on sparse coefficients by some supervised dictionary learning algorithms like [28] and [57].

The recently proposed TIDL method [59] is a supervised dictionary learning algorithm, which, moreover, aims to learn analytically represented atoms. The analytical representation of dictionaries is frequently done using analytical bases. The Transformation Invariant Dictionary Learning (TIDL) method is discussed in more detail in Section 3.1.

### 2.2.2.1 Discriminative K-SVD (D-KSVD)

Zhang and Li's Discriminative K-SVD (D-KSVD) [62] adds a discrimination term into the K-SVD algorithm discussed in Section 2.2.1.3. Labeling of training data

is used in order to learn a linear classifier with the help of a discrimination term. Moreover,  $l_2$  norm is used for regularization. D-KSVD proposes the following problem to optimize:

$$\arg \min_{D, W, C} \|Y - D C\|_2 + \gamma_1 \|H - W C\|_2 + \gamma_2 \|W\|_2 \text{ such that } \|C\|_0 \leq T \quad (2.26)$$

where the classifier parameters are represented by  $W$ . The matrix  $H$  has columns  $h_i = [0, \dots, 1, \dots, 0]$ , where the class labels are indicated by positions of the non-zero entries. Thus,  $\|H - W C\|_2$  shows the classification error and  $\|W\|_2$  is the regularization penalty.  $\gamma_1$  and  $\gamma_2$  are positive weight parameters.

### 2.2.2.2 Label Consistent K-SVD (LC-KSVD)

Jiang *et. al* [29] have proposed the Label Consistent K-SVD (LC-KSVD) method by introducing additional discrimination power to the dictionary learning objective. They define two methods in their study; namely, LC-KSVD1 and LC-KSVD2.

- LC-KSVD1

$$\arg \min \|Y - D C\|_2 + \gamma_1 \|Q - A C\|_2 \text{ such that } \|C\|_0 \leq s \quad (2.27)$$

where  $\|Q - A C\|_2$  is an additional discriminative term proposed as an improvement over the traditional dictionary learning objective. This term encourages the sparse codes to better capture the label information.  $A$  is a linear transformation matrix.  $Q = [q_1, q_2, \dots, q_N] \in \mathbb{R}^{k \times N}$  and  $q_i \in \mathbb{R}^k$  are the discriminative sparse codes of the input signals  $y_i$  which are the columns of  $Y$ . The non-zero values of  $q_i$  indicate the indexes where  $y_i$  and the atom  $d_k$  hold the same class label.  $\gamma_1$  is the control parameter of this term. For example, let  $D = [d_1, d_2, \dots, d_6]$ ,  $Y = [y_1, y_2, \dots, y_6]$  and  $Q$  be defined as follows:

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

This  $Q$  matrix is used to indicate that  $y_1, y_2, d_1$  and  $d_2$  are from class 1,  $y_3, y_4, d_3$  and  $d_4$  are from class 2 and likewise  $y_5, y_6, d_5$  and  $d_6$  are from class 3.

- LC-KSVD2

$$\arg \min \|Y - D C\|_2 + \gamma_1 \|Q - A C\|_2 + \gamma_2 \|H - W C\|_2 \text{ such that } \|C\|_0 \leq s \quad (2.28)$$

where  $\|H - W C\|_2$  is an additional term to the objective in eq. (2.27). The classification error is represented by this term.  $W$  is a linear predictive classifier matrix denoting the classifier parameter.  $H = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{m \times N}$  whose  $i^{th}$  column  $h_i = [0, 0, \dots, 1, 0, \dots, h_N]^T \in \mathbb{R}^m$  is the label vector corresponding to the input signal  $y_i$  which is the  $i^{th}$  column of  $Y$ . The non-zero value of  $h_i$  indicates the class of  $y_i$  and  $\gamma_2$  is the control parameter of this term.

## CHAPTER 3

### SUPERVISED DICTIONARY LEARNING WITH 1-SPARSE REPRESENTATIONS

In this chapter, we present the proposed method for supervised dictionary learning. We first give an overview of the motivation of our work and then describe our supervised dictionary learning method for fast classification with 1-sparse representations.

Recently supervised dictionary learning algorithms have shown promising results in classification applications but how to achieve good classification performance in a time-efficient manner is still an important open question. The computational constraints of classifying an input signal may be as important as the classification performance in some applications. Real-time object detection in images [48], web page ranking [8] are well-known examples where computational constraints have important roles.

In order to achieve fast and computationally efficient classification, we propose a supervised dictionary learning algorithm in this chapter. Our method builds on the recently proposed Transformation Invariant Dictionary Learning (TIDL) [59] method, which aims to learn supervised dictionaries for fast classification by using 1-sparsity. It also further decreases the memory requirements for storing the learnt atoms by representing them in an analytical basis. These are the main goals we share with TIDL. However, we further would like to improve the discrimination power of the learnt dictionaries considering computational needs and speed. Hence, we start with the TIDL algorithm and improve its performance by introducing suitable modifications in its objective function.

### 3.1 Transformation Invariant Dictionary Learning (TIDL)

Yuzuguler, Vural and Frossard designed the transformation invariant dictionary learning (TIDL) algorithm [59]. TIDL is a supervised dictionary learning method where analytically represented dictionaries are learnt using the Hermite basis. The representation of the dictionary atoms over an analytical basis permits the handling of geometric transformations more easily, thus, facilitates the learning of transformation-invariant dictionaries, which is one of the purposes of the TIDL method. The objective function of TIDL includes representation and discrimination terms. An index set having a predetermined number of images and the representation error are defined for each atom and the discrimination power of an atom is calculated over the training images within this index set rather than using all of the training images for all atoms.

Let  $Y = \{Y^m\}_{m=1}^M$  be a training set consisting of  $M$  classes. For example, class  $m$  has  $M_m$  labeled training images  $y_i^m \in \mathbf{R}^n$  where  $m \in \{1, 2, \dots, M\}$ .

TIDL method aims to learn a specific dictionary for each class instead of a global dictionary. These class-specific dictionaries are represented as  $D^m = \{d_i^m\}_{i=1}^N$  consisting of  $N$  atoms for class  $m \in \{1, 2, \dots, M\}$ . By the help of these dictionaries, the class label of a given test image  $y$  can be estimated as  $m^*$  with respect to class-specific reconstruction errors computed by the following formula:

$$m^* = \arg \min_{m=1,2,\dots,M} \left\| y - \sum_{i=1}^N c_i^m d_i^m \right\| \quad s.t. \quad \|c^m\|_0 = 1 \text{ so } c^m \text{ is 1-sparse.} \quad (3.1)$$

where  $d_i^m$  is the  $i^{th}$  atom in the dictionary of class  $m$ ,  $c^m = [c_1^m \ c_2^m \ \dots \ c_N^m]$  is the coefficient vector and  $N$  is the dictionary size which is equal for all classes  $m$ ,  $m \in \{1, 2, \dots, M\}$ . Moreover, the coefficient vector  $c^m$ ,  $m \in \{1, 2, \dots, M\}$  has a constraint of 1-sparsity by  $\|c^m\|_0 = 1$  for fast classification. To clarify, the estimated class label  $m^*$  is the class label that yields the minimum residual between the test image and its 1-sparse approximation.

The classification success rate is measured by the percentage of test images whose estimated class label  $m^*$  is accurate with the label assignment in eq. (3.1).



TIDL satisfies the fast classification purpose using 1-sparsity as a sparsity constraint in the algorithm. In addition to this, each atom is learnt as a discretized version of a two-dimensional analytical function over an analytical basis. This decreases the memory need for storing the dictionary atoms because each atom can be approximated with a relatively small number of coefficients in the analytical basis. The analytical representation is done over the Hermite 2D basis [51]. The Hermite basis forms an orthonormal basis for the square-integrable functions. Any atom can be represented as  $d$  with a pre-determined number  $s$  of the Hermite basis elements as follows:

$$d = \sum_{i=1}^s \alpha_i h_i \quad (3.2)$$

where  $\alpha_i$  are the coefficients of the basis vectors for  $i \in \{1, 2, \dots, s\}$  and  $\{h_i\}_{i=1}^{\infty}$  denotes the Hermite 2D basis with increasing degrees of Hermite polynomials used in their construction [51].

In TIDL, the atoms are analytically represented using the Hermite basis as follows:

$$d = H \alpha \quad (3.3)$$

where  $H \in \mathbf{R}^{n \times s}$  is the Hermite 2D matrix and  $\alpha$  is the coefficient vector  $\mathbb{R}^{s \times 1}$  whose  $i^{th}$  entry is  $\alpha_i$ . Figure 3.1 shows the visualizations of Hermite 2D basis vectors. The numbers written on the upper left corners of each image in Figure 3.1 denote the degrees of the basis vectors  $h_1$  to  $h_{10}$  in the Hermite 2D matrix  $H$ .

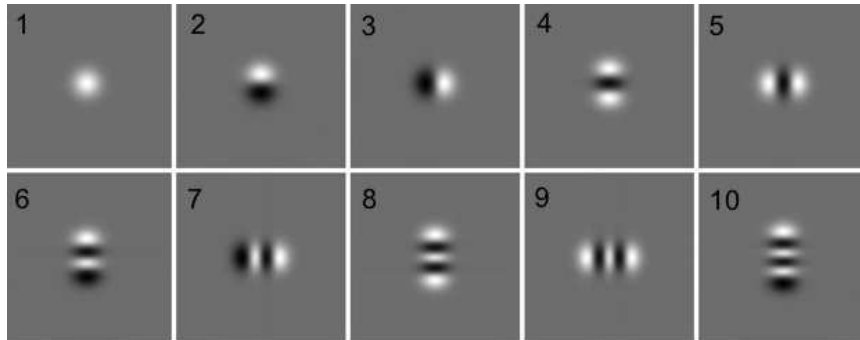


Figure 3.1: Hermite 2D Basis Visualizations

TIDL has the purpose of achieving invariance to geometric transformations. Training images are assumed to be geometrically transformed with respect to a transformation parameter vector  $\lambda \in \Lambda$  in transformation parameter domain  $\Lambda$ . The parameter vector  $\lambda$  can be any combination of geometric transformations such as rotation, scaling and affine transformation. Please note that the geometric transformations are assumed to constitute a linear operator on the images.

In order to counterbalance the effect of the geometric transformation on training image  $y$ , using a geometrically transformed form of  $H$  is enough due to the property of being linear.  $H_\lambda$  is the Hermite 2D basis [51] which is the geometrically transformed version of  $H$  by  $\lambda$ . The geometrically transformed atom  $d_\lambda$  can be represented over the basis  $H_\lambda$  with the same coefficient vector  $\alpha$  with following formula:

$$d_\lambda = H_\lambda \alpha \quad (3.4)$$

As already stated, Yuzuguler, Vural and Frossard use  $\|c^m\|_0 = 1$  as a sparsity constraint in TIDL. Due to this constraint, each image is represented by only one atom. Then, due to the usage of 1-sparsity, no constraints on the coefficient vector  $c$  are needed. The objective function of TIDL given in eq. (3.5) is used for learning an atom from the class  $m$ ,  $d_l^m \in D^m$ . The optimization problem is a function of  $\alpha$  which is the coefficient vector representing the learnt atom  $d_l^m$  in the Hermite basis.

$$f(\alpha) = \sum_{i \in I_l^{mm}} \|y_i^m - H_{\lambda_i^m} \alpha\|^2 - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i \in I_l^{mj}} \|y_i^j - H_{\lambda_i^j} \alpha\|^2 \quad (3.5)$$

Here,

- $I_l^{mj}$  and  $I_l^{mm}$  are the index sets that identify a predetermined number of images from  $j^{th}$  and  $m^{th}$  classes respectively, having the highest correlation with  $d_l^m$ .
- $\lambda \in \Lambda$  is a geometric transformation parameter vector and represents a combination of rotation, scaling, affine transformation parameters. The transformation parameter vector  $\lambda_i^m$  in eq. (3.5) represents the transformation applied to the training sample  $y_i^m$ . The transformation parameter of each training sample is assumed to be estimated

before the optimization of  $\alpha$ .

- $H_\lambda$  is the Hermite 2D basis [51] which is geometrically transformed by the parameter vector  $\lambda$ .
- $\eta$  is the parameter which encourages the learnt atom to be different from the samples from other classes.

The objective in eq. (3.5) aims to learn dictionaries whose atoms are similar to the training samples from the same class but dissimilar from the training samples from other classes. The diversity among the atoms in the dictionary of the same class can be enhanced by the help of random initialization and re-selecting index sets at each iteration.

If there is no transformation applied to the training image, the objective function to learn an atom  $d$  in the Hermite basis as a function of  $\alpha$  can be simplified as follows:

$$f(\alpha) = \sum_{i \in I_l^{mm}} \|y_i^m - H \alpha\|^2 - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i \in I_l^{mj}} \|y_i^j - H \alpha\|^2 \quad (3.6)$$

where  $I_l^{mj}$  and  $I_l^{mm}$  are the same index sets as in eq. (3.5).

After some operations, the following formula is obtained:

$$f(\alpha) = \alpha^T A \alpha - 2 b^T \alpha + c \quad (3.7)$$

where

$$\begin{aligned} A &= \sum_{i \in I_l^{mm}} H^T H - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i \in I_l^{mj}} H^T H \\ b &= \sum_{i \in I_l^{mm}} H^T y_i^m - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i \in I_l^{mj}} H^T y_i^j \end{aligned} \quad (3.8)$$

In order to minimize the objective in eq. (3.7), the coefficients  $\alpha$  are optimized by

solving  $\nabla f = 0$  as follows:

$$\alpha^* = A^{-1} b \quad (3.9)$$

When the matrix  $A$  is positive definite, the objective function  $f(\alpha)$  is strictly convex and the solution in eq. (3.9) finds the global minimum of the objective. In order to ensure the positive definiteness of  $A$ ,  $\eta$  is chosen as  $\frac{1}{4}$  of the smallest  $\eta$  value making the smallest eigenvalue of  $A$  vanish.

The TIDL algorithm can be summarized as follows:

- Given the training set  $Y = \{Y^m\}_{m=1}^M$ , the centroid of each set  $\{Y^m\}$  is calculated  $\forall m \in \{1, 2, \dots, M\}$ .
- The atoms of class  $m$  are initialized with the training images labeled with class  $j \neq m \forall j$  which are the most distant ones from the centroids of class  $m$ . This gives the advantage of taking the problematic regions close to class boundaries into account in order to decrease the misclassification rate.
- The selection of dictionary size  $N$  of each class  $j$  is important due to a trade-off between complexity and accuracy. The correct selection of  $N$  is achieved when the dictionaries of all classes cover the necessary information for classification within tolerable complexity.
- The sizes of index sets  $I_l^{mj}$  and  $I_l^{mm}$  must be selected carefully. If they are over selected, this will increase the similarity between the atoms in the same dictionary. If they are under selected, some useful information in training images may be discarded. Both of these improper selections may reduce the correct classification rate.
- TIDL determines the index sets  $I_l^{mj}$  and  $I_l^{mm}$  for atom  $d_l^m$  by selecting a predetermined number of training images from each class having the highest correlation with the atom  $d_l^m$ . In other words,  $I_l^{mj}$  is the set of indices of the training images from class  $j$  which are the most correlated ones with the  $l^{th}$  atom of the dictionary of class  $m$ .
- Transformation parameters  $\lambda_i^j$ 's are determined by the estimation of geometric transformations applied on the training images. This estimation can be done in several

ways. TIDL suggests the alignment of  $y_i^j$  with a reference image of each class for finding  $\lambda_i^j$ . This step is only required if the training images have undergone geometric transformations. Otherwise, it is not needed. In our experiments with TIDL, this step is skipped because we assumed that all images are geometrically aligned.

- TIDL iterates between the following steps  $\forall l \in \{1, 2, \dots, N\}$ :
  - Minimize the eq. (3.7) to find the Hermite coefficients  $\alpha$  for  $d_l^m$
  - Update atom  $d_l^m = H \alpha$

### 3.2 Our Method

Our method has the aim of attaining good classification performance with low computational cost. In order to reach this purpose, we build on some ideas from the TIDL method.

These are as follows:

- Using an analytical basis
- Using 1-sparsity as a sparsity constraint
- Using the idea of a discrimination term in the dictionary learning objective

If the test signals have high sizes, the computational power and storage capability of a device (computer or any embedded platform) have to be high enough to overcome the load of large matrix computations. To face this difficulty, memory consumption for the signal storage can be restricted by representing them over an analytical basis because oversized signals can be approximated with relatively fewer coefficients by using analytical basis representations without losing their characteristics.

In addition, using 1-sparsity reduces the computations thanks to reducing the number of coefficients in the coefficient vector  $c$  in eq. (3.1). Therefore, it is effective to use 1-sparsity to speed up the tests and thanks to that reach wide application areas where users want to get immediate classification results. In particular, the computational complexity of calculating a 1-sparse representation of a test signal in a dictionary of  $N$  atoms is only of  $O(N)$ , which can be easily found in any platform. Robust Real-Time

Face Detection [48], object detection [53] and image segmentation using dictionary learning [6] are just three of the numerous application areas where the speed and memory optimization are as crucial as the classification performance. To summarize, because of the fact that the computational and time complexity are important purposes of our method, it is useful to use an analytical basis and 1-sparsity.

The usage of a discrimination term improves the discriminative characteristics of atom, and consequently, the classification performance.

Although our method has some similar points with TIDL with the following modifications, which sketches the main features of our algorithm:

- Using all training images in the optimization of each atom, however, by including them with different weights in the objective (rather than using index sets as in TIDL)
- Addition of an incoherence term in the objective in order to enhance the variability between the atoms

In many applications, there is a limited number of training data or limited time for training. Due to such constraints, it is important to use all characteristic information from each training sample. Rather than defining index sets in the discrimination and representation terms as in TIDL, we use all training images for each step of dictionary learning in order to avoid any loss of characteristic information.

The usage of all training images can be useful. However, each of them carries information with different significance. For example, the training images labeled with the same class as the atom to be learnt are more important than others when the focus of a dictionary learning algorithm is accurate representation. For good classification, the training images belonging to other classes should also be taken into account according to their significance for the learnt atom in our algorithm.

With the weighted usage of all training images in the proposed method, the learnt atoms in the class-representative dictionaries tend to become similar. In order to avoid limited variability between atoms from the same class, we propose to introduce an incoherence term in the objective. The incoherence term is a term that measures the mutual coherence of the atoms labeled with the same class and thanks to it, the atoms

are encouraged to be different from each other. By reducing the similarity between the atoms in the dictionary, the contents of the class-representative dictionaries are enriched, and thus, the classification rate increases.

Our method aims to learn a specific dictionary for each class rather than learning a global dictionary. These class-specific dictionaries are represented as  $D^m = \{d_i^m\}_{i=1}^N$  consisting of  $N$  atoms for classes  $m \in \{1, 2, \dots, M\}$ . The atoms are represented over the 2D Hermite basis as in eq. (3.2) and eq. (3.3) where  $\alpha \in \mathbb{R}^{s \times 1}$  is the coefficient vector whose  $i^{th}$  entry is  $\alpha_i$ . In other words, the representation of each atom is as follows:

$$d = H \alpha = \sum_{i=1}^s \alpha_i h_i \quad (3.10)$$

Since the atoms can be written as a function of  $\alpha$  due to their representation in an analytical basis, the proposed dictionary learning method for learning an atom  $d = d_l^m$  from the  $m^{th}$  class has the following objective function of  $\alpha$  as follows:

$$\begin{aligned} f(\alpha) &= \sum_{i=1}^{M_m} \mu_i^m \|y_i^m - H \alpha\|^2 \\ &- \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} \mu_i^j \|y_i^j - H \alpha\|^2 \\ &+ \beta \|\alpha^T H^T H \bar{\alpha}\|^2 \end{aligned} \quad (3.11)$$

where

- $\bar{\alpha} = [\alpha_1^m \ \alpha_2^m \ \dots \ \alpha_{i-1}^m \ 0_{s,1} \ \alpha_{i+1}^m \ \alpha_{i+2}^m \ \dots \ \alpha_N^m]$
- $\alpha_i^m \in \mathbb{R}^{s \times 1}$  is coefficient vector representing the  $i^{th}$  atom of the dictionary belonging to class  $m$  in the Hermite basis.
- $0_{s,1}$  is the zero vector in  $\mathbb{R}^{s \times 1}$ .
- $s$  is the number of Hermite basis vectors.
- $\beta$  is the incoherence coefficient.

- $N$  is the dictionary size.
- $\mu_i^m$  and  $\mu_i^j$  are the weights of the training samples.
- $M_m$  and  $M_j$  are the number of training images labeled with class  $m$  and with class  $j$ , correspondingly.

The first term  $\mu_i^m \sum_{i \in M_m} \|y_i^m - H \alpha\|^2$  gives the total weighted distance of the learnt atom to the training samples labeled with same class. It can be called as an approximative term because it gathers information from training images from the same class with that atom.

The second term  $\sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i \in M_j} \mu_i^j \|y_i^j - H \alpha\|^2$  enforces discrimination between the atom and the training images labeled with different classes. Hence, this term can be called as a discriminative term.

The learnt atom needs to be a good representative of the training images from its class due to the first term. Moreover, it has the ability to discriminate itself from the other classes due to the second term. The effect of discriminative and representative properties are adjusted by the factor of  $\eta$ .

We set the weight parameters of the training samples using the Gaussian Kernel as follows:

$$\begin{aligned} \mu_i^j &= e^{-\|y_i^j - d\|^2 / \sigma^2} \\ &= e^{-(y_i^{jT} y_i^j + d^T d - 2 y_i^{jT} d) / \sigma^2} \end{aligned} \quad (3.12)$$

In TIDL, index sets are selected to determine which samples affects the learnt atoms. However in our method, the weight parameters of the training samples with the use of the Gaussian Kernel are selected with respect to the distances between the learnt atoms and the samples. Samples closer to the learnt atom affect it affects the atom more than the distant ones.

The last term is the incoherence term which increases the in-class variability. Thanks to this term, the atoms of the same class do not mimic each other anymore. Bao *et. al* [2], Barchiesi and Plumley [3] use the mutual coherence term as the maximum



absolute inner product of dictionary atoms as  $\max_{i \neq j} | \langle d_i, d_j \rangle |$  where  $d_i$  and  $d_j$  are the atoms with index  $i$  and index  $j$ , respectively. Ramirez uses the incoherence term as  $\|D^T D - I\|_F^2$  where  $I$  is the identity matrix and  $F$  denotes the Frobenius norm. Our incoherence term is given in (3.13).

$$\|\alpha^T H^T H \bar{\alpha}\|^2 \quad (3.13)$$

The weight of the incoherence term in the overall objective is adjusted by the parameter  $\beta$ . The effect of  $\beta$  will be discussed in Chapter 4.

Note that the objective in (3.11) is directly based on the pairwise distances between the atom and each training sample. This can be interpreted in the way that the 1-sparsity constraint  $\|c^m\|_0 = 1$  is inherently included in the learning objective.

We set the weight parameters of the training samples using the following equation by replacing  $\mu_i^j$  and  $\mu_i^m$  with their values in the eq. (3.14):

$$\begin{aligned} f(\alpha) = & \sum_{i=1}^{M_m} e^{-\|y_i^m - d\|^2 / \sigma^2} \|y_i^m - H \alpha\|^2 \\ & - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} e^{-\|y_i^j - d\|^2 / \sigma^2} \|y_i^j - H \alpha\|^2 \\ & + \beta \|\alpha^T B\|^2 \end{aligned} \quad (3.14)$$

where

- $B = H^T H \bar{\alpha}$
- $\bar{\alpha} = [\alpha_1^m \ \alpha_2^m \ \dots \ \alpha_{i-1}^m \ 0_{s,1} \ \alpha_{i+1}^m \ \alpha_{i+2}^m \ \dots \ \alpha_N^m]$
- $\alpha_i^m \in \mathbb{R}^{s \times 1}$  is the coefficient vector representing the  $i^{th}$  atom of the dictionary belonging to class  $m$  in the Hermite basis.
- $0_{s,1}$  is the zero vector in  $\mathbb{R}^{s \times 1}$ .
- $s$  is the number of Hermite basis vectors.
- $\beta$  is the weight of the incoherence term.

- $N$  is the dictionary size.
- $H$  denotes the Hermite Basis.

After some operations, the following formula is obtained:

$$f(\alpha) = \alpha^T A \alpha - 2 b^T \alpha + c \quad (3.15)$$

where

$$A = \begin{cases} \sum_{i=1}^{M_m} e^{-\|y_i^j - d\|^2 / \sigma^2} H^T H - \\ \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} e^{-\|y_i^j - d\|^2 / \sigma^2} H^T H + \\ \beta B B^T \end{cases} \quad (3.16)$$

$$b = \sum_{i=1}^M H^T y_i^m - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^M H^T y_i^j$$

where  $B = H^T H \bar{\alpha}$  and  $\bar{\alpha} = [\alpha_1^m \ \alpha_2^m \ \dots \ \alpha_{i-1}^m \ 0_{s,1} \ \alpha_{i+1}^m \ \alpha_{i+2}^m \ \dots \ \alpha_N^m]$

Solving  $\nabla f = 0$  yields the solution of the objective function in eq. (3.15) :

$$\alpha^* = A^{-1} b \quad (3.17)$$

The objective function has a unique global minimum given by the solution (3.17) when the matrix  $A$  is positive-definite. As in TIDL method, in order to ensure the positive definiteness of  $A$ ,  $\eta$  is chosen as  $\frac{1}{4}$  of the smallest  $\eta$  value making the smallest eigenvalue of  $A$  vanish.

Our dictionary learning method is sequential because it optimizes the dictionary as atom by atom. Also, there must be a well-designed initialization step before starting the iterations to achieve the best results. The algorithm is described as follows:

- Given the training set  $Y = \{Y^m\}_{m=1}^M$ , the centroids of each set  $\{Y^m\} \ \forall m \in$

$\{1, 2, \dots, M\}$  is calculated as follows:

$$\text{centroid of class } m = \frac{\sum_{i=1}^{M_m} y_i^m}{M_m} \quad (3.18)$$

where  $M_m$  is the number of images in training set belonging to class  $m$ .

- The atoms of class  $m$  are initialized with the training images labeled with class  $j \neq m \forall j$  which are the most distant ones from the centroids of class  $m$ .

The initialization of an atom with the most distant samples from the other classes gives the advantage of taking the problematic regions near class boundaries into account in order to decrease the misclassification rate.

Figure 3.2 gives a simple illustration of the initialization procedure where each class has a dictionary consisting of only one atom of dimension 2 with coordinates  $(x,y)$  and the training data, centers and atoms are not normalized.

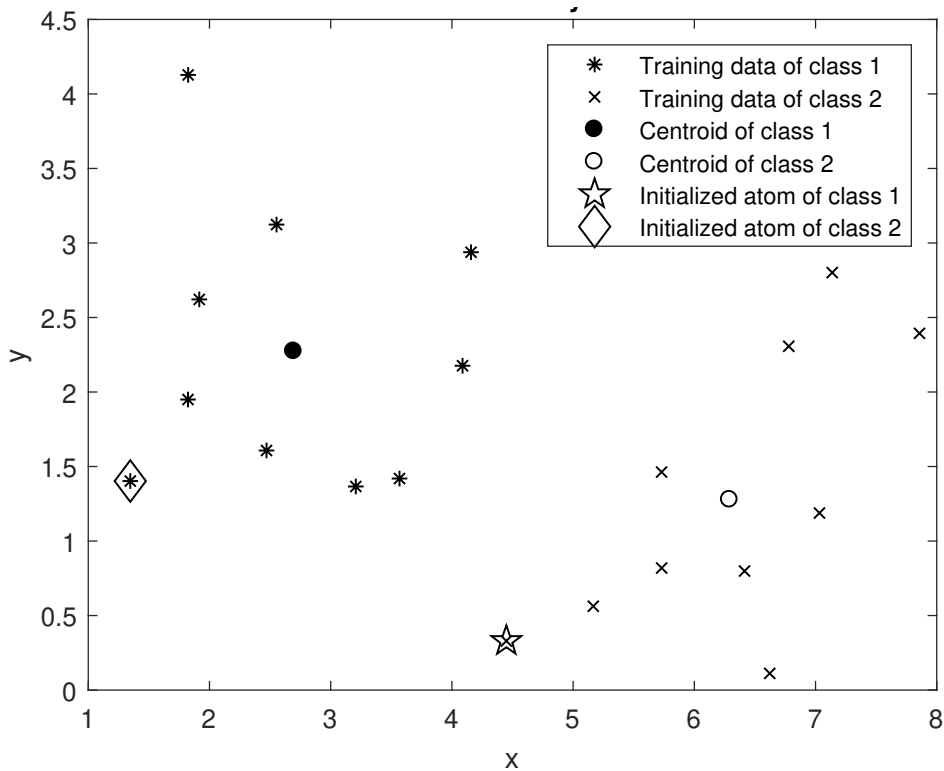


Figure 3.2: Illustration of Initialization Procedure

The outline of the proposed algorithm is as follows:

- Iterate between the following steps  $\forall i \in 1, 2, \dots, N$ :
  - Minimize the eq. (3.14) to find the Hermite basis coefficients  $\alpha$  of  $d_i^m$  by solving  $\nabla f = 0$  as discussed.
  - Update atom  $d_i^m = H \alpha$ .

The residual is the distance between the test image  $y$  and its best 1-sparse approximation within all class-representative dictionaries given in eq. (3.19).

$$\text{Residual} = \min_{m=1,2,\dots,M} \left\| y - \sum_{i=1}^N c_i^m d_i^m \right\| \quad s.t. \quad \|c^m\|_0 = 1 \text{ so } c^m \text{ is 1-sparse.} \quad (3.19)$$

where  $d_i^m$  is the  $i^{\text{th}}$  atom in the dictionary of class  $m$ ,  $c^m = [c_1^m \ c_2^m \ \dots \ c_N^m]$  is the coefficient vector and  $N$  is the dictionary size,  $m \in \{1, 2, \dots, M\}$  and  $\|c^m\|_0 = 1$  denotes the 1-sparsity constraint.

The class-representative dictionaries learnt with our algorithm can be used for the estimation of the class label  $m^*$  of a given test image  $y$  as follows:

$$m^* = \arg \min_{m=1,2,\dots,M} \left\| y - \sum_{i=1}^N c_i^m d_i^m \right\| \quad s.t. \quad \|c^m\|_0 = 1 \text{ so } c^m \text{ is 1-sparse.} \quad (3.20)$$

Using the class estimation in eq. (3.20), the residual can be formulated as follows:

$$\text{Residual} = \left\| y - \sum_{i=1}^N c_i^{m^*} d_i^{m^*} \right\| \quad s.t. \quad \|c^{m^*}\|_0 = 1 \quad (3.21)$$

## CHAPTER 4

### EXPERIMENTAL RESULTS

In this chapter, we present the performance evaluation of the proposed supervised dictionary learning method.

We first discuss the two data sets used; namely, MNIST Data Set [30] and Yale Face Data Set [22]. The MNIST data set consists of handwritten digit images. The Yale Face Data Set contains face images from different subjects with different light settings.

Secondly, we show how the algorithm parameters affect the classification performance and the residual measurement. The classification performance is measured by the misclassification rate, which is the mislabeling percentage of test images. Mislabeling is assigning a different class label to the test image rather than its true class label which is assigned with eq. (3.20) The residual of a test signal is the difference between its best 1-sparse approximation using a class specific dictionary and signal itself in accordance with eq. (3.19).

Finally, we compare our algorithm with the K-SVD [1], LC-KSVD [29] and TIDL [59] methods in terms of classification performance and residual measurement. K-SVD is a popular dictionary learning algorithm, which is known to provide accurate sparse representations. LC-KSVD is a supervised dictionary learning algorithm with discriminative properties proposed by Jiang *et. al* based on K-SVD. TIDL is the analytical dictionary learning method, which we have aimed to improve in this thesis. Hence, comparisons with these algorithms are helpful for evaluating the performance of our method.

## 4.1 Data Sets and Experimentation Settings

### 4.1.1 MNIST Data Set

The MNIST [30] data set contains handwritten digit images. It has a training set of 60,000 examples and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image frame.

In our experiments, we use 200 randomly selected training images and 200 randomly selected test images and resize them to a resolution of  $141 \times 141$  pixels. We use all 10 classes in all experiments conducted with the MNIST data set. The experiments are repeated for 100 Monte Carlo trials and the average is reported.

Some examples from different classes of the MNIST data set are shown in Figure 4.1.

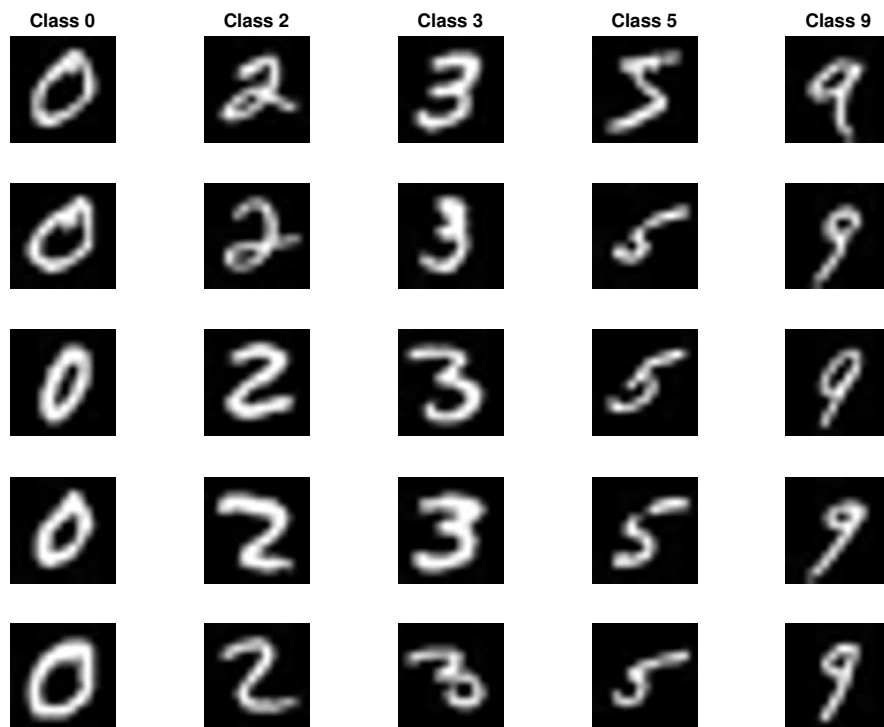


Figure 4.1: Mnist Data Set Samples

### 4.1.2 Yale Face Data Set

The data set consists of face images of 38 different people, where each person has 58 images with some variations on lighting angles and intensities from a single viewpoint [22].

In the Yale Face data set experiments, we randomly select 40 training images and 18 test images. We repeat this selection for 100 times (Monte Carlo trials) and report the average of the obtained results. Experiments are conducted with 3 and 5 classes which are randomly selected for each Monte Carlo trial.

Some examples from different classes of the Yale Face data set are shown in Figure 4.2.

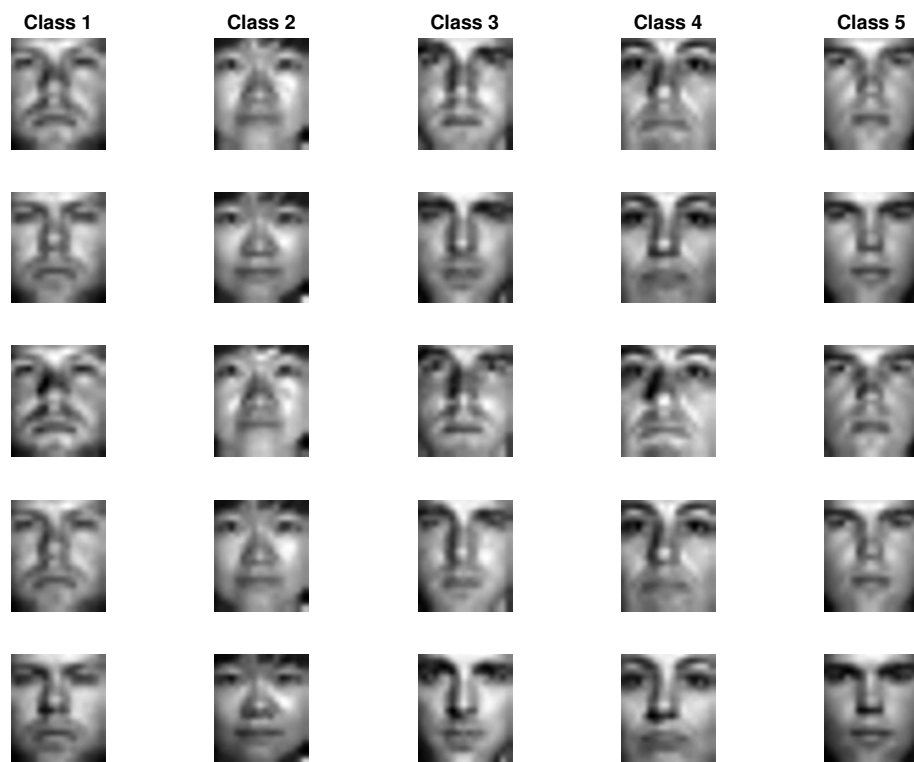


Figure 4.2: Yale Face Data Set Samples

## 4.2 Variation of the Performance with the Algorithm Parameters

### 4.2.1 Selection of the Weights of Training Samples

In TIDL, training images are either used or unused with regard to their selection in the index sets. Thus, the selection of the training images when learning an atom can be called as "hard". In our method, all training images are used in accordance with the selection of their weight parameters which can be called as a "soft" selection. To better understand the effect of the "hard" or "soft" selections, we consider a simplified version of our algorithm without the incoherence term in our experiments as follows:

$$f(\alpha) = \sum_{i=1}^{M_m} \mu_i^m \|y_i^m - H \alpha\|^2 - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} \mu_i^j \|y_i^j - H \alpha\|^2 \quad (4.1)$$

where  $\mu_i^m$  and  $\mu_i^j$  are the weights of the training images. Training images for class  $m$  are  $y_i^m$  and for class  $j$  are  $y_i^j \quad \forall i = 1, 2, \dots, M_j$  where  $M_j$  is the count of training images.  $H$  is the matrix representing the Hermite Basis.

The solution can be found as follows after making  $A$  positive definite with a suitable selection of  $\eta$ :

$$f(\alpha) = \alpha^T A \alpha - 2 b^T \alpha + c \Rightarrow \alpha^* = A^{-1} b \quad \text{where}$$

$$A = \sum_{i=1}^{M_m} \mu_i^m H^T H - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} \mu_i^j H^T H \quad (4.2)$$

$$b = \sum_{i=1}^{M_m} \mu_i^m H^T y_i^m - \eta \sum_{j \in \{1, \dots, M\} \setminus \{m\}} \sum_{i=1}^{M_j} \mu_i^j H^T y_i^j$$

The following kernels are examined for the choice of the training sample weight parameters:

- Kernel 1: Gaussian Kernel

$$\mu_i^j = e^{-\|y_i^j - d\|^2 / \sigma^2} = e^{-(y_i^j{}^T y_i^j + d^T d - 2 y_i^j{}^T d) / \sigma^2} \quad (4.3)$$



The Gaussian Kernel is related to the distance between the training sample and the learnt atom. With the selection of the Gaussian Kernel as weight parameters, the distant training images have a smaller effect on the learnt atom than the closer ones.

- Kernel 2: Correlation Kernel

$$\mu_i^j = e^{(-2 y_i^{jT} d)/\sigma^2} \quad (4.4)$$

In contrast to the Gaussian Kernel, the Correlation Kernel increases with the distance between the training sample and the learnt atom. With this selection, the atoms are affected more by the distant training images than closer ones.

The aim of this experiment examining the selection of the weight parameters is to determine the effect of the "soft" selection of weights and the effect of the distance between the training image and the atom on the classification performance. To conduct this experiment, 100 Monte Carlo runs are done with randomly selected 200 training and 200 test images of the MNIST data set. Then, Figures 4.3 and 4.4 are obtained from the above two kernels for dictionaries respectively of 10 and 1 atoms.

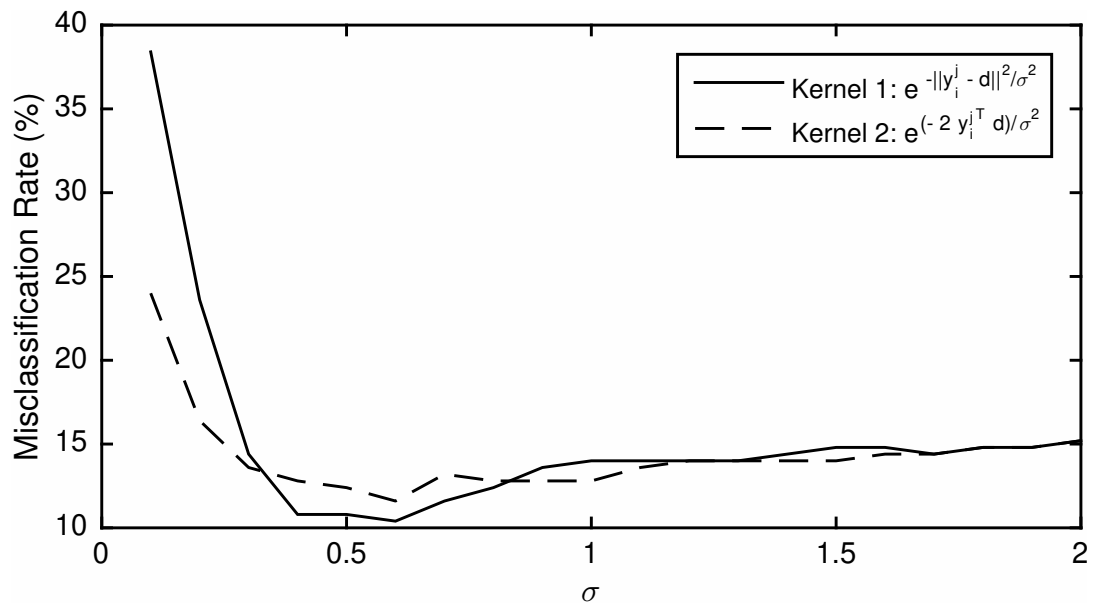


Figure 4.3: Misclassification Rate vs Kernel Scale  $\sigma$  for Dictionary Size 10 in MNIST Data Set

In Figures 4.3 and 4.4, the algorithm using the Gaussian Kernel in setting the weight parameter shows better classification performance than the one using the Correlation Kernel for both experiments conducted with different dictionary sizes. Thus, it can be concluded that the selection of the Gaussian Kernel is more appropriate for the classification purposes in the objective in eq. (4.2).

The atoms learnt by weight selection with the Correlation Kernel lose class-representative characteristics more than the atoms learnt by weight selection with the Gaussian Kernel because they are highly affected by the training images belonging to other classes. Thus, the above conclusions drawn from Figures 4.3 and 4.4 are as expected.

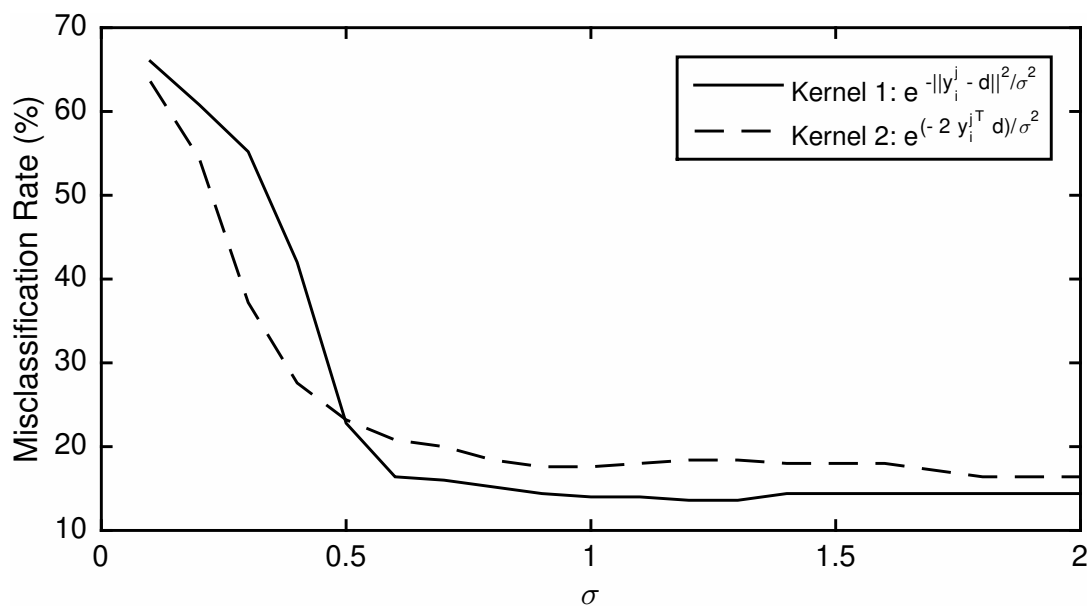


Figure 4.4: Misclassification Rate vs Kernel Scale  $\sigma$  for Dictionary Size 1 in MNIST Data Set

For a special and unusual dictionary consisting only one atom, we expect the misclassification rate to decrease with the increase in the kernel scale  $\sigma$  because consisting of only one atom, each dictionary needs to be affected by all training images. Figure 4.4 also confirms our expectations. However, for usual dictionaries consisting of sufficiently many atoms, the kernel scale  $\sigma$  has to be selected with respect to the dictionary size for creating a more comprehensive dictionary with number of atoms in the dictionary.

If  $\sigma$  is over selected, all training images affect all of the learnt atoms in the dictionary which decreases the variation in the dictionary. If  $\sigma$  is under selected, fewer training images have an effect on the learnt atoms so the class-representativeness of the dictionary decreases.

Also, the effect of "hard" or "soft" selection of the weights is also questioned with this experiment. The following table shows the optimized results of TIDL and the simplified version of our method for dictionary sizes of 1 and 10.

Table 4.1: Misclassification Rate (%) of Algorithms

Number of Atoms	1	10
TIDL	19,4	12,65
Simplified Method	<b>18,8</b>	<b>11</b>

The "soft" selection of weight parameters of training images creates a more informative dictionary so yields better classification performance than the "hard" one as used in TIDL as shown in Table 4.1.

#### 4.2.2 Effect of the Number of Iterations

The effect of the number of iterations is studied in our dictionary learning approach in this section. The number of Monte Carlo runs and the number of training and test images are the same as those of Section 4.2.1 in this experiment. Number of atoms in each class specific dictionary is 10 and the residual definition in eq. (3.19). The results are shown in Figures 4.5, 4.6 and 4.7.

In our dictionary learning algorithm, the atoms are sequentially learnt, where each atom is optimized by fixing all the others. In several dictionary learning algorithms such as K-SVD, once all atoms are optimized, the algorithm comes back to the beginning to have another pass over all atoms and these iterations are continued until convergence. We thus study in this experiment whether it is useful to employ our algorithm in an iterative manner, i.e., by re-optimizing all atoms sequentially several times in the learning. Figures 4.5 and 4.6 show the misclassification rate and the

residual respectively with the change on the number of iterations, i.e. the number of times the whole dictionary is re-optimized in the learning.

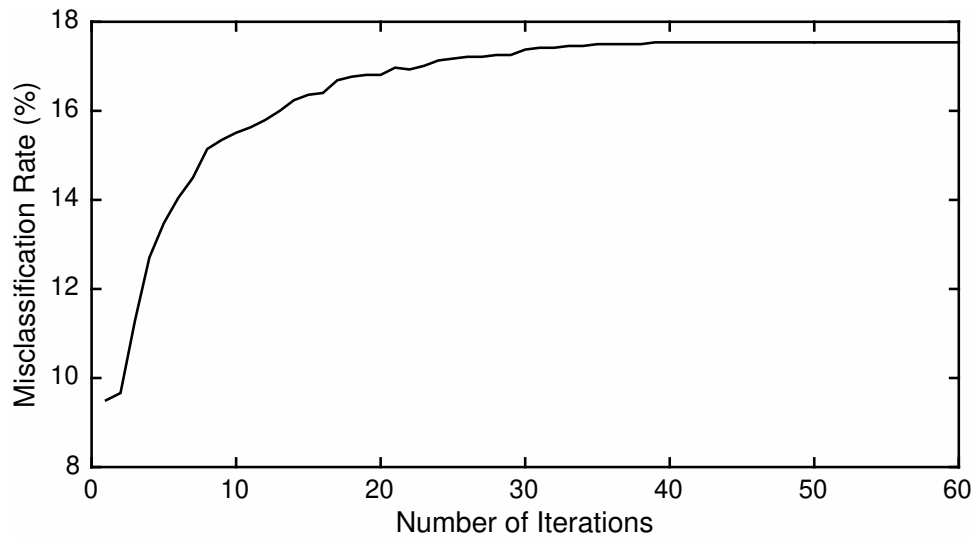


Figure 4.5: Misclassification Rate (%) vs Number of Iterations in MNIST Data Set

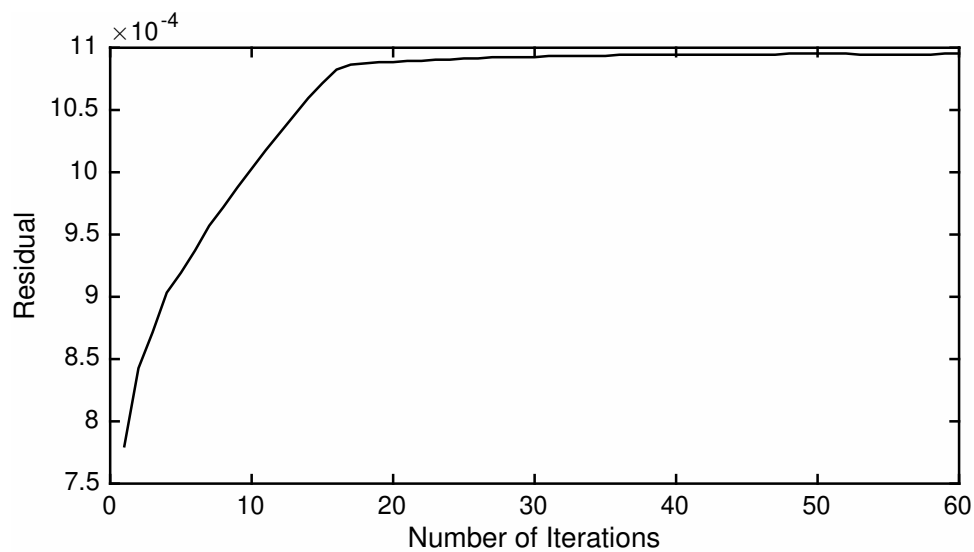


Figure 4.6: Residual vs Number of Iterations in MNIST Data Set

It is seen in Figure 4.7 that as the number of iterations increases, atoms lose their classification characteristics and atoms within the same class also tend to get more

similar to each other. This visual inspection of the atoms is in line with the observation drawn from Figure 4.5 and 4.6 that terminating the algorithm after a single iteration gives better classification and representation performance.






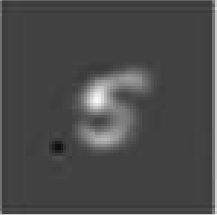

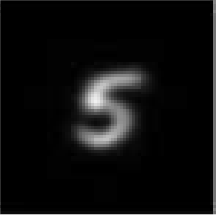

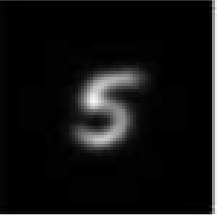
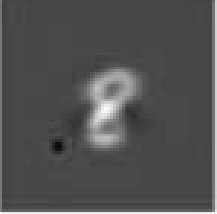
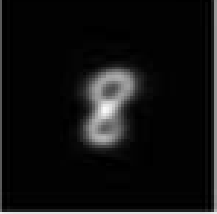
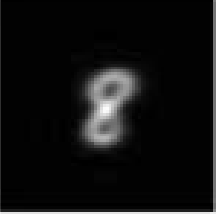
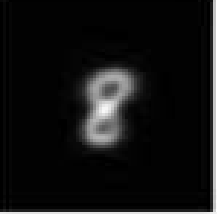
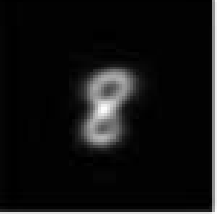
Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
				
				
				

Figure 4.7: Atoms vs Number of Iterations in MNIST Data Set

**4.2.3 Optimization of  $\beta$**

In this section, we examine the effect of the parameter  $\beta$  which weights the incoherence term. With the increase of  $\beta$ , the mutual coherence of atoms is decreased. This leads to an increase in the diversity of the atoms from the same class.

**4.2.3.1 Optimization of  $\beta$  in MNIST Data Set**

The experiments are conducted with the setup described in Section 4.1.1 where the  $\sigma$  values are fixed in the objective function in eq. (3.14). Figure 4.8 shows the variation

of the classification performance with respect to  $\beta$  in the MNIST data set.

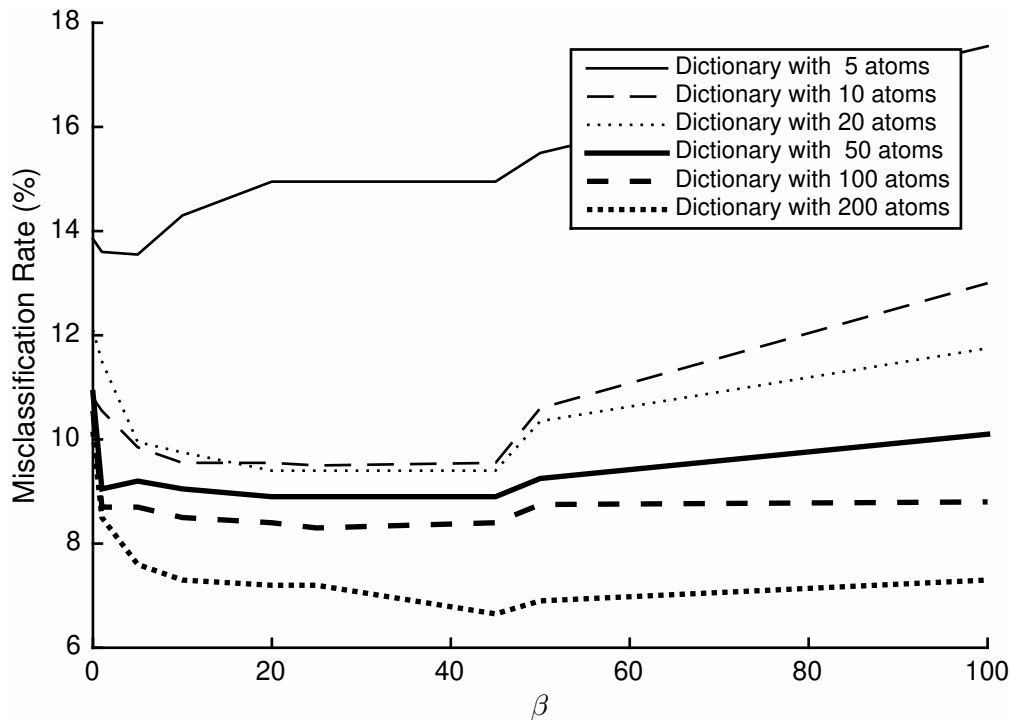


Figure 4.8: Misclassification Rate (%) vs  $\beta$  in MNIST Data Set

Since the estimation of the class labels is done by eq. 3.20, the increase in the number of atoms in the dictionary increases the chances of the selection of a suitable class-representative atom for each test sample. Thus, higher dictionary sizes perform better in classification. The results in Figure 4.8 are in line with these expectations.

Table 4.2 shows the  $\beta$  values optimizing the classification performance for the MNIST data set:

Table 4.2: Selection of  $\beta$  for Different Dictionary Sizes in MNIST Data Set

Number of Atoms	1	5	10	20	50	100	200
$\beta$	1	0,7	25	20	20	25	45
Misclassification Rate (%)	18,6	13,3	9,5	9,4	8,9	8,3	6,55

Figure 4.9 shows the residual change with respect to  $\beta$  in the MNIST data set and it

suggests that there is an optimal  $\beta$  value around 10 that minimizes the residual for all dictionary sizes. If the  $\beta$  is under-selected, the atoms used in the representation of the test image have limited diversity. If  $\beta$  is over selected, the incoherence term becomes the most dominant term in the objective function in eq. (3.14). Thus, the approximation capability of the atoms degrade when  $\beta$  is increased too much beyond the optimum  $\beta$ .

The selection of the  $\beta$  value with respect to the residual has some differences with the one with respect to the misclassification performance in Figure 4.8. This can be explained by the fact that the residual is only related to the representation so only the atoms from the dictionary of the same class, whereas the classification performance is related to all dictionaries from different classes. Consequently, when the size of the dictionary increases, the similarity between atoms should be decreased for better correct classification rates unlike finding the optimum  $\beta$  for better residuals. Since we aim the best correct classification rate, we select  $\beta$  values as in Table 4.2 for the experiments conducted for the MNIST data set.

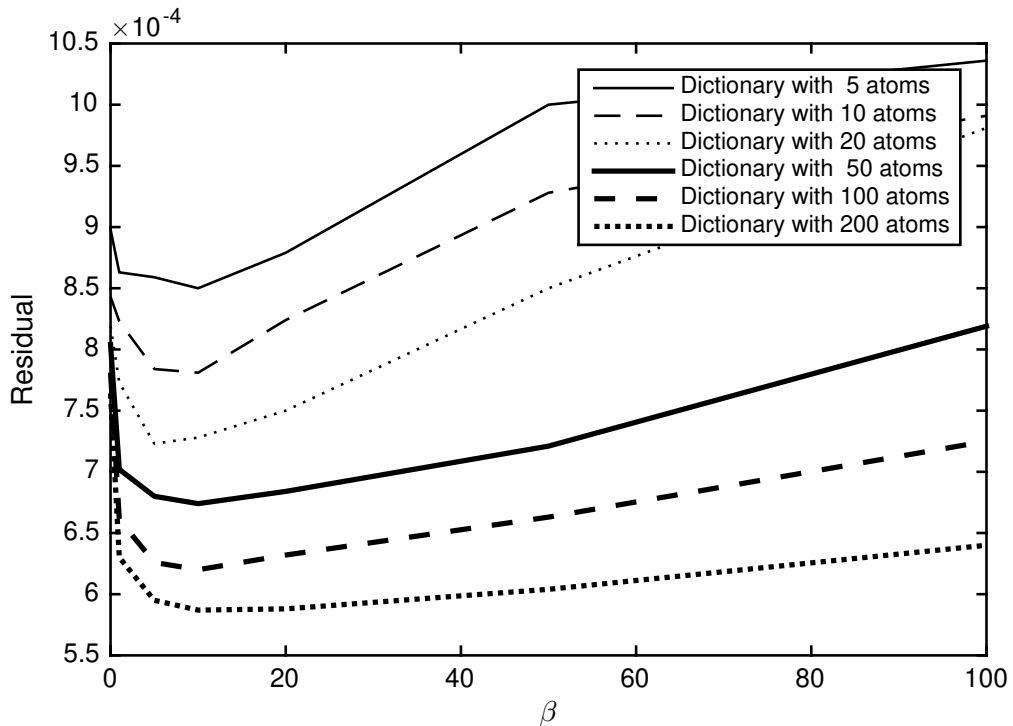


Figure 4.9: Residual vs  $\beta$  in MNIST Data Set

### 4.2.3.2 Optimization of $\beta$ in Yale Data Set

For the Yale data set, we conduct two main experiments, by choosing respectively 3 and 5 classes from the data set. The classes are randomly chosen in each Monte Carlo trial and the setup in Section 4.1.1 is used to study the effect of  $\beta$  on the performance of our method.

- **Experiments on 3 different classes:**

Figures 4.10 and 4.11 show the results obtained by experimenting on 3 classes from the data set. It is observed in Figure 4.10 that the classification performance increases with the size of the dictionaries. Also, Figure 4.11 shows that larger dictionaries produce more accurate representations by reducing the residual as expected.

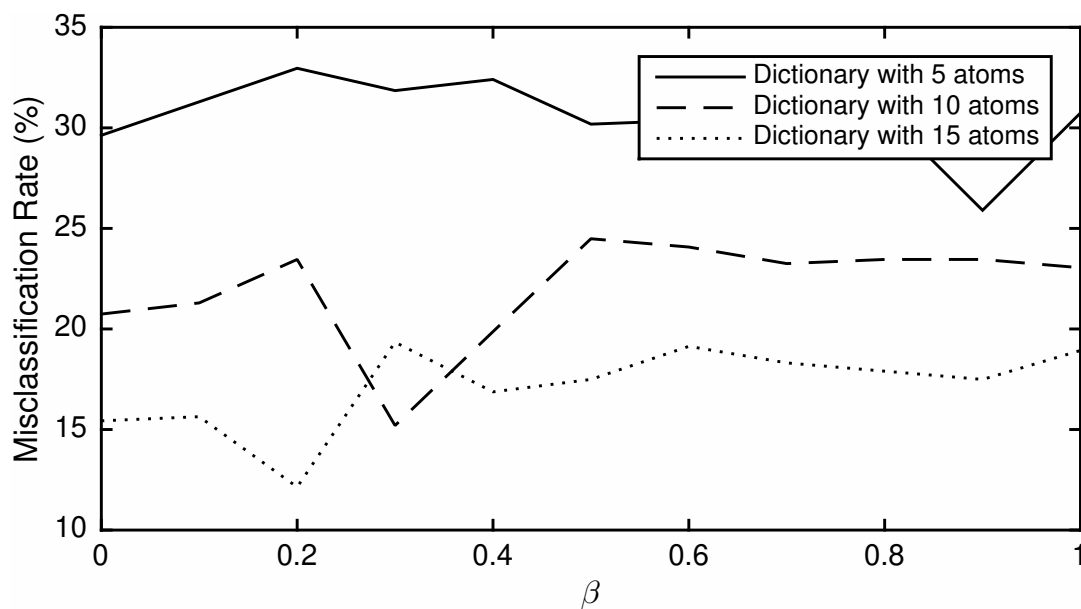


Figure 4.10: Misclassification Rate (%) vs  $\beta$  in Yale Face Data Set for 3 Classes



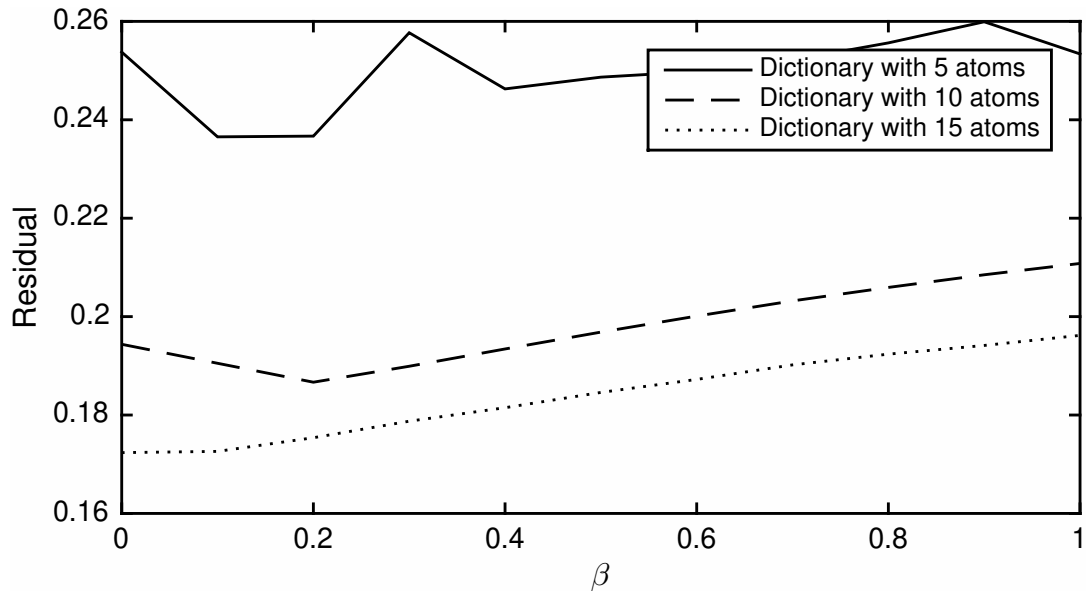


Figure 4.11: Residual vs  $\beta$  in Yale Face Data Set for 3 Classes

From Figure 4.10, it is difficult to draw the conclusion about the relationship of incoherence term and the increase in the size of the dictionaries like the one from the experiments conducted with the MNIST data set where increasing  $\beta$  for larger dictionaries resulted in better correct classification rates. The training images belonging to the same class have larger variability in the Yale Face data set than the ones in the MNIST data set. Hence, the dissimilarity between the atoms has a natural tendency to increase when there are more atoms in the dictionary in the Yale Face data set unlike in the MNIST data set. In other words, the atoms learnt in the Yale Face data set are not similar by the nature of the training images so the need for the incoherence term decreases by the increase in the size of the dictionaries.

The residual increases with respect to the increase in  $\beta$  as shown in Figure 4.11. With the increase in  $\beta$ , the atoms become as dissimilar from each other as they become dissimilar from the training images labeled with same class as the atoms. This causes an increase in the residual so a decrease in the representation power.

The optimized results for 3 classes in the Yale data set are obtained with the  $\beta$  values given in Table 4.3:

Table 4.3: Selection of  $\beta$  for Different Dictionary Sizes on 3 Classes in Yale Face Data Set

Number of Atoms	1	5	10	15
$\beta$	0,5	0,9	0,3	0,2
Misclassification Rate (%)	32,8	25,9	15,2	12,14

• **Experiments on 5 different classes:**

Figure 4.12 and Figure 4.13 show the results obtained on 5 different classes from the Yale data set.

The same conclusion with the experiments on 3 classes in the same data set is drawn about the variation of the misclassification rate and the residual change with respect to the size of the dictionary. The misclassification rate and the residual decrease with when the size of the dictionary increases as shown in Figure 4.12 and Figure 4.13.

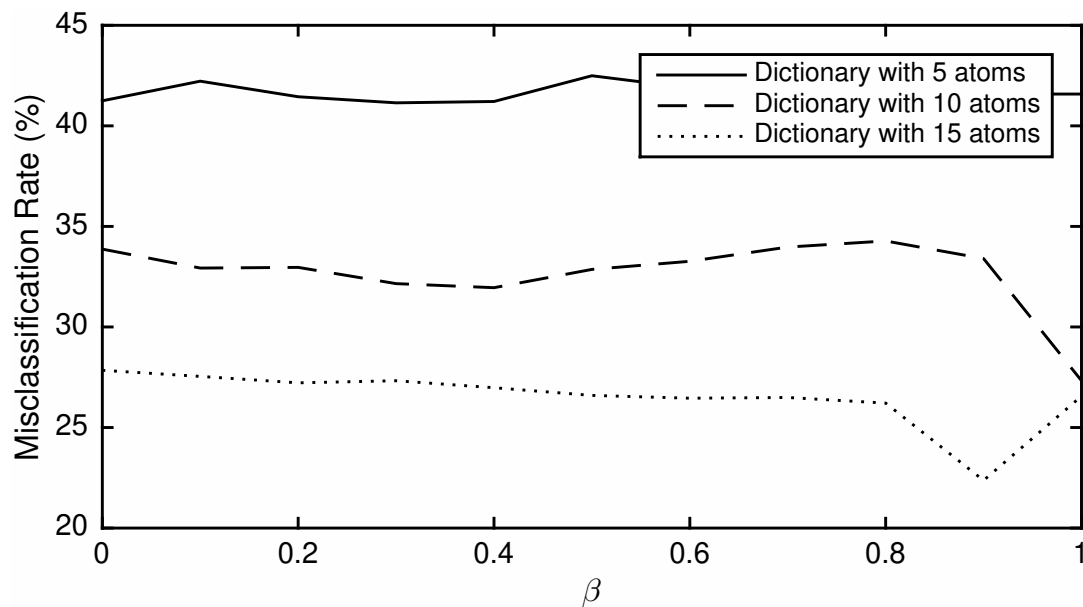


Figure 4.12: Misclassification Rate (%) vs  $\beta$  in Yale Face Data Set for 5 Classes

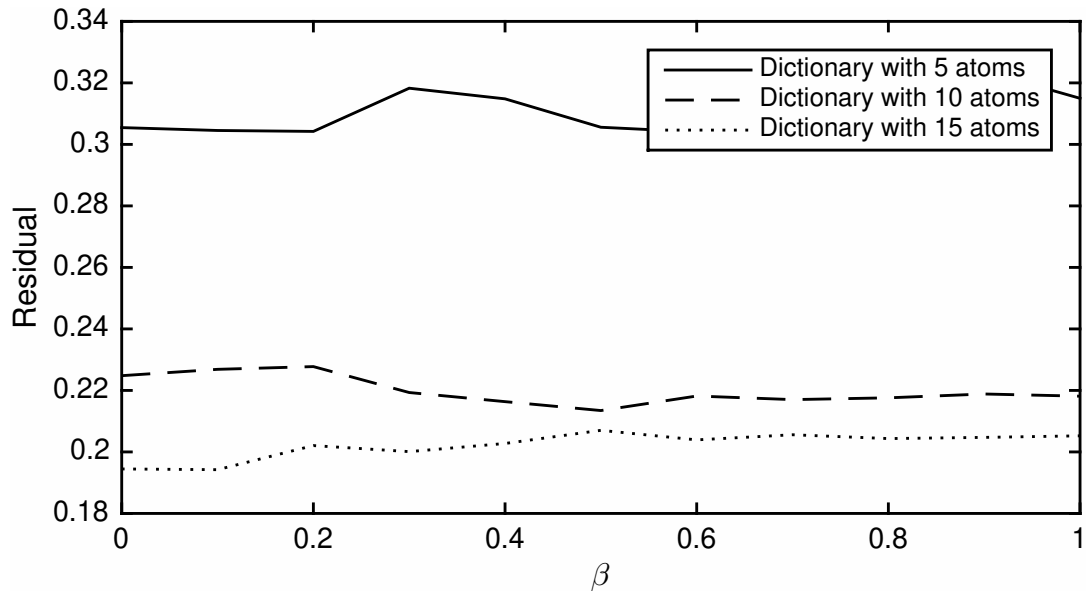


Figure 4.13: Residual vs  $\beta$  in Yale Face Data Set for 5 Classes

The mutual coherence of the atoms from the same dictionary should decrease to increase the discriminative power of the dictionaries, as a result of the increase in the number of classes. For this reason, the experiments with 5 classes reported in Figure 4.12 suggest that the increase in  $\beta$  reduces the misclassification rate unlike the experiments with 3 classes. Also, the misclassification rate and the residual increase with the increase in the classified classes as expected. As in Figure 4.11, Figure 4.13 also confirms the expectation that if  $\beta$  increases the residual also increases because of the increase in the atom dissimilarity causing a decrease in representation power.

The optimal beta values for different dictionary sizes with 5 classes are reported in Table 4.4.

Table 4.4: Selection of  $\beta$  for Different Dictionary Sizes on 5 Classes in Yale Face Data Set for 5 Classes

Number of Atoms	1	5	10	15
$\beta$	0,8	0,8	1	0,9
Misclassification Rate (%)	59,84	38,8	27,33	22,37

The visualizations of dictionaries in Figure 4.14 show the effect of  $\beta$  under with the experiments for 5 classes.

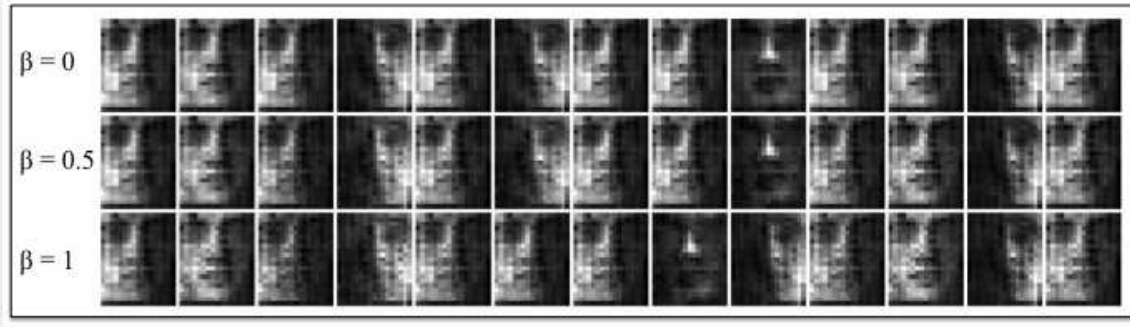


Figure 4.14: Atoms with the change of  $\beta$  in Yale Face Data Set

Figure 4.14 shows the decrease in the mutual coherence of atoms with the increase in  $\beta$  value which increases the power of incoherence term on the objective function in eq. (3.14) and confirms the conclusion arrived from Figure 4.12 and 4.13.

### 4.3 Comparison with Other Algorithms

After these optimizations of parameters in our algorithm, the proposed method is compared with reference supervised and unsupervised dictionary learning methods; namely, K-SVD [1], LC-KSVD [29] and TIDL [59] , with respect to the following performance measures:

- The classification performance is measured with the percentage of test samples misclassified with 1-sparse representations in the learnt dictionaries. The test sample is assigned the label of the atom which gives the minimum residual as in eq. (3.20).
- The representation performance is measured with the norm of the residual vector, which is the difference between the test signal and its best 1-sparse representation as formulated in eq. (3.19).

To summarize, the comparisons show the classification and representation performances of compared algorithms with 1-sparsity constraint. Using 1-sparsity reduces the

computational complexity and increases the speed of all algorithms so it helps to achieve the aim of our method which is fast and memory efficient classification.

The experiments are done with the setup described in Section 4.1. For the proposed method, we use the optimized values of  $\beta$  and  $\sigma$  given in the Section 4.2 to solve the eq. (3.14). In the training of the K-SVD, we set the number of iterations as 50, the sparsity of K-SVD,  $s$  as 40 in the eq. (2.22) which gives the optimum results for the data sets. An individual dictionary is computed for each class with K-SVD, by learning the dictionary with the training samples of that class. In training step of LC-KSVD1, we use  $\gamma_1$  as 0.002 and sparsity  $s$  as 40 for optimum results in eq. (2.27). Likewise for LC-KSVD2, we set  $\gamma_1$  as 0.002,  $\gamma_2$  as 0.004 and sparsity  $s$  as 40 for optimum results in eq. (2.28). For both LC-KSVD methods, the number of iterations is set to 50. To learn dictionaries with the TIDL method, we use the optimized  $\sigma$  value as used in the proposed method which gives the optimum results in eq. (3.6). As stated above, 1-sparsity is used as given in eq. (3.20) in all tests to measure the performance differences among these methods.

Please note that in all experiments in this section similar results are obtained from LC-KSVD1 and LC-KSVD2 so the plots in this section legends LC-KSVD1 as LC-KSVD for the simplicity of the plots.

### 4.3.1 Results in MNIST Data Set

The setup described in Section 4.1.1 is used with the optimized parameters for each method given in Section 4.3 for the experiments in this section. Figure 4.15 shows the classification performance of compared methods.

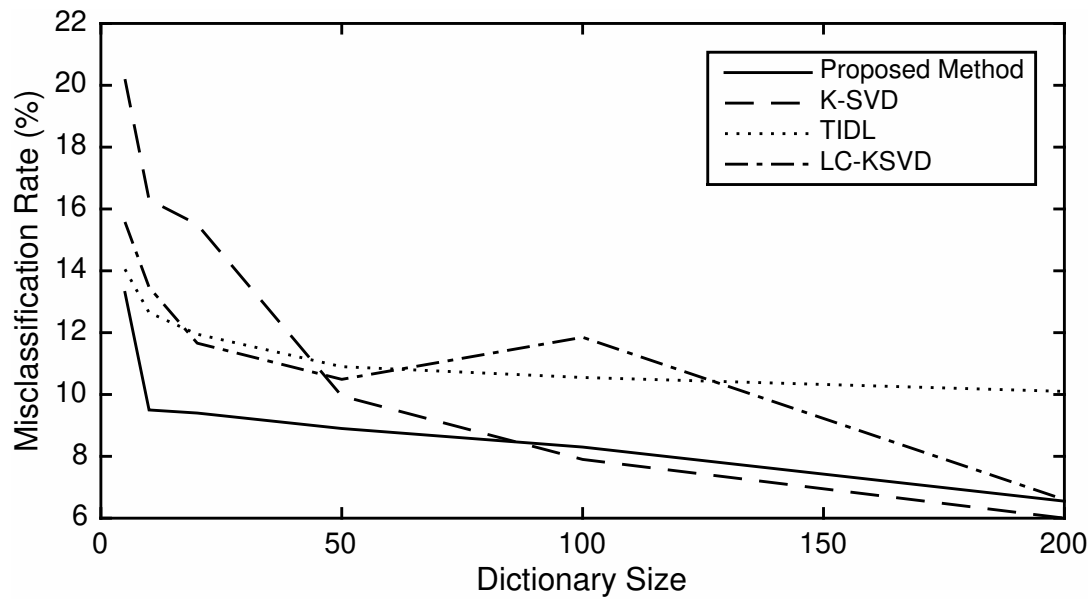


Figure 4.15: Misclassification Error (%) Comparisons of Algorithms in MNIST Data Set

The representation performance of compared methods is shown in Figure 4.16.

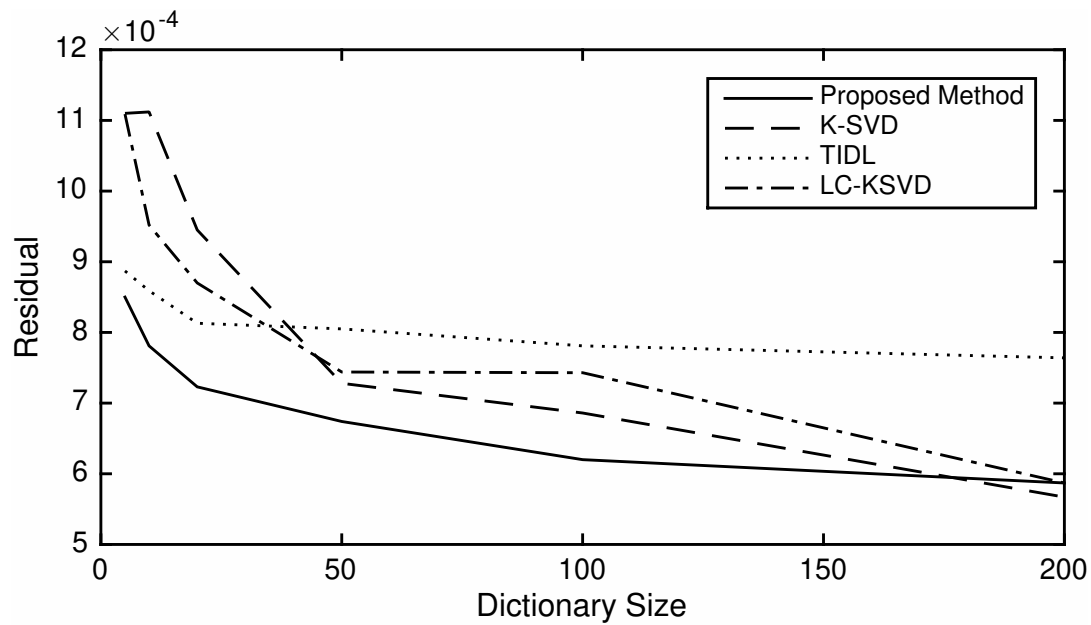


Figure 4.16: Residual Comparisons of Algorithms in MNIST Data Set

It is observed in Figures 4.15 and 4.16 that the classification performance and the representation accuracy increase with the size of the dictionaries in all compared algorithms. The proposed method and its preceding version TIDL have the lowest misclassification rate and residual for dictionaries having up to 20 atoms. Moreover, the proposed method continues the leading performance of classification and representation for larger dictionaries having up to 100 atoms. This leading performance of our method demonstrates the success of the ideas of training using different weighted training images and the incoherence term added to the objective of the TIDL method as observed in Figures 4.15 and 4.16.

The results of these experiments are given in Tables 4.5 and 4.6 in more detail.

Table 4.5: Misclassification Rate (%) Comparisons of Algorithms in MNIST Data Set

Dictionary Size	1	5	10	20	50	100	200
Proposed Method	20,50	<b>13,30</b>	<b>9,50</b>	<b>9,40</b>	<b>8,90</b>	8,30	6,55
K-SVD	20,10	20,20	16,30	15,50	9,95	<b>7,90</b>	<b>6,00</b>
TIDL	<b>18,6</b>	14,05	12,65	11,95	10,90	10,55	10,10
LC-KSVD1	29,15	15,58	13,47	11,66	10,49	11,85	6,4
LC-KSVD2	29,15	15,49	13,56	11,92	10,28	9,4	6,35

Table 4.6: Residual Comparisons of Algorithms in MNIST Data Set

Dictionary Size	1	5	10	20	50	100	200
Proposed Method	0,00147	<b>0,00085</b>	<b>0,00078</b>	<b>0,00072</b>	<b>0,00067</b>	<b>0,00062</b>	0,00059
K-SVD	0,00111	0,00111	0,00111	0,00095	0,00073	0,00069	<b>0,00057</b>
TIDL	<b>0,00108</b>	0,00089	0,00086	0,00081	0,00081	0,00078	0,00076
LC-KSVD1	0,00112	0,00111	0,00095	0,00087	0,00074	0,00074	0,00059
LC-KSVD2	0,00112	0,00111	0,00095	0,00084	0,00075	0,00072	<b>0,00057</b>

The leading classification performance of our algorithm stands for the lower dictionary sizes in the MNIST data set as seen in Table 4.5. This may help our algorithm to

be used in platforms with low storage capability because for dictionary sizes up to 100 for the MNIST data set, our algorithm outperforms the compared algorithms. However, for large dictionaries, the K-SVD method captures the lead and is followed by LC-KSVD, the proposed method and TIDL. This result can be explained by the visualization of some example atoms in Figures 4.17 and 4.19 which are taken from dictionaries of size 200. The K-SVD method learns more comprehensive dictionaries than our method and the advantage of these comprehensive dictionaries are observed more for larger dictionaries.

For the representation performance, all methods end with similar residual errors for large dictionaries but when the dictionary size is small, the LC-KSVD1 and LC-KSVD2 have the most accurate representations followed by the proposed algorithm, TIDL and K-SVD as seen in the Table 4.6.

10 sample atoms selected from 3 classes are shown in Figures 4.17 - 4.21, respectively for the proposed method, TIDL, K-SVD, LC-KSVD1 and LC-KSVD2.

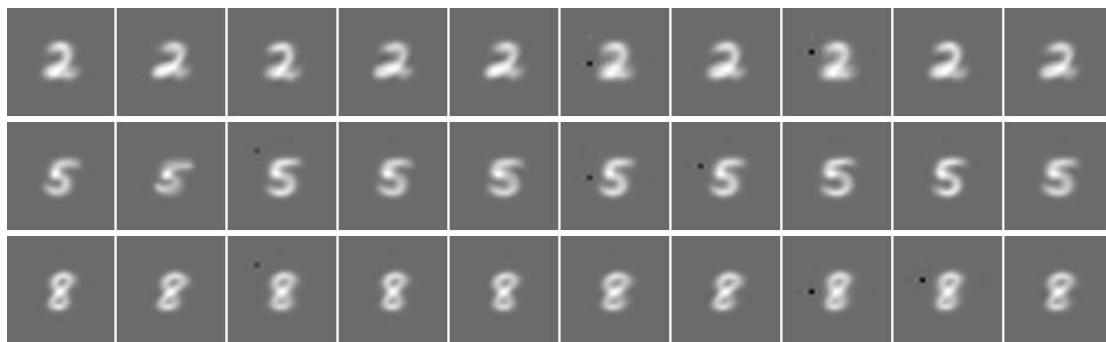


Figure 4.17: Dictionaries Learnt with Proposed Method in MNIST Data Set for 3 Classes





Figure 4.18: Dictionaries Learnt with TIDL in MNIST Data Set for 3 Classes



Figure 4.19: Dictionaries Learnt with K-SVD in MNIST Data Set for 3 Classes



Figure 4.20: Dictionaries Learnt with LC-KSVD1 in MNIST Data Set for 3 Classes



Figure 4.21: Dictionaries Learnt with LC-KSVD2 in MNIST Data Set for 3 Classes

Figure 4.19 shows that the K-SVD method learns a representative dictionary with atoms that rather look like the training images belonging to the same class. It can be observed in Figures 4.20 and 4.21, that LC-KSVD1 and LC-KSVD2 learn atoms that also have dissimilarities from the training images belonging to other classes. This causes the LC-KSVD methods to have better classification performance than K-SVD as shown in Table 4.5. The TIDL method learns dictionaries having atoms different from each other which also bear dissimilarities from the training images of the other classes. This causes the TIDL method to have better classification performance than K-SVD. Although the proposed method learns atoms that looks similar to each other, these atoms are particularly adapted to the setting of 1-sparse representations with the help strategies such as including all training images in the learning of each atom. Consequently, the proposed algorithm outperforms the others in most settings when 1-sparse representations are used as observed in Table 4.6. It also yields the best classification performance over the compared algorithms until the size of the dictionary exceeds about 100 atoms. The LC-KSVD1 and LC-KSVD2 methods have the lowest misclassification rates for larger dictionary sizes due to the dissimilarities of the atoms in the dictionaries.

### 4.3.2 Results in Yale Face Data Set

We experiment with the setup in Section 4.1.2 for 3 and 5 randomly selected classes, correspondingly. For each experiment, 100 Monte Carlo trials are done with the random selection of classes and the training images of corresponding classes. Each method uses their optimized parameters given in Section 4.3.

- **Experiments on 3 different classes:**

Figure 4.22 shows the misclassification results of all methods for 3 different classes in the Yale Face data set.

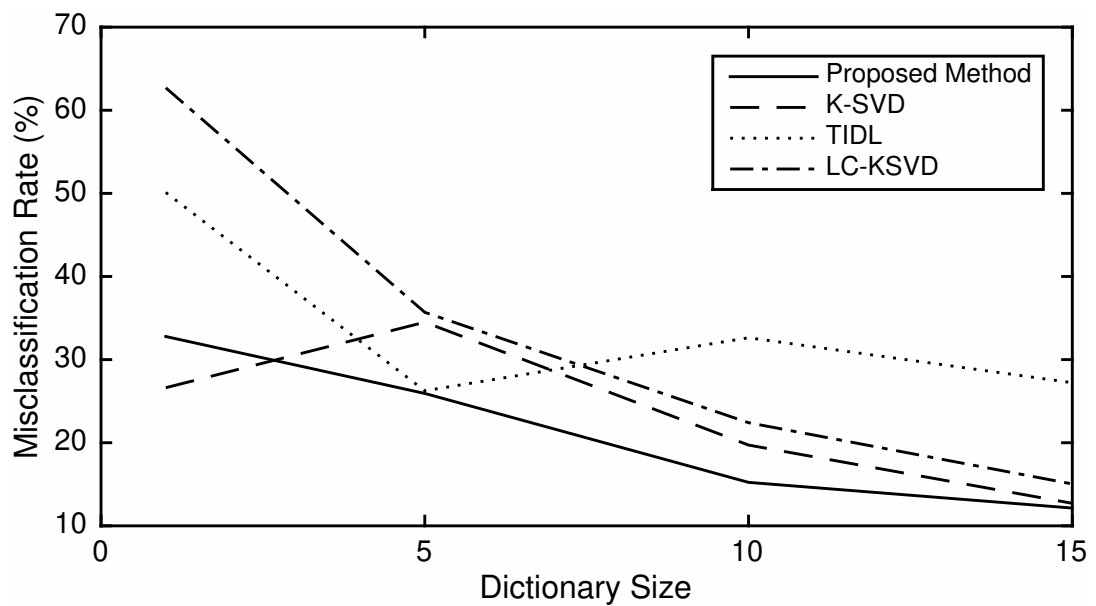


Figure 4.22: Misclassification Error (%) Comparisons of Algorithms in Yale Face Data Set for 3 Classes

For the experiments on 3 classes from Yale data set, Table 4.22 shows that the classification performance of the proposed algorithm leads the compared algorithms. The KSVD, LC-KSVD2, LC-KSVD1, and TIDL methods follow our algorithm, respectively. Consequently, the aim of fast and efficient classification of our algorithm is achieved for this experiment. K-SVD has the second lowest misclassification rate because it performs well when the number of classes is low.

Figure 4.23 shows the residual results of all methods for 3 different classes in the Yale Face data set. In terms of the accuracy of 1-sparse representations, K-SVD is the leading algorithm followed by LC-KSVD1, LC-KSVD2, the proposed algorithm and TIDL, respectively as seen in Figure 4.23. It is an expected result because the K-SVD method owes its popularity to its representation power and LC-KSVD is a supervised and extended version of K-SVD. The proposed method performs better than the TIDL method for all dictionary sizes because it incorporates more information in the learning of the atoms by using all training images with different weights, unlike the TIDL method.

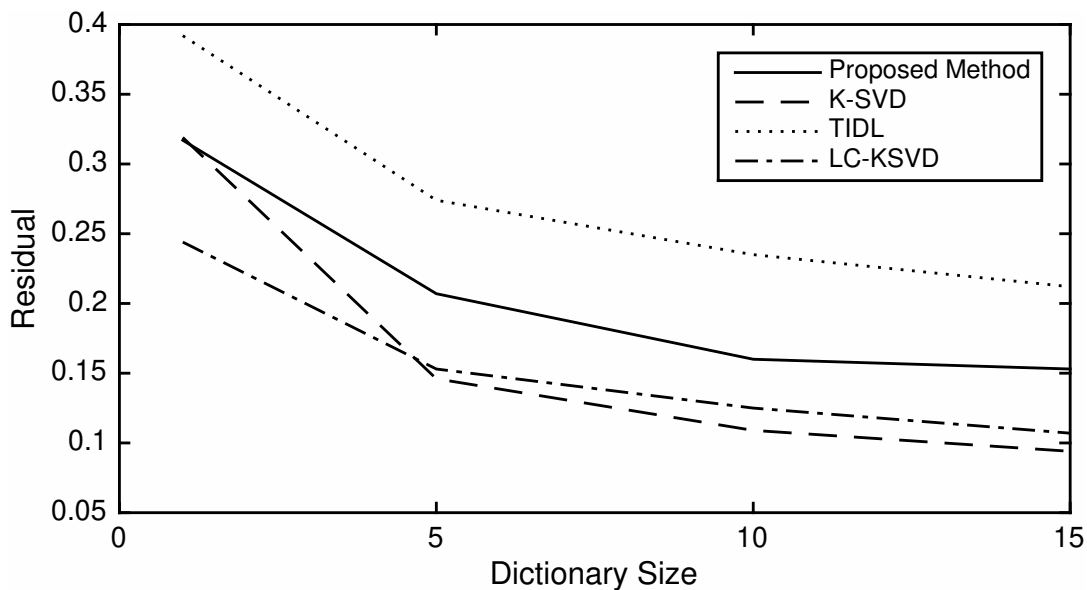


Figure 4.23: Residual Comparisons of Algorithms in Yale Face Data Set for 3 Classes

The experimental results of this section are also given in Tables 4.7 and 4.8.

As seen in Table 4.7, the proposed algorithm outperforms the compared algorithms in terms of classification performance. The experiments done for dictionaries having 10 atoms, the proposed method is 4% better than the KSVD which is the runner-up algorithm with respect to the misclassification rate. However, this big difference among the algorithms is not valid for the experiments with dictionaries having 15 atoms. For this experiment, all of the algorithms except the TIDL method performs similarly in terms of classification. The TIDL method is powerful for the experiments

conducted with dictionary sizes of 5. As stated in [59], the TIDL method shows its classification ability for lower dictionary sizes which is also seen in Table 4.7.

Table 4.7: Misclassification Rates (%) on 3 Classes in Yale Face Data Set

Dictionary Size	1	5	10	15
Proposed Method	32,78	<b>25,93</b>	<b>15,23</b>	<b>12,14</b>
KSVD	<b>26,62</b>	34,50	19,72	12,71
TIDL	50,11	26,25	32,63	27,22
LC-KSVD1	62,70	35,69	22,41	15,04
LC-KSVD2	62,70	35,63	22,43	14,41

Table 4.8: Residual Error on 3 Classes in Yale Face Data Set

Dictionary Size	1	5	10	15
Proposed Method	0,317	0,205	0,172	0,161
KSVD	0,319	<b>0,142</b>	<b>0,110</b>	<b>0,095</b>
TIDL	0,333	0,239	0,193	0,180
LC-KSVD1	0,255	0,150	0,117	0,101
LC-KSVD2	<b>0,254</b>	0,150	0,118	0,101

Being one of the most powerful dictionary learning algorithms in terms of sparse representation, the K-SVD method takes the lead when the 1-sparse representation accuracies are studied as seen in Table 4.8. K-SVD is followed by LC-KSVD, the proposed method and TIDL, respectively in terms of approximation accuracy.

Here we display some atoms from the dictionaries having 10 atoms learnt from 3 classes of the Yale Face data set.

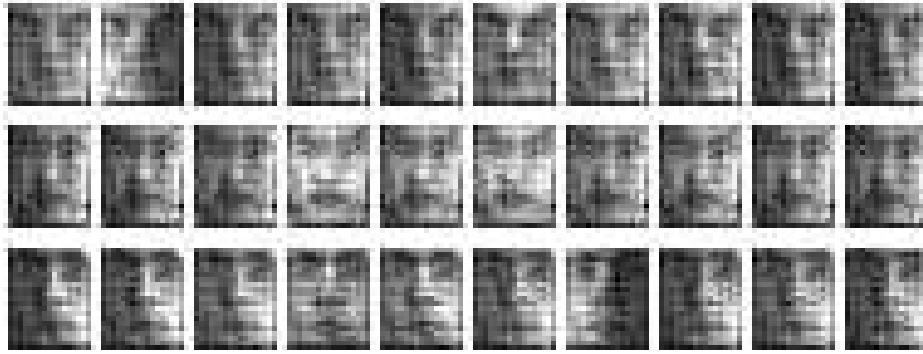


Figure 4.24: Dictionaries Learnt with Our Method in Yale Face Data Set for 3 Classes

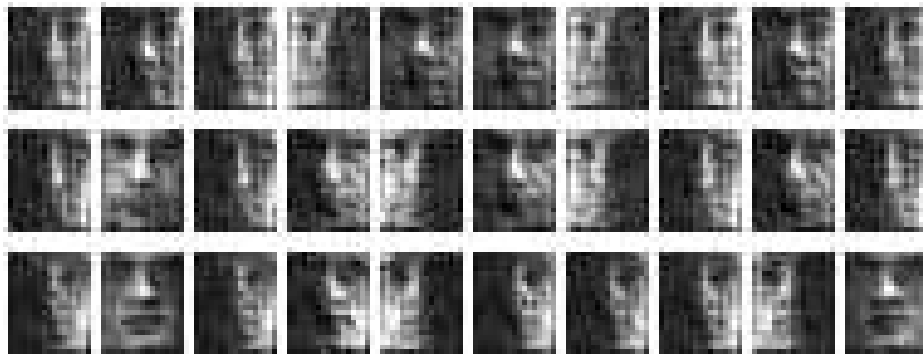


Figure 4.25: Dictionaries Learnt with TIDL in Yale Face Data Set for 3 Classes



Figure 4.26: Dictionaries Learnt with K-SVD in Yale Face Data Set for 3 Classes



Figure 4.27: Dictionaries Learnt with LC-KSVD1 in Yale Face Data Set for 3 Classes



Figure 4.28: Dictionaries Learnt with LC-KSVD2 in Yale Face Data Set for 3 Classes

It is surprising that K-SVD outperforms the LC-KSVD methods which are the extended and supervised version of K-SVD in classification. The cause of this unexpected result is related to the learnt atoms of these method seen in Figures 4.26, 4.27 and 4.27. The visual examination of the dictionaries shows that the atoms learnt by the K-SVD method have more class-specific characteristics than the ones learnt by the LC-KSVD1 and LC-KSVD2 methods. The atoms learnt with the proposed method also have the same characteristic information as seen in Figure 4.24. Atoms learnt by the proposed method also have some features from the other classes which results in the increase in the classification performance but the decrease in the accuracy of representation. Although the TIDL method learns some atoms having both characteristics

from the training images labeled with the same class and the other classes, some atoms learnt by TIDL seem disrupted that may be caused by unsuccessful atom initialization and finding index sets corresponding to this initialization.

- **Experiments on 5 different classes:**

The 1-sparse classification and representation performances of the compared methods for randomly selected 5 different classes in the Yale Face data set can be seen in Figures 4.29 and 4.30, respectively.

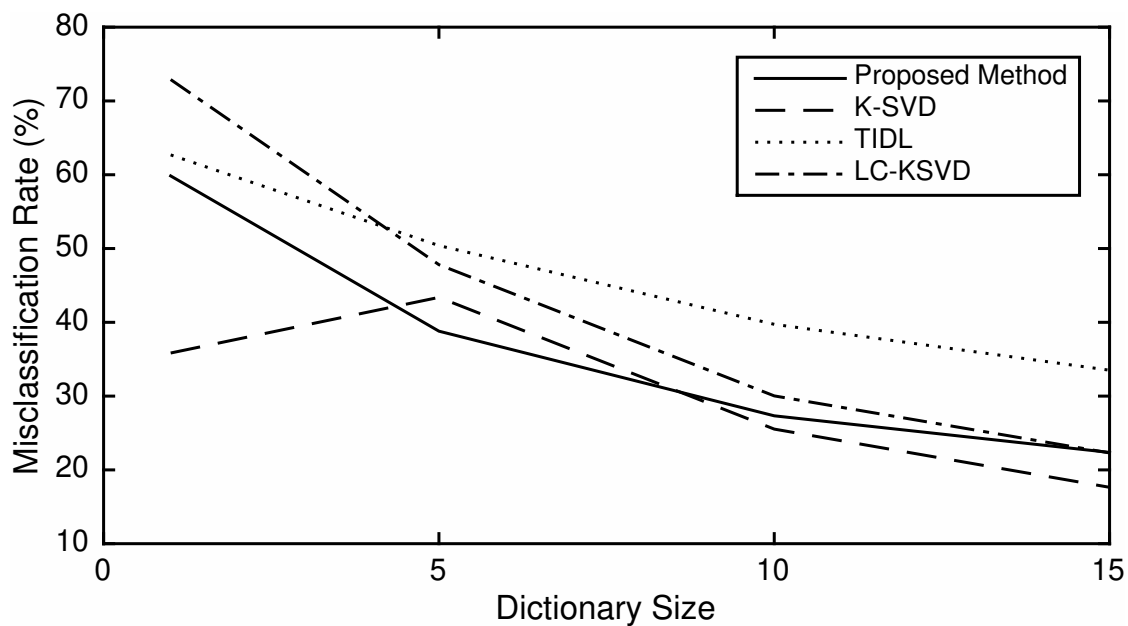


Figure 4.29: Misclassification Error (%) Comparisons of Algorithms in Yale Face Data Set for 5 Classes



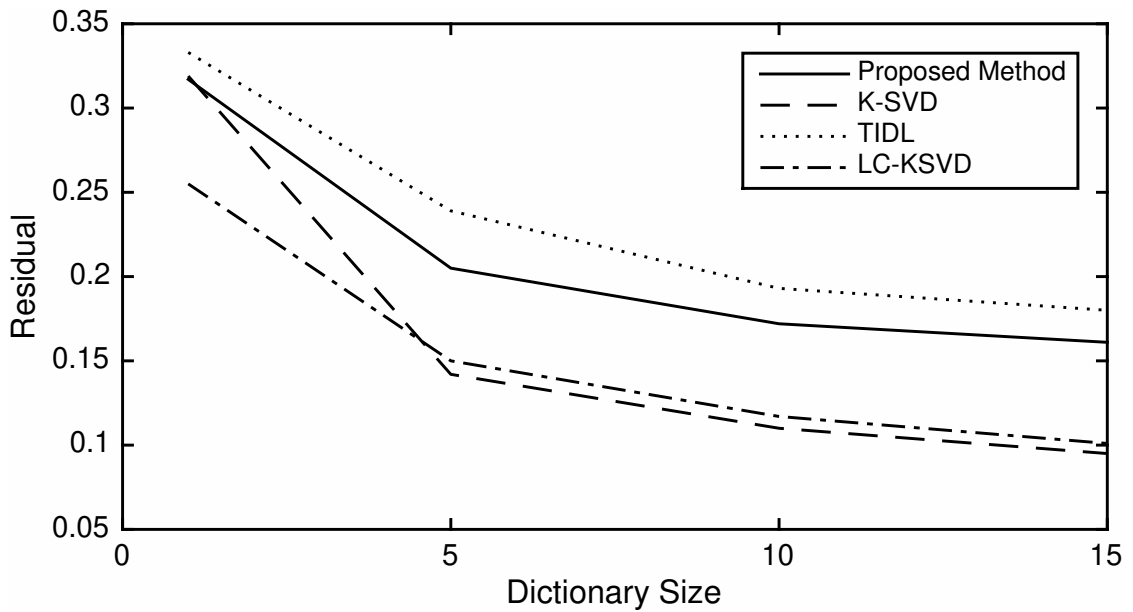


Figure 4.30: Residual Comparisons of Algorithms in Yale Face Data Set for 5 Classes

There are some differences among Figure 4.22 and Figure 4.29. First of all, the misclassification rates are higher in this setup because of the increase in the number of classes. This is an expected result because the difficulty of classifying more classes is directly proportional to the number of classes which can be inferred from eq. (3.20). Secondly, the proposed method is observed not to outperform the others in this setup with 5 classes. Although it has better results than its preceding version TIDL for all dictionary sizes, this result shows that the improvement of its performance for more classes can be a future work. Thirdly, the K-SVD method outperforms all compared methods including the proposed method.

Figure 4.30 presents similar results to those of Figure 4.23 so it can be concluded that the change in the number of classes does not change the representation performance. Also, the same conclusions related to the residual drawn from the experiments with 3 classes on Yale Face data set are valid for this experiment.

Table 4.9: Misclassification Rates (%) on 5 Classes in Yale Face Data Set

Dictionary Size	1	5	10	15
Proposed Method	59,84	<b>38,80</b>	27,33	22,37
K-SVD	<b>35,84</b>	43,38	<b>25,53</b>	<b>17,64</b>
TIDL	62,69	50,40	39,73	33,51
LC-KSVD1	72,90	47,81	30,03	22,30
LC-KSVD2	73,06	47,50	30,06	22,37

Table 4.10: Residual Error on 5 Classes in Yale Face Data Set

Dictionary Size	1	5	10	15
Proposed Method	0,317	0,207	0,160	0,153
K-SVD	0,319	<b>0,146</b>	<b>0,109</b>	<b>0,094</b>
TIDL	0,392	0,274	0,235	0,212
LC-KSVD1	<b>0,244</b>	0,153	0,125	0,107
LC-KSVD2	<b>0,244</b>	0,152	0,125	0,108

In terms of 1-sparse representations, it can be inferred from Table 4.10 that K-SVD has the best performance as in Table 4.8. K-SVD owes the success behind its classification performance to its representation power. The proposed method cannot outperform K-SVD in this experiment. The proposed method cannot maintain its success with increasing class number due to the usage of all training images in the training step.

The dictionaries having 10 atoms learnt with 5 classes from the Yale Face data set are displayed in Figures 4.31 - 4.35.



Figure 4.31: Dictionaries Learnt with Proposed Method in Yale Face Data Set for 5 Classes



Figure 4.32: Dictionaries Learnt with TIDL in Yale Face Data Set for 5 Classes



Figure 4.33: Dictionaries Learnt with K-SVD in Yale Face Data Set for 5 Classes



Figure 4.34: Dictionaries Learnt with LC-KSVD1 in Yale Face Data Set for 5 Classes



Figure 4.35: Dictionaries Learnt with LC-KSVD2 in Yale Face Data Set for 5 Classes

From Figures 4.31 - 4.35, it is inferred that the K-SVD method learns the most comprehensive dictionary among all methods so that it takes the lead in both the classification and the representation performance for this experiment. The LC-KSVD1 and LC-KSVD2 methods learn similar dictionaries so they result in similar misclassification rates. The proposed method outperforms the TIDL method in terms of the classification performance and the accuracy of representations as it learns more diverse dictionaries.



## CHAPTER 5

### CONCLUSION

In this thesis, we proposed a supervised dictionary learning method for fast and efficient classification of test samples. Particularly focusing on applications over platforms with limited memory and computation resources, we have aimed to develop a dictionary learning algorithm that minimizes the computational complexity of the classification of test images. We have adopted an analytical representation of supervised dictionaries over a two-dimensional Hermite basis in order to decrease the computational complexity and the need for memory. In order to speed up the classification of test samples, we have learnt the dictionaries in a way to allow an accurate classification of test samples with 1-sparse representations. Also, when learning each atom, we use all of the training images with different coefficients based on their distances to the learnt dictionary atom to increase the variability between the atoms. We have also used an incoherence term in our learning objective that discourages the similarity between the dictionary atoms from the same classes.

This thesis starts with a brief review of sparse representations and dictionary learning concepts. In Chapter 2, we have given an overview of the literature with some examples of related supervised and unsupervised dictionary learning algorithms and also on some sparse representation methods. In Chapter 3, the proposed method has been explained in detail. We have started this chapter with a review of the TIDL algorithm that has inspired our method, with the description of the proposed algorithm by specifying its common and different points with respect to TIDL. Modifications on the training sample selection and weighting strategies and the addition of an incoherence term for atom variability have also been widely discussed in the same chapter. In Chapter 4, we have given the results of the experiments conducted on

several data sets in order to measure the classification and representation performance of the proposed method. We have compared our method with the K-SVD [1], LC-KSVD [29] and TIDL [59] methods for classification with 1-sparse signal representations.

The experiments on a simplified version of our method without the incoherence term show that training with all training images with different weights performs better than the hard selection of training images via index sets as done in the TIDL method. Also, the comparison of the results obtained with and without the incoherence term in our method shows the effectiveness of the incoherence term. For example, the experiments conducted in the MNIST data set with the dictionaries having 10 atoms shows that TIDL has 12.65%, the simplified method has 11% and finally the proposed method has 9.5% misclassification rates which are inferred from Tables 4.1 and 4.5. Please note that the experiments for the proposed method use the optimized weights of the incoherence term which is also needed for these successful results.

Table 5.1: Computational Complexities of Classification Algorithms

Algorithm	Complexity
Proposed Method	$O(MN)$
Nearest Neighbor	$O(K)$
Support Vector Machine	$O(M)$

The computational complexities of classification algorithms when calculating a 1-sparse representation of a test signal in  $M$  class-specific dictionaries of each having  $N$  atoms and the number of training samples of all classes  $K$  is given in Table 5.1. The proposed method estimates the class label of a given test image  $y$  with the eq. (3.20). Thus, the computational complexity of the proposed method on test step can be expressed with  $O(MN)$  where  $M$  is the class number and each class-specific dictionary have  $N$  atoms. The Nearest Neighbor [14] approach searches the minimum distance between the test image  $y$  and the training sample  $y_i, \forall i \in \{1, 2, \dots, K\}$ . NN estimates the class-label of test image with the label of the training image satisfying  $\arg \min \|y - y_i\| \forall i \in \{1, 2, \dots, K\}$ . For this reason, NN computes every distance between test image and all training samples. Thus, it has a test time complexity of  $O(K)$  where  $K$  is the number of all training samples. Support Vector Machine



[13] defines boundaries for each class  $m$  to separate it from the other classes. SVM estimates the class label of test image  $y$  with  $w_m^T y + b_m, \forall m \in \{1, 2, \dots, M\}$  where  $M$  is the class number and  $w_m$  and  $b_m$  define the boundary of class  $m$ . Thus, the computational complexity of SVM is  $O(M)$ . The Support Vector Machine method only depends on the class number. For small dictionaries, the proposed method has complexity getting closer to the complexity of SVM. The design aim of the proposed algorithm which is applicability over the platforms having critical resources is supported with this conclusion. Moreover, training with large training sets limits the usage of Nearest Neighbor unlike the proposed method.

Experiments are conducted with the comparison of classification and representation performances of our method with different algorithms using the 1-sparsity constraint. 1-sparsity speeds up the test stage of all algorithms and reduces their memory need. Thus, the experiments show the performance of compared algorithms for the applications where memory and computational resources are limited.

The experiments with large dictionary sizes can be conducted with the MNIST data set. The results of these experiments show that the proposed method has considerably better classification results at small dictionary sizes. Consequently, the proposed method has benefits for applications with particular restrictions on the dictionary size and demands for fast classification of test images.

The classification experiments with different number of classes in the Yale Face data set shows that the classification performance of all algorithms decreases with the increase in the number of classes. However, these experiments also show that the representation performance of the algorithms is not as affected as the classification performance.

The training images are more similar to each other in the MNIST data set compared to the Yale Face data set. This causes the learnt atoms to be more similar atoms in the MNIST data set. Hence, the need for an incoherence term is smaller in the Yale Face data set because of having more diverse training images. Since our method brings an innovation to the TIDL method through the incoherence term whose success is shown by the experiments, our method is more favorable among the compared methods in the MNIST data set than the Yale data set due to the need of atom incoherence.

In the proposed method, some weight parameters are set experimentally. To begin with, the Gaussian Kernel is used for setting the weight parameters of training images in the training step of our algorithm. An important parameter to tune for setting the weights is then the kernel scale. We tuned the kernel scale empirically. The tuning of the kernel scale with more developed strategies might be the first of future works. Also, the incoherence parameter is optimized experimentally in our work. The second future work can be the automatic selection of the incoherence parameter. The incoherence parameter should be optimized in accordance with the dictionary size, the number of classes and the data set. In addition to the automatic optimization of the weight parameters, there are still some room for improving this work like enhancing its performance for a large number of classes, the addition of some terms to increase the classification rate in large dictionaries. Also, using an analytical basis would facilitate the extension of our algorithm to be transformation-invariant like TIDL. However, the transformation invariance of our algorithm is not examined in this work. Thus, improvements for the geometric invariance can be another future work. Moreover, basis selection can be optimized to increase speed and efficiency of the algorithm because with an optimized selection of basis, signals of interest can be represented with fewer coefficients which reduces the complexity and run time of the algorithm. Also, it may increase the classification rate. Lastly, comparisons with the baseline classification algorithms like Support Vector Machine and Nearest Neighbor might be another future work.

To conclude, the experimental results show that, in comparison with the TIDL algorithm, the proposed modifications over the selection of the training sample weights and the introduction of the incoherence term have allowed the proposed method to achieve successful classification performance. The comparisons with reference methods also show that our method achieves its fast and efficient classification purpose as it often gives more accurate results than the other methods in comparison, especially at small dictionary sizes with 1-sparse representations. Consequently, the potential of our algorithm for usage on platforms with low storage and computational capability is demonstrated.

## REFERENCES

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 54(11):4311–4322, Nov. 2006.
- [2] C. Bao, Y. Quan, and H. Ji. A convergent incoherent dictionary learning algorithm for sparse coding. In *ECCV*, 2014.
- [3] D. Barchiesi and M. D. Plumbley. Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. *Trans. Sig. Proc.*, 61(8):2055–2065, Apr. 2013.
- [4] W. J. Beksi and N. Papanikolopoulos. Object classification using dictionary learning and rgb-d covariance descriptors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1880–1885, May 2015.
- [5] O. Bryt and M. Elad. Compression of facial images using the k-svd algorithm. *J. Vis. Comun. Image Represent.*, 19(4):270–282, May 2008.
- [6] G. Bull, J. Gao, and M. Antolovich. Image segmentation using dictionary learning and compressed random features. In *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, Nov 2014.
- [7] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 155–168, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [8] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research - Proceedings Track*, 14:1–24, 2011.
- [9] S. Chen and D. Donoho. Basis pursuit. Technical report, Stanford University, 1994.

- [10] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit, 1995.
- [11] Y. Chen and J. Su. Sparse embedded dictionary learning on face recognition. *Pattern Recognition*, 64:51 – 59, 2017.
- [12] R. R. Coifman and M. V. Wickerhauser. Adapted waveform analysis as a tool for modeling, feature extraction, and denoising. *Optical Engineering*, 33(7):2170–2174, July 1994. Special issue on Adapted Wavelet Analysis.
- [13] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.
- [14] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, Sept. 2006.
- [15] W. Dai, Y. Shen, X. Tang, J. Zou, H. Xiong, and C. W. Chen. Sparse representation with spatio-temporal online dictionary learning for promising video coding. *IEEE Transactions on Image Processing*, 25(10):4580–4595, Oct 2016.
- [16] T. A. Davis and Y. Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38:1:1–1:25, 2011.
- [17] I. Diamant, E. Klang, M. Amitai, E. Konen, J. Goldberger, and H. Greenspan. Task-driven dictionary learning based on mutual information for medical image classification. *IEEE Transactions on Biomedical Engineering*, 64(6):1380–1392, June 2017.
- [18] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec 2006.
- [19] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.
- [20] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proceedings of the Acoustics, Speech, and Signal Processing*,

1999. *On 1999 IEEE International Conference - Volume 05, ICASSP '99*, pages 2443–2446, Washington, DC, USA, 1999. IEEE Computer Society.
- [21] N. Eslahi, A. Aghagolzadeh, and S. M. H. Andargoli. Compressive video sensing via dictionary learning and forward prediction. *CoRR*, abs/1508.07640, 2015.
- [22] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, Jun 2001.
- [23] P. R. Gill, A. Wang, and A. Molnar. The in-crowd algorithm for fast basis pursuit denoising. *IEEE Transactions on Signal Processing*, 59(10):4595–4605, Oct 2011.
- [24] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *Trans. Sig. Proc.*, 45(3):600–616, Mar. 1997.
- [25] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for  $l_1$ -minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.
- [26] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12*, pages 1183–1194, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.
- [27] E. J. C and D. Donoho. Curvelets - a surprisingly effective nonadaptive representation for objects with edges. 04 2000.
- [28] R. Jiang, H. Qiao, and B. Zhang. Efficient fisher discrimination dictionary learning. *Signal Processing*, 128:28 – 39, 2016.
- [29] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11*, pages 1697–1704, Washington, DC, USA, 2011. IEEE Computer Society.

- [30] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.
- [31] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.
- [32] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, April 2012.
- [33] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [34] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *CoRR*, abs/0809.3083, 2008.
- [35] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008.
- [36] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [37] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, Dec 1993.
- [38] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, Apr. 1995.
- [39] N. M. Nayak and A. K. Roy-Chowdhury. Learning a sparse dictionary of video structure for activity modeling. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4892–4896, Oct 2014.
- [40] E. L. Pennec and S. Mallat. Bandelet image approximation and compression. *SIAM JOURNAL OF MULTISCALE MODELING AND SIMULATION*, 4:2005, 2005.
- [41] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In

- 2010 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508, June 2010.
- [42] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
- [43] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. 2008.
- [44] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [45] I. Tomic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, March 2011.
- [46] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theor.*, 53(12):4655–4666, Dec. 2007.
- [47] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.
- [48] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.
- [49] J. Wang, L. Zhuang, and N. Yu. Online shared dictionary learning for visual tracking. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, ICIMCS '15, pages 22:1–22:5, New York, NY, USA, 2015. ACM.
- [50] D. P. Wipf and B. D. Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, pages 2153–2164, 2004.
- [51] A. W̄Ensche. Hermite and laguerre 2d polynomials. *Journal of Computational and Applied Mathematics*, 133(1):665 – 678, 2001. 5th Int. Symp. on Orthogonal Polynomials, Special Functions and their Applications.
- [52] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, Feb. 2009.

- [53] Y. Xie, C. Huang, T. Song, J. Ma, and J. Jing. Object co-detection via low-rank and sparse representation dictionary learning. In *2013 Visual Communications and Image Processing (VCIP)*, pages 1–6, Nov 2013.
- [54] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang. Low-dose x-ray ct reconstruction via dictionary learning. *IEEE Transactions on Medical Imaging*, 31(9):1682–1697, Sept 2012.
- [55] Y. Xu, Z. Li, J. Yang, and D. Zhang. A survey of dictionary learning algorithms for face recognition. *IEEE Access*, 5:8502–8514, 2017.
- [56] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *Trans. Img. Proc.*, 19(11):2861–2873, Nov. 2010.
- [57] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 543–550, Washington, DC, USA, 2011. IEEE Computer Society.
- [58] Y. Yankelevsky and M. Elad. Structure-aware classification using supervised dictionary learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4421–4425, March 2017.
- [59] A. C. Yuzuguler, E. Vural, and P. Frossard. Transformation-invariant dictionary learning for classification with 1-sparse representations. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3562–3566, May 2014.
- [60] G. Zhang, H. Sun, Z. Ji, Y.-H. Yuan, and Q. Sun. Cost-sensitive dictionary learning for face recognition. *Pattern Recognition*, 60:613 – 629, 2016.
- [61] J. Zhang, Q. Li, R. J. Caselli, J. Ye, and Y. Wang. Multi-task dictionary learning based convolutional neural network for computer aided diagnosis with longitudinal images. *CoRR*, abs/1709.00042, 2017.
- [62] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698, June 2010.



- [63] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144, Jan 2012.
- [64] N. Zhou, Y. Shen, J. Peng, and J. Fan. Learning inter-related visual dictionary for object recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3490–3497, June 2012.
- [65] P. Zhou, C. Zhang, and Z. Lin. Bilevel model-based discriminative dictionary learning for recognition. *IEEE Transactions on Image Processing*, 26(3):1173–1187, March 2017.