SPARSE REPRESENTATION BASED CLASSIFICATION: TOWARDS EFFICIENCY AND ACCURACY

by

SOHEIL SHAFIEE

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2015

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Farhad Kamangar for his thoughtful ideas, comprehensive knowledge regarding my research, constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies at UTA. I wish to thank members of The Vision-Learning-Mining Research Lab (VLM) and specifically, Dr. Vassilis Athitsos for his positive attitude, generous supports and encouragements during my research and serving in my Ph.D. dissertation committee. I would also thank one of my best teachers, Dr. Manfred Huber, and my academic adviser, Dr. Jean Gao for their interest in my research and for taking time to serve in my dissertation committee.

I am grateful to all the teachers who taught me during the years I spent in school, first in Iran and then in the Unites States. I would also like to express my deep gratitude to all my friends and fellow graduate students who always helped and supported me back in Iran and in the United States.

I would especially thank my amazing family for their constant support and love. In particular, my parents, Jafar and Saeideh and my dear sister Sanaz. Finally, to my lovely wife, Laleh, thank you for your support and constant encouragement during my graduate studies and for always being there for me. There is no doubt that this work could not have been done without you.

March 18, 2015

ABSTRACT

SPARSE REPRESENTATION BASED CLASSIFICATION: TOWARDS EFFICIENCY AND ACCURACY

Soheil Shafiee, Ph.D.

The University of Texas at Arlington, 2015

Supervising Professor: Farhad Kamangar

With the fast growing deployment of machine intelligence in several real-life applications, there are always increasing needs for faster and more precise machine learning algorithms, especially classification and object recognition. One of the most recent methods proposed for this purpose is Sparse Representation-based Classification (SRC) which works based on the emerging theory of Compressive Sensing. SRC shows excellent classification results in comparison to many well-known classification approaches. However, despite its high recognition power, SRC suffers from high computational and memory costs as it directly uses all original ground truth data as representatives to build its training model. Given high recognition rates of SRC, it becomes important to reduce the time and memory requirements of this method while preserving its accuracy. These improvements help SRC to be a more practical solution especially to be used on portable devices. This research investigates different representative reduction approaches in the SRC context on multiple heterogeneous datasets and proposes a training model to be used along with SRC by using fewer but more informative representatives for the training space. We also investigate how incorporating multiple modalities of the data helps to improve SRC outcomes by extending efficient SRC implementations to multi-modality schemes and introducing three different approaches for this purpose. Experimental results show the proposed methods not only perform faster, but they also improve the classification accuracy on different datasets.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Classification Problem

Machine learning is a scientific approach which makes computers able to gain information to perform a specific task such as classification. Learning is a data-driven process which gains information from some available data and *learns* how to discriminate or predict the behavior of the data. Learning is categorized into *un-supervised* and *supervised* learning. In un-supervised learning, training data are not associated with labels and instead, their statistical properties help the system to gain knowledge over the data. The most common example of un-supervised learning are clustering algorithms [1] such as k-means [2], k-medoids [3], etc. On the other side, supervised learning benefits from labeled data (annotated measurements or ground truth) in the training phase which helps for more accurate and efficient learning in comparison to un-supervised learning. Classification is one of the most challenging problems in the category of supervised machine learning and it is defined as the process of assigning a class label to an unknown test sample. When computers solve the classification problems, they will be able to simulate human recognition system which leads to the optimal goal of automation and machine intelligence. A large number of methods and algorithms have been presented to solve the classification problem and they are becoming more accurate and applicable in different real-life applications. Some examples of these applications are object recognition such as face

[4–7] and character recognition [8–10], signal classification such as voice recognition [11, 12] and biomedical signals classification [13, 14], etc.

Classification algorithms use data representation approaches or *features* to characterize their objects of interest. In supervised classification, these features are extracted from a set of known samples are pre-labeled by an automatic labeling algorithm or an expert to form the training samples which are used to train the classifier in the next phase. When training data is sufficiently available and well-distributed, simple classifiers such as Nearest Neighbor (NN) [15] can classify an unknown test sample by simply assigning it to the class of its nearest neighbor. To reduce the error, the classification process can be performed by assigning a test sample to the class of majority of its k nearest neighbors (k-NN). At a higher level, Nearest Subspace classifiers, NS or k-NS, identify an unknown test sample by assigning it to its nearest subspace or k subspaces, respectively. A subspace defined as a low-dimensional space on which samples of a single class lie. Several learning algorithms, such as Neural Networks [16], Decision Trees [17], and Support Vector Machines (SVM) [18] are also shown to be effective when employed as classifiers.

1.2 Sparse Representation-based Classification

Recently a classification approach is presented and shown to be effective in many applications specially face recognition. Wright et al. proposed this method in 2009 and called it *Sparse Representation-based Classification* (SRC) [7] and reported interesting results in automatic face recognition application. The main idea of SRC derived from the assumption that in the face recognition domain, a particular sample face from a specific person can be represented as a linear combination of the other samples from the same person. Considering all the training samples from all the classes, this representation is naturally sparse, i.e. only a few face samples involve in reconstruction of the test sample. While this assumption holds, the problem of classification is converted into the recovery problem of a sparse vector which is defined as a vector with a few non-zero entries.

The emerging theory of *Compressive Sensing* (CS) [19,20] proves that it is possible to recover a sparse signal from only a few measurements. For this purpose, the signal and the measurement transform must satisfy certain conditions. It has been shown that when these conditions are satisfied, the original signal can be efficiently recovered via an ℓ^1-norm minimization process. In the context of classification using SRC, the unknown object can be identified by recovering a coefficient vector which represents that object as a linear combination of other objects from multiple classes. When multiple classes along with several samples are available for each class, this coefficient vector will be in the form of a sparse signal which may be recovered according to CS theory.

1.3 Contributions

In this thesis, we challenged SRC algorithm from two standpoint of efficiency and accuracy. Despite the high recognition rates Wright et al. reported for SRC, it suffers from time and space complexity. The fastest solution for ℓ^1-norm optimization problem which is utilized by SRC is shown to be of quadratic time complexity in the dimension of the sparse vector to be recovered. In other words, when the coefficient vector doubled in length, the time needed to solve the optimization problem quadrupled. In Chapter 2 we will show that the length of the coefficient vector to be recovered is equal to the number of columns in the SRC training matrix. Since the original SRC algorithm uses the training samples directly to form its training matrix columns, given large datasets along with SRC quadratic time complexity, the classification process gets expensive in terms of execution time.

A number of approaches have been proposed so far to increase the speed of the classification task using SRC. These methods -which are called *Sample Reduction* in this thesis- try to reduce the number of columns in the training matrix by either building representative but abstract models [21, 22] or selecting the best representatives from the original training samples [23]. *Dictionary Learning* (DL) approaches -which belong to the first category- are usually employed to form smaller training matrices while preserve the hidden discriminative information of the data. In this thesis, different variants of efficient SRC-based approaches are investigated and an efficient sample reduction method is proposed which improves the efficiency and accuracy of the SRC in face and digit recognition applications. The proposed method employs a clustering algorithm on each individual class of the training data separately and ends up with sub-dictionaries whose number of atoms (dictionary columns) depends on the diversity of the original samples from that class. Experiments on face and digit recognition applications show the effectiveness of the proposed method in comparison to SRC, in its original implementation as well as when combined with other sample reduction methods.

Another drawback of SRC is the fact that it looks at data as a single modality entity. In many real-life problems, the data is available from different feature spaces. For example, a color image consists of three red, green and blue components and each component contains certain information which might be desirable and effective for classification purposes in the visual object classification problem. As another example, in a face recognition problem, multiple feature vectors, such as Local Binary Patterns and down-sampled gray-scale (described in Section 2.2), may be extracted from face images. Each feature vector can be considered as an individual modality with its specific data representation and discrimination power. In this thesis, we investigate a multi-modality approach based on SRC [24] which also suffers from performance issues when large number of training samples are available. To tackle this problem, we propose three category of approaches to accelerate multi-modality classification task in SRC framework and compared these approaches in different classification problems. In the first category, the original samples are first fed into sample reduction algorithms and then modalities are extracted to be used in a multi-modal SRC framework for classification. In the second and third approach, the original training samples are first subjected to modality extraction and then their number of representatives are reduced using a sample reduction algorithm introduced in Chapter 3. The difference between second and third approaches is that the former forces the number of representatives for each class to be the same over different modalities while the latter is more flexible and can handle different number of atoms per class over different modalities. We performed several experiments to show the improvements achieved by these approaches in comparison to other well-known classification algorithms.

1.4 Thesis Outline

The rest of this document is organized as follows. The basics of the Compressive Sensing theory as well as the state of the art Sparse Representation-based Classification are reviewed in Chapter 2. We also introduce the mathematical notations and the datasets used for the experiments in this chapter. In Chapter 3, we first review some of the techniques presented for efficient SRC implementation, then we introduce our suggested sample reduction method. Experiments are conducted and reported to show improvements of the proposed method over the original SRC. Our results are also compared to when SRC is combined with other sample reduction methods. Multi-modality extensions for SRC are studied in Chapter 4. We also introduced our proposed efficient multi-modality approaches along with the experiments in this chapter. Finally, our contributions are summarized in Chapter 5.

CHAPTER 2

BACKGROUND

2.1 Mathematical Notation and Definitions

2.1.1 Mathematical Notations

In this document, vectors and matrices are assumed to have real entries. Scalar numbers are denoted by regular letters such as a and vectors are denoted by boldface lower-case letters, such as $\boldsymbol{v}$ which is called an m-dimensional vector when it is a column array consists of m real entries or equivalently, $\boldsymbol{v} \in \mathbb{R}^m$. Matrices are denoted by regular upper-case letters, such as M and $M \in \mathbb{R}^{m \times n}$ indicates matrix M has m rows and n columns with entries m_{ij} located in the i^{th} row and j^{th} column. M^{T} denotes the transpose of matrix M where the entry at row j and column i of $M^{\mathsf{T}} \in \mathbb{R}^{n \times m}$ is equal to the entry at row j and column i of matrix $M \in \mathbb{R}^{m \times n}$.

According to the above notation, a vector can be decomposed into its entries by $\boldsymbol{v} = [v_1, v_2, \ldots, v_m]^{\mathsf{T}}$. Matrix $M \in \mathbb{R}^{m \times n}$ can be decomposed into n vectors or $M = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n]$. The notations $\boldsymbol{m}_i$ and $\boldsymbol{m}^j$ represent the i^{th} column and j^{th} row of matrix M, respectively. In a same way, matrix $M \in \mathbb{R}^{m \times n}$ can be decomposed into C sub-matrices $M_c \in \mathbb{R}^{n \times m_c}$s such that $M = [M_1, M_2, \ldots, M_C]$ or it can be decomposed into K sub-matrices $M^k \in \mathbb{R}^{n_k \times m}$s such that $M = \left[(M^1)^{\mathsf{T}}, (M^2)^{\mathsf{T}}, \ldots, (M^K)^{\mathsf{T}}\right]^{\mathsf{T}}$.

Subscripts and superscripts are also used frequently for scalars, vectors and matrices and each of them will be made clear where needed.

2.1.2 Vector and Matrix Operations

For vector $\boldsymbol{v} \in \mathbb{R}^m$ the ℓ^p-norm is denoted by $\|\boldsymbol{v}\|_p$ and

$$\|\boldsymbol{v}\|_p \triangleq \left(\sum_{i=1}^{m} |v_i|^p\right)^{\frac{1}{p}}. \tag{2.1}$$

For the special cases where $p = 0$ and $p = \infty$, we have

$$\|\boldsymbol{v}\|_0 \triangleq \sum_{i=1}^{m} \{v_i \neq 0\}, \tag{2.2}$$

or count of non-zero entries of $\boldsymbol{v}$ and

$$\|\boldsymbol{v}\|_\infty \triangleq \max_i |v_i|. \tag{2.3}$$

For matrix $M \in \mathbb{R}^{m\times n}$, the ℓ^p-norm is defined as

$$\|m\|_p \triangleq \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |m_{ij}|^p\right)^{\frac{1}{p}}. \tag{2.4}$$

The special case of $p = 2$ is known as the Frobenius norm and denoted by both $\|M\|_2$ or $\|M\|_F$. The mixed $\ell^{p,q}$-norm of $M = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n]$ is defined as

$$\|M\|_{p,q} \triangleq \left(\sum_{i=1}^{n} \|\boldsymbol{m}_i\|_q^p\right)^{\frac{1}{p}}. \tag{2.5}$$

for a square matrix $M \in \mathbb{R}^{n\times n}$, its trace is defined as

$$\operatorname{tr}(M) = m_{11} + m_{22} + \cdots + m_{nn} = \sum_{i=1}^{n} m_{ii}. \tag{2.6}$$

2.2 Datasets

One of the main challenges in most research studies in machine learning is the datasets which are used to train and the proposed methods and evaluate them against other available approaches. In this thesis, different face and digit datasets are used which are summarized as follows.

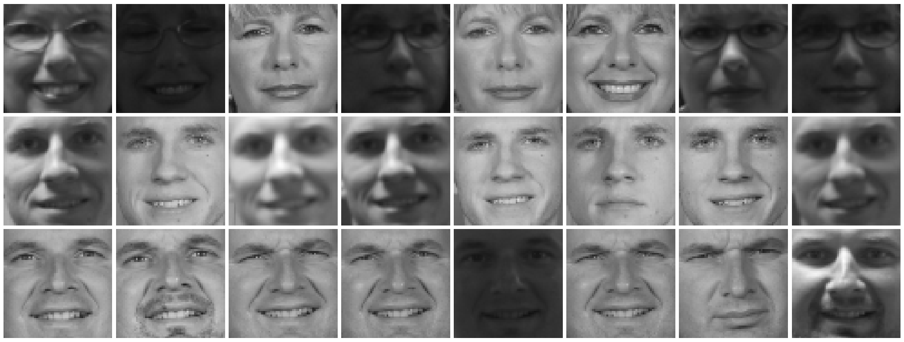


Figure 2.1: Samples from FRGC dataset after pre-processing (face images in each row are from same class).

FRGC face dataset This dataset is presented in [25] under the name of Face Recognition Grand Challenge for the first time. Face images are captured in different times, poses and situations. The main data contains 36817 face images from 535 persons. The original resolution of the images is either 1704×2272 or 1200×1600. For our experiments, 100 classes are randomly selected from the dataset. All images are converted to 8-bit grayscale, normalized and cropped into smaller 60×60 pixels (Figure 2.1). For each class, 80 and 30 face images were randomly selected as training and testing sets, respectively which results in a total number of 8000 training and 3000 testing samples. Images in testing set were selected to be different from training images. For the experiments of multi-modality approaches, we used two modalities:

Down-sampled gray-scale level (GS) All images are resized into 32×32 images and the corresponding 1024 vectors were used as one modality.

Local Binary Patterns (LBP) These features are widely used in face recognition literature. LBP operator is originally designed for textures description. This operator labels all pixels of an image after applying a threshold value to its 3×3 neighborhood and consider the result as a sequence of true-false values. [26] used this idea to extract

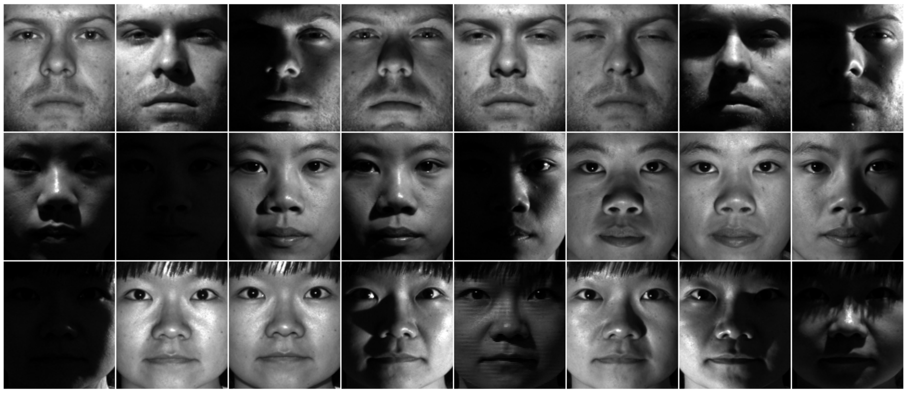


Figure 2.2: Samples from Extended Yale B dataset (face images in each row are from same class).

face descriptors for face recognition purposes. We also use LBP in our experiments on face recognition by incorporating a neighborhood of 8 pixels and radius of 1 pixel. For a down-sampled 32×32 grayscale face image, the LBP modality will be a 2 dimensional 30×30 image which is then vectorized to be used as 900 dimensional feature vectors.

Extended Yale B dataset The second face dataset used for the experiments were selected from the Extended Yale B face dataset [27,28]. This dataset contains a total of 2414 face images from 38 subjects which are cropped and normalized into 192×168 frontal face images. Images are captured under various controlled lighting conditions in the laboratory (Figure 2.2). Half of these images (1207) were selected randomly for training and the remaining were used as test samples. The same GS and LBP modalities which used for FRGC face dataset are also applied to the Extended Yale B dataset in our multi-modality experiments.

Cedar Buffalo digit dataset (USPS) To evaluate different methods in a context other than face recognition, a number of experiments are also performed on handwritten digits datasets. The first digit dataset which is used in this study is the

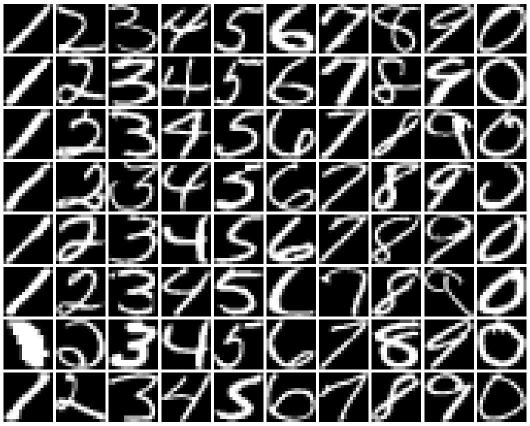


Figure 2.3: Samples from USPS dataset (images in each row are from same class).

Cedar Buffalo binary digits dataset (USPS) [29]. This dataset is taken from zip codes, contains 10 classes with a total number of 11000 8 bits images, down-sampled and Gaussian smoothed to 16×16 digit bitmaps (1100 images for each digit $0, 1, \ldots, 9$). Images are thus represented by 256 dimensional vectors. Among these samples (Figure 2.3), half of them in each class are selected as training samples and the rest are used to test the classification algorithm.

Multi-feature digit dataset (UCI)[1] The second digit dataset used in this study is the UC Irvine [30] multi-feature digit dataset (UCI) which is extracted from a collection of original Dutch public utility maps. A slightly different version of the dataset is used in [31]. This dataset contains 10 classes with a total number of 2000 samples (200 samples for each of digits $0, 1, \ldots, 9$). The original maps were scanned in 8 bits grey value in 400 dpi, sharpened [32] and thresholded to automatically extract [33] the digits. Images were normalized to size 30×48 and 6 different modalities were extracted and available for all samples:

1. FOU: 76 Fourier coefficients of the character shapes,

[1] Available online at `https://archive.ics.uci.edu/ml/datasets/Multiple+Features`

2. FAC: 216 profile correlations,
3. KAR: 64 Karhunen-Love coefficients [34], is the result of a linear transform which is the projection of images onto the eigenvectors of a covariance matrix of the training images.
4. PIX: 240 pixel averages in 2×3 windows (15×16),
5. ZER: 47 rotation invariant Zernike moments [35], are the projection of the image onto a set of orthogonal bases functions,
6. MOR: 6 morphological features, like the number of endpoints of the skeleton.

More details on the features is found in [31, 36]. Half of the data in each class was randomly selected as training samples and the rest were used to test the classification algorithms.

2.3 Recovering a Sparse Signal

Compressive Sensing (CS) theory was introduced by Candès et. al [20] and Donoho [19] in 2006 for the first time. According to this theory, under certain conditions, it is possible to reconstruct an unknown signal or a vector of coefficients from some measurements with a dimensionality far less than the original vector. Compressive sensing, i.e. recovering a vector or signal from a lower dimensional measurement signal, exceeds the traditional Shannon/Nyquist sampling theorem [37, 38] and hence has been recently received considerable attention in many applied areas such as compressive imaging [39], medical imaging [40, 41], remote sensing [42], communication [43], etc[2]. In order to achieve this recovery, the signal of interest needs to be *sparse* and the measurement matrix must also satisfy certain conditions. A signal or vector is called sparse when the majority of its entries are either zero or very close

[2] A comprehensive taxonomy on applications of CS is available at http://dsp.rice.edu/cs.

to zero. More scientifically, a k-sparse signal is a signal whose entries, except for k of them, are zero or close to zero.

The CS problem can be stated as follows. Given a measurement signal $\boldsymbol{y} \in \mathbb{R}^m$, the goal is to recover the original sparse signal $\boldsymbol{x} \in \mathbb{R}^n$. Vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are related by a system of linear equations

$$\boldsymbol{y} = A\boldsymbol{x}, \tag{2.7}$$

where $A \in \mathbb{R}^{m \times n}$ is called the measurement matrix. Let's assume there are more equations than unknowns or $m > n$. The system of linear equations (2.7) with $m > n$ is called an over-determined system and most of the time, no exact solution can be found for this problem. A conventional approximate solution to this problem which yields to a unique answer is to minimize the ℓ^2-norm of the signal of interest or mathematically,

$$\hat{\boldsymbol{x}}_2 = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \|\boldsymbol{x}\|_2 \text{ subject to } \ A\boldsymbol{x} = \boldsymbol{y}\,. \tag{2.8}$$

Optimization problem (2.8) has a closed-form solution of $\hat{\boldsymbol{x}} = A^{\mathsf{T}}(AA^{\mathsf{T}})^{-1}$ but this solution is not necessarily sparse.

Now, let's consider the case where the dimensionality of the measurement signal $\boldsymbol{y}$ is smaller than the one for the original signal $\boldsymbol{x}$, i.e. $m < n$. An illustration of this situation is shown in Figure 2.4. Given the measurements $\boldsymbol{y}$ and the measurement matrix A with $m < n$, equation (2.7) represents an under-determined system of linear equations which has infinitely many solutions for $\boldsymbol{x}$. However, since the signal $\boldsymbol{x}$ is known to be sparse, it is possible to search for a sparse solution among many solutions of (2.7) or equivalently, solve the ℓ^0-norm optimization problem

$$\hat{\boldsymbol{x}}_0 = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \|\boldsymbol{x}\|_0 \text{ subject to } \ A\boldsymbol{x} = \boldsymbol{y}\,, \tag{2.9}$$

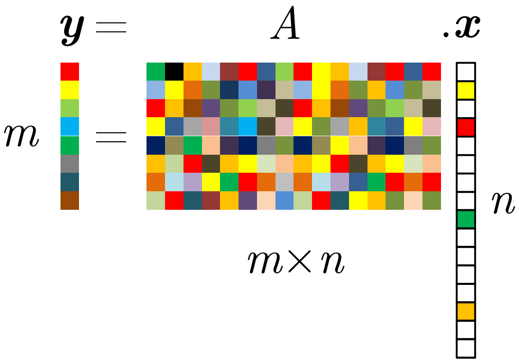


Figure 2.4: Compressive sensing theory: it is possible to recover a k-sparse signal $\boldsymbol{x}$ (in this example, $k = 4$) given measurements $\boldsymbol{y}$ when an under-determined dictionary A exists such that $\boldsymbol{y} = A\boldsymbol{x}$. Color and white entries correspond to non-zero and zero entries, respectively.

where the term $\|\cdot\|_0$ enforces sparsity of the solution. The optimization problem (2.9) is shown to be an NP-hard problem [44] and to solve for a k-sparse signal $\boldsymbol{x}$, one should exhaustively search for all possible k-sparse signals in n dimension.

According to CS theory, [45, 46], it is shown that under certain conditions, minimizer of (2.9) is equal to the one of

$$\hat{\boldsymbol{x}}_1 = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \|\boldsymbol{x}\|_1 \text{ subject to } A\boldsymbol{x} = \boldsymbol{y}\,. \tag{2.10}$$

Unlike (2.9), polynomial complexity solutions for (2.10) are presented in the optimization literature [47].

One of the conditions that must be satisfied for the equivalency of the solutions of (2.9) and (2.10), is that $\boldsymbol{x}$ should be a k-sparse signal such that $k \ll n$. Another condition for this equivalency is that the measurement matrix A should satisfy the Restricted Isometry Property (RIP) [48]. Matrix A satisfies RIP if there is an $\epsilon_k \in (0, 1)$ such that

$$1 - \epsilon_k \leq \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \leq 1 + \epsilon_k \tag{2.11}$$

holds for all k-sparse signals $\boldsymbol{x} \in \mathbb{R}^n$ and $n > k$. In the above inequality, ϵ_k is called the restricted isometry constant and it is shown that if $\epsilon_{2k} < \sqrt{2}-1$, (2.9) and (2.10) converge to equal solutions [48].

An example of a matrix which satisfies RIP with high probability is an $m \times n$ random matrix whose entries are independently and identically distributed (iid) random variables from a Gaussian probability density function with mean zero and variance $\frac{1}{n}$ [49, 50]. In this case, a k-sparse signal of length n can be recovered from a measurement vector of length m where $m \geq ck\text{log}\frac{n}{k}$ with c being a small constant. If RIP condition ($\epsilon_{2k} < \sqrt{2} - 1$) holds for matrix A, the k-sparse signal $\boldsymbol{x}$ can be recovered from the measurements $\boldsymbol{y}$ via ℓ^1-norm optimization which is a convex optimization problem and several methods, such as Orthogonal Matching Pursuit (OMP) [51], Regularized Orthogonal Matching Pursuit (ROMP) [52], Compressive Sampling Matching Pursuit (CoSaMP) [53] and Subspace Pursuit [54], have been proposed to efficiently solve this problem.

In many applications, the original signal of interest ($\boldsymbol{x}$) is not sparse. In these situations, if a domain which represents $\boldsymbol{x}$ as a sparse signal exists, it is possible to solve the recovery problem by substituting $\boldsymbol{x}$ by the (k-sparse) vector $\boldsymbol{s}$ such that $\boldsymbol{s} = \Psi^{-1}\boldsymbol{x}$ where the square matrix $\Psi \in \mathbb{R}^{n\times n}$ is a transform matrix [49]. By substituting $\boldsymbol{x}$ in (2.7) with $\Psi\boldsymbol{s}$, we will have

$$\boldsymbol{y} = A\boldsymbol{x} = A\Psi\boldsymbol{s} = \Theta\boldsymbol{s}, \tag{2.12}$$

where $\Theta = A\Psi \in \mathbb{R}^{m\times n}$. Similar to (2.7), equation (2.12) represents an underdetermined system of equations with a k-sparse vector $\boldsymbol{s}$ and the whole system (2.12) satisfies the CS conditions. Some examples for Ψ can be listed as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) or Wavelet transform.

2.4 Compressive Sensing for Classification

The idea of incorporating CS for classification purposes was developed in 2009 for the first time in face recognition application [7]. In this study, Wright et al. mapped the classification problem into a CS problem using the basic equation (2.7) and called it Sparse Representation-based Classification (SRC). They also reported interesting results on face recognition application which were outstanding in both recognition rates and robustness to noise when compared to other face recognition methods. Since then, SRC was used in different applications and moreover, lots of supplement algorithms were developed and published on top of the original SRC method. In this section, we introduce the state-of-the-art SRC along with some experiments to have fundamental information regarding the rest of this dissertation in chapters 3 and 4.

2.4.1 Sparse Representation-based Classification (SRC)

Wright et al. demonstrated the effectiveness of SRC in a face recognition application [7]. A face dataset consists of some face images each of which can be represented as a matrix $I \in \mathbb{R}^{h \times w}$ where h and w represent the height and the width of the face image, respectively. In many face recognition algorithms, the vectorized version of matrix I which is denoted by $\boldsymbol{v} \in \mathbb{R}^m$ and $m = w \times h$, is used to represent the face image data. Assuming there are n_i training images from subject (class) i in the dataset, the matrix $V_i = [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$ represents all face images from subject i. These face images can be considered as some points in an m dimensional space which span a face subspace for class i [55]. In an ideal situation, a test image from this class, noted by $\boldsymbol{y}_i \in \mathbb{R}^m$ and $\boldsymbol{y}_i \nsubseteq V_i$, can be represented as a linear combination of the training images from the same class (Figure 2.5) or mathematically

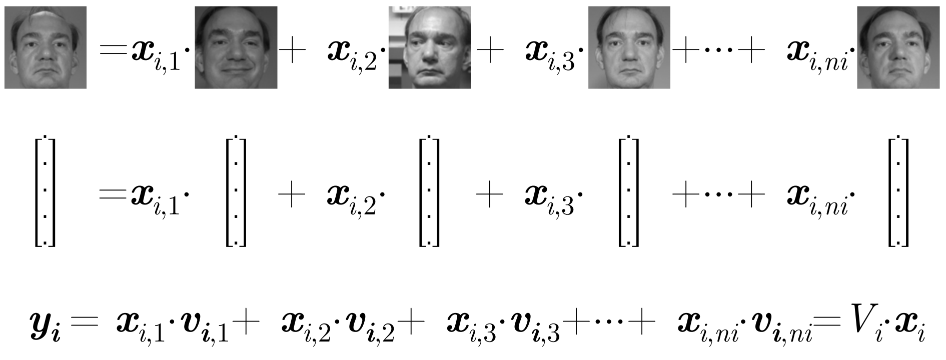

$=\boldsymbol{x}_{i,1}\cdot + \boldsymbol{x}_{i,2}\cdot + \boldsymbol{x}_{i,3}\cdot + \cdots + \boldsymbol{x}_{i,ni}\cdot$
$=\boldsymbol{x}_{i,1}\cdot + \boldsymbol{x}_{i,2}\cdot + \boldsymbol{x}_{i,3}\cdot + \cdots + \boldsymbol{x}_{i,ni}\cdot$
$\boldsymbol{y_i} = \boldsymbol{x}_{i,1}\cdot\boldsymbol{v}_{\boldsymbol{i},1} + \boldsymbol{x}_{i,2}\cdot\boldsymbol{v}_{\boldsymbol{i},2} + \boldsymbol{x}_{i,3}\cdot\boldsymbol{v}_{\boldsymbol{i},3} + \cdots + \boldsymbol{x}_{i,ni}\cdot\boldsymbol{v}_{\boldsymbol{i},ni} = V_i\cdot\boldsymbol{x}_i$

Figure 2.5: A face image of one class can be ideally represented as a linear combination of other samples from the same class (face images are from FRGC dataset).

$$\boldsymbol{y}_i = x_{i,1}\boldsymbol{v}_{i,1} + x_{i,2}\boldsymbol{v}_{i,2} + \cdots + x_{i,n_i}\boldsymbol{v}_{i,n_i}, \tag{2.13}$$

where $x_{i,p}$s $\in \mathbb{R}$ and $p \in \{1, 2, \ldots, n_i\}$ are the coefficients representing $\boldsymbol{y}_i$ in the domain of V_i. Denote the coefficient vector associated with class i by $\boldsymbol{x}_i$=[$x_{i,1}$, $x_{i,2}$, $\ldots$, x_{i,n_i}]$^{\mathsf{T}}$, (2.13) can be formulated in the matrix form of

$$\boldsymbol{y}_i = V_i\boldsymbol{x}_i. \tag{2.14}$$

In a more general and multi-class scenario, considering all $n = n_1 + n_2 + \cdots + n_C$ training images from C classes, the entire training set can be represented by the training matrix $V = [V_1, V_2, \ldots, V_C] \in \mathbb{R}^{m\times n}$ where

$$V = [\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2}, \ldots, \boldsymbol{v}_{1,n_1}, \boldsymbol{v}_{2,1}, \ldots, \boldsymbol{v}_{2,n_2}, \ldots, \boldsymbol{v}_{C,n_C}]. \tag{2.15}$$

Figure 2.6 shows an illustration of this multi-class scenario. Given (2.15), equation (2.14) can be generalized as

$$\boldsymbol{y}_i = V\boldsymbol{x}^\star, \tag{2.16}$$

where $\boldsymbol{x}^\star \in \mathbb{R}^n$ is a coefficient vector with its all entries equal to zero except for the ones associated with class i. Given several classes and many face images from

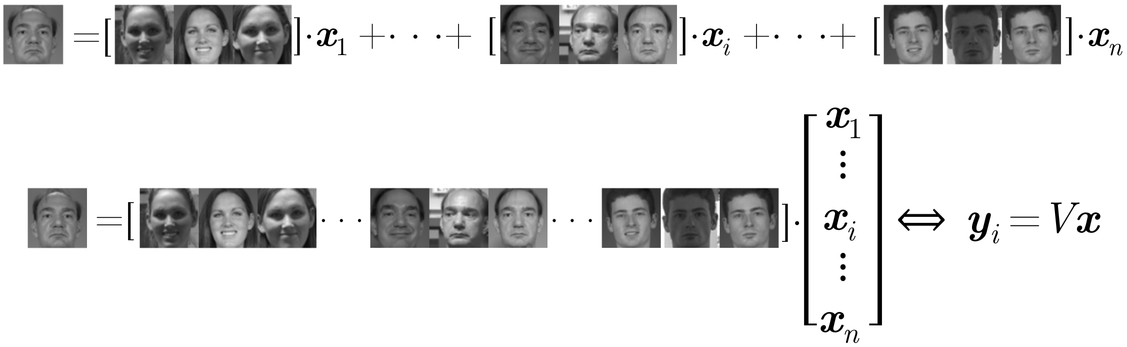

$= [\ ] \cdot \boldsymbol{x}_1 + \cdots + [\ ] \cdot \boldsymbol{x}_i + \cdots + [\ ] \cdot \boldsymbol{x}_n$
$\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \Leftrightarrow \boldsymbol{y}_i = V\boldsymbol{x}$

Figure 2.6: Generalization of the problem illustrated in Figure 2.5. In an ideal situation, a face image from a specific class i can be represented as a linear combination of training faces from all classes such that the coefficient corresponding other classes such that $\forall j \neq i, \boldsymbol{x}_j = \vec{\mathbf{0}}$.

each class in the training matrix, entries in the corresponding coefficient vector $x^\star$ are mostly zero hence $x^\star$ can be considered as a sparse vector. More specifically, this vector is an n_i-sparse vector of dimension n. If in matrix V, the number of image samples is larger than the dimensionality of the sample space ($m < n$), (2.16) represents an under-determined system of equations. The main idea of SRC proposed in [7] comes from this fact that an unknown test sample $\boldsymbol{y}_i$ can be classified by recovering the sparse coefficient vector $\boldsymbol{x}^\star$ in (2.16). Based on the discussion in Section 2.3, a sparse vector can be efficiently recovered from an under-determined system of equations by solving the ℓ^1-norm optimization problem (2.10) with a quadratic complexity.

Up to this point, the assumption was an ideal situation, where a test face image from one class is exactly equal to a linear combination of some pre-known samples from the same class. In real applications, due to the existence of noise and modeling errors in the data, this assumption is not valid. To deal with this problem, it can be assumed that (2.16) holds by considering a small amount of noise, i.e.

$$\boldsymbol{y} = V\boldsymbol{x} + \boldsymbol{e}, \tag{2.17}$$

where $\boldsymbol{e} \in \mathbb{R}^m$ represents the noise term, $\boldsymbol{y}$ is a general test sample from an un-known class and vector $\boldsymbol{x}$ is the general coefficient vector to be recovered. In $\boldsymbol{x}$, in contrast to $\boldsymbol{x}^\star$, entries associated with classes other than i are not necessarily zero but usually they are small real values in comparison to the entries correspond to class i. Having this assumption, the CS optimization problem (2.10) can be reformulated as

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \|\boldsymbol{x}\|_1 \text{ subject to } \|V\boldsymbol{x} - \boldsymbol{y}\|_2 \leq \boldsymbol{\epsilon}, \tag{2.18}$$

where $\hat{\boldsymbol{x}}$ is the recovered coefficient vector and $\boldsymbol{\epsilon} > \|\boldsymbol{e}\|_2$ is the relaxation term imposed to allow some noise in the recovery process. The optimization problem (2.18) is known as Least Absolute Shrinkage and Selection Operator (LASSO) [56] and categorized as a convex optimization problem and can be efficiently solved by polynomial solutions such as [47] which is a second-order cone programming approach. We used the implementation of primal-dual algorithm for linear programming to solve (2.18) optimization problem [57, 58].

Figure 2.7 shows an example of the result of applying SRC algorithm on face images from FRGC dataset. In this experiment, 5000 face images from 100 classes formed the matrix V which is used as the training matrix in the optimization process (2.18) to recover the coefficient vector. The test sample $\boldsymbol{y}$ (Figure 2.7a) is selected from class 4 and fed into (2.18) along with the training matrix V. After optimization, the coefficients associated with the first 500 training samples are shown in Figure 2.7b by thin-red lines. The coefficient associated with the same class of $\boldsymbol{y}$ are marked by thick-blue lines in the recovered vector and the training samples associated with the 2 largest coefficients are shown in this figure. It is clearly seen that largest coefficients among all 500 coefficients happens to be from the same class as the test sample and moreover, they look very similar in terms of face expressions.

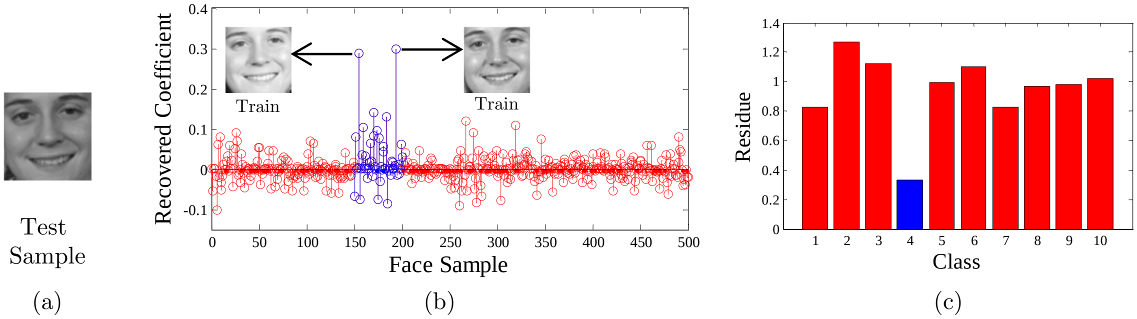


Figure 2.7: A result of running SRC algorithm on (a) an input test sample using a dictionary containing 500 face images (10 subjects, 50 face images each). (b) Recovered coefficient vector entries. It can be clearly seen that largest values in the recovered vector are associated to class 4 (blue-thick entries) which is the test sample's class. (c) Residual values for all 10 classes. Class 4 (solid blue) has the minimum residue.

The second step in SRC is to identify the class of the test sample $\boldsymbol{y}$ by analyzing the recovered coefficient vector $\boldsymbol{x}$. Basically, any kind of classifier can be used in this step. One can simply assign the test sample to the class which includes the largest coefficient in $\boldsymbol{x}$ (the right-most training sample in Figure 2.7b). However, to get more benefits from the face subspace structures, it is needed to consider all training samples from a class to identify the class associated with $\boldsymbol{y}$. For this purpose, given the recovered vector $\hat{\boldsymbol{x}}$, a vector $\boldsymbol{\delta}_i \in \mathbb{R}^n$ is defined for every class i $(i = 1, 2, \ldots, C)$ with all zero entries except for the ones associated with class i which are equal the corresponding ones in the recovered vector $\hat{\boldsymbol{x}}$. An approximation of the input test sample $\boldsymbol{y}$ for class i can be calculated by $\hat{\boldsymbol{y}}_i = V\boldsymbol{\delta}_i$. Then, a class label i is assigned to the object by selecting the class which minimizes the error between the actual test sample $\boldsymbol{y}$ and the approximations $\hat{\boldsymbol{y}}_i$ vectors or

$$class\,(\boldsymbol{y}) = \hat{i} = \underset{i}{\operatorname{argmin}}\ \left\|V\boldsymbol{\delta}_i - \boldsymbol{y}\right\|_2. \tag{2.19}$$

In the above equation the optimization objective function, $\left\|V\boldsymbol{\delta}_i - \boldsymbol{y}\right\|_2$ is called *residue*. Figure 2.7c shows the residual values calculated for 10 classes. As can be seen, resid-

Algorithm 2.1: SRC Algorithm [7].

```
input  : Training samples matrix $V = [V_1, V_2, \dots, V_C] \in \mathbb{R}^{m\times n}$ for $C$ classes;
         Test sample $\boldsymbol{y} \in \mathbb{R}^m$;
         Error tolerance $\boldsymbol{\epsilon} > 0$;
output : $class(\boldsymbol{y})$;
- Normalize the columns of training matrix to have unit $\ell^2$-norm;
- Solve the optimization problem (2.18) to recover the coefficient vector $\hat{\boldsymbol{x}}$;
for $i = 1, 2, \dots, C$ do
    - $\boldsymbol{\delta}_i \in \mathbb{R}^n = \vec{\mathbf{0}}$ except for the sub-vector with class $i$ which is equal to $\hat{\boldsymbol{x}}_i$;
    - Compute the residuals $r_i(\boldsymbol{y}) = \|V\boldsymbol{\delta}_i - \boldsymbol{y}\|_2$;
$class(\boldsymbol{y}) = \underset{i}{\mathrm{argmin}}\; r_i(\boldsymbol{y})$;
```

ual value for class 4, which is the class to which test sample $\boldsymbol{y}$ belongs, is significantly smaller than the ones correspond to other classes, so SRC selects class 4 to be the class associated with the test sample $\boldsymbol{y}$. The above discussion on the recognition process via SRC is summarized in Algorithm 2.1.

2.4.2 Dimensionality Reduction (DR)

In real-life applications of face recognition, there are usually lots of subjects and face images available for training. Face images are usually from high dimensionality. For instance, face images which are used for the SRC experiments illustrated in Figure 2.7 are of the size of 60×60 and form 3600 dimensional vectors and a total number of 100 subjects each contains 50 face images are analyzed. In the experiments which are introduced in [7], face images are even larger and of the resolution of 640×480. An SRC framework with a training set containing large number of images with high dimensionalities not only needs large storage, but also introduces high computational complexity which leads to longer execution time of the classification task. For example, classification of a single test sample in the experiment of Figure 2.7 took approximately 320 seconds which is not desirable in real applications. On the other hand, high dimensionality of the data imposes another disadvantage to the problem. As described in Section 2.3, sparse signal recovery problem of (2.9) is an NP-hard problem. However, under certain conditions, the convex optimization problem (2.10) will efficiently converge to an equivalent solution to the minimizer of (2.9). For this purpose, the training matrix V is needed to represent an under-determined system of equations i.e. V must have more columns than rows. In SRC, dimension of face images and number of training face samples determine the number of rows and columns in matrix V, respectively. Training matrices with face images from high dimensionality forces to select a large number of training samples to make an under-determined system of linear equations. Large number of images might not be available in many real-life applications and moreover increases the time needed for the classification task. In Chapter 3, we will discuss the time complexity of the fastest available algorithms to solve (2.18) is quadratic to the number of training samples. Therefore, increasing number of samples to make V sufficiently under-determined, imposes large amount of delays to the classification process.

In order to overcome the above mentioned issues, dimensionality reduction methods can be used on both training and test samples. In the context of machine learning and specifically, face recognition, a variety of feature extraction methods are investigated which helps to deal with the problem of high dimensionality of the samples. Some of these methods result in holistic face features such as Eigenfaces [4], Fisherfaces [59] and Laplacianfaces [60]. Another category of feature extraction techniques look for local and partial descriptors such as eyes, nose, etc. [61, 62]. Most feature extraction methods use linear operations on the original images, hence the

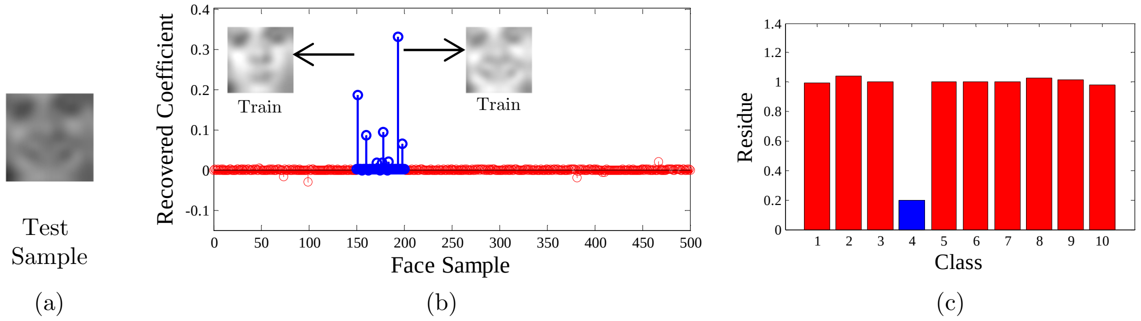


Figure 2.8: A result of running SRC algorithm on (a) an input test sample using a dictionary containing 500 face images (10 subjects, 50 face images each) and down-sample features. (b) Recovered coefficient vector entries. It can be clearly seen that largest values in the recovered vector are associated to class 4 (blue-thick entries) which is the test sample's class. (c) Residual values for all 10 classes. Class 4 (solid blue) has the minimum residue.

feature extraction process for a feature vector $\hat{\boldsymbol{y}}$ can be easily formulated by a matrix multiplication $\hat{\boldsymbol{y}} = R\boldsymbol{y}$, where $R \in \mathbb{R}^{r \times m}$ is the feature extraction operator. On the training side, the dimension of matrix V is also reduced to r by the multiplication $RV = \hat{V} \in \mathbb{R}^{r \times n}$. In this situation, by selecting $r \ll n$, the system of linear equations $\hat{\boldsymbol{y}} = \hat{V}\boldsymbol{x}$ will represent an under-determined system and since the coefficient vector to be recovered is still sparse, it can be recovered using ℓ^1-norm optimization problem. Simply by substituting $\hat{\boldsymbol{y}}$ and $\hat{V}$ in the SRC algorithm (Algorithm 2.1) the optimization problem (2.18) is converted to

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmin}} \ \|\boldsymbol{x}\|_1 \text{ subject to } \left\| \overbrace{\hat{V}}^{RV} \boldsymbol{x} - \overbrace{\hat{\boldsymbol{y}}}^{R\boldsymbol{y}} \right\|_2 \leq \boldsymbol{\epsilon}. \tag{2.20}$$

Several linear dimensionality reduction methods such as Eigenfaces, Fisherfaces, Laplacianfaces, down-sampling and random projection are studied in [7]. We conduct an experiment with a same settings as the one reported in Figure 2.7 using down-sampling feature extraction and the result of running SRC to classify a test subject from class 4 is illustrated in Figure 2.8. As shown in Figure 2.8b, the recovered vector in this experiment shows more sparsity than the one recovered in the first experiment (Figure 2.7). This happens because dimensionality reduction makes the number of rows in the training matrix to be far smaller than the ones when original images are used. In this specific experiment, without applying dimensionality reduction, matrix V is of the size 3600×5000 while after applying down-sampling, matrix V will become of size 100×5000 which represents a highly under-determined system of equations. Comparing the residual plot in figures 2.7 and 2.8 also shows that using dimensionality reduction leads to larger difference in the residual values of the real class of $\boldsymbol{y}$ and other classes which helps SRC to identify the class associated with the test sample in a more accurate way when dimensionality reduction is used.

CHAPTER 3

EFFICIENT IMPLEMENTATIONS FOR SRC

As discussed in Section 2.4, Wright et al. proposed a classification method called Sparse Representation-based Classification (SRC) which is shown to have interesting recognition rates on face recognition applications.

SRC solves the ℓ^1-norm optimization problem (2.18) or it's Lagrange multiplier equivalent, i.e.

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \left\{ \|V\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \right\}, \tag{3.1}$$

to recover the coefficient vector $\boldsymbol{x}$ and uses this vector for classification. One of the major limitations of SRC is its speed which depends on the speed of solving ℓ^1-norm optimization problem (3.1). It is shown that solving this optimization problem, in its fastest way, has a time complexity which is quadratic to the number of columns in matrix V [63]. This implies that if the number of training samples or, equivalently, the number of columns in matrix V get doubled, the time required for solving (3.1) is quadrupled. Figure 3.1 shows the relationship between the execution time and the size of the SRC training matrix in a face recognition application. In this example, columns in matrix V are 100 dimensional vectors formed by vectorizing down-sampled face images. It is shown that how increasing number of samples in matrix V leads to the classification running time increment in SRC.

In most practical applications, for example, a face recognition engine, the problem confronts with so many classes and there are usually lots of training samples available in each class. Moreover, as discussed in Section 2.4.2 face images usually come from a relatively large dimensional space. The dimensionality problem can

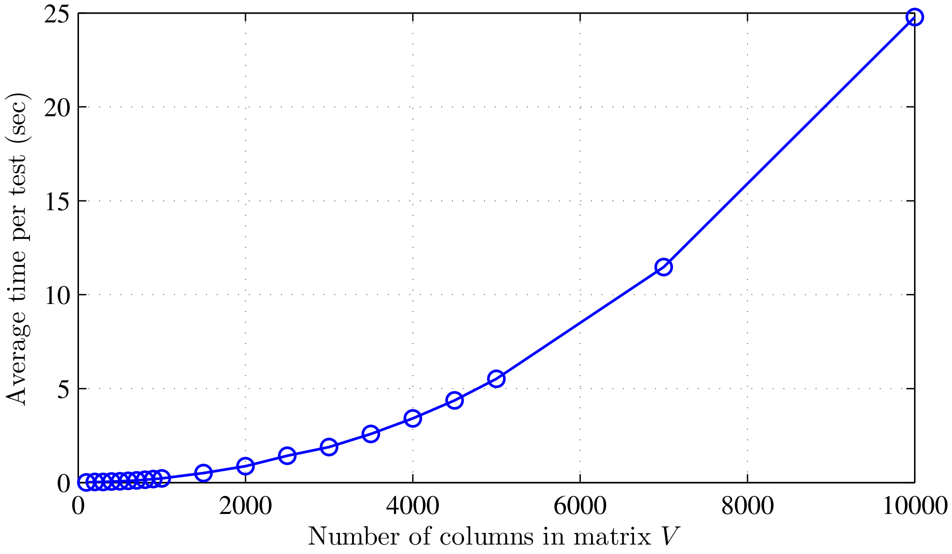


Figure 3.1: Average time per test in SRC is super-linear to the number of columns in matrix V.

be solved efficiently by dimensionality reduction methods described in Section 2.4.2. However, the large number of training samples is still a big challenge in SRC efficiency from both time and space point of view. Given the high recognition accuracy of SRC, it becomes important to reduce the time and memory requirements of this method. Reducing the number of training samples by a relatively modest factor results in a significant decrease in SRC running time. Improvements in time and memory efficiency help to make SRC a more practical solution for portable devices and can also significantly decrease the computational load of SRC when running on more powerful hardware.

According the above discussion, reducing the number of atoms in SRC training matrix -which is called *Sample Reduction* (SR) in this document- is a big challenge towards its efficiency. In [7], authors suggest to randomly select a subset of training samples in order to reduce the size of the training matrix and as a result, reduce the execution time and space requirements. This solution is not necessarily optimum since there may exist discriminative information in some samples which have not been selected. On the other hand, some of the selected samples may contain redundancies and information overlaps and can be merged or ignored by introducing a minimum loss to the recognition outcome. A number of studies investigate different methods to tackle this issue which are introduced in the following section.

3.1 A Review of Efficient SRC Algorithms

In this section, we discuss some methods which are presented to improve the efficiency of the SRC algorithm by using more efficient and in the meantime precise data representation. In SRC context, two main categories for this purpose are presented so far.

The first category includes approaches which use the training samples to build a representative matrix whose columns are not selected from the original face images. These methods usually searches for domains which represent training data precisely and in the same time be discriminative in order to be used for classification purposes. *Dictionary learning* (DL) algorithms can be categorized in this group. Primary DL methods were developed to encode a data collection in a more abstract form for communication and storage purposes. Usually, one of the objectives in DL is to design a dictionary such that it can represent the data as sparse as possible.

An effective approach to build a dictionary is presented in [64] in which original training samples are used in an optimization process to learn a dictionary. Assume all the n training samples (vectors $\boldsymbol{v}_i$) are stored in matrix $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n]$ and the goal is to come up with a dictionary $D \in \mathbb{R}^{m\times d}$ $(d \leq n)$ by solving

$$\begin{aligned}\{D, W\} =& \underset{D,W}{\operatorname{argmin}} \left\{ \sum_{j=1}^{n} \left\| \boldsymbol{v}_j - D\boldsymbol{w}_j \right\|_2^2 + \lambda \left\| \boldsymbol{w}_j \right\|_1 \right\} \\ =& \underset{D,W}{\operatorname{argmin}} \left\{ \left\| V - DW \right\|_F^2 + \lambda \left\| W \right\|_1 \right\} \\ & \text{such that} \quad \forall \ell \in \{1, 2, \dots, d\} : \left\| \boldsymbol{d}_\ell \right\|_2^2 \leq 1, \end{aligned} \tag{3.2}$$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_n] \in \mathbb{R}^{d\times n}$ is the coefficient matrix which can be considered as the representation of training samples V over the dictionary D. The notation $\boldsymbol{d}_\ell$ shows the ℓ^{th} atom (column) in the dictionary D, λ is the regularization parameter which holds a trade-off between the sparsity level of the columns of W and the error term $\left\| V - DW \right\|_F$. Further dictionary learning algorithms are presented in [65, 66] which may be used for classification purposes. A dictionary should have two important properties in order to be used as the SRC training matrix:

1. It should represent the training data precisely.
2. It should have discriminative power which makes it appropriate to be utilized for classification.

In this document, we will discuss the details of two DL methods which are specifically designed to be used in an SRC framework: Metaface dictionary learning (MF) [21] and Fisher Discrimination Dictionary Learning (FDDL) [22].

The next category of methods to approach sample reduction is called *sample selection (SS)* in this thesis. These methods try to find the best representatives among all training samples and build the training matrix V. For example, simply selecting a random subset of samples from training data can be considered as a sample selection method. To have a more justified example from this category, details for the Sparse Modeling Representative Selection (SMRS) method [23] will be discussed in Section 3.1.3.

3.1.1 Metaface Dictionary Learning

Yang et al. [21] introduced a dictionary learning method based on metagenes in gene expression data analysis [67], and used it along with SRC in a face recognition framework. In this approach, dictionary D is considered to be a collection of sub-dictionaries representing each class separately, i.e. $D = [D_1, D_2, \ldots, D_C]$, where each $D_i \in \mathbb{R}^{m \times d_i}$ is learned for each class i separately by using training samples from that class, V_i. Atoms in the sub-dictionary D_i are denoted by $\boldsymbol{d}_{i,j}$ where $D_i = [\boldsymbol{d}_{i,1}, \boldsymbol{d}_{i,2}, \ldots, \boldsymbol{d}_{i,d_i}]$ and $d_i \leq n_i$ is the total number of atoms in the sub-dictionary D_i. Each atom in the dictionary required to be a unit vector, i.e. $\forall i, j : \boldsymbol{d}_{i,j}^{\mathsf{T}} \boldsymbol{d}_{i,j} = 1$. Metaface dictionary for class i will be determined by solving

$$\left\{ \hat{D}_i, \hat{W}_i \right\} = \underset{D_i, W_i}{\operatorname{argmin}} \; \left\{ \left\| V_i - D_i W_i \right\|_F^2 + \lambda \left\| W_i \right\|_1 \right\}$$

$$\text{such that } \forall j \in \{1, 2, \ldots, d_i\} : \boldsymbol{d}_{i,j}^{\mathsf{T}} \boldsymbol{d}_{i,j} = 1, \tag{3.3}$$

where $W_i \in \mathbb{R}^{d_i \times n_i}$ is a sub-dictionary associated with i^{th} class. Equation (3.3) is a multi-variable optimization problem and could be solved by alternatively optimizing D_i and W_i while the other one is fixed. When fixing D_i, the objective function is reduced into

$$\hat{W}_i = \underset{W_i}{\operatorname{argmin}} \; \left\{ \left\| V_i - D_i W_i \right\|_F^2 + \lambda \left\| W_i \right\|_1 \right\}. \tag{3.4}$$

The optimization problem (3.4) is solved using the standard convex optimization approach presented in [68]. In the next step, W_i is fixed and the objective function will be in the form of

$$\hat{D}_i = \underset{D_i}{\operatorname{argmin}} \; \left\| V_i - D_i W_i \right\|_F^2 \quad \text{such that} \quad \boldsymbol{d}_{i,j}^{\mathsf{T}} \cdot \boldsymbol{d}_{i,j} = 1 \quad \forall j. \tag{3.5}$$

To solve (3.5), matrix W_i can be decomposed into $W_i = [(\boldsymbol{w}_i^1)^\mathsf{T}, (\boldsymbol{w}_i^2)^\mathsf{T}, \dots, (\boldsymbol{w}_i^{d_i})^\mathsf{T}]^\mathsf{T}$, where $\boldsymbol{w}_i^j \in \mathbb{R}^{1\times n}$ represents the j^{th} row of W_i. Now, each column of D_i (i.e. $\boldsymbol{d}_{i,j}$s) will be updated one by one and when updating one, all others are fixed. For each j, the following optimization problem is solved

$$\begin{aligned}
\hat{\boldsymbol{d}}_{i,j} =& \underset{\boldsymbol{d}_{i,j}}{\operatorname{argmin}} \left\| V_i - \sum_{\ell\neq j} \boldsymbol{d}_{i,\ell}\boldsymbol{w}_i^\ell - \boldsymbol{d}_{i,j}\boldsymbol{w}_i^j \right\|_F^2 \quad \text{such that} \quad \boldsymbol{d}_{i,j}^\mathsf{T} \cdot \boldsymbol{d}_{i,j} = 1 \\
=& \underset{\boldsymbol{d}_{i,j}}{\operatorname{argmin}} \left\| \gamma - \boldsymbol{d}_{i,j}\boldsymbol{w}_i^j \right\|_F^2 \quad \text{such that} \quad \boldsymbol{d}_{i,j}^\mathsf{T} \cdot \boldsymbol{d}_{i,j} = 1, \qquad (3.6)
\end{aligned}$$

where $\gamma = V_i - \sum_{\ell\neq j} \boldsymbol{d}_{i,\ell}\boldsymbol{w}_i^\ell$. Optimization problem (3.6) can be solved using Lagrange multiplier and has the closed form solution of

$$\boldsymbol{d}_{i,j} = \frac{\gamma\left(\boldsymbol{w}_i^j\right)^\mathsf{T}}{\gamma\left\|\left(\boldsymbol{w}_i^j\right)^\mathsf{T}\right\|_2}. \qquad (3.7)$$

After updating all $\boldsymbol{d}_{i,j}$s, the whole dictionary D_i will be updated. This process is repeated for each individual class $i \in \{1, 2, \dots, C\}$ to form the final dictionary $D = [D_1, D_2, \dots, D_C]$. The overall optimization steps are summarized in Algorithm 3.1. This dictionary will be used in SRC framework in the next step to perform the classification task. Figure 3.2 (right) illustrates the first 3 Metaface dictionary atoms calculated for three different datasets. An original sample from each class is shown on the left. In this experiment, the total number of calculated dictionary atoms for the illustrated class was set to 11, 10 and 45 for FRGC, Extended Yale B and USPS digits, respectively. The performance of SRC using Metaface dictionary is evaluated in [21] and compared to nearest neighbor classifier and when SRC uses a random subset of training data as its matrix. Recognition results on three different face datasets are reported in this study and show higher classification accuracies for metaface in comparison to the other two classification methods.

Algorithm 3.1: Metaface optimization process [21].

```
input  : Training samples matrix $V = [V_1, V_2, \dots, V_C] \in \mathbb{R}^{m \times n}$ for $C$ classes;
         Regularization term $\lambda$;
         Number of dictionary atoms for each class $d_i|_{i=1}^{C}$;
for $i \in \{1, 2, \dots, C\}$ do
    Each column of $D_i \in \mathbb{R}^{m \times d_i}$ is initialized as a random $\boldsymbol{d}_{i,j}|_{j=1}^{d_i}$, $\|\boldsymbol{d}_{i,j}\|_2 = 1$;
    repeat
        Fix $D_i$ and solve (3.4) for $W_i$;
        Fix $W_i$ and solve (3.5) for $D_i$;
    until convergence or maximum number of iterations;
output : Dictionary $D$ and coefficients $W$;
$D = [D_1, D_2, \dots, D_C]$;
$W = [{W_1}^\mathsf{T}, {W_2}^\mathsf{T}, \dots, {W_C}^\mathsf{T}]^\mathsf{T}$;
```

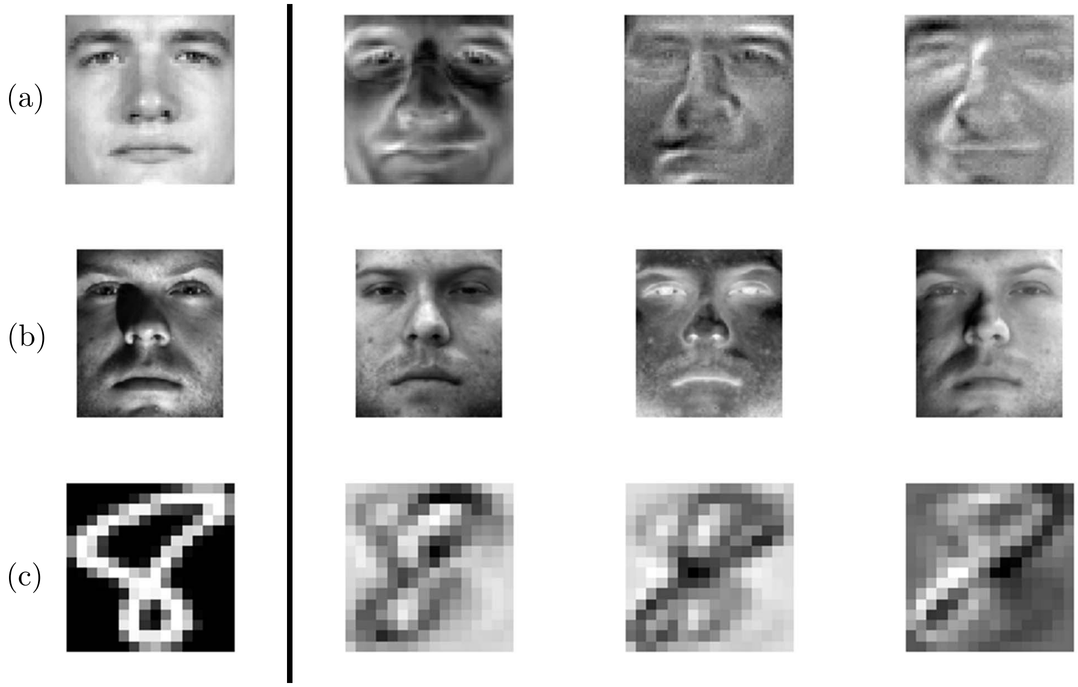


Figure 3.2: Metaface dictionary atoms (right) calculated for original face data (left) on (a) FRGC dataset, (b) YaleB dataset and (c) USPS digits dataset.

3.1.2 Fisher Discrimination Dictionary Learning (FDDL)

Yang et al. [22] proposed a dictionary learning method which builds a dictionary containing class-labeled atoms and in the meanwhile uses Fisher discrimination criterion to make the dictionary more discriminative. FDDL solves

$$\left\{\hat{D}, \hat{W}\right\} = \underset{D,W}{\operatorname{argmin}} \; \{r(V, D, W) + \lambda_1 \left\|W\right\|_1 + \lambda_2 f(W)\}, \tag{3.8}$$

to come up with dictionary $\hat{D}$ and the matrix of coefficients $\hat{W}$. The ℓ^1-norm term implies the sparsity, the function $r(V, D, W)$ is the discriminative fidelity term, and $f(\cdot)$ is the Fisher discrimination constraint term. The regularization parameters λ_1 and λ_2 set a trade-off between the sparsity and discrimination power.

The discriminative fidelity term is actually the summation of fidelity terms over all classes or equivalently,

$$r(V, D, W) = \sum_{i=1}^{C} r\left(V_i, D, W_i\right), \tag{3.9}$$

where each element $W_i \in \mathbb{R}^{d \times n_i}$ is the representation of V_i over D and can be decomposed into matrices $W_i^j \in \mathbb{R}^{d_j \times n_i}, j \in \{1, 2, \ldots, C\}$ representing mapping of original samples V_i over the sub-dictionary D_j. In other words, W_i=[$(W_i^1)^\mathsf{T}$, $(W_i^2)^\mathsf{T}$, $\ldots$, $(W_i^C)^\mathsf{T}]^\mathsf{T}$. In order for the dictionary D to be a good representative for the original training samples V, The discriminative fidelity term, $r(\cdot)$, forces 3 constraints on both dictionary and coefficient matrix.

C1: The whole dictionary D to be a good representative for the samples from class i i.e. V_i. In other words, r tries to make DW_i as close as possible to V_i.

C2: The sub-dictionary D_i to be a good representative for samples V_i. Equivalently, $D_i W_i^i$ is forced to be a good approximation of V_i.

C3: The coefficients which correspond to classes other than i should not represent class i. In other words, $W_j^i (i \neq j)$ should be close to zero to make $D_j W_i^j$ as small as possible.

The above constraints, mathematically formulate the fidelity term for class i as

$$r(V_i, D, W_i) = \|V_i - DW_i\|_F^2 + \left\|V_i - D_i W_i^i\right\|_F^2 + \sum_{\substack{j=1 \\ j \neq i}}^{C} \left\|D_j W_i^j\right\|_F^2. \tag{3.10}$$

This function is substituted in the objective function of (3.8).

The function $f(\cdot)$ in (3.8) is the Fisher discrimination criterion which increases the discrimination power of the dictionary by considering the within-class and between-class scattering ($S_\omega(W)$ and $S_\beta(W)$, respectively) and implemented as

$$f(W) = \mathrm{tr}(S_\omega(W)) - \mathrm{tr}(S_\beta(W)) + \eta\|W\|_F^2, \tag{3.11}$$

where the last term imposed to provide convexity weighted by η and the $\mathrm{tr}(\cdot)$ is the matrix trace operator defined in Section 2.1.2. More details on the calculation of Fisher discrimination criteria can be found in [22].

The functions (3.10), (3.11) and sparsity constraint are convex functions [22], so (3.8) is categorized as a multi-variable convex optimization problem which is solved by alternatively optimizing of D and X. The authors of [22], approached to (3.8) in a similar way they approached Metaface dictionary learning (Algorithm 3.1) i.e. alternative optimization. The optimization steps are discussed in more details in [22].

Solving (3.8) for all classes, results in a discriminative dictionary whose atoms are labeled for different classes. Figure 3.3 shows 3 atoms of the FDDL dictionary on the same three datasets and subjects of Figure 3.3. FDDL dictionary is used in the next step instead of the training matrix in SRC algorithm for the classification purpose. Experimental results in [22] shows significant improvements of classification when using FDDL dictionary comparing to use a subset of the original training samples with the same number of columns in SRC framework and other two classifiers of nearest neighbor and SVM. The SRC accuracy evaluation is also performed using FDDL and other two DL methods of discriminative KSVD and dictionary learning

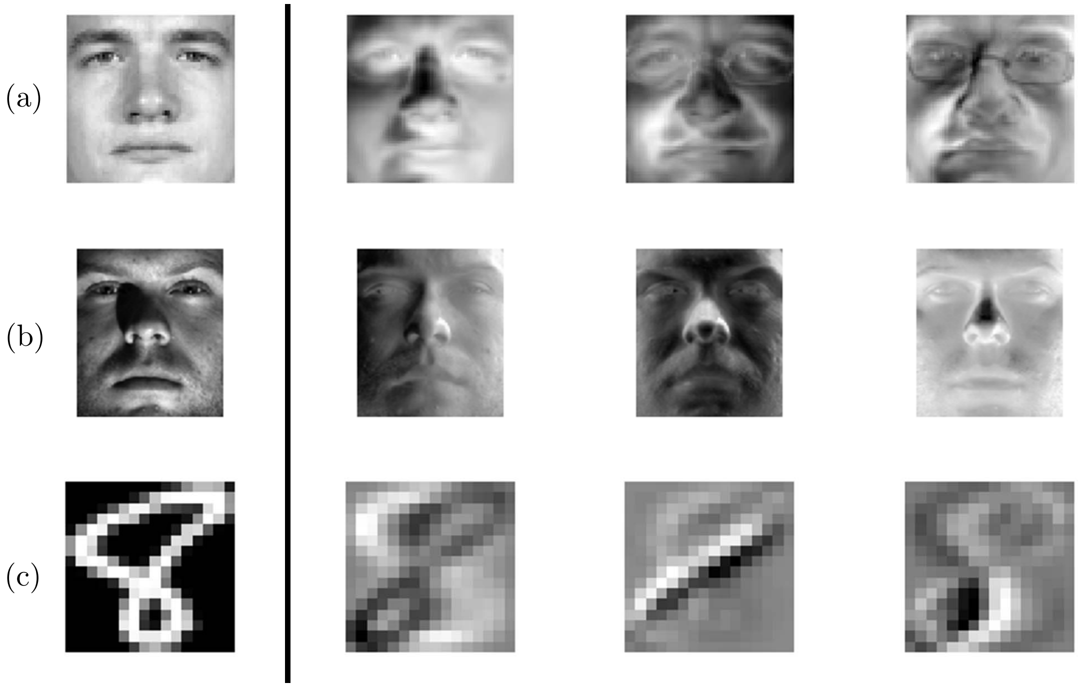


Figure 3.3: FDDL dictionary atoms (right) calculated for original face data (left) on (a) FRGC dataset, (b) YaleB dataset and (c) USPS digits dataset.

with structure incoherence (DLSI) methods. Experiments on face and digit recognition and gender classification confirm that applying FDDL dictionary to SRC will result in significantly higher recognition rates.

3.1.3 Sparse Modeling Representative Selection (SMRS)

While Metaface and FDDL methods try to build a new dictionary by processing and modifying the training data, Sparse Modeling Representative Selection (SMRS) proposes to form the dictionary by selecting its atoms from the original training samples [23]. Basically, this approach searches for a few numbers of training samples which represents all the training data as precise as possible. To achieve this objective at the first step, [23] tries to represent each training sample as a linear combination of all others by optimizing

$$\|V - VW\|_F^2 , \tag{3.12}$$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n] \in \mathbb{R}^{n \times n}$ is the coefficient matrix. To choose d representative samples among all samples, another force should be imposed into the objective function of (3.12). The authors in [23] enforced

$$\|W\|_{0,q} \leq k, \tag{3.13}$$

to (3.12), where k is the maximum number of representatives of interest and $\|W\|_{0,q}$ is the mixed $\ell^{0,q}$-norm operator and counts the nonzero rows of W. In other words, the nonzero rows of the solution W, indicate the indices of the representatives among training data V. In order to make the solution translation-invariant, affine constraint $1^{\mathsf{T}}W = 1^{\mathsf{T}}$ is also added to the objective function. As described in Section 2.3, solving an ℓ^0-norm optimization is NP-hard, so it is usually replaced by its relaxed alternative, i.e. ℓ^1-norm. With this consideration, the sparsity term will be substitute by $\|W\|_{1,q}$ or the sum of ℓ^q-norms of the rows of the coefficient matrix W. This framework is performed for each individual class, so finally, representatives for the i^{th} class of the training data are selected using

$$\hat{W}_i = \underset{W_i}{\operatorname{argmin}} \left\{ \lambda \|W_i\|_{1,q} + \frac{1}{2} \|V_i - V_i W_i\|_F^2 \right\} \quad \text{such that} \quad 1^{\mathsf{T}} W_i = 1^{\mathsf{T}}, \tag{3.14}$$

where λ is the sparsity regularization parameter. The parameter q is selected to be greater than one to make the optimization problem convex [23]. The optimization problem (3.14) is solved using Alternating Direction Method of Multipliers (ADMM) method from [69].

The above scheme also indicates which samples are more informative to represent class i by directly comparing ℓ^q-norms of the selected samples. Moreover, [23]

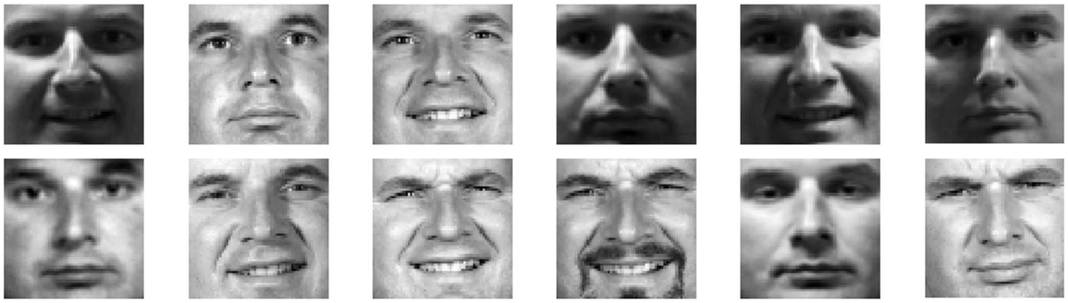


Figure 3.4: Representative selection on one of the subjects FRGC face dataset using SMRS algorithm.

suggests an interesting method to improve the results by detecting outliers after performing an analysis on the coefficient matrix W. The rows of W which correspond to outliers should have a few non-zero entries. Based on this fact, a measure called row-sparsity index (RSI) is introduced which is used to remove outlier representatives. This study also made some discussions on how to efficiently update the representatives for each class when new training samples are introduced. This capability makes SMRS to be a good representative selection algorithm in applications faces dynamically updating datasets.

Figure 3.4 shows the representatives selected using SMRS algorithm. In this experiment, a total number of 80 face images from one of the subjects of FRGC dataset were selected as training samples, q is selected to be 2 and the parameter λ is selected such that final number of selected representatives is equal to 10.By looking at the selections, we can conclude that SMRS has selected face images with a high diversity.

Authors in [23] demonstrate the effectiveness of their proposed approach by conducting experiments in two main categories. First evaluation is choosing best representatives in video sequences to summarize the scenes. Results show that SMRS algorithm selects the most distinctive frames as representatives and rejects frames which are visually similar to the selected ones. The second set of experiments uses the representatives of two datasets (Extended Yale B face and USPS digits) in different classification methods including nearest neighbor, nearest subspace, SVM and SRC. Recognition rate results confirm the effectiveness of SMRS representatives comparing to other methods for selecting representatives (random, k-medoids [3], which is a variant of k-means, and RRQR[1]).

3.2 Efficient SRC using Adaptive Clustering

In this section a method is proposed to reduce the number of columns in matrix V by using an efficient replacement of the original training samples. This method along with its experiments is published as a full conference paper [72].

According to the discussion in Section 2.4.1, face images from one subject form a subspace in the original m dimensional space. Considering the fact that many training face images might contain similar information, using all samples to represent a sub-space is not efficient. In this case, it would be more rational to characterize each class by a more representative and smaller set of sample vectors. For example, consider a dataset with 100 classes with 150 sample per class or a total number of 15000 training face images. Assuming each face is a 60×60 image, the size of the training matrix V will be 3600×15000. If a matrix of size 100×3600 is used for feature extraction (R in (2.20)), then the final size of the matrix $\hat{V}$ in (2.20) will be 100×15000. For comparison, if the average number of samples per class is selected to be 10, the size of the matrix $\hat{V}$ will be reduced into 100×1000. Although random selection of the original training samples may reduce the number of columns in $\hat{V}$, it cannot necessarily represent each class well.

[1] Rank Revealing QR Factorization [70, 71]

Algorithm 3.2: k-means clustering algorithm [1].

```
input        : Data matrix $V_i$ and number of clusters $Q_i$;
- Initialize a $Q_i$-partition randomly (or based on some prior knowledge;)
- Calculate mean vector for all clusters ($\boldsymbol{m}_j, \forall j = 1, \ldots, Q_i$) and form the cluster prototype matrix $M = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{Q_i}]$;
while there is a change in at least one cluster do
    - Re-partition: Assign each object in the dataset to the nearest (minimum Euclidean distance) cluster with the center $\boldsymbol{d}_i^j, j = 1, \ldots, Q_i$;
    - Recalculate cluster prototype matrix;
output       : Cluster Centers $\boldsymbol{d}_i^j$s, $j = 1, \ldots, Q_i$;
```

A widely-used method to represent a group of samples by a smaller number of representatives is clustering. k-means clustering [1, 2] has been frequently used in pattern recognition and machine learning applications and is a method to partition a dataset into k groups. k-means selects k set of cluster centers in data domain which is a more compact representation of the original dataset. This algorithm works based on squared error metric and summarized in Algorithm 3.2.

In the proposed approach, training samples are clustered into a number of groups using an adaptive scheme of k-means clustering. The centers of the clusters are then selected to form the columns of the matrix D which is further substitutes V in the SRC framework. For this purpose, the number of clusters for each class is adaptively selected based on the variability of the training samples for that class. The variability of a cluster is defined as the maximum within-cluster sums of point-to-centroid distance measure

$$MaxDist_i = \max_j \left\{ \sqrt{\sum_{\ell=1}^{N_i^j} \left(\boldsymbol{v}_i^{j,\ell} - \boldsymbol{d}_i^j \right)^2} \right\}, \tag{3.15}$$

where, N_i^j is the number of samples in cluster j of class i, $\boldsymbol{v}_i^{j,\ell}$ is the ℓ^{th} sample in j^{th} cluster of class i and $\boldsymbol{d}_i^j$ represents cluster center of the j^{th} cluster in class i.

In practice, the number of clusters for class i, Q_i, is constrained by a predetermined maximum number of clusters, Q_{max}. Q_i, starts from one and is incremented as long as $Q_i \leq Q_{max}$ and $MaxDist_i$ is larger than a fixed predetermined threshold, τ. Since different classes may contain different variety of samples, applying this approach on the training dataset will result in different number of clusters for each class. This clustering scheme will be called *Adaptive Clustering* (AC) in this document and is summarized in Algorithm 3.3.

The two parameters Q_{max} and τ allow the system to control the number of columns in the final training matrix D. This adjusting feature is a parameter to trade off efficiency for the accuracy. AC algorithm partitions the training samples space to clusters such that classes with high variability end up with more representatives than the classes with lower variability. Moreover, similar training samples are efficiently clustered into one group and represented by a single representative.

3.2.1 Experiments

Experiments on the efficiency of the proposed AC algorithm are conducted in two main stages. First we show the efficiency of the SRC method using AC training matrix comparing to the original SRC method, where all or a random selection of the original training samples were used to form the training matrix. A face recognition application was selected to show how the proposed method works and how efficient and accurate it is comparing to the original SRC method. The next set of experiments

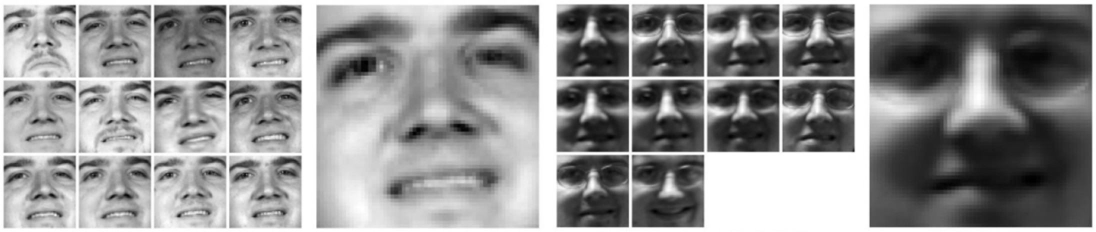
Algorithm 3.3: The proposed *Adaptive Clustering* (AC) algorithm.

```
input  : Training samples $V = [V_1, V_2, \ldots, V_C]$;
         The maximum number of clusters $Q_{max}$ and threshold $\tau$;
for class $i \in \{1, 2, \ldots, C\}$ do
    $Q_i = 2$;
    $M_i = \frac{1}{n_i} \sum_{\ell=1}^{n_i} \boldsymbol{v}_{i,\ell}$;
    $MaxDist_i = \sqrt{\sum_{\ell=1}^{n_i} (\boldsymbol{v}_{i,\ell} - M_i)^2}$;
    while $MaxDist_i > \tau$ or $Q_i \leq Q_{max}$ do
        Run Algorithm 3.2 on samples $V_i$ to form $Q_i$ clusters with centers $\boldsymbol{d}_i^j|_{j=1}^{Q_i}$;
        $MaxDist_i = \max_j \left\{ \sqrt{\sum_{\ell=1}^{N_i^j} \left( \boldsymbol{v}_i^{j,\ell} - \boldsymbol{d}_i^j \right)^2} \right\}$;
        $Q_i$++;
    $D_i = \left[ \boldsymbol{d}_i^1, \boldsymbol{d}_i^2, \ldots, \boldsymbol{d}_i^{Q_i - 1} \right]$;
output : Dictionary $D = [D_1, D_2, \ldots, D_C]$;
```

which are studied in Section 3.3 includes a deep study on using different dictionary learning and representative selection methods which were introduced in Section 3.1. Three different datasets were examined to compare the effectiveness of these methods.

The first set of experiments show the effectiveness of the proposed sample reduction method on FRGC face dataset. Figure 3.5 shows two examples of face cluster centers from one class along with the face images belong to each cluster. As can be seen in this figure, face images which form the cluster on the left and right contain similar information in terms of face expressions and lighting conditions. Here, samples are 60×60 face images or 3600 dimensional vectors.

Figure 3.6 shows the histogram of number of clusters for 100 classes after applying adaptive clustering algorithm. In this experiment, 100 classes are randomly selected from the FRGC face dataset to which AC is applied. Total number of training samples for all classes is 14794 face images and each training sample is a cropped



Original face images in cluster 4 Cluster 4 center Original face images in cluster 6 Cluster 6 center

Figure 3.5: Illustrations of two cluster centers and their members for one subject from FRGC face dataset.

to a 60×60 gray face image, i.e. a 3600 dimensional vector. Adaptive clustering parameters Q_{max} and τ are selected such that after clustering the average number of clusters for all classes is $\sim$10 ($Q_{max} = 20$ and $\tau = 0.42$). As a result, the total number of training samples in this experiment is 1000 which forms a 3600×1000 dictionary D to be used as the SRC training matrix. Changing τ changes the average number of clusters per class which results in a different number of columns in D. It is seen from this figure that most classes ended up with 7-14 cluster per class and there are a few classes with less than 7 and more than 14 clusters which shows the low and high diversity (in terms of pixel values) of the original samples in these classes, respectively.

Figure 3.7 shows the formed cluster centers for one of the face classes in the training dataset. This class has a total number of 112 training samples. AC with $\tau = 0.42$ and $Q_{max} = 20$ was applied to this class and as a result, a total number of 10 clusters are formed. Clusters are formed by different number of face samples which are also included in Figure 3.7 for each cluster.

In next experiment, we investigated the effect of changing parameter τ on the size of the training matrix. This parameter acts as a tradeoff between running time and recognition accuracy. The original training samples contain a total of 14794 face

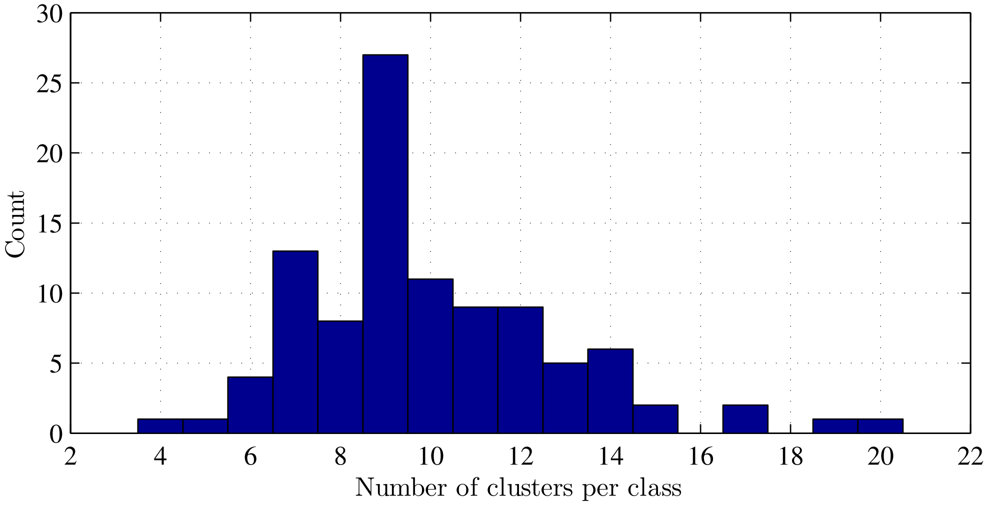


Figure 3.6: Histogram of number of clusters for 100 classes using AC method ($Q_{max} = 20$ and $\tau = 0.42$).

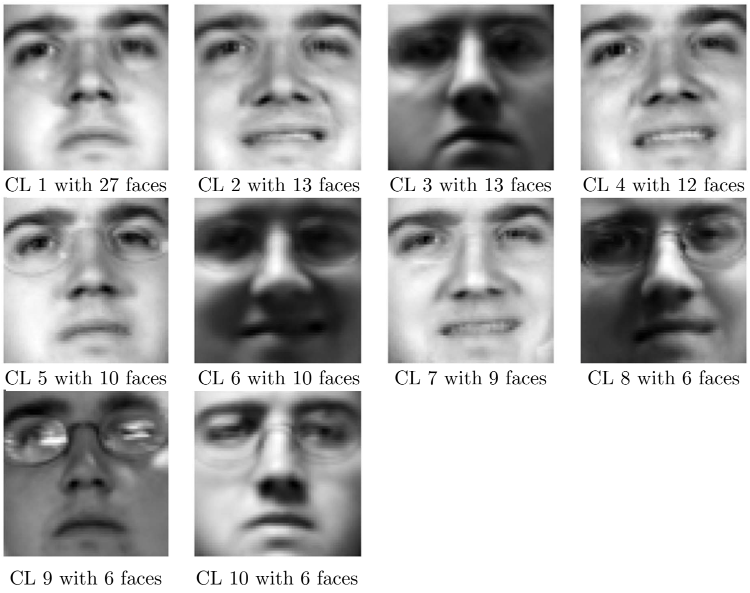

CL 1 with 27 faces
CL 2 with 13 faces
CL 3 with 13 faces
CL 4 with 12 faces
CL 5 with 10 faces
CL 6 with 10 faces
CL 7 with 9 faces
CL 8 with 6 faces
CL 9 with 6 faces
CL 10 with 6 faces

Figure 3.7: Cluster centers formed by AC algorithm for one class from FRGC face dataset.

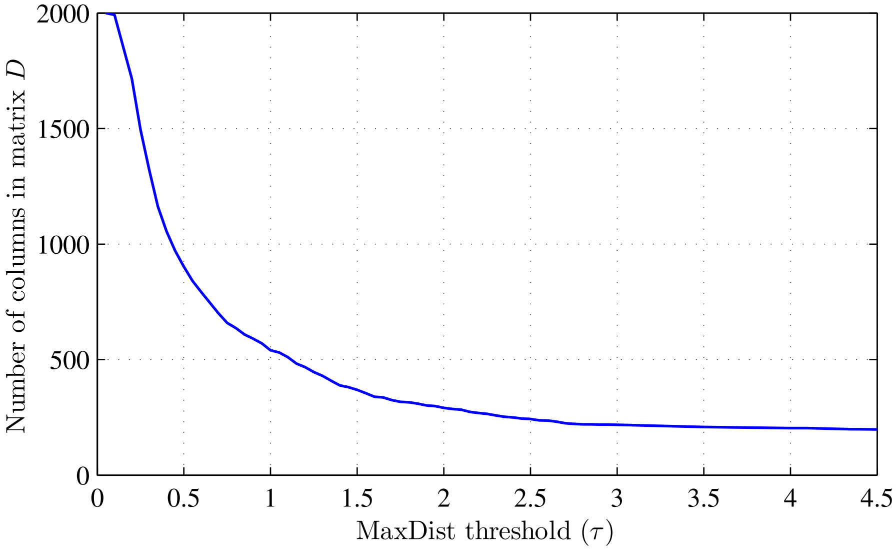
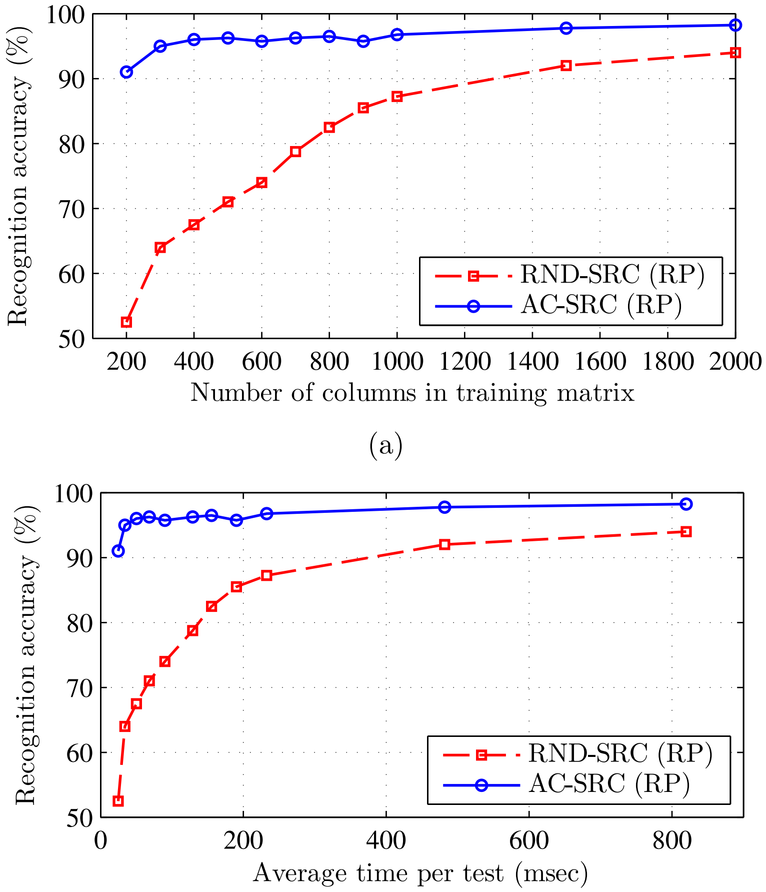


Figure 3.8: The relationship between τ and number of columns in matrix D.

samples from 100 classes. All classes were clustered separately using different values for the parameter τ. Figure 3.8 shows how the size of training matrix D varies by increasing this parameter. Parameter Q_{max} was set to 20 in this experiment. As it can be seen in Figure 3.8, when τ is increased from 0.05 to 4, the number of columns in dictionary D decreases from 2000 to 204.

SRC performance when using random subsets of training data (RND-SRC) and when using the proposed method (AC-SRC) is measured and compared with different number of columns in dictionary D. For the experiments in this section, parameters τ and Q_{max} are selected such that different number of columns are formed in D. Recognition accuracy simulations are performed for 3 different DR methods introduced in Section 2.4.2, random projections (RP), Eigen faces (EIG), and down-sampling (DS). Feature extraction matrix R is selected to have 100 rows in all simulations. Primal-dual algorithm introduced in [57] was used to solve the optimization problem (3.1). In order to compare the recognition accuracy for the two methods, AC parameters (Q_{max} and τ) are tuned to achieve some pre-determined total number of clusters



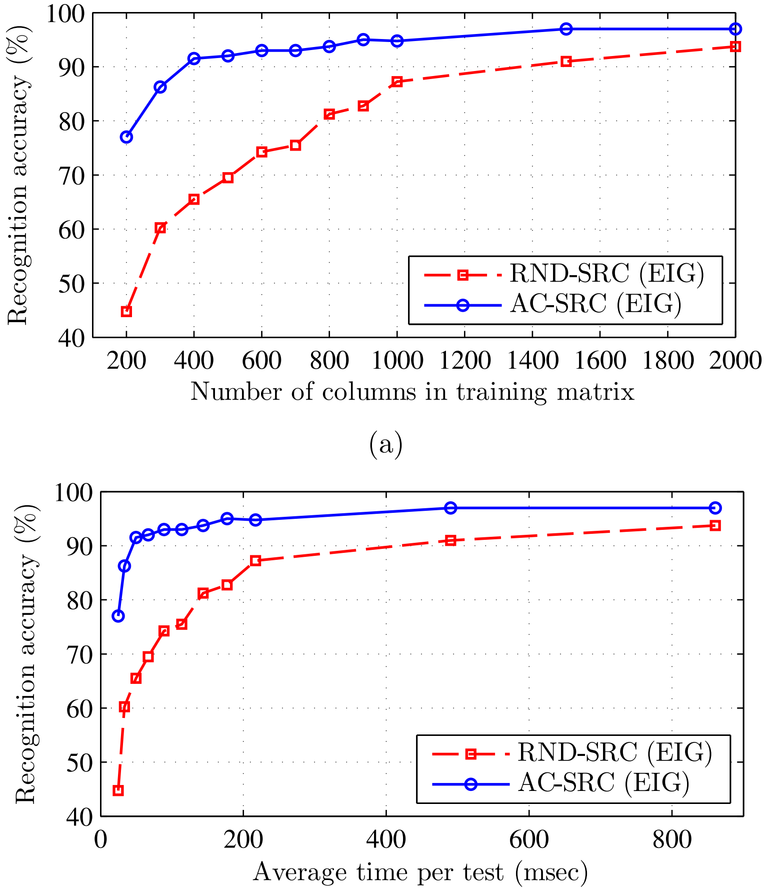
(a)

(b)

Figure 3.9: Recognition rate for AC-SRC using Random projection dimensionality reduction (blue solid line) and original SRC (red dashed line) method (a) using different sizes of matrix D and (b) versus the average execution time for each test.

$(200, 300, \dots)$ for AC-SRC experiments. Then a similar number of training face images were randomly selected from the whole training dataset to form the RND-SRC training matrix to be used for recognition. With this scheme, it is possible to compare the recognition rate of both methods given equal computational load.

Figure 3.9a shows the recognition rates for different sizes of training matrix using random projection. It can be seen that for the same number of columns in the training matrix, using AC dictionary, SRC introduces higher recognition rates com-

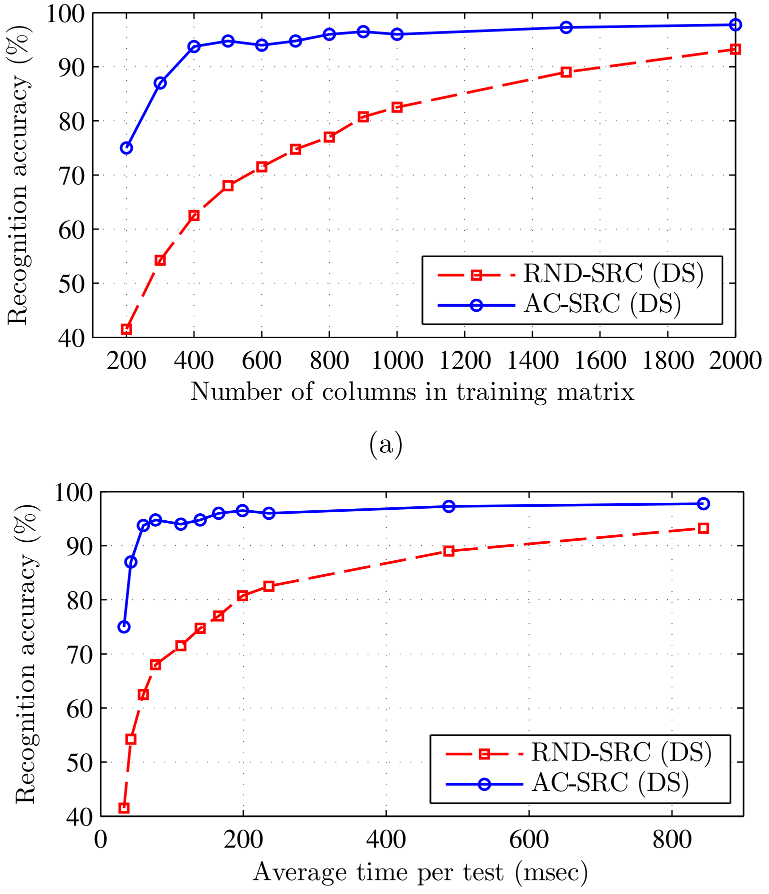


(a)

(b)

Figure 3.10: Recognition rate for AC-SRC using Eigen faces dimensionality reduction (blue solid line) and original SRC (red dashed line) method (a) using different sizes of matrix D and (b) versus the average execution time for each test.

pared to when it uses a random subset of the original training samples. For instance, with a 100×1000 dictionary D (an average of 10 clusters per class), recognition accuracy is %95.75 which is higher than the %85.5 accuracy for the RND-SRC method with the same size of training matrix (average of 10 training sample per class). Note that for the RND-SRC implementation, samples from the training dataset were randomly selected to form matrix D. This random selection is performed 10 times and an average recognition rate is reported as the result.



(a)

(b)

Figure 3.11: Recognition rate for AC-SRC using Down-Sampling dimensionality reduction (blue solid line) and original SRC (red dashed line) method (a) using different sizes of matrix D and (b) versus the average execution time for each test.

Figure 3.10a and Figure 3.11a also show similar results but when EF and DS dimensionality reduction methods were used along in conjunction with SRC, respectively. Again, it is clearly seen that SRC when using AC outperforms SRC when using randomly-selected original training samples. Figure 3.9b, Figure 3.10b and Figure 3.11b show the recognition rate of both approaches versus the average SRC running time per test for the three dimensionality reduction methods. Examination of these results indicates that for the same average time per test, SRC using AC outperforms the RND-SRC. For example, at the 100 msec average time per test, the recognition rates of the AC-SRC are approximately %96, %93, %95 while the recognition rates for the RND-SRC are %76, %75, %70 for the RP, EF, and DS dimensionality reduction, respectively.

The efficiency of using the proposed AC dictionary along with SRC is also shown in Table 3.1. For example, using a subset of 3500 images (an average number of 35 samples per class) from the original dataset and a DS dimensionality reduction, results in a matrix size of 100×3500 for which RND-SRC method achieves %94.2 recognition rate. In contrast, to achieve the same recognition rate, the proposed method only needs an average number of 5 clusters per class, which leads to a 100×500 training matrix. This is a reduction of number of columns in matrix D by a factor of 7 (3500 for RND-SRC to 500 for AC-SRC). Considering the fact that the ℓ^1-norm minimization solution has a quadratic computational complexity, the proposed AC-SRC introduces a significant improvement in the running time for recognition of a single test image. This improvement is reflected in Table 3.1, where the SRC-AC with DS dimensionality reduction leads to a speed improvement of factor 33 (2.64s for RND-SRC to 0.08s for AC-SRC) while achieving even better recognition rates (%94.7 for AC-SRC compared to %94.2 for RND-SRC). Similar improvements are made when RP and EF dimensionality reduction methods are used along with SRC.

Table 3.1: Recognition rate, running time per test and number of columns in matrix D for AC-SRC and RND-SRC methods for classifications of faces from 100 subjects of FRGC dataset using Random Projection (RP), Eigen (EIG) and Down-Sampling (DS) features.

DR	AC-SRC			RND-SRC		
Method	Acc(%)	Time(s)	V Cols	Acc(%)	Time(s)	V Cols
RP	95.0	0.03	300	94.0	0.81	2000
EIG	95.0	0.18	900	94.2	1.33	2500
DS	94.7	0.08	500	94.2	2.64	3500

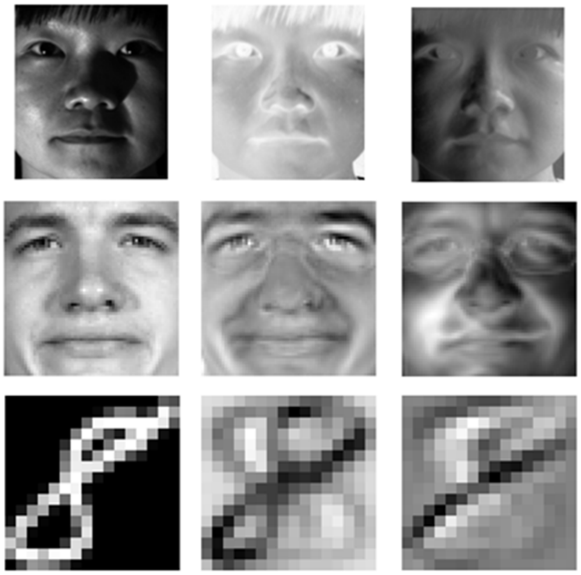


Figure 3.12: SMRS(left), Metaface(middle) and FDDL (right) representatives selected on subjects 3, 4514 and 8 from Extended Yale B (top), FRGC (middle), and Cedar Buffalo (bottom) datasets, respectively.

3.3 Evaluation Studies on the Role of Sample Reduction in SRC

As discussed in Section 3.1, some studies suggested alternatives for large training matrices in SRC to make the whole process faster and space efficient. These sample reduction methods are categorized into dictionary learning based methods and sample selection methods. Three of these approaches i.e. Metaface DL, FDDL, and SMRS are introduced in Section 3.1. Our proposed AC method can also be categorized as a dictionary learning based method.

These sample reduction approaches are evaluated in SRC frameworks on a variety of datasets and the corresponding results are reported in [73]. Experiments are conducted on FRGC and Extended Yale B face and USPS digits datasets introduced in Section 2.2. Figure 3.12 shows a column of SMRS, Metaface (MF) and FDDL dictionaries for three different datasets. As can be seen, SMRS dictionary (left column) selected some of the original data as representatives. On the other hand, Metaface and FDDL methods learn their own representatives which are different from original images.

In this evaluation study, we first build dictionaries for different sample reduction methods. SRC classifies test samples in conjunction with each individual dictionary and their recognition rates and testing time are compared. We also compare these methods to when all the original data and a randomly selected subset of the data is used as the training matrix in SRC. Results are presented separately for Extended Yale B, FRGC and USPS datasets.

3.3.1 Experiments on Extended Yale B Dataset

In the first experiment, SMRS algorithm with a fixed predetermined λ (equation (3.14)) for all classes was applied to training images to select the best representatives to be used as a dictionary. This algorithm selected 8, 9, 10 or 11 (with an average number of 9.58) representatives for each class (364 total representatives). Total running time for SMRS algorithm to build the dictionary was around 20 seconds. After this step, original image vectors of length 32256 were down-sampled into 120 dimensional vectors. Recognition rate using these representatives in an SRC framework is %91.53 while the average classification time for a test sample is 50.87 milliseconds.

Number of selected representatives from the above SMRS experiment, are forced into Metaface and FDDL dictionary learning methods. Metaface dictionary learning process takes approximately 1300 seconds for all classes while FDDL learning time is about 19400 seconds which is dramatically slow in comparison to the other two methods. For the same dimensionality reduction approach (i.e. down-sampling), SRC recognition rates are %86.60 and %92.52 for Metaface and FDDL dictionaries, respectively. Finally, the best recognition rate is reported for our proposed method (AC-SRC) which is equal to %93.09 while the AC learning time is roughly 174 seconds which is acceptable in comparison to other methods learning time. In this experiment, the values for τ and Q_{max} were set such that the average number of formed representatives per class is close to the one from SMRS algorithm.

In order to test the effect of dimensionality reduction on classification results, the above experiments are repeated using random projection dimensionality reduction which projects face images into a 120 dimensional space. The classification is repeated 10 times with different random projection matrices and the average recognition rate using SMRS, Metaface, FDDL and AC dictionary learning methods are %93.25, %88.17, %94.04 and %94.42, respectively.

To complete the experiments, SRC is also deployed as suggested in [7], i.e. with randomly-selected samples (RND-SRC) and also with using all available training samples as the training matrix (ALL-SRC). Random selection of training samples is repeated 10 times and the average recognition rate is reported. Results confirm that using SMRS, FDDL and the proposed AC dictionaries can improve the recognition accuracy especially when DS dimensionality reduction is used. When using all training samples, as expected, SRC achieves highest recognition rates of %97.70 and %98.37 for DS and RP dimensionality reduction processes, respectively. Note that although this approach dominates in terms of accuracy, it is much slower comparing to when selecting a subset or learn a dictionary of samples. Specifically, while matrices as small as 10 representatives per class (a total of 380 samples) can classify a test sample in approximately 50 msec, classification time using a training matrix of all training samples (a total of 1207 samples) is about 390 msec which is approximately 8 times slower. Table 3.2 shows the summary of the learning and classification results on the Extended Yale B dataset.

3.3.2 Experiments on FRGC Dataset

SMRS algorithm is employed first to select training representatives and form the first dictionary. The parameter λ is selected such that the average number of representatives is 12.5 with the total dictionary learning time of 38 seconds. The same numbers of representatives are forced to Metaface and FDDL dictionary learning methods which introduce far longer learning processes (8200 and 91000 seconds respectively). On the other side, AC learning time is 172 seconds which is a relatively acceptable learning time. Again, learning parameters for the adaptive AC method are selected such that the average number of representatives per class is similar to the one for SMRS method.

Learned dictionaries using the four methods are used for SRC classification along with down-sampling dimensionality reduction. This dimensionality reduction changes sample vectors length from 3600 to 100. Recognition rates using SMRS, Metaface FDDL and AC-SRC dictionaries are %94.30, %90.77, %94.10 and %97.06, respectively. For this dataset, similar to Extended Yale B dataset, SRC accuracy is the highest when AC dictionary is used and SMRS and FDDL dictionaries perform better comparing to when Metaface dictionary is employed. The same number of representatives per class is also forced to select 10 random subsets of the training

Table 3.2: Recognition rate (using down-sampling (DS) and random projection(RP)) and learning time for different sample reduction methods on Extended Yale B face dataset.

	%Acc (DS)	%Acc (RP)	Learn Time (s)	Test Time (s)
RND-SRC	85.79	92.49	N/A	0.051
SMRS-SRC	91.53	93.25	20	0.050
MF-SRC	86.60	88.17	1300	0.051
FDDL-SRC	92.52	94.04	19400	0.049
AC-SRC	**93.09**	**94.42**	174	0.045
ALL-SRC	97.70	98.37	N/A	0.39

data. These matrices are used for SRC classification and the average recognition rate of %90.23 is obtained which is about %7 smaller than when AC dictionary is used.

In the next step, random projection is used to reduce the dimensionality of the samples. The average recognition rates are %85.76, %80.65, %88.02 and %91.93 for SMRS, Metaface, FDDL and AC-SRC dictionaries respectively. The average recognition rate reported by randomly selected representatives in this experiment is %83.10 which is again far smaller than most of the dictionary-based classifications.

Regarding testing time, sample reduction methods introduce approximately 0.32 second to classify a test sample when the total number of atoms in the training matrix is about 1250. Although incorporating all samples to form the SRC training matrix ends up in high recognition rates (%96.80 and %97.70 for DS and RP, respectively), the classification of every single FRGC face sample takes about 15 seconds which is about 47 times slower than when smaller dictionaries are used for classification. The results of using different sample reduction methods for classification of FRGC face dataset are summarized in Table 3.3.

3.3.3 Experiments on USPS Digit Dataset

At the first step, SMRS dictionary learning is employed to build the dictionary matrix. Parameter λ is selected to create an average of 24.7 representatives per class

Table 3.3: Recognition rate (using down-sampling (DS) and random projection(RP)) and learning time for different sample reduction methods on FRGC face dataset.

	%Acc (DS)	%Acc (RP)	Learn Time (s)	Test Time (s)
RND-SRC	90.23	83.10	N/A	0.32
SMRS-SRC	94.30	85.76	38	0.32
MF-SRC	90.77	80.65	8200	0.31
FDDL-SRC	94.10	88.02	91000	0.34
AC-SRC	**97.06**	**91.93**	172	0.30
ALL-SRC	96.80	97.70	N/A	15.14

and the dictionary learning time is 85 seconds. Similar to face datasets, Metaface and FDDL dictionary learning methods are incorporated with the same number of representatives per class as SMRS dictionary. Dictionary learning running time is 5377 and 2298 seconds for Metaface and FDDL learning methods, respectively. AC dictionary is built under parameters setting such that the average number of representatives per class is approximately equal to the one from SMRS. Learning time for the AC dictionary learning is 82 seconds which represents the fastest dictionary learning among all methods for digits dataset.   DS and RP are used to reduce the dimensionality of the data from 256 to 64. Recognition rates using down-sampled SMRS, Metaface, FDDL and AC dictionaries are %88.82, %85.93, %90.16 and %87.69, respectively while the average recognition rate over 10 runs of random projection dimensionality reduction are reported %80.83, %79.40, %85.08 and %80.13, respectively. An average testing time of 24 milliseconds is reported for each digit test sample in the above classification framework. Table 3.4 shows SRC recognition rate and dictionary learning time when using SMRS, Metaface, FDDL and AC dictionary learning methods as well as using all and random selections over the training samples for the digit recognition problem.

Table 3.4: Recognition rate (using down-sampling (DS) and random projection(RP)) and learning time for different sample reduction methods on USPS digit dataset.

	%Acc (DS)	%Acc (RP)	Learn Time (s)	Test Time (s)
RND-SRC	82.49	76.20	N/A	0.02
SMRS-SRC	88.82	80.83	85	0.02
MF-SRC	85.93	79.40	5377	0.02
FDDL-SRC	**90.16**	**85.08**	2298	0.02
AC-SRC	87.69	80.13	82	0.02
ALL-SRC	97.50	95.58	N/A	5.22

3.3.4 Interpretation of the Results

From the learning point of view, in the experiments on face data, SMRS is the fastest method. While Metaface and FDDL methods need more than an hour to build the dictionary, the learning time for SMRS is less than a minute. Our proposed adaptive clustering method introduces a learning time about 3 minutes which is way faster than Metaface and FDDL but not as fast as SMRS. In digit recognition experiments, AC introduces the fastest learning time and we can conclude that it is a fast learning method when the dimensionality of the samples is lower but there are a large number of training samples available for each class. SMRS learning time is still reasonable and slightly longer than AC but Metaface and FDDL, introduce very slow learning process on digit recognition experiments. These differences in the learning phase make SMRS and AC the best choice for dynamic situations where the dictionary is regularly updated with new samples. While FDDL method introduces the longest learning process on Extended Yale B and FRGC face datasets, Metaface has the slowest learning curve when classifying digits. This fact implies that Metaface learning time highly depends on the number of training images rather than the dimensionality of the training samples. On the other hand, FDDL needs more time to learn when number of classes and the dimensionality of the data is higher.

Figure 3.13 shows SRC accuracy using different sample and dimensionality reduction methods on the three datasets. Investigation of the results for the four different sample reduction methods show that for both Extended YaleB and FRGC face data, AC-SRC introduces the best recognition rates. The next accurate results obtained when FDDL dictionary is used (except for one case where SMRS introduces %0.2 higher recognition rate than FDDL on FRGC face dataset). SRC recognition rates when using SMRS dictionary are, in general, slightly lower than when FDDL dictionary is applied. For digit recognition experiments, FDDL-SRC is the best clas-

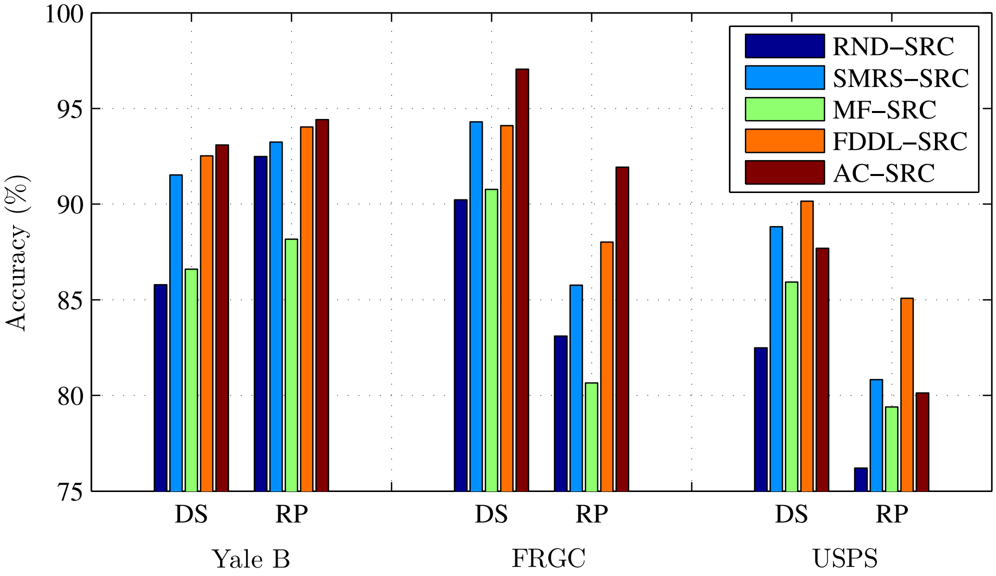


Figure 3.13: Recognition accuracy on different datasets using dimensionality reduction methods down-sampling (DS) and random projection (RP).

sifier in terms of accuracy, while SMRS and AC stand after this method. Metaface dictionary learning method accuracy is lower comparing the other two approaches and even in some cases it is less accurate than when simple random selection of the training data is used as training model (Using RP dimensionality reduction on Extended Yale B and FRGC datasets). As expected, using all the training samples as matrix V in SRC results in the best recognition rates in all experiments but classification of a test sample by this approach is far slower than when any one of the 5 sample reduction methods used in conjunction with SRC. The long classification time makes this approach non-useable in real-life applications and proves the effectiveness of incorporating sample reduction.

Analysis of the SRC recognition rates using down-sampling and random projection dimensionality reduction gives us some interesting results where for the FRGC and USPS datasets, DS is more effective than random projection for all 5 sample reduction methods but this is not the case for the Extended Yale B dataset. This difference may be the result of the characteristics of these datasets. As introduced in Section 2.2, Extended Yale B dataset contains face images which are similar in pose and expressions but only captured in different controlled lighting conditions while the other two datasets were not captured within a controlled environment.

As a summary, one can conclude that using reduced-samples dictionaries in an SRC framework, leads to faster classification process comparing to when all training images are used to form the matrix V in SRC optimization (3.1). Among the selected sample reduction methods, the proposed AC and FDDL introduced the highest recognition rates while the learning curve for FDDL was much slower than AC and SMRS. In comparison to best recognition results, the ones for the SMRS are also acceptable in all datasets. This makes FDDL to be more applicable in off-line applications and AC and SMRS to be more practical when training data is always updating and learning is performed in a dynamic manner.

CHAPTER 4

MULTI-MODAL SPARSE REPRESENTATION CLASSIFICATION

The SRC algorithm and its variants described in Section 2.4 and 3.1 looks at the data space as a single modality (feature) space. In most of classification problems, different modalities of a data sample may contain different discriminative information and using a combination of these modalities would result in better classification rates comparing when focusing on only one modality. There are many classification approaches especially in object recognition which consider multiple feature spaces to increase the classification accuracy [74, 75]. In this chapter, we first review a study which focuses on multi-modality implementation of SRC and then propose 3 methods to improve the efficiency and accuracy of this framework.

4.1 Related Work

4.1.1 Classification Using Multi-task Joint Sparse Representation

In [24], a supplement method was presented on top of SRC which combined the multi-modality property of multitask sparse linear regression [76] and the high classification power of the SRC method. Authors of [24], called their approach Multi-task Joint Sparse Representation and Classification (MTJSRC). MTJSRC utilizes different tasks (modalities/feature spaces) of the training dataset and makes several training matrices to be used in the SRC framework. The test sample may be also represented as multiple feature vectors. An example for this configuration is an image classification problem where R, G and B components of test and training images are

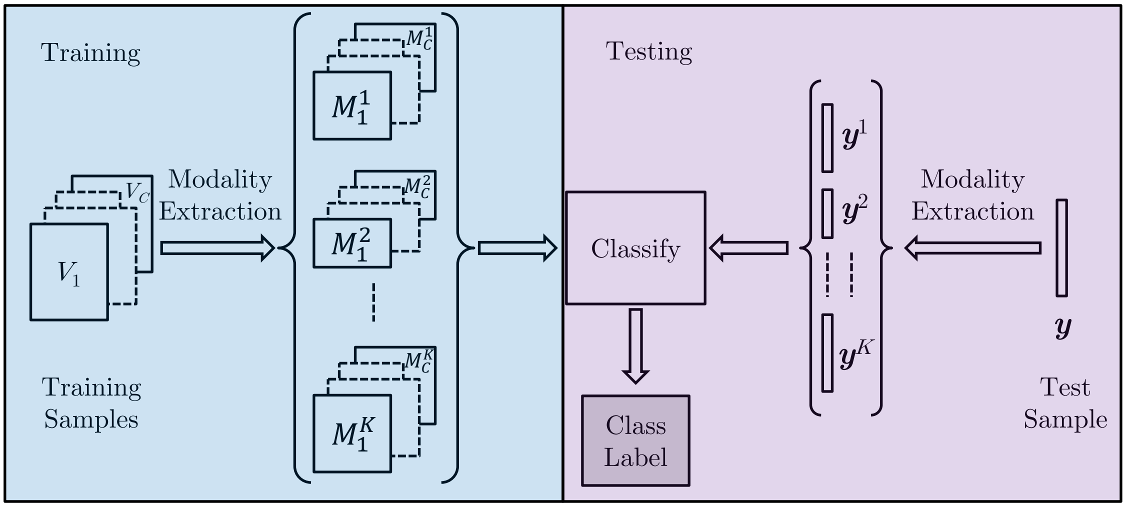


Figure 4.1: An illustration for MTJSRC algorithm.

separately used to form 3 training matrices. A schematic of this method is illustrated in Figure 4.1.

MTJSRC uses a modified version of the original SRC algorithm. Assume the test sample, $\boldsymbol{y}$, and any of the training samples from C classes, V, are represented by K different modalities. Denote by $M^k = [M_1^k, M_2^k, \ldots, M_C^k] \in \mathbb{R}^{m^k \times n}$ the k^{th} modality of the training samples V. Each sub-matrix M_i^k is an $m^k \times n_i$ matrix with m^k and n_i represent the dimensionality of k^{th} modality and the number of training samples in class i, respectively. Similarly, a test sample $\boldsymbol{y}$ can be represented by K vectors, each representing a modality. For the k^{th} modality of $\boldsymbol{y}$, i.e. $\boldsymbol{y}^k \in \mathbb{R}^{m^k}$, equation (2.17) can be reformulated as

$$\boldsymbol{y}^k = M^k \boldsymbol{x}^k + \boldsymbol{e}^k, \tag{4.1}$$

where $\boldsymbol{x}^k \in \mathbb{R}^n$ $(n = \sum_{i=1}^{C} n_i)$ is the representation of $\boldsymbol{y}^k$ of over the matrix M^k and $\boldsymbol{e}^k \in \mathbb{R}^{m^k}$ is the corresponding error vector. The coefficient vector $\boldsymbol{x}^k$ can be decomposed into smaller vectors $\boldsymbol{x}_i^k \in \mathbb{R}^{n_i}$ and each of them is the reconstruction coefficient vector associated with class i ($\boldsymbol{x}^k = \left[(\boldsymbol{x}_1^k)^\mathsf{T}, (\boldsymbol{x}_2^k)^\mathsf{T}, \ldots, (\boldsymbol{x}_C^k)^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^n$). Denote by

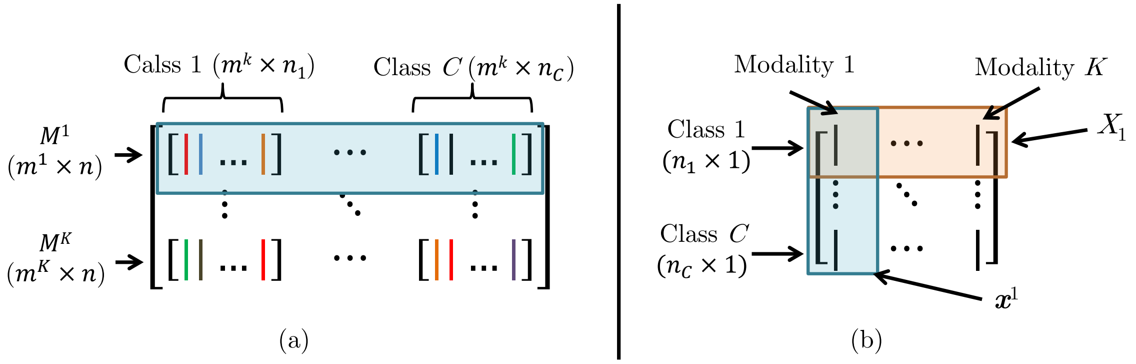


Figure 4.2: A demonstration of the training samples matrix M (a) and the coefficients matrix X (b) in (4.2).

$X = [\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K] \in \mathbb{R}^{n \times K}$ the coefficients matrix contains the coefficient vectors for all K tasks. For a better understanding, Figure 4.2 shows the training matrices M and its corresponding coefficient matrix X. Here, the training matrix is defined as M=$[(M^1)^\intercal, (M^2)^\intercal, \ldots, (M^K)^\intercal]^\intercal$.

In order to classify the test sample, similar to SRC, it is required to recover the coefficient vectors $\boldsymbol{x}^k$s or equivalently, the coefficient matrix X. According to the sparsity constraint in SRC, here, the ideal recovered matrix X for a test sample $\boldsymbol{y}_i$ from class i should be a row-sparse matrix which contains zero entries except for the rows associated with class i, or equivalently entries in sub-matrix X_i. This sub-matrix is called $X_i \in \mathbb{R}^{n_i \times K}$ and $X_i = [\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^K]$ and contains the coefficients associated with all modalities of training samples from class i. Figure 4.2b illustrates this sub-matrix corresponds to class 1 within the overall coefficient matrix X.

In its first step and to recover the coefficient matrix X, MTJSRC solves the optimization problem

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{k=1}^{K} \left\| \boldsymbol{y}^k - M^k \boldsymbol{x}^k \right\|_2^2 + \lambda \left\| X \right\|_{1,2} \right\}, \tag{4.2}$$

which is also known as *group-LASSO*. The notation $\|\cdot\|_{1,2}$ is the mixed $\ell^{1,2}$-norm operator to impose raw-sparsity and is defined as

$$\|X\|_{1,2} = \sum_{i=1}^{C} \|X_i\|_2. \tag{4.3}$$

An explanation on how imposing $\ell^{1,2}$-norm to the objective function helps to find the class associated with the test sample is as follows. first, to combine the strength of all the atoms within class i, ℓ^2-norm (Frobenius norm) is applied over X_i. So there will be a single number (μ_i) associated with each class. All these numbers form the vector

$$\boldsymbol{\mu} = \left[\|X_1\|_2, \|X_2\|_2, \ldots, \|X_C\|_2\right] = \left[\mu_1, \mu_2, \ldots, \mu_C\right]. \tag{4.4}$$

The objective of sparse coding in SRC needs to find a solution in which as few as possible classes get involved in the reconstruction of the test sample. To consider this constraint in the objective function, the term $\|\boldsymbol{\mu}\|_0$ is added. However, as discussed in Section 2.3, solving the non-convex ℓ^0-norm optimization is NP-hard and its relaxed version, i.e. ℓ^1-norm optimization is used or equivalently

$$\|\boldsymbol{\mu}\|_0 = \left\|\left[\|X_1\|_2, \ldots, \|X_C\|_2\right]\right\|_0 = \sum_{i=1}^{C} I\left(\|X_i\|_2 \neq 0\right) \xrightarrow{relaxed} \sum_{i=1}^{C} \|X_i\|_2. \tag{4.5}$$

The optimization problem (4.2) is shown to have an iterative solution which is known as Accelerated Proximal Gradient (APG) method [77, 78]. In this method, optimization is done in two alternative steps which is called *Generalized gradient mapping* and *Aggregation*. A summary of this algorithm which is used by the authors in [24] is shown in Algorithm 4.1. Note that the running cost of this algorithm is mostly come from the gradient calculation (4.6) and the cost for other steps are negligible comparing to this step. If T is the average number of iterations and k_L is the index of the modality with the largest dimension, the computational complexity of

Algorithm 4.1: APG algorithm proposed in [24] to solve (4.2).

```
input         : Modality matrices $M^k|_{k=1}^K$,   Test sample modalities $\boldsymbol{y}^k|_{k=1}^K$
                Sparsity regulizer $\mu$ and step-size $\eta > 0$;
initialization: $t = 0$ ; $\alpha_0 = 0$ ; $G^0 = \boldsymbol{g}^{k,0}|_{k=1}^K = 0_{n \times K}$;
repeat
    Step 1: Given $G^t$ update $X^{t+1}$:
    for modality $k \in \{1, 2, \ldots, K\}$ do
        $\nabla^k = -\left(M^k\right)^\mathsf{T} \boldsymbol{y}^k + \left(M^k\right)^\mathsf{T} M^k \boldsymbol{g}^{k,t};$    (4.6)
        $\boldsymbol{x}^{k,t+1} = \boldsymbol{g}^{k,t} - \eta \nabla^k;$
    for class $i \in \{1, 2, \ldots, C\}$ do
        $\boldsymbol{x}^{k,t+1} = \max\left(\left[1 - \frac{\lambda\eta}{\|\boldsymbol{x}_i^{t+1}\|_2}\right], 0\right);$
    Step 2: Given $X^t$ and $X^{t+1}$ update $G^{t+1}$:
    $\alpha_{t+1} = \frac{2}{t+3};$    $\gamma = \frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t};$
    for modality $k \in \{1, 2, \ldots, K\}$ do
        $\boldsymbol{g}^{k,t+1} = \boldsymbol{x}^{k,t+1} + \gamma\left(\boldsymbol{x}^{k,t+1} - \boldsymbol{x}^{k,t}\right);$
    $t \leftarrow t + 1;$
until Convergence or Maximum Iteration;
output        : Matrix $X = [\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K]$;
```

(4.6) is $\mathcal{O}\left(Knm^{k_L} + 2TKnm^{k_L}\right)$ [24] which can also be considered as the complexity for the whole Algorithm 4.1.

Given recovered coefficient vectors for different tasks, the class of the unknown test sample can be calculated in a similar way to (2.19) i.e.

$$class(i) = \hat{i} = \underset{i}{\operatorname{argmin}} \sum_{k=1}^{K} \left\|\boldsymbol{y}^k - M_i^k \boldsymbol{x}_i^k\right\|_2^2. \tag{4.7}$$

Authors of [24] showed while the original SRC method achieves acceptable recognition rates by using only one feature of the data, combining different modalities using MTJSRC improves the results in face and object recognition applications. Experiments in this study were conducted on Extended Yale B face datasets using two features and on Oxford flower dataset [79] with 7 features. Classification accuracies were better when using multi-feature SRC comparing to when single feature SRC classifiers were used. Moreover, they compared their method to other multi-feature classifiers such as SVM and nearest subspace where the results confirmed higher accuracy for the proposed method.

MTJSRC challenges the accuracy of SRC by incorporating multiple modalities. It is shown that when modalities are selected in such a way that they cover different aspects of the data space, MTJSRC outperforms SRC in different applications such as face and object recognition [24]. MTJSRC uses the modalities from all samples directly to form its models, thus its time complexity depends on both the number of modalities and their dimensionality. Despite the good results reported in [24], similar to SRC, MTJSRC's performance degrades when there are a large number of training samples available as training model. In comparison to SRC, this problem is even more intense in this case since there are multiple training matrices each of them imposing its own time complexity to the solution. This complexity imposes some limitations on using this method in practical applications facing a large number of training data with high dimensional modalities. Moreover, different modalities may contain redundancy, un-used information, and noisy data which decrease the classification robustness.

4.2 Efficient and Accurate Classifiers Using Sparse Representation

In this section, we propose methods based on sparse representation which classify test samples in a more accurate way by incorporating multiple modalities and in the same time performs in a more efficient manner (both time and space) by substituting the original data by an efficient representation of the data in the training phase. For this purpose, we employ the sample reduction approaches discussed in Chapter 3.

In this study, incorporating sample reduction methods into MTJSRC to tackle its limitation on large number of training samples is approached in three ways.

1- Red-Mod: (First sample reduction, then modality extraction) In this approach, training samples are fed into a sample reduction method to come up with a more abstract representation and then, modalities are extracted to be used in the optimization process and final classification. Since all modalities are extracted from a same set, they end up with equal number of columns and the problem can be solved in a similar way of MTJSRC by employing all modality matrices directly in the optimization problem (4.2) and recovering the sparse coefficients. This approach is described in details in Section 4.2.1 and is published in [80].

2- Uniform Mod-Red: (First modality extraction, then uniform sample reduction) Due to the fact that different modalities come from different natures and the data in each modality space has its own information overlaps and redundancies, the optimum number of representatives may vary among different modalities. By extracting the features in the first step, Approach 1 ignores this fact. To consider this issue, second approach extracts the modalities from the original training samples at the first step and then apply sample reduction methods to come up with abstract modality matrices. These matrices are then used in a multi-modality objective function which is optimized to recover the coefficient matrix. From this category, in Section 4.2.2, we propose a multi-modal SRC-based method with Fisher discrimination sample reduction which is published in [81].

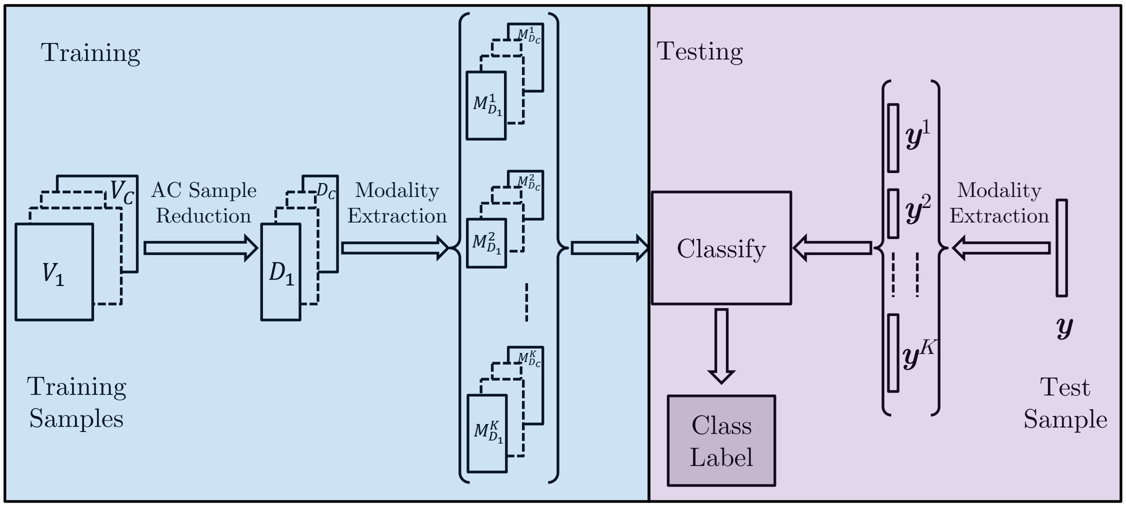


Figure 4.3: An illustration for CMSRC method.

3- Non-Uniform Mod-Red: (First modality extraction, then non-uniform sample reduction) Although approach 2 applies dictionary learning algorithms directly on modality matrices, it enforces same number of representatives on all modality matrices from a same class which can be considered as a limitation. In real world data, over a same set of data, the representative power of a class may differ among different modalities. To address this limitation, in Section 4.2.3 we present a multi-modality sparse representation-based classification method which can handle modality matrices with different number of atoms in each class. This approach is also published in [82].

4.2.1 Cluster-Based Multi-task Classification Using Sparse Representation

In this section, a multi-modal dictionary-based framework is presented for a more efficient and, in the same time, accurate classification which is called Cluster-based Multi-task Sparse Representation Classification (CMSRC). This method is categorized to be a method from approach 1 and its block diagram is shown in Figure 4.3.

Method In the first step, adaptive clustering algorithm (Algorithm 3.3) is applied to the training matrix V to build a dictionary with different number of representatives for each class. Representatives' cardinality depends on the variability of the training samples in each class and is tuned by the two parameters τ and Q_{max}. After processing the training images by this algorithm, sub-dictionaries D_i with different number of atoms are formed for each class. These sub-dictionaries then form the super dictionary D which contains representatives for all samples from all classes.

The next step is to extract modalities from each atom in the dictionary D. Different modality matrices will be formed which are noted by $M_D^k \in \mathbb{R}^{m^k \times d}$. Note that modality matrices extracted from dictionaries can be decomposed into class-modality matrices or $M_D^k = \left[M_{D_1}^k, M_{D_2}^k, \ldots, M_{D_C}^k\right]$. These modality matrices are then used in a framework to classify an unknown test sample. A test sample $\boldsymbol{y}$ which is a single-modality vector can be also represented as a multi-modal vector of different dimensions noted by $\boldsymbol{y}^k$. For the k^{th} modality we have

$$\boldsymbol{y}^k = M_D^k \boldsymbol{x}^k + \boldsymbol{e}^k, \tag{4.8}$$

where $\boldsymbol{x}^k \in \mathbb{R}^d$ is the representation of the k^{th} modality of the test sample $\boldsymbol{y}$ over the modality matrix M_D^k and $\boldsymbol{e}^k \in \mathbb{R}^{m_k}$ represents the error term. The coefficient vector $\boldsymbol{x}^k$ is formed by sub-vectors $\boldsymbol{x}_i^k \in \mathbb{R}^{d_i}$ associated with class i ($\boldsymbol{x}^k = \left[\left(\boldsymbol{x}_1^k\right)^\mathsf{T}, \left(\boldsymbol{x}_2^k\right)^\mathsf{T}, \ldots, \left(\boldsymbol{x}_C^k\right)^\mathsf{T}\right]^\mathsf{T}$). Similar to what discussed in Section 4.1.1, the coefficient matrix X can be recovered by solving

$$\underset{X}{\operatorname{argmin}} \left\{\frac{1}{2}\sum_{k=1}^{K} \left\|\boldsymbol{y}^k - M_D^k \boldsymbol{x}^k\right\|_2^2 + \lambda \left\|X\right\|_{1,2}\right\}. \tag{4.9}$$

To solve (4.9), we used Algorithm 4.1 with substituting M^k by M_D^k. In the classification step, the class of the unknown test sample is determined by placing the recovered coefficient matrix X in (4.7).

Experiments We performed face recognition experiments to show the effectiveness of the proposed CMSRC method in comparison to other common classifiers including SRC, AC-SRC, MTJSRC, Nearest Subspace (NS) Classifier [83], Nearest Neighbor (NN) Classifier [15] and Support Vector Machines (SVM) [18] with linear kernel.

Comparison to Single-Modality SRC-based Approaches We first compare the recognition rate of the proposed method (CMSRC) to the single modality RND-SRC (where a random subset of training images are used as training matrix) and AC-SRC. To exactly mimic SRC process in [7], we randomly select a subset from the training set and then use their GS and LBP matrices in an SRC framework separately. For fixed number of samples in each class (4∼12 for Extended Yale B and 2∼12 for FRGC datasets), the sample selection and classification is repeated 10 times and the average and standard deviation of recognition rates are reported. To compare with AC-SRC, Algorithm 3.3 is separately applied to GS and LBP modalities of all training samples. Tuning parameters τ and Q_{max} are selected such that the average number of clusters per class for each modality become equal to what used for the SRC experiment. Finally, the learned dictionaries are fed into SRC for classification.

In the next step, to run our proposed CMSRC on the data, first, adaptive clustering algorithm is applied to all training samples. Again, we tune the algorithm to come up with the same number of representatives per class as what is used for previous simulations. Then, feature vectors are extracted from the representatives to form the modality matrices. Before using the modality matrices M_D^ks in (4.9), all their columns are down-sampled to 100 dimensional vectors. This dimensionality reduction is necessary to convert the classification problem to an under-determined system of linear equations (details are discussed in Section 2.4.2). After the optimization

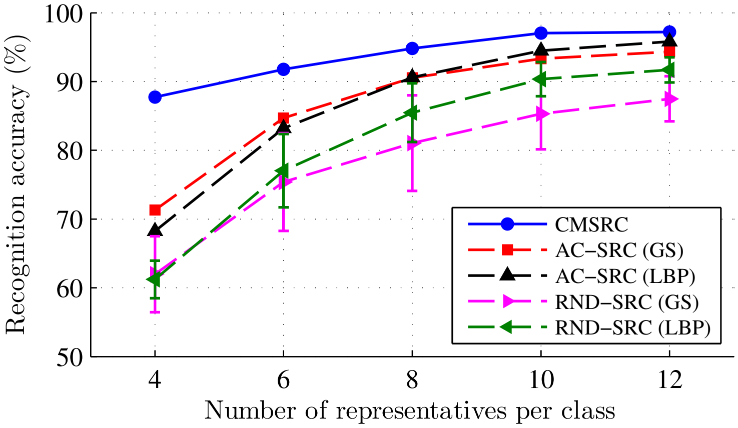


Figure 4.4: Recognition rates of single modality RND-SRC, AC-SRC and CMSRC on Extended Yale B dataset.

process, given the recovered coefficients X, classification task is completed using (4.7).

Figures 4.4 and 4.5 show the recognition rates for Extended Yale B and FRGC face datasets, respectively. As can be seen for both datasets and with different number of representatives per class, the proposed method performs more accurately than the original SRC and AC-SRC which are two single-modality approaches.

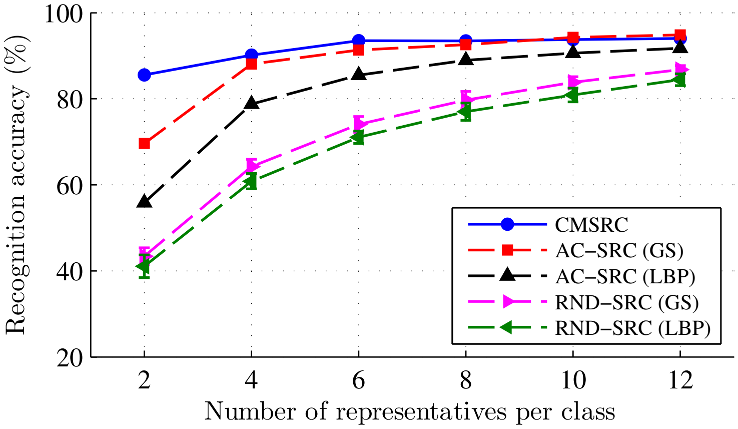


Figure 4.5: Recognition rates of single modality RND-SRC, AC-SRC and CMSRC on FRGC dataset.

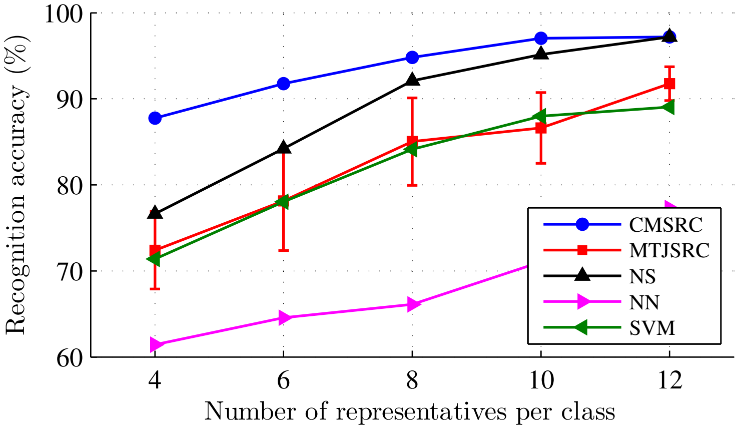


Figure 4.6: Recognition rates of multi-modality classifiers on Extended Yale B dataset.

Comparison to Multi-Modality Approaches The proposed method is also evaluated against other multi-modality classification approaches. Similar to previous experiment, we use fixed number of representatives for all classifiers. MTJSRC, as performed in [24], is driven by randomly selected training samples. We repeat this selection 10 times and report the average recognition rate. For other classifiers, including Nearest Subspace (NS), Nearest Neighbor (NN) and Support Vector Machine (SVM), similar to our previous experiment on CMSRC, first the number of representatives is reduced by applying Algorithm 3.3 and then used the final representatives as training models for classification. Figures 4.6 and 4.7 show the classification rates for this experiment on Extended Yale B and FRGC datasets, respectively. It is clearly seen that the proposed method achieves higher recognition rates when compared to other classifiers. In particular, for small number of training representatives, CMSRC shows significant improvements in comparison to other classifiers.

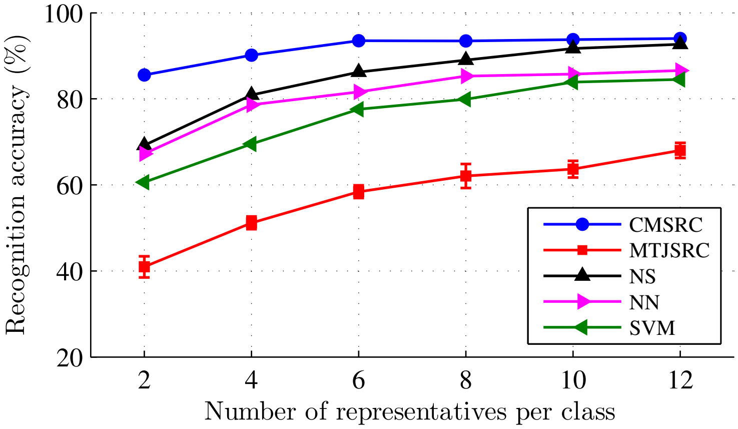


Figure 4.7: Recognition rates of multi-modality classifiers on FRGC dataset.

4.2.2 Multi-modal Sparse Representation Classification with Fisher Discrimination Sample Reduction

In this section, we propose a multi-modal dictionary-based classification method called Multi-modal Fisher discrimination Sparse Representation Classification. Our approach in this section is to first extract the modalities and then perform sample reduction on the extracted modalities individually. Since for all modality matrices, sample reduction is employed by enforcing equal number of representatives per class, this method is considered to belong to approach 2, Modality/Uniform Reduction. The block diagram of this method is illustrated in Figure 4.8.

Method Given the matrix of training samples V, a total number of K modalities are first extracted to form matrices $M^k, k = 1 \ldots K$ which are then compressed into dictionaries $D^{M^k} \in \mathbb{R}^{m_k \times d}$, where $d = \sum_{i=1}^{C} d_i$ is the number of atoms in the dictionary (d_i is the number of representatives of class i in D^{M^k}). To achieve this

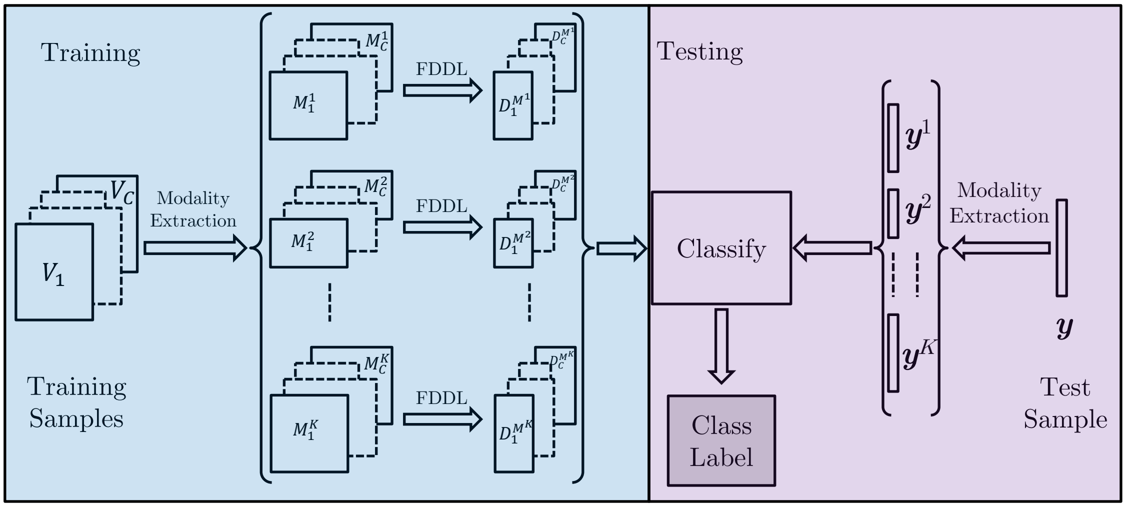


Figure 4.8: An illustration for MMFSRC method.

compression, FDDL (Section 3.1.2) is employed in a multi-modality scheme which solves

$$\underset{D^{M^k}, W^k}{\operatorname{argmin}} \left\{ \sum_{i=1}^{C} r\left(M_i^k, D^{M^k}, \left[W^k\right]_i\right) + \lambda_1 \left\|W^k\right\|_1 + \lambda_2 f\left(W^k\right) \right\}. \tag{4.10}$$

The modality dictionary D^{M^k} is formed by class-specific sub-dictionaries $D_i^{M^k} \in \mathbb{R}^{m^k \times d_i}|_{i=1}^{C}$ and correspondingly, the coefficient sub-matrix $\left[W^k\right]_i$ (derived from the matrix of all coefficients W^k) represents modality matrix for class i, M_i^k over the dictionary D^{M^k}. $\left[W^k\right]_i$ can be considered as a row-concatenation of sub-matrices $\left[W^k\right]_i^j \in \mathbb{R}^{d_j \times n_i}$ (Figure 4.9). Similar to what discussed in Section 3.1.2, dictionary D^{M^k} is a good representative for the modality matrix M^k if 3 constraints are satisfied. First, the whole dictionary D^{M^k} should represent all atoms associated with class i or mathematically, $M_i^k \cong D^{M^k}\left[W^k\right]_i$. Second, $M_i^k \cong D_i^{M^k}\left[W^k\right]_i^i$ to make the sub-dictionary $D_i^{M^k}$ to be a good representative for atoms associated with class i. Finally, for $j \neq i$, a good discriminative dictionary may keep $D_j^{M^k}\left[W^k\right]_i^j$ as small as possible to make the sub-dictionary for class i a "not good" representatives for other classes. Therefore, discriminative fidelity term $r(M_i^k, D^{M^k}, \left[W^k\right]_i)$ is defined as

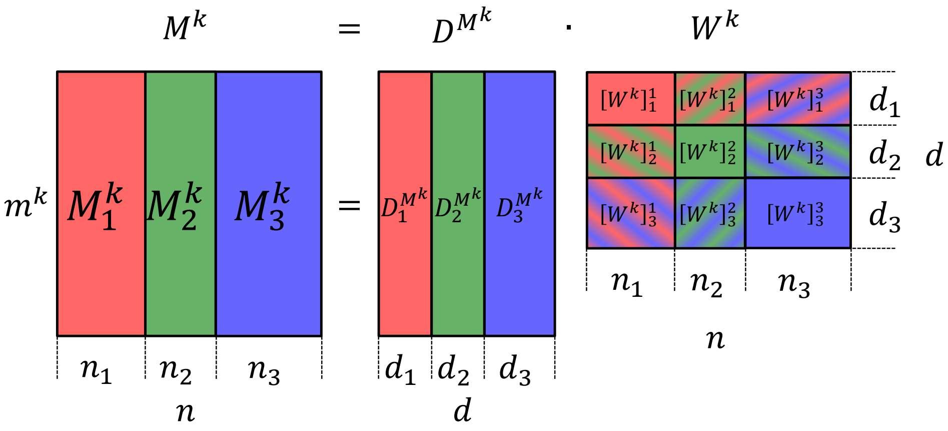


Figure 4.9: A visualization of original modality matrix M^k, dictionary D^{M^k} and the coefficient matrix W^k.

$$\left\| M_i^k - D^{M^k} \left[W^k\right]_i \right\|_F^2 + \left\| M_i^k - D_i^{M^k} \left[W^k\right]_i^i \right\|_F^2 + \sum_{\substack{j=1 \\ j \neq i}}^{C} \left\| D_j^{M^k} \left[W^k\right]_i^j \right\|_F^2 . \tag{4.11}$$

The Fisher discrimination criterion $f(\cdot)$ is forced to make the classes within the dictionary D^{M^k} as discriminative as possible by minimizing the within-class and maximizing the between-class scatter of matrix W^k [22]. These two scatter matrices are denoted by S_ω and S_β, respectively and defined as

$$S_\omega(W^k) = \sum_{i=1}^{C} \sum_{\boldsymbol{w} \in \left[W^k\right]_i} \left(\boldsymbol{w} - \boldsymbol{m}_i^k\right) \left(\boldsymbol{w} - \boldsymbol{m}_i^k\right)^\mathsf{T} \tag{4.12}$$

and

$$S_\beta(W^k) = \sum_{i=1}^{C} n_i \left(\boldsymbol{m}_i^k - \boldsymbol{m}^k\right) \left(\boldsymbol{m}_i^k - \boldsymbol{m}^k\right)^\mathsf{T}, \tag{4.13}$$

where $\boldsymbol{m}_i^k$ and $\boldsymbol{m}^k$ are the mean vector of $\left[W^k\right]_i$ and W^k, respectively. Given above definitions, the Fisher discriminative term is defined as

$$f(W^k) = \operatorname{tr}(S_\omega(W^k)) - \operatorname{tr}(S_\beta(W^k)) + \eta \left\| W^k \right\|_F^2 . \tag{4.14}$$

Algorithm 4.2: Multi-modal Fisher discrimination DL.

```
input   : training samples $V$;
          desired number of representatives for all classes, $d_i|_{i=1}^{C}$;
          regularization parameters $\lambda_1$, $\lambda_2$;
for $k \in \{1, 2, \dots, K\}$ do
    Extract modalities $M^k$ from original samples $V$;
    Initialize $D^{M^k}$s with random coulumns with unit $\ell^2$-norm;
repeat
    for $k \in \{1, 2, \dots, K\}$ do
        for $i \in \{1, 2, \dots, C\}$ do
            Fix $D^{M^k}$ and solve (4.10) for $\left[W^k\right]_i$;
        for $i \in \{1, 2, \dots, C\}$ do
            Fix $W^k$ and solve (4.10) for $D_i^{M^k}$;
until Convergence or Maximum Iteration;
output  : Dictionaries $D^{M^k}|_{k=1}^{K}$;
          Coefficients $W^k|_{k=1}^{K}$;
```

Authors in [22] show that (4.14) is a convex function which helps the overall (4.10) to be a convex optimization problem.

The main optimization problem (4.10) is solved for every modality k and can be approached as an alternative optimization problem where W^k is updated while D^{M^k} is fixed in the first step and vice versa in the second step. This process is iteratively repeated until convergence [22]. Algorithm 4.2 summarizes the optimization process of (4.10).

Given the unknown test sample and the dictionaries for all modalities, D^{M^k}s, the objective is to reconstruct the coefficient matrix X in

$$\underset{X}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{k=1}^{K} \left\| \boldsymbol{y}^k - D^{M^k} \boldsymbol{x}^k \right\|_2^2 + \lambda \left\| X \right\|_{1,2} \right\}. \tag{4.15}$$

Since numbers of dictionary atoms per class (d_is) are equal for all modalities, it is possible to use the APG algorithm (Algorithm 4.1) by only replacing M^k with D^{M^k}. Given the recovered coefficient matrix, one can determine the class of test sample $\boldsymbol{y}$ by placing X and D^{M^k} in (4.7).

Experiments Several experiments are conducted to show the efficiency and accuracy of the proposed MMFSRC method in face recognition applications. SRC-based methods and more specifically, when using FDDL sample reduction are shown to be effective in other applications such as object and digit recognition [22]. In this section, MMFSRC is compared to both single modality approaches SRC and FDDL-SRC Section 3.1.2 and other multi-feature approaches including MTJSRC, Nearest Subspace, Nearest Neighbor and Support Vector Machines with linear kernel. Experiments are conducted on YaleB and FRGC face datasets introduced in Section 2.2.

Comparison to Single-Modality SRC-based Approaches These set of experiments compares the accuracy of MMFSRC to the single modality approaches of SRC and FDDL-SRC. Modalities are extracted from all training samples which ends in matrices of size 1024×1216 (GS) and 900×1216 (LBP) for Extended Yale B and 1024× 5000 (GS) and 900×5000 (LBP) for FRGC face samples. At first, the original SRC is executed on each individual modality separately. For this purpose, we randomly selected fixed number of columns (4∼12 for Extended Yale B and 2∼12 for FRGC) from GS and LBP training matrices which are then separately used as SRC training matrices

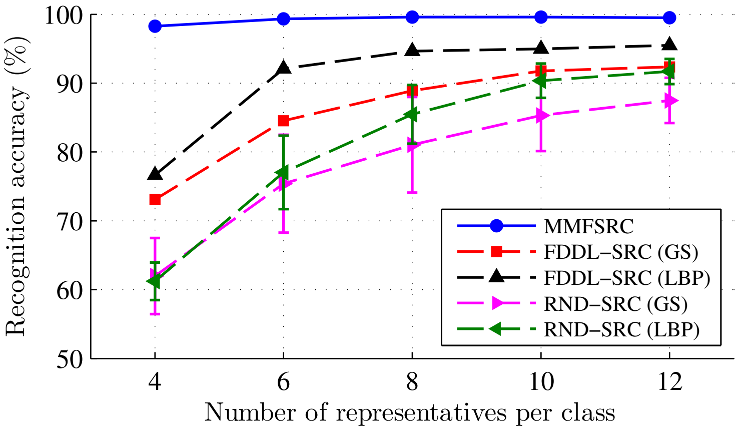


Figure 4.10: Recognition rates of single modality RND-SRC, FDDL-SRC and MMF-SRC on Extended Yale B dataset.

(V in (3.1)). In order for the optimization problem to recover a sparse coefficient vector, we need to satisfy the compressive sensing conditions (Section 2.3) and for this purpose, the system of linear equations (2.16) is needed to be an under-determined system of linear equations. To achieve this, down-sampling is used to reduce the dimensionality of the samples to 100. Finally, the coefficient vector $\boldsymbol{x}$ is recovered by solving (3.1) and the identity of test samples $\boldsymbol{y}$ is determined by applying (2.19). This random selection of training modalities is repeated 10 times and after running SRC, average recognition rates for different sizes of training matrices are reported in Figures 4.10 and 4.11 for Extended Yale B and FRGC datasets, respectively.

Equation (3.8) is then used to form two dictionaries while λ_1 and λ_2 are set to 0.005 and 0.05, respectively. As discussed previously, to build FDDL dictionaries, it is possible to force the number of atoms per class. To achieve a fair comparison with SRC experiments in terms of time complexity, 4∼12 and 2∼12 (for Yale B and FRGC) atoms per class are enforced to the dictionaries. These dictionaries are then used individually instead of SRC training matrix and after recovering the coefficient

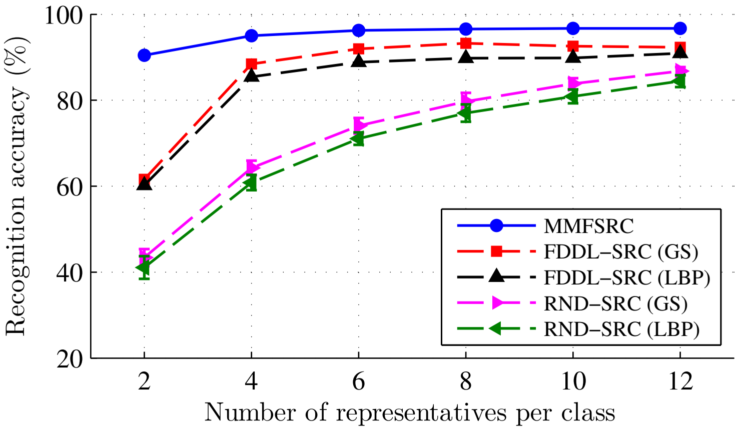
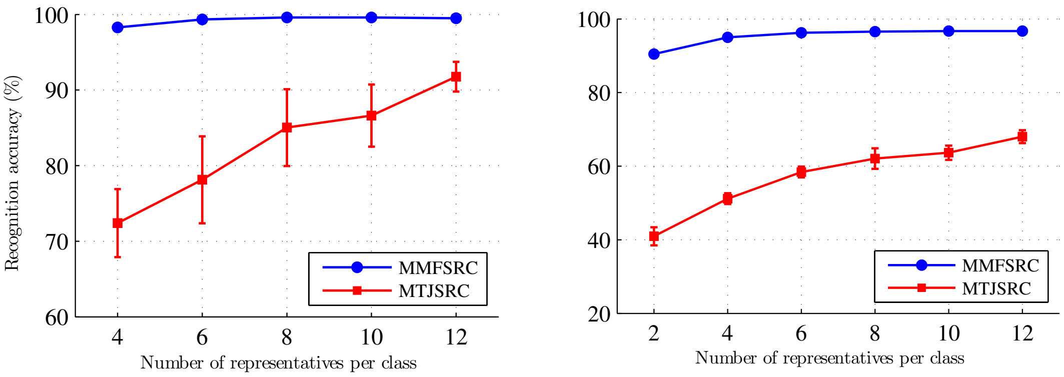


Figure 4.11: Recognition rates of single modality RND-SRC, FDDL-SRC and MMF-SRC on FRGC dataset.

vectors, classification is done over the test dataset. The corresponding results are reflected by FDDL-SRC(GS) and FDDL-SRC(LBP) in Figures 4.10 and 4.11.

Finally, the proposed MMFSRC method, i.e. equation (4.15), is employed given the two modality dictionaries and the set of test data. Solid blue line in Figures 4.10 and 4.11 illustrates the recognition rates of this method given different number of representatives per class. It can be seen that for similar number of columns, higher recognition rates are achieved using MMFSRC comparing to both SRC and FDDL-SRC which run in a single modality scheme. Note that, for example in Extended Yale B experiments, MMFSRC achieves %98.27 recognition rate by only using a total number of 152 representatives (4 per class) while the second best classifier, i.e. FDDL-SRC with LBP features, at its best accuracy (a total number of 456 representatives), has a recognition rate of %95.47.

Comparison to Multi-Modality Approaches The next evaluation illustrates the accuracy of MMFSRC in comparison to MTJSRC. Since MTJSRC uses a selection of training set to form its training matrix, we repeated this algorithm 10 times us-



(a) (b)

Figure 4.12: Recognition rates on (a) YaleB and (b) FRGC datasets, for MMFSRC and MTJSRC.

ing randomly selected training samples. The average of the recognition rates using this method is compared to MMFSRC results (using the same setup as previous experiment) in Figures 4.12a and 4.12b for Extended Yale B and FRGC datasets, respectively. It can be seen that, especially when the size of the training matrix is very small, the proposed method outperforms MTJSRC. Even when larger number of representatives (12 per class) are selected to form the training matrix in MTJSRC, it only achieves %91.76 and %68.01 rates for Extendd Yale B and FRGC, respectively while with the same number of representatives (and same running time), MMFSRC achieves %99.50 and %96.73 accuracy.

In final experiment, we used the modalities from all the training samples to train different classifiers while MMFSRC uses dictionaries with only 12 representatives per class. The corresponding classification rates are reported in Table 4.1 where the proposed method achieves recognition rates of %99.50 and %96.73 while MTJSRC even when all training features are used to build its training matrices introduces smaller accuracies of %99.01 and %95.43 for Extended Yale B and FRGC datasets, respectively. Note that, for example, in experiments on FRGC, MMFSRC uses two

100×1200 matrices while MTJSRC uses two 100×5000 matrices which imposes a far larger time complexity to the algorithm. More specifically, given the group-LASSO computational complexity (Section 4.1.1) and with the same number of iterations, the proposed method is not only more accurate than MTJSRC, but also classifies each FRGC test sample more than 4 times faster. Experiments also show that when using all samples, MTJSRC needs more iterations to converge which results in a much slower classification. Table 4.1 also shows the recognition rates using other classification approaches using all training samples in training phase. Among these classifiers, nearest subspace is shown to perform the best where its recognition rates are slightly smaller and larger on Extended Yale B and FRGC datasets, respectively.

4.2.3 Non-Uniform Multi-Modal Sparse Representation Classification

In the context of multi-task sparse coding (Section 4.1.1), all the assumptions and formulations are valid for the cases where the training model (matrices M^k) are directly formed by the original samples. In this situation, given the training samples V, different modalities are extracted from individual samples (columns of V) and stored in modality matrices M^ks. Therefore each column in any of the modality matrices M^k has a corresponding column in all other modality matrices ($M^{k'}, k' \neq k$) and as a result, all modality matrices have equal number of columns for each

Table 4.1: Classification accuracy for different classifiers by using all training samples for training.

Classifier	Accuracy (%)	
	Extended Yale B	FRGC
SVM	94.32	93.23
NN	94.32	90.40
NS	98.60	**96.93**
MTJSRC	99.01	95.43
MMFSRC	**99.50**	96.77

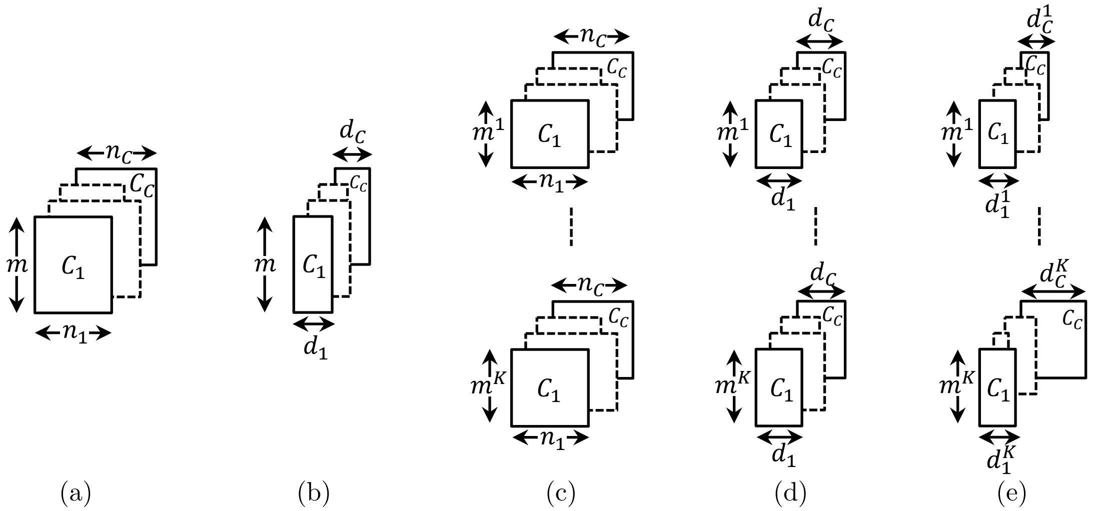


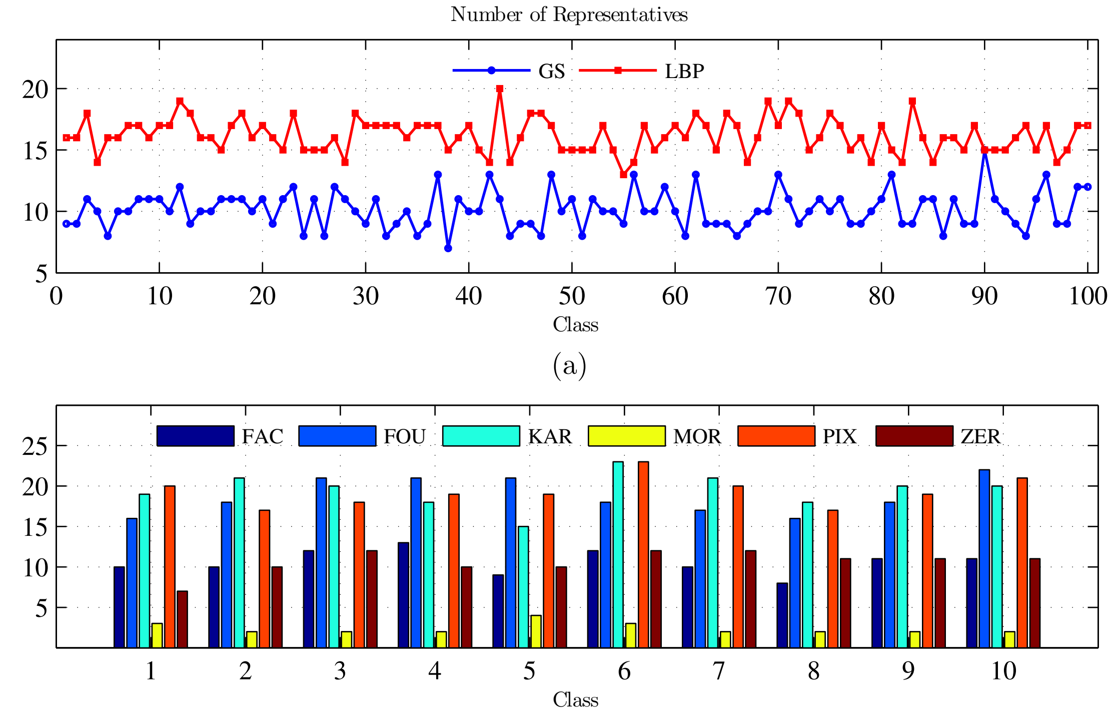
Figure 4.13: Training matrices in (a) original SRC, (b) SRC with dictionary learning, (c) CMSRC, (d) MMFSRC and (e) SRC using dictionaries with different number of atoms.

class (Figure 4.13c). The corresponding coefficient vectors, $\boldsymbol{x}^k$, and specifically the ones associated with each class i corresponding to all modalities, $\boldsymbol{x}_i^k$, are from equal dimensionality or mathematically,

$$\left\{ \boldsymbol{x}_i^k, \boldsymbol{x}_i^{k'} \in \mathbb{R}^{n_i} \longmapsto \boldsymbol{x}^k, \boldsymbol{x}^{k'} \in \mathbb{R}^{n} \right\}, \forall i|_{i=1}^{C}, \forall k, k'|_{\substack{k=1 \\ k \neq k'}}^{K}. \tag{4.16}$$

As a result, the column-concatenation of the coefficient vectors over all modalities, i.e. $[\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^K]$, will form matrix $X \in \mathbb{R}^{n \times K}$ which is then recovered by (4.2) and used for classification. This property is also valid for the proposed multi-modality classifier, CMSRC and MMFSRC (Figure 4.13d). In the case of CMSRC, at the first step, sample reduction is employed and then modalities M^{D^k}s are extracted, so for the coefficient vectors from different modalities, the assumption (4.16) is still valid (n_i and n are replaced with d_i and d). In MMFSRC, modalities are extracted first, but when learning dictionaries $D_i^{M^k}$s, the same number of representatives, d_i is imposed to all modalities for each class. This leads for the coefficients to be in a same situation as CMSRC where assumption (4.16) is satisfied. In all above mentioned situations, it is possible to break the coefficient matrix X to sub-matrices X_i for all classes i the classification task may be perform by reconstructing a row-sparse X. Therefore, the group LASSO optimization (4.15) can be utilized for the reconstructing process. In this section, we look at the classification problem using a sparse representation framework from a different point of view. In the proposed method, modalities are extracted from the training samples and stored in different modality matrices which are then separately fed into a sample reduction process. This approach allows the number of representatives (dictionary atoms) for each modality to be flexible and different from other modalities depending on the variability, information content and discriminative power of that specific modality over the dataset. The classification problem under this setting is mapped into a new optimization and an algorithm is proposed to recover the coefficient vectors. This approach is considered from the third approach, Modality/Nun-Uniform Reduction.

Method In some sample reduction methods the final number of representatives is not specified. For example, among the methods introduced in 3.1, SMRS and AC determine the optimum number of representatives by using some parameters. When K modalities are extracted from the training set, there are K datasets containing different information and redundancies and if the objective is to reduce samples in these matrices, the optimum solution may leads to a different number of representatives for each single modality in each class ($n_i^k \neq n_i^{k'}$, $k, k' \in \{1, \ldots, K\}, k \neq k'$). For example, in a face recognition application, after the sample reduction process, there might be different number of representatives for GS and LBP modalities for a specific class. As an example, Figure 4.14a shows the number of samples selected by SMRS for two modalities on 100 classes of the FRGC face dataset. It can be clearly seen that the



(a)

(b)

Figure 4.14: Number of representatives selected by SMRS on (a) two modalities of 100 classes from FRGC face dataset and (b) 6 modalities of 10 classes from UCI digit dataset.

number of representatives in each class is different for the two extracted modalities. Another example of variation of number of representatives is shown in Figure 4.14b where SMRS was applied to 6 modalities of hand written digits samples from 10 classes (0∼9) of UCI digit dataset. In these cases, the training matrices associated with each modality have different number of columns per class (Figure 4.13e) and their corresponding coefficient vectors, $\boldsymbol{x}_i^k$s for different values of k are from different dimensionalities. So, they cannot form a matrix anymore and the group LASSO optimization (4.2) can not be employed. Figure 4.15b shows the coefficient vectors for an example of this situation. It can be seen that for a specific class, coefficient vectors corresponding to different modalities have different dimensionalities.

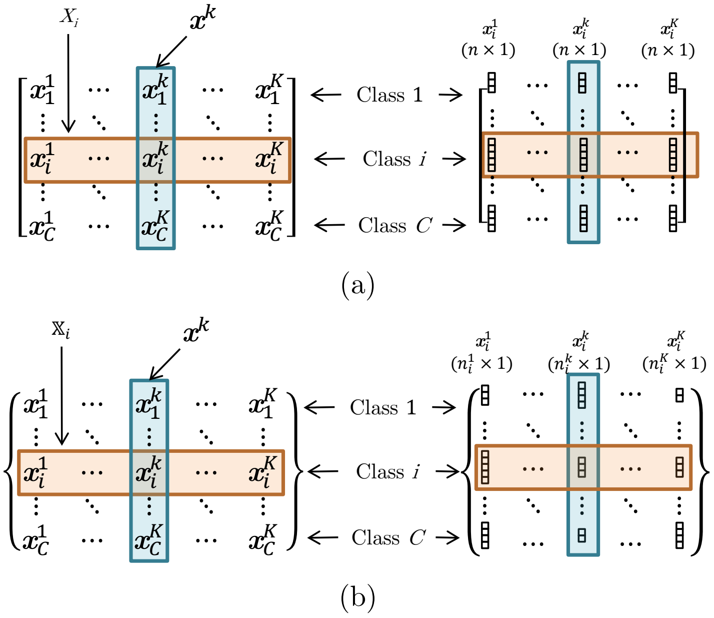

X_i
$\boldsymbol{x}^k$
Class 1
Class i
Class C
$\boldsymbol{x}_i^1$ $(n \times 1)$
$\boldsymbol{x}_i^k$ $(n \times 1)$
$\boldsymbol{x}_i^K$ $(n \times 1)$
(a)
$\mathbb{X}_i$
$\boldsymbol{x}^k$
Class 1
Class i
Class C
$\boldsymbol{x}_i^1$ $(n_i^1 \times 1)$
$\boldsymbol{x}_i^k$ $(n_i^k \times 1)$
$\boldsymbol{x}_i^K$ $(n_i^K \times 1)$
(b)

Figure 4.15: Multi-task coefficients representation (a) matrix X in MTJSRC and (b) array $\mathbb{X}$ include vectors with different length after applying sample reduction to modality matrices. Note that unlike $\mathbb{X}_i$ in (b), X_i in (a) can be represented in a matrix form.

In order to incorporate a method to handle modality matrices with different number of atoms, we introduce the 1,2-normalized mixed norm

$$\|\mathbb{X}\|_{1,N_2} = \sum_{i=1}^{C} \|\mathbb{X}_i\|_{N_2} = \sum_{i=1}^{C} \sum_{k=1}^{K} \frac{\left\|\boldsymbol{x}_i^k\right\|_2}{n_i^k}, \tag{4.17}$$

where $\mathbb{X}_i$ represents an array of the coefficient vectors $\boldsymbol{x}_i^k$s ($\mathbb{X}_i = \{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^K\}$) and $\mathbb{X} = \{\mathbb{X}_1, \ldots, \mathbb{X}_C\}$. This norm is used to enforce sparsity constraint to the multi-task optimization function which is now reformulated as

$$\hat{\mathbb{X}} = \underset{\mathbb{X}}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{k=1}^{K} \left\| \boldsymbol{y}^k - \sum_{i=1}^{C} D_i^{M^k} \boldsymbol{x}_i^k \right\|_2^2 + \lambda \|\mathbb{X}\|_{1,N_2} \right\}. \tag{4.18}$$

We derive the solution to the optimization problem (4.18) by incorporating a modified version of the well-known Accelerated Proximal Gradient (APG) algorithm [77, 78] where the coefficient vector array sequence $\widehat{\mathbb{X}}^t$ and an aggregation vector array sequence $\widehat{\mathbb{G}}^t = \{\boldsymbol{g}_1^{1,t}, \ldots, \boldsymbol{g}_2^{1,t}, \ldots, \boldsymbol{g}_2^{K,t}, \ldots, \boldsymbol{g}_C^{K,t}\}$ are alternatively updated in two steps.

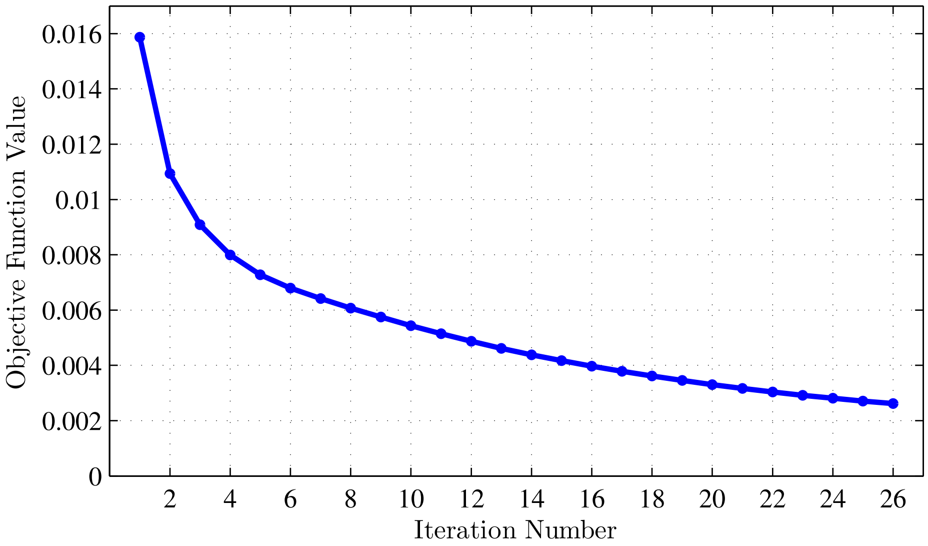


Figure 4.16: An example of the NMSRC optimization process (Algorithm 4.3). The curve shows the value of the objective function (4.18) in 26 iterations.

Details of this algorithm is shown in Algorithm 4.3. Convergence of the objective function for the proposed algorithm is shown in Figure 4.16. After convergence, the recovered array $\mathbb{X}$ is used in (4.7) to find the identity of the unknown test sample $\boldsymbol{y}$.

Experiments The proposed Non-uniform Multi-modal SRC method (NMSRC) is evaluated through several experiments on face and digit recognition. First, modalities are extracted from the original training data. To reduce the number of samples, AC which introduced in Section 3.2 and SMRS [23] on the modality matrices. These two approaches are from both categories, dictionary learning and sample selection and unlike FDDL and Metaface, they automatically tune the number of representatives in every modality of each class depending on the statistics of the data. Improvements achieved by the multi-modality property of the proposed method are first illustrated by comparing it to single-modality SRC-based approaches, i.e. original SRC, SMRS-SRC and AC-SRC. The next set of experiments are performed to show the effec-

Algorithm 4.3: Proposed algorithm based on APG to solve (4.18).

```
input          : Modality matrices $M^k|_{k=1}^K$,   Test sample modalities $\boldsymbol{y}^k|_{k=1}^K$;
                 Sparsity regulizer $\mu$ and step-size $\eta > 0$;
initialization: $t = 0$ ; $\alpha_0 = 0$ ; $\mathbb{G}^0 = \boldsymbol{g}^{k,0}|_{k=1}^K = 0$;
repeat
    Step 1: Given $\mathbb{G}^t$ update $\mathbb{X}^{t+1}$:
    for modality $k \in \{1, 2, \ldots, K\}$ do
        $\nabla^k = -\left(M^k\right)^\mathsf{T} \boldsymbol{y}^k + \left(M^k\right)^\mathsf{T} M^k \boldsymbol{g}^{k,t}$; $\boldsymbol{x}^{k,t+1} = \boldsymbol{g}^{k,t} - \eta\nabla^k$;
    for class $i \in \{1, 2, \ldots, C\}$ do
        $\Omega_i = \sum_{k=1}^K \frac{\left\|\boldsymbol{x}_i^{k,t+1}\right\|_2}{n_i^k}$;
        for modality $k \in \{1, 2, \ldots, K\}$ do
            $\boldsymbol{x}^{k,t+1} = \max\left(\left[1 - \frac{\lambda\eta}{\Omega_i}\right], 0\right)$;
    Step 2: Given $\mathbb{X}^t$ and $\mathbb{X}^{t+1}$ update $G^{t+1}$:
    $\alpha_{t+1} = \frac{2}{t+3}$;   $\gamma = \frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t}$;
    for modality $k \in \{1, 2, \ldots, K\}$ do
        $\boldsymbol{g}^{k,t+1} = \boldsymbol{x}^{k,t+1} + \gamma\left(\boldsymbol{x}^{k,t+1} - \boldsymbol{x}^{k,t}\right)$;
    $t \leftarrow t + 1$;
until Convergence or Maximum Iteration;
output         : Coefficients $\mathbb{X} = \left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K\right\}$;
```

tiveness of the proposed method in comparison to its multi-modality ancestor, i.e. MTJSRC.

Face Recognition Experiments We used the previously introduced Extended Yale B and FRGC face datasets with GS and LBP modalities to show the multi-modality nature of the proposed algorithm.

In the first step, adaptive clustering (AC) (Algorithm 3.3) is utilized to build dictionaries out of the extracted modalities, individually. According to Section 3.2, number of clusters formed for each class in AC is controlled by two input parameters: the maximum within-cluster sums of point-to-centroid distance (τ) and maximum number of clusters (Q_{max}). These parameters are tuned such that the final dictionaries include average numbers of 2 (only on FRGC), 4, 6, 8, 10 and 12 columns per class for each modality and as a result, the formed dictionaries on Extended Yale B dataset are of sizes 1024×(152∼456) (GS) and 900×(152∼456) (LBP) and on FRGC dataset are of size 1024×(200∼1200) (GS) and 900×(200∼1200) (LBP).

SMRS algorithm is then applied to the modality matrices of Extended Yale B dataset, where the average number of 9.13 and 11.63 representatives per class were selected on GS and LBP modalities, respectively. These numbers are 10.15 and 16.20 for FRGC dataset. Figure 4.14a shows the number of representatives selected for each class of FRGC dataset individually.

For the optimization problem (4.18) to recover a row-sparse coefficient matrix, the training matrix must represent an under-determined system of equations and this is achieved by a matrix which has more columns than rows. In other words, for each modality, the number of representatives should be larger than the modality dimension. To satisfy this condition, down-sampling is used to reduce the GS and LBP modalities dimensionality to 100. These models are then fed into the proposed optimization problem (4.18) and its accuracy is compared to when these dictionaries are individually used in an SRC framework and when the multi-modality MTJSRC is applied by a training matrix of randomly selected samples.

The proposed multi-modality method with non-uniform modality matrices from two sample reduction approaches (AC-NMSRC and SMRS-NMSRC) is evaluated against the single-modality SRC methods of AC-SRC and SMRS-SRC. Figure 4.17

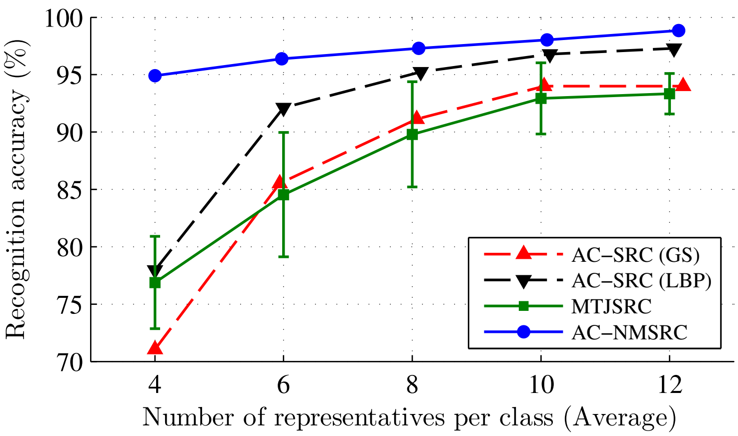


Figure 4.17: Recognition rates of NMSRC comparing to MTJSRC and single-modality SRC with AC sample reduction on FRGC face dataset.

shows the face recognition accuracies for different methods on Extended Yale B datasets. Single-modality dictionaries (GS and LBP) are first generated by adaptive clustering algorithm with different number of representatives per class and independently imposed to the original SRC for final classification. These two dictionaries are then used in the proposed optimization (4.18) simultaneously to classify unknown test samples. In the next experiment, MTJSRC randomly selects samples from the original training set, extract modalities and classifies test samples through its multi-modality framework. Random selection on samples is repeated 10 times and the average and standard deviation of the recognition rates are reported. Figure 4.18 shows the results from similar experiments on FRGC dataset. It can be seen that especially for smaller dictionaries, the proposed method improves the recognition accuracy on both datasets comparing to other SRC-based classifiers. As an example, to achieve about ∼95% accuracy on Extended Yale B dataset, AC-NMSRC needs training matrices as small as ≈4 columns per class, while SRC with LBP dictionary need ≈10 columns per class and MTJSRC needs dictionaries with more than 12 samples per class. Note that the execution time in all SRC-based methods directly affected by the number of

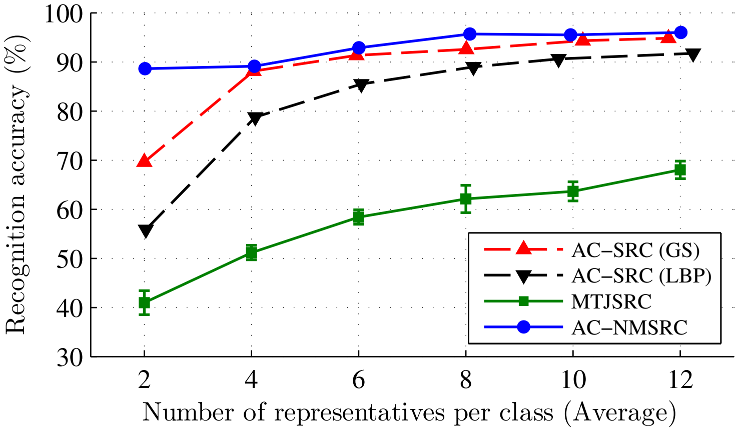


Figure 4.18: Recognition rates of NMSRC comparing to MTJSRC and single-modality SRC with AC sample reduction on FRGC dataset.

columns in the training matrices and this makes our proposed method more applicable in large scale problems. Table 4.2 compares the two variants (AC-NMSRC and SMRS-NMSRC) of the proposed method with SRC and MTJSRC. For an average of around 10 (Extended Yale B) and 12 (FRGC) samples per class, it is clearly seen that both the proposed methods outperforms other classification approaches.

Digit Recognition Experiments The proposed method is evaluated over UCI multimodality digit dataset with 10 classes of hand written numerals (0∼9) which in-

Table 4.2: Recognition rates and the average number of representatives per class for different classifiers on face datasets.

Method	Extended Yale B		FRGC	
	#Reps.	Acc.(%)	#Reps.	Acc.(%)
AC-SRC (GS)	10.05	94.00	11.78	94.83
AC-SRC (LBP)	10.14	96.79	12.24	91.73
SMRS-SRC (GS)	9.13	89.64	10.15	90.17
SMRS-SRC (LBP)	11.63	96.13	16.20	91.57
MTJSRC	10.00	92.94 ±3.11	12.00	68.01 ±1.77
AC-NMSRC	10.10	**98.03**	12.01	**96.00**
SMRS-NMSRC	10.38	**98.77**	13.18	**95.97**

troduced in Section 2.2. Similar to face recognition experiments, AC with different parameters is used as the first approach to reduce the number of samples for all modalities. SMRS is also applied where an average of 10.60, 18.80, 19.50, 2.40, 19.30 and 10.60 samples per class are selected among all the training data on the modalities 1∼6, respectively.

For dimensionality reduction process, random projection is employed on modalities 1, 2, 3 and 5 where these vectors are randomly projected to 25 dimensional vectors. Pixel value image (Modality 4) is down-sampled into a 5×5 (25 dimensional) image and the last 6 dimensional modality is directly used for classification. Two proposed approaches (AC-NMSRC and SMRS-NMSRC) are employed for the experiments on UCI digit recognition and the corresponding results are reflected in Table 4.3. 12 different training matrices are built by applying AC and SMRS on the available 6 features. These matrices are individually used in SRC framework and then simultaneously fed into our proposed method. Results confirm that the proposed multi-modality methods outperforms single-modality SRC in all trials with the recognition rates of %96.90 and %93.60 for AC-NMSRC and SMRS-NMSRC, respectively. MTJSRC is then applied to the original training data and achieves an average of %90.92 recognition rate out of its 10 run of training sample selections (14 samples

Table 4.3: Recognition rates and the average number of representatives per class for different classification approaches on UCI digit dataset.

Method	#Reps.	Acc.(%)	Method	#Reps.	Acc.(%)
AC-SRC (FAC)	13.80	89.50	SMRS-SRC (FAC)	10.60	86.00
AC-SRC (FOU)	15.20	70.40	SMRS-SRC (FOU)	18.80	68.70
AC-SRC (KAR)	14.50	89.30	SMRS-SRC (KAR)	19.50	86.40
AC-SRC (MOR)	12.80	51.30	SMRS-SRC (MOR)	2.40	36.60
AC-SRC (PIX)	13.70	93.40	SMRS-SRC (PIX)	19.30	93.30
AC-SRC (ZER)	15.00	78.10	SMRS-SRC (ZER)	10.60	73.20
MTJSRC	14.00	90.92	AC-NMSRC	14.10	**96.90**
		±0.70	SMRS-NMSRC	13.50	**93.60**

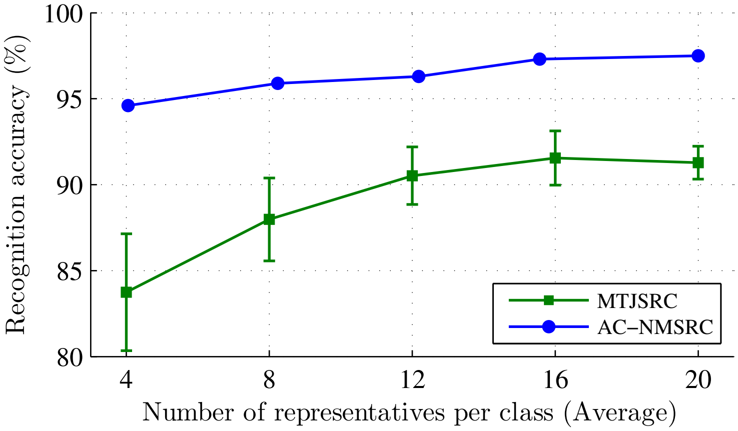


Figure 4.19: Recognition rates achieved by NMSRC comparing to MTJSRC (random selection of samples) with training matrices with different sizes on UCI digit dataset.

per class). The last experiments (MTJSRC and two proposed methods) are also repeated for different training matrix sizes and Figure 4.19 reflects the corresponding results. It can be seen that AC-NMSRC achieves outstanding recognition rate of %96.50 by employing dictionaries as small as ≈40 atoms, while MTJSRC achieves an average of %91.28 even when it uses 200 atoms in its training matrices.

CHAPTER 5

CONCLUSION

In this thesis, we presented a comprehensive study on the state-of-the-art Sparse Representation-based Classification (SRC) method by investigating different methods proposed to improve SRC from both efficiency and accuracy points of view. SRC is inspired by the emerging theory of Compressive Sensing (CS) which recently attracted considerable attentions. According to CS, under certain conditions, a sparse signal may be efficiently recovered from a few number of measurements by incorporating an ℓ^1-norm optimization process. SRC maps the problem of classification of an unknown test sample $\boldsymbol{y}$ to an under-determined system of linear equations $\boldsymbol{y} = V\boldsymbol{x}$ where V contains the given training samples and $\boldsymbol{x}$ is the sparse coefficient vector of interest. By recovering $\boldsymbol{x}$ based on the CS theory, SRC can identify the class to which test sample $\boldsymbol{y}$ belongs. Despite the interesting results reported for SRC in different applications, it suffers from high computational costs when dealing with large training data.

We first studied a variety algorithms to accelerate SRC optimization process by substituting the original training model with more abstract ones. We also proposed a method to reduce the number of representatives for each class based on the diversity and hidden information in the training samples of that class. This approach uses an adaptive scheme of k-means clustering to reduce the number of representatives for each class of the training data. Experiments on face and digit recognition show the effectiveness of the proposed method in comparison to the original implementation of SRC and using other sample reduction approaches along with SRC.

In the next step, we focused on multi-modality implementations of SRC by proposing three variants of multi-task joint sparse representation (MTJSRC) which is a complementary approach based on SRC. In MTJSRC, several modalities of the input space can be incorporated for classification. A joint optimization variant of the one used in SRC is utilized to calculate and recover the coefficient vectors which are then used to find the class associated with a given test sample. In this study, we investigated some methods to efficiently solve this multi-class multi-modality problem. Our first proposed approach -which is called *RedMod* in this thesis- applies sample reduction to the training samples at the first step. Then, after extracting modalities from the new representatives, this method forms training models to be used by MTJSRC for the classification task. To improve the accuracy by considering each modality separately, the second approach -which is called *Uniform ModRed* in this thesis- first extracts the modalities from all the training samples and then sample reduction is separately incorporated on different modalities. A limitation of this approach is that we have to enforce same number of representatives to all modalities from specific classes in order to be able to use MTJSRC optimization process. As a more flexible method, the third approach -which is called *Non-Uniform ModRed* in this thesis- can handle modality matrices with different number of representatives per class by using a novel optimization algorithm. We evaluated the proposed approaches by performing several experiments on face and digit recognition applications. Experimental results show our proposed methods achieve higher recognition rates in a more efficient structures in terms of both computational cost and required space.

REFERENCES

[1] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[2] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the* 5^{th} *Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[3] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis.* John Wiley & Sons, 2009, vol. 344.

[4] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 1991, pp. 586–591.

[5] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von Der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 19, no. 7, pp. 775–779, 1997.

[6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[7] J. Wright, A. Yang, A. Ganesh, and S. Sastry, “Robust face recognition via sparse representation,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 31, no. 2, pp. 210–227, 2009.

[8] Ø. D. Trier, A. K. Jain, and T. Taxt, “Feature extraction methods for character recognition-a survey,” *Pattern recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[9] C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, “Handwritten digit recognition: benchmarking of state-of-the-art techniques,” *Pattern Recognition*, vol. 36, no. 10, pp. 2271–2285, 2003.

[10] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “recaptcha: Human-based character recognition via web security measures,” *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[11] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] A. Subasi and M. I. Gursoy, “EEG signal classification using PCA, ICA, LDA and support vector machines,” *Expert Systems with Applications*, vol. 37, no. 12, pp. 8659–8666, 2010.

[14] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, “Feature reduction and selection for emg signal classification,” *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification.* John Wiley & Sons, 2012.

[16] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural network design.* Pws Pub. Boston, 1996.

[17] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[18] V. Kecman, *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models.* MIT press, 2001.

[19] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[20] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, 2006, pp. 1433–1452.

[21] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," *IEEE International Conference on Image Processing (ICIP)*, pp. 1601–1604, 2010.

[22] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," *IEEE International Conference on Computer Vision (ICCV)*, pp. 543–550, 2011.

[23] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1607, 2012.

[24] X. T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing (tIP)*, vol. 21, no. 10, pp. 4349–4360, 2012.

[25] P. Phillips, P. Flynn, T. Scruggs, and K. Bow, "Overview of the face recognition grand challenge," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[26] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 28, no. 12, pp. 2037–2041, 2006.

[27] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 27, no. 5, pp. 684–698, 2005.

[28] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 23, no. 6, pp. 643–660, 2001.

[29] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 16, no. 5, pp. 550–554, 1994.

[30] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[31] M. Van Breukelen, R. P. Duin, D. M. Tax, and J. Den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.

[32] J. Kittler, M. Hatef, and R. P. W. Duin, "Combining classifiers," in *International Conference on Pattern Recognition (ICPR)*, 1996, pp. 897–901.

[33] J. Den Hartog, T. Ten Kate, and J. J. Gerbrands, "Knowledge-based interpretation of utility maps," *Computer Vision and Image Understanding (CVIU)*, vol. 63, no. 1, pp. 105–117, 1996.

[34] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C. L. Wilson, "NIST form-based handprint recognition system," Technical Report NISTIR 5469 and CD-ROM, National Institute of Standards and Technology, Tech. Rep., 1994.

[35] A. Khotanzad and Y. H. Hong, "Rotation invariant pattern recognition using zernike moments," in *International Conference on Pattern Recognition (ICPR)*. IEEE, 1988, pp. 326–328.

[36] D. M. Tax, M. Van Breukelen, R. P. Duin, and J. Kittler, "Combining multiple classifiers by averaging or by multiplying?" *Pattern recognition*, vol. 33, no. 9, pp. 1475–1485, 2000.

[37] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[38] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[39] J. Romberg, "Imaging via compressive sampling [introduction to compressive sampling and recovery via convex programming]," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, 2008.

[40] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[41] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly, "k-t sparse: High frame rate dynamic MRI exploiting spatio-temporal sparsity," in *Proceedings of the 13th Annual Meeting of ISMRM, Seattle*, vol. 2420, 2006.

[42] J. Ma and F. Le Dimet, "Deblurring from highly incomplete measurements for remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 792–802, 2009.

[43] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Transactions on Communications*, vol. 50, no. 3, pp. 374–377, 2002.

[44] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, 1998.

[45] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[46] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

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

[48] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[49] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[50] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[51] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[52] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[53] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[54] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[55] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 25, no. 2, pp. 218–233, 2003.

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

[57] E. Candès and J. Romberg, "L1-magic : Recovery of sparse signals via convex programming," 2005. [Online]. Available: http://users.ece.gatech.edu/~justin/l1magic/downloads/l1magic.pdf

[58] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[59] P. Belhumeur, J. Hespanda, and D. Kriegman, "Eigenfaces versus fisherfaces: Recognition using class specific linear projection," *IEEE Transaction on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 19, no. 7, pp. 711–720, 1997.

[60] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using laplacianfaces," *IEEE Transaction on Pattern Analysis and Machine Intelligence (tPAMI)*, vol. 27, no. 3, pp. 328–340, 2005.

[61] M. Savvides, R. Abiantun, J. Heo, S. Park, C. Xie, and B. Vijayakumar, "Partial & holistic face recognition on FRGC-II data using support vector machine," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW).* IEEE, 2006, pp. 48–48.

[62] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.

[63] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ^1-norm minimization problems when the solution may be sparse," *IEEE transactions on information theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[64] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[65] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling and Simulation*, vol. 7, p. 214, 2008.

[66] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2691–2698.

[67] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.

[68] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ^1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[69] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[70] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 968–977.

[71] T. F. Chan, “Rank revealing QR factorizations,” *Linear Algebra and Its Applications*, vol. 88, pp. 67–82, 1987.

[72] S. Shafiee, F. Kamangar, V. Athitsos, and J. Huang, “Efficient sparse representation classification using adaptive clustering,” *International Conference on Image Processing,Computer Vision, and Pattern Recognition (IPCV)*, 2013.

[73] S. Shafiee, F. Kamangar, V. Athitsos, and J. Huang, “The role of dictionary learning on sparse representation-based classification,” in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA).* ACM, 2013, p. 47.

[74] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *IEEE International Conference on Computer Vision (ICCV).* IEEE, 2009, pp. 221–228.

[75] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001, pp. I–511.

[76] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[77] Y. Nesterov, “Gradient methods for minimizing composite objective function,” 2007.

[78] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell, “Accelerated gradient method for multi-task sparse learning problem,” in *IEEE International Conference on Data Mining (ICDM).* IEEE, 2009, pp. 746–751.

[79] M. E. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2006, pp. 1447–1454.

[80] S. Shafiee, F. Kamangar, and L. S. Ghandehari, “Cluster-based multi-task sparse representation for efficient face recognition,” in *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2014, pp. 125–128.

[81] S. Shafiee, F. Kamangar, V. Athitsos, J. Huang, and L. Ghandehari, “Multimodal sparse representation classification with fisher discriminative sample reduction,” in *IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5192–5196.

[82] S. Shafiee, F. Kamangar, and V. Athitsos, “A multi-modal sparse coding classifier using dictionaries with different number of atoms,” in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2015, pp. 518–525.

[83] A. Majumdar and R. K. Ward, “Compressive classification for face recognition,” in *Face recognition*, M. Oravec, Ed. InTech Publishers, 2010, pp. 47–64.

BIOGRAPHICAL STATEMENT

Soheil Shafiee was born in Gorgan, Iran, in 1981. He obtained his B.Sc. and M.Sc. degrees in Biomedical Engineering from Amirkabir University of Technology, Tehran, Iran, in 2004 and 2007, respectively. He entered the department of Computer Science and Engineering of the University of Texas at Arlington in 2010 to pursue his Ph.D. in Computer Science. In his Ph.D. research, he mainly focused on machine learning and pattern recognition and their applications in computer vision.