Abstract

Title of dissertation: Sparse Methods for Robust and Efficient
Visual Recognition
Sumit Shekhar, Doctor of Philosophy, 2014

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Visual recognition has been a subject of extensive research in computer vision. A vast literature exists on feature extraction and learning methods for recognition. However, due to large variations in visual data, robust visual recognition is still an open problem. In recent years, sparse representation-based methods have become popular for visual recognition. By learning a compact dictionary of data and exploiting the notion of sparsity, start-of-the-art results have been obtained on many recognition tasks. However, existing data-driven sparse model techniques may not be optimal for some challenging recognition problems. In this dissertation, we consider some of these recognition tasks and present approaches based on sparse coding for robust and efficient recognition in such cases.

First we study the problem of low-resolution face recognition. This is a challenging problem, and methods have been proposed using super-resolution and machine learning-based techniques. However, these methods cannot handle variations like illumination changes which can happen at low resolutions, and degrade the performance. We propose a generative approach for classifying low resolution faces, by exploiting 3D face models. Further, we propose a joint sparse coding framework for robust classification at low resolutions. The effectiveness of the method is demonstrated on different face datasets.

In the second part, we study a robust feature-level fusion method for multimodal biometric recognition. Although score-level and decision-level fusion methods exist in biometric literature, feature-level fusion is challenging due to different output formats of biometric modalities. In this work, we propose a novel sparse representation-based method for multimodal fusion, and present experimental results for a large multimodal dataset. Robustness to noise and occlusion are demonstrated.

In the third part, we consider the problem of domain adaptation, where we want to learn effective classifiers for cases where the test images come from a different distribution than the training data. Typically, due to high cost of human annotation, very few labeled samples are available for images in the test domain. Specifically, we study the problem of adapting sparse dictionary-based classification methods for such cases. We describe a technique which jointly learns projections of data in the two domains, and a latent dictionary which can succinctly represent both domains in the projected low-dimensional space. The proposed method is efficient and performs on par or better than many competing state-of-the-art methods.

Lastly, we study an emerging analysis framework of sparse coding for image classification. We show that the analysis sparse coding can give similar performance as the typical synthesis sparse coding methods, while being much faster at sparse encoding. In the end, we conclude the dissertation with discussions and possible future directions.

Sparse Methods for Robust and Efficient Visual Recognition

by

Sumit Shekhar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Ramani Duraiswami
Professor Behtash Babadi
Professor David Jacobs
Professor Amitabh Varshney

Dedication

To my *parents* and *sisters*.

Acknowledgments

I would like to express my deepest gratitude to my adviser, Prof. Rama Chellappa for providing me opportunity to pursue graduate studies in computer vision and mentoring through out my graduate life. I am thankful to Prof. Chellappa for providing a wonderful research environment and encouragement to pursue challenging problems, while giving valuable advice at times of doubt. The discussions we had were always enlightening and enjoyable, broadening my perspectives about research and the field of computer vision. His undying passion towards the field of computer vision and high professional integrity have always enthused me to work hard for my dissertation and will guide me in my future career pursuits.

It is an honor to have Prof. Ramani Duraiswami, Prof. Amitabh Varshney, Prof. David Jacobs and Prof. Behtash Babadi in my dissertation committee. I am thankful to them for serving in my committee and providing insightful suggestions to improve this dissertation.

During my student life, I was fortunate to take courses under several outstanding professors - Prof. André L. Tits, Prof. David Jacobs, Prof. Hal Daume, Prof. Nuno Martins, Prof. Min Wu, Prof. Sennur Ulukus, Prof. Prakash Narayan, Prof. Wojciech Czaja, Prof. Abram Kagan, Prof. Subhasis Chaudhuri and Prof. Chellappa. I am thankful to them for helping me develop insights in computer vision and machine learning as well as incorporate mathematical rigor in my research.

I am deeply indebted to my mentors - Dr. Vishal Patel and Dr. Nasser Nasrabadi without whom this dissertation would not have been possible. I am grateful to Dr. Ashwin

Swaminathan for guiding me through a wonderful internship at Qualcomm. I would also like to express my gratitude to my seniors - Prof. Pavan Turaga, Dr. Raghuram Gopalan, Dr. Kaushik Mitra, Dr. Nitesh Shroff, Dr. Dikpal Reddy, Dr. Ming Yu Liu, Dr. Jaishanker Pillai and Dr. Qiang Qiu for their mentoring throughout my Ph.D. I am also grateful to Prof. Ajay Kumar, who mentored me through my undergraduate internship, and introduced me to computer vision research.

It was a pleasure to work with Hien Nguyen, Ashish Srivastava and Yi-Chen Chen over the last few years. I would also like to thank my labmates and colleagues including Ravi Teja, Garrett Warnell, Huy Tho Ho, Kota Hara, Xavier Gibert Serra, David Shaw, Chris Reale, Jie Ni, Arijit Biswas, Abhishek Sharma, Jonghyun Choi, Anne Jorstad, Ming Du, Sima Taheri, Swaminathan Sankaranarayanan, Ching-Hui Chen, Boyu Lu, Jingjing Zheng, Priyanka Vageeswaran and Arthita Ghosh for making my graduate life memorable. I am also grateful to the Computer Vision Seminar Series (CVSS) for providing a good place of interaction for vision students.

I would like to thank the staff in UMIACS, ECE and CFAR. In particular, special thanks to Janice Perone, Arlene Schenk, Maria Hoo, Melanie Prange and Dr. Tracy Chung.

I would like to acknowledge my friends - Gunjan Srivastava, Sumit Kumar, Vaibhav Singh, Ravi Garg, Krishna Chaitanya, Praneeth Boda, Madhura Joglekar, Shantanu Debnath, Kapil Anand, Shalabh Jain, Kaustubh Jain, Anirban Gangopadhyay, Anirban Ghosh(s), Rajibul Islam, Souvik Bhattacharjee, friends at IIT Bombay, who have always encouraged me and have helped me immeasurably throughout my graduate life.

My parents and family have been a great source of inspiration and support through out my life. This dissertation would not have been possible with out the constant support, encouragement and love of my entire family - my parents Shashi Bhushan Prasad and Punam Rani, sisters Puja Shekhar and Pallavi Shekhar, brother-in law Abhijat Sinha, niece Aardra, respected elders and my wife-to-be Shradha Sinha. Finally, I would like to thank God Almighty.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Visual recognition has been a subject of extensive research in computer vision. A vast literature exists on feature extraction and learning methods for recognition. However, in real world, captured images show myriad variations, which can be caused by changes in cameras, object viewpoint, lighting conditions, etc. Dealing with these variations is challenging, and the problem holds considerable research interest.

Low resolution can be a significant challenge in many practical recognition systems, e.g., surveillance, where the person can be standing far away from camera, thus, his/her image has few pixels. Many face recognition algorithms, which work well when image has sufficient resolution, break down at low resolutions. Image enhancement techniques, like super-resolution do not increase recognition performance. Further, low resolution is usually coupled with other variations like illumination, blur, etc making the problem harder.

An approach to robust visual classification is through fusion of multiple cues. While individual features may not work well for different variations in data, a joint inference can give improved results. Fusion is typically done at feature-level or output score-level. Ranked outputs of individual classifiers are also fused in some applications. Fusion of raw features can be more robust, however there can be challenges due to different feature formats and high feature dimension. Another important problem is to devise a way to weigh different features at the test time.

Recently, Saenko *et al* [109] introduced the problem of domain adaptation to the

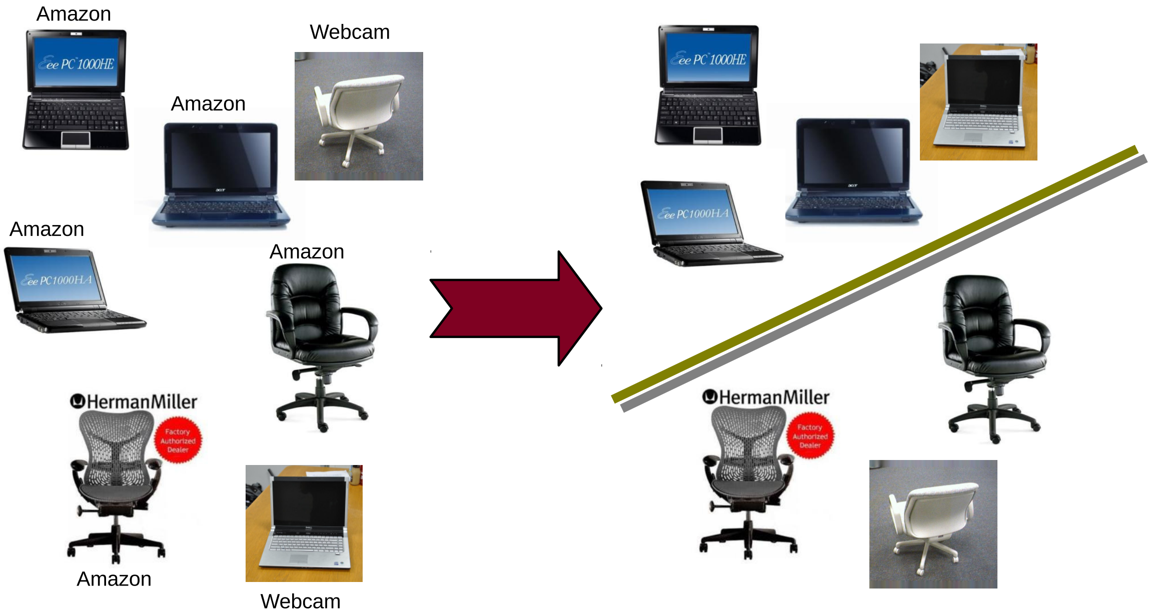


Figure 1.1: Example of domain adaptation problem. We want to adapt a classifier to perform well on images from Webcam with a few labeled examples available while training. However, sufficient number of labeled samples are available from the Amazon dataset.

vision community. The problem of domain adaption deals with the situation where the test images come from a different distribution than the training data, as shown in Figure 1.1. This change can be caused due to the variations in data capture as mentioned above. Further, due to high cost of annotation, only few labeled samples are usually available in the test domain for adapting the classifier. The challenge here is to learn a robust classifier which can perform well in testing conditions. Applications of domain adaptation methods include robust object recognition (e.g. matching high resolution images to low resolution), using unlabeled videos (on YouTube) for improving image recognition, etc.

Lastly, with large amount of data available, efficient classification has become an important problem. With the above challenges in mind, we now present some effective solutions in this dissertation.

1.2 Proposed Algorithms and Contributions:

1. **Low resolution face recognition:**

   In the first part, we consider the challenging problem of recognition of low resolution face images. As the recognition becomes difficult at low resolutions, we assume that a high resolution training image is available for recognition. We propose a generative approach [115, 116] for classifying the low resolution image, by exploiting the information available in high resolution training through 3D face models. An important feature of our algorithm is that it can handle resolution changes along with illumination variations. The effectiveness of the proposed method is demonstrated using standard datasets and a challenging outdoor face dataset. It is shown that our method is efficient and can perform significantly better than many competing low resolution face recognition algorithms.

2. **Robust feature-level fusion:**

   Traditional biometric recognition systems rely on a single biometric signature for authentication. While the advantage of using multiple sources of information for establishing the identity has been widely recognized, computational models for multimodal biometrics recognition have only recently received attention. In the second part of the dissertation, we propose a multimodal sparse representation method [114, 117], which represents the test data by a sparse linear combination of training data, while constraining the observations from different modalities of the test subject to share their sparse representations. Thus, we simultaneously take into account correlations as well as coupling information among biometric modalities. A multimodal quality measure is also proposed to weigh each modality as it gets fused. Furthermore, we also kernelize the algorithm to handle non-linearity in data. The optimization problem is solved using an efficient alternative direction method. Various experiments show that the proposed method compares favorably with competing fusion-based methods.

3. **Domain-adaptive dictionary learning:**

Data-driven dictionaries have produced state-of-the-art results in various classification tasks. However, when the target data has a different distribution than the source data, the learned sparse representation may not be optimal. In this part of the dissertation, we investigate if it is possible to optimally represent both source and target by a common dictionary. Specifically, we describe a technique [118, 119] which jointly learns projections of data in the two domains, and a latent dictionary which can succinctly represent both the domains in the projected low-dimensional space. An efficient optimization technique is presented, which can be easily kernelized and extended to multiple domains. The algorithm is modified to learn a common discriminative dictionary, which can be further used for classification. The proposed approach does not require any explicit correspondence between the source and target domains, and shows good results even when there are only a few labels available in the target domain. Various recognition experiments show that the method performs on par or better than competitive state-of-the-art methods.

4. **Analysis Sparse Coding:**
   Data-driven sparse models have been shown to give superior performance for image classification tasks. Most of these works depend on learning a synthesis dictionary and the corresponding sparse code for recognition. However in recent years, an alternate analysis coding based framework (also known as co-sparse model) has been proposed for learning sparse models. In this work [113], we study this framework for image classification. We demonstrate that the proposed approach is robust and efficient, while giving a comparable or better recognition performance than the traditional synthesis-based models.

Finally, we present discussions and describe extensions of the proposed approaches.

1.3 Organization

The dissertation is organized as follows. The method for synthesis-based low resolution face recognition is presented in Chapter 2. In Chapter 3, we present a robust feature-level fusion method for multimodal recognition. The method for domain-adaptive dictionary learning is presented in Chapter 4. The analysis sparse coding model is discussed in Chapter 5. A summary and future research directions are presented in Chapter 5.

Chapter 2: Low-Resolution Face Recognition

2.1 Introduction

Face recognition has been an active field of research in biometrics for over two decades [146]. Current methods work well when the test images are captured under controlled conditions. However, quite often the performance of most algorithms degrades significantly when they are applied to the images taken under uncontrolled conditions where there is no control over pose, illumination, expressions and resolution of the face image. Image resolution is an important parameter in many practical scenarios such as surveillance where high resolution cameras are not deployed due to cost and data storage constraints and further, there is no control over the distance of faces from the camera. Figure 2.1 illustrates a practical scenario where one is faced with a challenging problem of recognizing humans when the captured face images are of very low resolution (LR).

Many methods have been proposed in the vision literature that can deal with this resolution problem in FR. Most of these methods are based on application of super-resolution (SR) technique to increase the resolution of images so that the recovered higher-resolution (HR) images can be used for recognition. One of the major drawbacks of applying SR techniques is that there is a possibility that the recovered HR images may contain some serious artifacts. This is often the case when the resolution of the image is very low. As a result, these recovered images may not look like the images of the same person and the recognition performance may degrade significantly.

In practical scenarios, the resolution change is also coupled with other variations such as pose change, illumination and expression. Algorithms specifically designed to

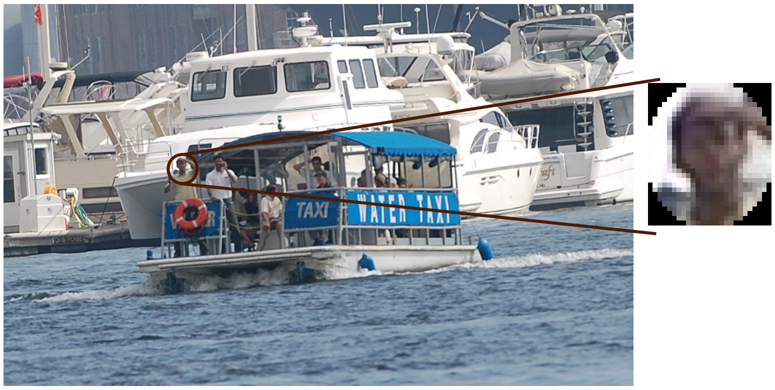


Figure 2.1: A typical image in remote face recognition.

deal with LR images quite often fail in dealing with these variations. Hence, it is essential to include these parameters while designing a robust method for low-resolution FR. To this end, in this dissertation, we present a generative approach to low-resolution FR that is also robust to illumination variations based on learning class specific dictionaries. One of the major advantages of using generative approaches is that they have reduced sensitivity to noise than the discriminative approaches [146]. Furthermore, we kernelize the learning algorithm to handle non-linearity in the data samples and present a joint sparse coding framework for robust recognition.

The training stage of our method consists of three main steps. In the first step, given HR training samples from each class, we use an image relighting method to generate multiple images of the same subject with different lighting so that robustness to illumination changes can be realized. In the second step, the resolution of the enlarged gallery images from each class is matched with that of the probe image. Finally, in the third step, class and resolution specific dictionaries are trained. For the testing phase, a novel LR image is projected onto the span of the atoms in each learned dictionary. The residual vectors are then used to classify the subject. A flowchart of the proposed algorithm is shown in Figure 2.2.

The key contributions of this work are:

1. We propose a synthesis-based method for LR FR that is robust to illumination variations, and a dictionary learning framework for classification at low resolutions.

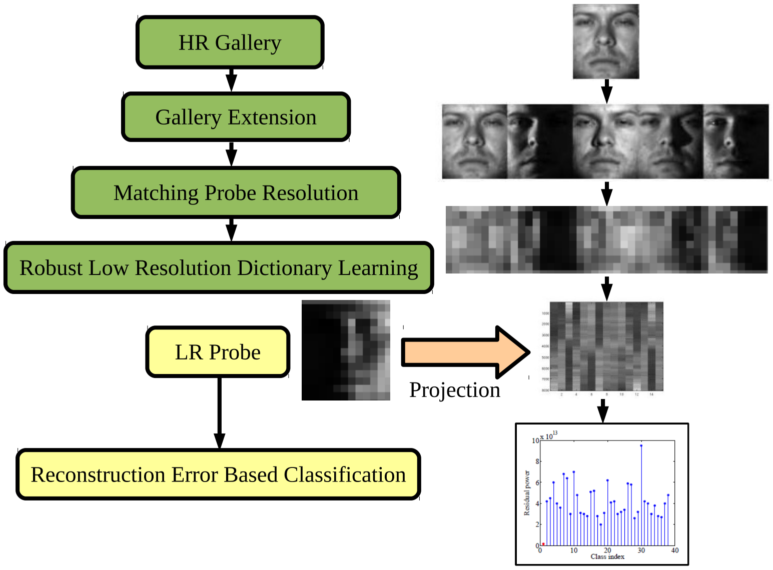

HR Gallery
Gallery Extension
Matching Probe Resolution
Robust Low Resolution Dictionary Learning
LR Probe
Reconstruction Error Based Classification
Projection
x 10^13
Residual power
Class index
0
2
4
6
8
10
0
10
20
30
40

Figure 2.2: Overview of the proposed low resolution face recognition framework.

2. We extend our method from linear to non-linear case by learning a dictionary in the high-dimensional feature space using kernel methods.

3. A joint non-linear dictionary learning method is proposed for LR FR that shares common sparse codes between HR and LR dictionaries.

2.1.1 Chapter organization

The rest of the chapter is organized as follows: In Section 2.2, we review some related works. The proposed approach is described in Section 2.3 and experimental results are presented in Section 2.4. The computational efficiency of the proposed approaches is analyzed in Section 2.5. Finally, Section 2.6 concludes the chapter with a brief summary and discussion.

2.2 Previous Work

In this section, we review some of the recent FR methods that can deal with low resolution. We also briefly discuss the relevant sparse coding literature.

2.2.1 SR-based approaches

SR is the method of estimating HR image $\mathbf{x}$ given downgraded image $\mathbf{y}$. The LR image model is often given as

$$\mathbf{y} = \mathbf{BHx} + \eta,$$

where $\mathbf{B}, \mathbf{H}$ and η are the downsampling matrix, the blurring matrix and the noise, respectively. Earlier works for solving the above problem were based on taking multiple LR inputs and combining them to produce the HR image. A classical work by Baker and Kanade [6] showed that the methods using multiple LR images using smooth priors often fail to produce good results as the resolution factor increases. They also proposed a face hallucination method for super-resolving face images. Subsequently, there have been works using single image for SR such as example-based SR [35], SR using neighborhood embedding [20] and sparse representation-based SR [137].

While these methods can be used for super-resolving the face images, that can be subsequently recognized, methods have also been proposed for specifically handling the problem for faces. In particular, an eigen-face domain SR method for FR was proposed by Gunturk *et al* in [45]. This method proposes to solve the FR at LR using SR of multiple LR images using their PCA domain representation. Given an LR face image, Jia and Gong [52] propose to directly compute a maximum likelihood identity parameter vector in the HR tensor space that can be used for SR and recognition. Hennings-Yeomans *et al.* [47] presented a Tikhonov regularization method that can combine the different steps of SR and recognition in one step. Wilman *et al.* [149] proposed a relational learning approach for super-resolution and recognition of low resolution faces.

2.2.2 Metric learning-based approaches

Though LR face images are directly not suitable for face recognition purpose, it is also not necessary to super-resolve the images before recognition, as the problem of recognition is not the same as SR. Based on this motivation, some different approaches to this problem have been suggested. The method of Coupled Metric Learning [64] attempts to solve this problem by mapping the LR image to a new subspace, where higher recognition can be achieved. A similar approach for improving the matching performance of the LR images using multidimensional scaling was recently proposed by Biswas *et al.* in [10–12]. Further, Ren *et al.* [102] used coupled kernel methods for low-resolution face recognition. A coupled Fisher analysis method was proposed by Sienna *et al.* [124]. Lei *et al.* [63]. also proposed a coupled discriminant analysis framework for heterogenous face recognition.

2.2.3 Other methods

There have been several attempts to solve the problem of unconstrained FR using videos. In particular, Arandjelovic and Cipolla [3] use a video database of LR face images with variations in pose and illumination. Their method combines a photometric model of image formation with a statistical model of generic face appearance variation to deal with illumination. To handle pose variation, it learns local appearance manifold structure and a robust same-identity likelihood.

A change in resolution of the image changes the scale of the image. Scale change has a multiplicative effect on the distances in image. Hence, if the image is represented in log-polar domain, a scale change will lead to a translation in the said domain. Based on this, an FR approach has been suggested by Hotta *et al.* in [48] to make the algorithm scale invariant. This method proposes to extract shift-invariant features in the log-polar domain.

Additionally a support vector data description method for LR FR has been described in [62]. 3D face modeling has also been used to address the LR face recognition problem [71] [98]. Choi *et al.* [23] present an interesting study on the use of color for degraded face recognition.

2.2.4 Sparse Coding

In recent years, sparse representation-based classification method (SRC) has emerged as a powerful tools for various classification problems. Wright *et al.* [135] proposed the seminal SRC algorithm for face recognition. It was shown that by exploiting the inherent sparsity of data, one can obtain improved recognition performance over traditional methods especially when data are contaminated by various artifacts such as illumination variations, disguise, occlusion, and random pixel corruption. A review of linear and non-linear dictionary-based algorithms for face recognition is presented in Patel *et al.* [87]. Further, a framework for joint sparse coding has been used for various tasks, like super-resolution [137] and cross-view recognition [53]. The motivation for using joint sparse coding in such tasks is due to being able to transfer the sparse codes between high and low resolution image patches [137] or combine information from multiple views [53]. In this chapter, we propose a method for learning joint dictionaries for HR and corresponding LR gallery images for robust recognition at low resolutions.

2.3 Proposed Approach

In this section, we present the details of the proposed low-resolution FR algorithm based on learning class specific dictionaries.

2.3.1 Image Relighting

As discussed earlier, the resolution change is usually coupled with other parameters such as illumination variation. In this section, we introduce an image relighting method that can deal with this illumination problem in LR face recognition. The idea is to capture various illumination conditions using the HR training samples, and subsequently use the expanded gallery for recognition at low resolutions.

Assuming the Lambertian reflectance model for facial surface, the HR intensity

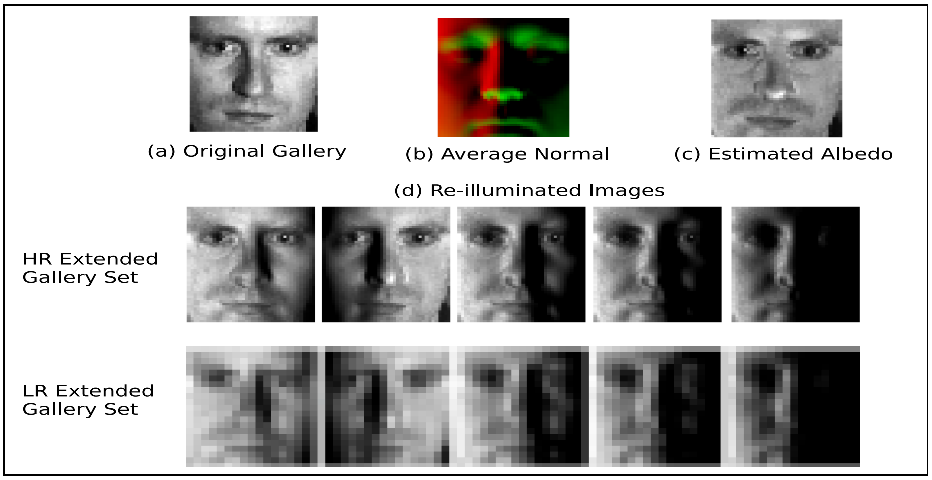


Figure 2.3: Examples of the (a) original image, (b) average normal used for calculation, (c) estimated albedo and (d) re-illuminated HR and LR gallery images.

image $\mathbf{I}^H$ is given by the Lambert's cosine law as follows:

$$\mathbf{I}^H(i,j) = \boldsymbol{\rho}(i,j)\max(\mathbf{n}(i,j)^T\mathbf{s}, 0), \tag{2.1}$$

where $\mathbf{I}^H(i,j)$ is the pixel intensity at location (i,j), $\mathbf{s}$ is the light source direction, $\boldsymbol{\rho}(i,j)$ is the surface albedo at location (i,j), $\mathbf{n}(i,j)$ is the surface normal of the corresponding surface point. Given the face image, $\mathbf{I}^H$, image relighting involves estimating $\boldsymbol{\rho}$, $\mathbf{n}$ and $\mathbf{s}$, which is an extremely ill-posed problem. To overcome this, we use 3D facial normal data [13] to first estimate an average surface normal, $\bar{\mathbf{n}}$. Further, the model is non-linear due to the $\max$ term in (2.1). However, the shadow points do not reveal any information about albedo. Hence, we neglect the $\max$ term in further discussion. The albedo, $\boldsymbol{\rho}$ and source directions $\mathbf{s}$ can now be estimated as follows:

- The source direction can be estimated using $\bar{\mathbf{n}}$ and assuming unit albedo following a linear Least Squares approach [16]:

$$\hat{\mathbf{s}} = \left(\sum_{i,j}\bar{\mathbf{n}}(i,j)\bar{\mathbf{n}}(i,j)^T\right)^{-1}\sum_{i,j}\mathbf{I}^H(i,j)\bar{\mathbf{n}}(i,j).$$

- An inital estimate of albedo, $\boldsymbol{\rho}^0$ can be obtained as:

$$\boldsymbol{\rho}^0(i,j) = \frac{\mathbf{I}^H(i,j)}{\bar{\mathbf{n}}(i,j)^T\hat{\mathbf{s}}}.$$

- The final albedo estimate is obtained using minimum mean square approach based on Wiener filtering framework [9]:

$$\hat{\boldsymbol{\rho}} = E(\boldsymbol{\rho}|\boldsymbol{\rho}^0),$$

where, $E(\boldsymbol{\rho}|\boldsymbol{\rho}^0)$ denotes the minimum mean square estimate (MMSE) of the albedo.

Using the estimated albedo map, $\hat{\boldsymbol{\rho}}$ and average normal, $\bar{\mathbf{n}}$ we can generate new images under any illumination condition using the image formation model (2.1). It was shown in [61] that an image of an arbitrarily illuminated face can be approximated by a linear combination of face images in the same pose, illuminated by nine different light sources placed at pre-selected positions.

Hence, the image formation equation can be rewritten as

$$\mathbf{I}^H = \sum_{k=1}^{9} a_k \mathbf{I}_k^H, \tag{2.2}$$

where

$$\mathbf{I}_k^H(i,j) = \boldsymbol{\rho}(i,j) \max(\mathbf{n}(i,j)^T \mathbf{s}_k, 0),$$

and $\{\mathbf{s}_1, \cdots, \mathbf{s}_9\}$ are pre-specified illumination directions. Since, the objective is to generate HR gallery images which will be sufficient to account for any illumination in the probe image, we generate images under pre-specified illumination conditions and use them in the gallery. Figure 2.3 shows some relighted HR images along with the corresponding LR images and the estimated albedo. Furthermore, as the condition is true irrespective of the resolution of LR image, the same set of gallery images can be used for all resolutions.

2.3.2 Low Resolution Dictionary Learning

In LR face recognition, given labeled HR training images, the objective is to identify the class of a novel probe LR face image. Suppose that we are given C distinct face classes and a set of m_i HR training images per class, $i = \{1, \cdots, C\}$. Here, m_i corresponds to the total number of images in class i including the relighted images. We identify an $l_H \times q_H$

grayscale image as an N_H-dimensional vector, $\mathbf{x}_H$, which can be obtained by stacking its columns, where $N_H = r_H \times q_H$. Let

$$\mathbf{X}_i^H = [\mathbf{x}_{i,1}^H, \cdots, \mathbf{x}_{i,m_i}^H] \in \mathbb{R}^{N_H \times m_i}$$

be an $N_H \times m_i$ matrix of training images corresponding to the i^{th} class. For resolution and illumination robust recognition, the matrix $\mathbf{X}_i^H$ is pre-multiplied by downsampling $\mathbf{B}$ and blurring $\mathbf{H}$ matrices. Here, $\mathbf{H}$ has a fixed dimension of $N_H \times N_H$ and $\mathbf{B}$ will be of size $N_L \times N_H$, where $N_L = r_L \times q_L$, the LR probe being a grayscale image of $r_L \times q_L$. The resolution specific training matrix, $\mathbf{X}_i^L$ is thus created as

$$\mathbf{X}_i^L = \mathbf{B}\mathbf{H}\mathbf{X}_i^H \triangleq (\mathbf{X}_i^H) \downarrow . \tag{2.3}$$

Given this matrix, we seek the dictionary that provides the best representation for each elements in this matrix. One can obtain this by finding a K-atom dictionary $\mathbf{D}_i \in \mathbb{R}^{N_L \times K}$, and a sparse matrix $\mathbf{\Gamma}_i \in \mathbb{R}^{K \times m_i}$ that minimizes the following representation error

$$\begin{aligned}(\hat{\mathbf{D}}_{\mathbf{i}}, \hat{\mathbf{\Gamma}}_i) = \underset{\mathbf{D}_i, \mathbf{\Gamma}_i}{\arg\min} \ \|\mathbf{X}_i^L - \mathbf{D}_i \mathbf{\Gamma}_i\|_F^2 \ \text{ subject to} \\ \|\boldsymbol{\gamma}_k\|_0 \leq T_0 \quad \forall\, k,\end{aligned} \tag{2.4}$$

where $\boldsymbol{\gamma}_k$ represent the columns of $\mathbf{\Gamma}_i$ and the ℓ_0 sparsity measure $\|.\|_0$ counts the number of nonzero elements in the representation. Here, $\|\mathbf{A}\|_F$ denotes the Frobenius norm defined as $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j |\mathbf{A}(i,j)|^2}$. Many approaches have been proposed in the literature for solving such optimization problems. We adapt the K-SVD algorithm [2] for solving (2.4) due to its simplicity and fast convergence. The K-SVD algorithm alternates between sparse-coding and dictionary update steps. In the sparse-coding step, $\mathbf{D}_i$ is fixed and the representation vectors $\boldsymbol{\gamma}_k$s are found for each example $\mathbf{x}_{i,j}^L$. Then, with fixed a $\mathbf{\Gamma}_i$, the dictionary is updated atom-by-atom in an efficient way. See [2] for more details on the K-SVD dictionary learning algorithm.

2.3.2.1 Classification:

Given an $r_L \times q_L$ LR probe, it is column-stacked to give the column vector $\mathbf{y}$. It is projected onto the span of the atoms in each $\mathbf{D}_i$ of the C class dictionary, using the orthogonal projector

$$\mathbf{P}_i = \mathbf{D}_i(\mathbf{D}_i^T\mathbf{D}_i)^{-1}\mathbf{D}_i^T.$$

The approximation and residual vectors can then be calculated as

$$\hat{\mathbf{y}}_i = \mathbf{P}_i\mathbf{y} = \mathbf{D}_i\boldsymbol{\alpha}_i \tag{2.5}$$

and

$$\begin{aligned}\mathbf{r}_i(\mathbf{y}) &= \mathbf{y} - \hat{\mathbf{y}}_i \\ &= (\mathbf{I} - \mathbf{P}_i)\mathbf{y},\end{aligned} \tag{2.6}$$

respectively, where $\mathbf{I}$ is the identity matrix and

$$\boldsymbol{\alpha}_i = (\mathbf{D}_i^T\mathbf{D}_i)^{-1}\mathbf{D}_i^T\mathbf{y} \tag{2.7}$$

are the coefficients. Since the K-SVD algorithm finds the dictionary, $\mathbf{D}_i$, that leads to the best representation for each examples in $\mathbf{X}_i^L$, $\|\mathbf{r}_i(\mathbf{y})\|_2$ will be small if $\mathbf{y}$ were to belong to the i^{th} class and large for the other classes. Based on this, we can classify $\mathbf{y}$ by assigning it to the class, $d \in \{1, \cdots, C\}$, that gives the lowest reconstruction error, $\|\mathbf{r}^i(\mathbf{y})\|_2$:

$$\begin{aligned}d &= \text{identity}(\mathbf{y}) \\ &= \arg\min_i \|\mathbf{r}_i(\mathbf{y})\|_2.\end{aligned} \tag{2.8}$$

2.3.2.2 Generic Dictionary Learning:

The class-specific dictionary, $\mathbf{D}_i, i = 1, \cdots, C$ learnt above can be extended to use features other than intensity images. Specifically, the dictionary can be learnt using features like Eigenbasis, $\mathbf{F}_i^H$ extracted from training matrix $\mathbf{X}_i^H$. However, as equation (2.3) does not hold for $\mathbf{F}_i^H$, the resolution specific feature matrix $\mathbf{F}_i^L$ is directly extracted using $\mathbf{X}_i^L$. Our Synthesis-based LR FR (SLRFR) algorithm is summarized in Figure 2.4.

Given a LR test sample $\mathbf{y}$ and C training matrices $\{\mathbf{X}_i^H\}_{i=1}^C$ corresponding to HR gallery images.

Procedure:

- **Gallery Extension**: For each training image, use the relighting approach described in section 2.3.1 to generate multiple images with different illumination conditions and use them in the gallery.
- Learn the best dictionaries $\mathbf{D}_i$, to represent the resolution specific enlarged training matrices, $\mathbf{X}_i^L$, using the K-SVD algorithm, where $\mathbf{X}_i^L = (\mathbf{X}_i^H) \downarrow$, $i = 1, \cdots, C$.
- Compute the approximation vectors, $\hat{\mathbf{y}}^i$, and the residual vectors, $\mathbf{r}^i(\mathbf{y})$, using (2.5) and (2.6), respectively for $i = 1, \cdots, C$.
- Identify $\mathbf{y}$ using (2.8).

Figure 2.4: The SLRFR algorithm.

2.3.3 Non-linear Dictionary Learning

The class identities in the face dataset may not be linearly separable. Hence, we also extend the SLRFR framework to the kernel space. This essentially requires the dictionary learning model to be non-liner [129].

Let $\boldsymbol{\phi}^L : \mathbb{R}^{N_L} \to G$ be a non-linear mapping from N_L dimensional space into an inner product space G. A non-linear dictionary can be trained in the feature space G by solving the following optimization problem

$$(\hat{\mathbf{A}}_i, \hat{\mathbf{\Gamma}}_i) = \underset{\mathbf{A}_i, \mathbf{\Gamma}_i}{\arg\min} \ \|\boldsymbol{\phi}^L(\mathbf{X}_i^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\mathbf{\Gamma}_i\|_F^2 \ \text{ subject to}$$
$$\|\boldsymbol{\gamma}_k\|_0 \leq T_0 \quad \forall\, k, \tag{2.9}$$

where

$$\boldsymbol{\phi}^L(\mathbf{X}_i^L) = [\boldsymbol{\phi}^{\boldsymbol{L}}(\mathbf{x}_{i,1}^L), \cdots, \boldsymbol{\phi}^L(\mathbf{x}_{i,m_i}^L)].$$

In (2.9) we have used the following model for the dictionary in the feature space,

$$\tilde{\mathbf{D}}_i = \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i,$$

Since it can be shown that the dictionary lies in the linear span of the samples $\boldsymbol{\phi}^L(\mathbf{X}_i^L)$, where $\mathbf{A}_i \in \mathbb{R}^{m_i \times K}$ is a matrix with K atoms [129]. This model provides adaptivity via modification of the matrix $\mathbf{A}_i$. Through some algebraic manipulations, the cost function in (2.9) can be rewritten as,

$$\begin{aligned}\|\boldsymbol{\phi}^L(\mathbf{X}_i^L) - &\boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\mathbf{\Gamma}_i\|_F^2 \\ &= \mathbf{tr}((\mathbf{I} - \mathbf{A}_i\mathbf{\Gamma}_i)^T \boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L, \mathbf{X}_i^L)(\mathbf{I} - \mathbf{A}_i\mathbf{\Gamma}_i)),\end{aligned} \tag{2.10}$$

where $\boldsymbol{\mathcal{K}}^L$ is a kernel matrix whose elements are computed from

$$\kappa(i,j) = \boldsymbol{\phi}^L(\mathbf{x}_i^L)^T \boldsymbol{\phi}^L(\mathbf{x}_j^L).$$

It is apparent that the objective function is well-defined since it only involves a matrix of finite dimension $\boldsymbol{\mathcal{K}}^L \in \mathbb{R}^{m_i \times m_i}$, instead of dealing with a possibly infinite dimensional dictionary.

An important property of this formulation is that the computation of $\boldsymbol{\mathcal{K}}^L$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\boldsymbol{\phi}^L$. Some commonly used kernels include polynomial kernels

$$\boldsymbol{\mathcal{K}}^L(\mathbf{x}, \mathbf{y}) = \langle(\mathbf{x}, \mathbf{y}\rangle + c)^d$$

and Gaussian kernels

$$\boldsymbol{\mathcal{K}}^L(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right),$$

where c, d and σ are parameters.

Similar to the optimization of (2.4) using the linear K-SVD [2] algorithm, the optimization of (2.9) involves sparse coding and dictionary update steps in the feature space which results in the kernel dictionary learning algorithm [129]. Details of the optimization algorithm can be found in [129] and Appendix A.

2.3.3.1 Classification:

Let $\{\mathbf{A}_i\}_{i=1}^C$ denote the learned dictionaries for C classes. Let $\mathbf{z} \in \mathbb{R}^{N_L}$ be a vectorized LR probe image z of size $r_L \times q_L$. We first find coefficient vectors $\boldsymbol{\gamma}_i \in \mathbb{R}^K$ with at most T non-zero coefficients such that $\boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_i$ approximates $\mathbf{z}$ by minimizing the following problem

$$\min_{\boldsymbol{\gamma}_i} \; \|\boldsymbol{\phi}^L(\mathbf{z}) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_i\|_2^2 \;\; s.t \;\; \|\boldsymbol{\gamma}_i\|_0 \leq T, \tag{2.11}$$

for all $i = 1, \cdots, C$. The above problem can be solved by the Kernel Orthogonal Matching Pursuit (KOMP) algorithm [129]. The reconstruction error is then computed as

$$\begin{aligned} \mathbf{r}_i &= \|\boldsymbol{\phi}^L(\mathbf{z}) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_i\|^2 \\ &= \boldsymbol{\mathcal{K}}^L(\mathbf{z}, \mathbf{z}) - 2\boldsymbol{\mathcal{K}}^L(\mathbf{z}, \mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_i + \\ \boldsymbol{\gamma}_i^T\mathbf{A}_i^T&\boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L, \mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_i, \end{aligned} \tag{2.12}$$

where,

$$\boldsymbol{\mathcal{K}}^L(\mathbf{z}, \mathbf{X}_i^L) = [\kappa(\mathbf{z}, \mathbf{x}_{i,1}^L), \kappa(\mathbf{z}, \mathbf{x}_{i,2}^L), \cdots, \kappa(\mathbf{z}, \mathbf{x}_{i,m_i}^L)].$$

Similar to the linear case, once the residuals are found, we can classify $\mathbf{z}$ by assigning it to the class, $d \in \{1, \cdots, C\}$, that gives the lowest reconstruction error, $\|\mathbf{r}^i(\mathbf{y})\|_2$:

$$\begin{aligned} d &= \text{identity}(\mathbf{y}) \\ &= \arg\min_i \|\mathbf{r}_i(\mathbf{y})\|_2. \end{aligned} \tag{2.13}$$

Our kernel Synthesis-based LR FR (kerSLRFR) algorithm is summarized in Figure 2.5.

Given a LR test sample $\mathbf{y}$ and C training matrices $\{\mathbf{X}_i^H\}_{i=1}^C$ corresponding to HR gallery images.

Procedure:

- Gallery extension as described in Algorithm 2.4.
- Learn non-linear dictionaries $\mathbf{A}_i$, to represent the resolution specific enlarged training matrices, $\mathbf{X}_i^L$, using the kernel dictionary learning algorithm 2.9, where $\mathbf{X}_i^L = (\mathbf{X}_i^H) \downarrow$, $i = 1, \cdots, C$.
- Compute the sparse codes, $\boldsymbol{\gamma}_i$ and the residual vectors, $\mathbf{r}^i$, using (2.11) and (2.12), respectively for $i = 1, \cdots, C$.
- Identify $\mathbf{y}$ using (2.13).

Figure 2.5: The kerSLRFR algorithm.

2.3.4 Joint Non-linear Dictionary Learning

In the previous sections, we described methods to learn resolution-specific dictionaries for linear and non-linear cases. However, even though dictionaries can capture class-specific variations, the recognition performance would go down at low resolutions. Hence, information available in the HR training images must be exploited to make the method robust. To enable this, we propose a framework of learning joint dictionaries

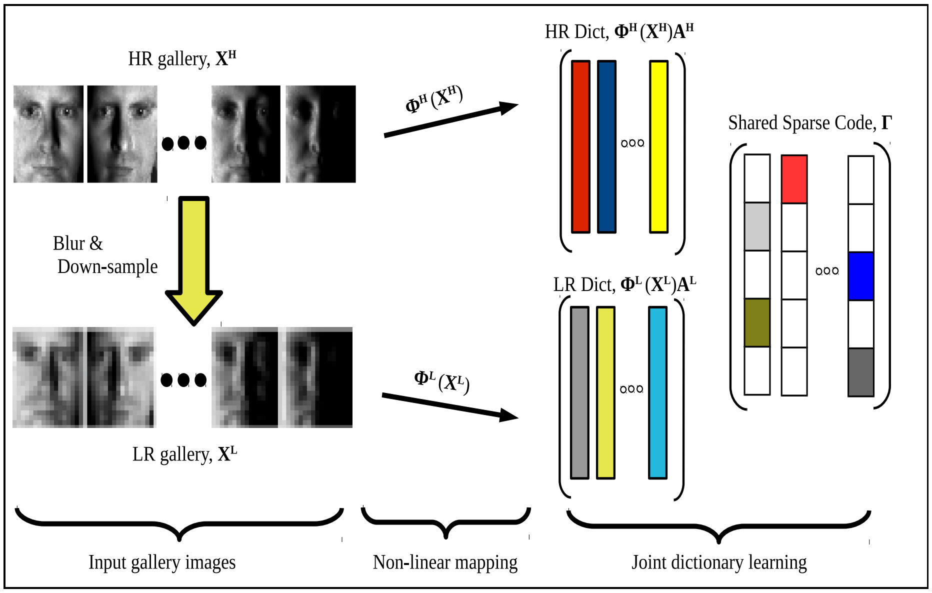

HR gallery, $\mathbf{X^H}$
Blur & Down-sample
LR gallery, $\mathbf{X^L}$
$\Phi^H(X^H)$
$\Phi^L(X^L)$
HR Dict, $\mathbf{\Phi^H(X^H)A^H}$
LR Dict, $\mathbf{\Phi^L(X^L)A^L}$
Shared Sparse Code, $\mathbf{\Gamma}$
Input gallery images
Non-linear mapping
Joint dictionary learning

Figure 2.6: Overview of the proposed joint non-linear dictionary learning approach. We constrain the LR and HR dictionaries to share sparse codes to learn robust dictionaries at low resolutions.

for HR and corresponding LR images. We achieve this through sharing sparse codes between HR and LR dictionaries. This regularizes the learned LR dictionary to output similar sparse codes as HR dictionary, thus, making it robust. The proposed formulation is described as follows. An overview of the proposed approach is also shown in Figure 2.6.

Let $\boldsymbol{\phi}^H : \mathbb{R}^{N_H} \rightarrow G$ be a non-linear mapping from N_H dimensional space into a dot product space G. We seek to learn dictionaries $\mathbf{A}^H \in \mathbb{R}^{m_i \times K}$ and $\mathbf{A}^L \in \mathbb{R}^{m_i \times K}$ by solving the optimization problem:

$$\begin{aligned}(\hat{\mathbf{A}}_i^H, \hat{\mathbf{A}}_i^L, \hat{\mathbf{\Gamma}}_i) = \underset{\mathbf{A}_i^H, \mathbf{A}_i^L, \mathbf{\Gamma}_i}{\arg\min}\ & \|\boldsymbol{\phi}^H(\mathbf{X}_i^H) - \boldsymbol{\phi}^H(\mathbf{X}_i^H)\mathbf{A}_i^H\mathbf{\Gamma}_i\|_F^2 \\ & + \lambda\|\boldsymbol{\phi}^L(\mathbf{X}_i^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i^L\mathbf{\Gamma}_i\|_F^2 \\ & \text{subject to } \|\boldsymbol{\gamma}_k\|_0 \leq T_0\ \forall\ k, \end{aligned} \tag{2.14}$$

where, $\lambda > 0$ is a hyperparameter. This can be re-formulated as:

$$(\hat{\tilde{A}}_i, \hat{\mathbf{\Gamma}}_i) = \underset{\tilde{\mathbf{A}}, \mathbf{\Gamma}_i}{\arg\min} \ \|\mathbf{\Phi}_1(\mathbf{X}_i^H, \mathbf{X}_i^L) - \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\tilde{\mathbf{A}}_i\mathbf{\Gamma}_i\|_F^2$$
$$\text{subject to } \|\boldsymbol{\gamma}_k\|_0 \leq T_0 \,\forall\, k, \tag{2.15}$$

where,

$$\mathbf{\Phi}_1(\mathbf{X}_i^H, \mathbf{X}_i^L) = \begin{bmatrix} \phi(\mathbf{X}^H) \\ \sqrt{\lambda}\phi(\mathbf{X}^L) \end{bmatrix}, \ \tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_i^H \\ \mathbf{A}_i^L \end{bmatrix},$$

$$\mathbf{\Phi}_1(\mathbf{X}_i^H, \mathbf{X}_i^L) = \begin{bmatrix} \phi(\mathbf{X}^H) & \mathbf{0} \\ \mathbf{0} & \sqrt{\lambda}\phi(\mathbf{X}^L) \end{bmatrix}.$$

The optimization problem (2.14) can be solved in a similar way as (2.9) using a modified version of kernel K-SVD algorithm [129]. Details of the method are presented in Appendix A A.

2.3.4.1 Classification:

Let $\{\mathbf{A}_i^L\}_{i=1}^C$ denote the learned dictionaries for C classes. Then a low resolution probe $\mathbf{z} \in \mathbb{R}^{N_L}$ can be classified using the KOMP algorithm [129], as described in (2.11), (2.12) and (2.13), by substituting $\{\mathbf{A}_i^L\}_{i=1}^C$ for dictionary term. The proposed algorithm referred to as joint kernel SLRFR (jointKerSLRFR) is summarized in Figure 2.7.

2.4 Experiments

To demonstrate the effectiveness of our method, in this section, we present experimental results on various face recognition datasets. We demonstrate the effectiveness of the proposed recognition framework, as well as compared with metric learning [11, 64] and SR- based [47, 149] methods. For all the experiments, we learnt the dictionary elements using the PCA features.

Given a LR test sample $\mathbf{y}$ and C training matrices $\{\mathbf{X}_i^H\}_{i=1}^C$ corresponding to HR gallery images.

Procedure:

- Gallery extension as described in Algorithm 2.4.
- Learn the dictionaries $\mathbf{A}_i^H$ and $\mathbf{A}_i^L$ to jointly represent the HR and LR training matrices, $\mathbf{X}_i^H$ and $\mathbf{X}_i^L$, where $\mathbf{X}_i^L = (\mathbf{X}_i^H) \downarrow$, $i = 1, \cdots, C$, respectively using the joint kernel dictionary algorithm.
- Using the learnt dictionary $\mathbf{A}_i^L$, compute the the sparse codes, $\boldsymbol{\gamma}_i$ and the residual vectors, $\mathbf{r}^i$, using (2.11) and (2.12) respectively for $i = 1, \cdots, C$.
- Identify $\mathbf{y}$ using (2.13).

Figure 2.7: The jointKerSLRFR algorithm.

2.4.1 FRGC Dataset

We present results on Experiment 1 of the FRGC dataset [91]. It consists of 152 gallery images, each subject having one gallery and 608 probe images under controlled setting. A separate training set of 183 images is also available which was used to learn the PCA basis.

2.4.1.1 Implementation

The resolution of the HR image was fixed at 48×40 and the probe images at resolutions of 10×8 and 7×6 were created by smoothening and downsampling the HR probe images. From each gallery image, 5 different illumination images were produced, which were flipped to give 10 images per subject. The experiments were done at resolutions of 10×8 and 7×6, thus validating the method across resolutions. We also tested the CLPM algorithm [64] and PCA performances on the expanded gallery to get a fair comparison. We also report the recognition rate for PCA using the original gallery image to demonstrate the utility of gallery extension at low resolutions. Results from other algorithms are also tabulated. We chose RBF kernel for testing kerSLRFR and jointKerSLRFR and set $\lambda = 1$ for jointKerSLRFR. The kernel parameter, σ was obtained through cross-validation for both HR and LR data. The dictionary size, K was set to 7 and the sparsity, T_0 was taken as 4. We used the nearest neighbor method for classification using PCA features and CLPM [64] method.

2.4.1.2 Observations

Figure 2.8 and Table 2.1 show that the proposed methods clearly outperforms previous algorithms. The proposed algorithm, SLRFR improves the CLPM algorithm for all the resolutions, while kerSLRFR further boosts the performance. The jointKerSLRFR shows the best performance for all the methods. The joint sparse coding framework, clearly helps in improving performance at low resolutions. Further, PCA based on the extended gallery set also improves the performance over using a single gallery image. This shows that our method of gallery extension can be coupled with the existing face recognition algorithms to improve performance at low resolutions.

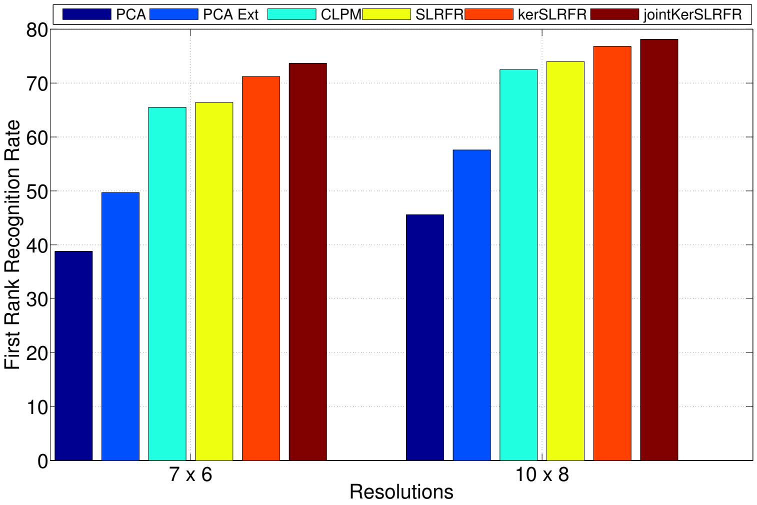


Figure 2.8: Recognition Rates for FRGC data with probes at low resolutions

Resolution	MDS [11]	S2R2 [47]	VLR [149]	PCA Ext	CLPM	SLRFR	kerSLRFR	jointKerSLRFR
6×6	-	55.0%	-	45.1%	60.7%	62.9%	64.7%	$\mathbf{65.2}\%$
7×6	-	-	55.5%	49.7%	65.5%	66.4%	71.2%	$\mathbf{73.6}\%$
9×7	58.0%	-	-	56.1%	70.2%	72.2%	76.4%	$\mathbf{78.1}\%$

Table 2.1: Comparisons for rank one recognition rate of FRGC dataset

2.4.1.3 Sensitivity to noise:

Low resolution images are often corrupted by noise. Thus, senstivity to noise is critical in assessing the performance of different algorithms. Figure 2.9 shows the recognition rates for different algorithms with increasing noise level. It can be seen that CLPM shows a sharp decline with increasing noise, but the proposed approaches SLRFR, kerSLRFR and jointKerSLRFR are stable with noise. This is because the CLPM algorithm learns a model tailored to noise-free low resolution images, whereas the generative approach in the proposed methods leads to stable performance with increasing noise.

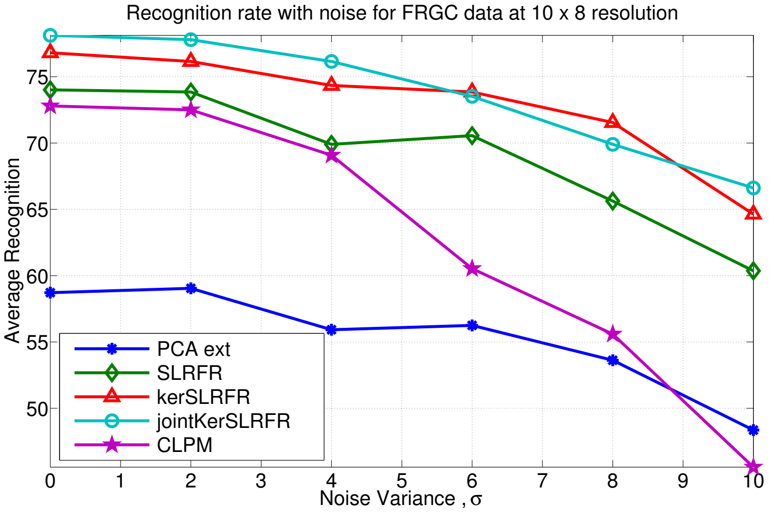


Figure 2.9: Recognition Rates for FRGC data across increasing noise levels at 10×8 LR probe resolutions

2.4.2 CMU-PIE dataset

The PIE dataset [125] consists of 68 subjects in frontal pose and under different illumination conditions. Each subject has 21 face images under different illumination conditions.

2.4.2.1 Implementation

We chose the first 34 subjects with 6 randomly chosen illuminations as the training set to learn PCA basis. For the remaining 34 subjects and the 15 illumination conditions, the experiment was done by choosing one gallery image per subject and taking the remaining as probe images. The procedure was repeated for all the images and the final recognition rate was obtained by averaging over all the images. The size of the HR images was fixed to 48×40. The LR images were obtained by smoothening followed by downsampling the HR images. For each galley image, 10 images under different illuminations produced using gallery extension method and the corresponding flipped images were added to the gallery set. The RBF kernel was chosen for kerSLRFR and jointKerSLRFR and the kernel parameter, σ was set through cross-validation. We set $\lambda = 1$ for all the experiments.

Resolution	MDS [11]	VLR* [149]	PCA ext	CLPM [64]	SLRFR	kerSLRFR	jointKerSLRFR
7×6	55.0%	74%	51.7%	64.6%	73.3%	76.5%	**76.9%**
12×10	73.0%	–	63.5%	73.5%	83.8%	86.8%	**87.4%**
19×16	78.0%	–	83%	85.6%	87.1%	89.7%	**90.0%**

Table 2.2: Comparisons for rank one recognition of PIE dataset rate. Note that VLR* [149] uses multiple gallery images while training.

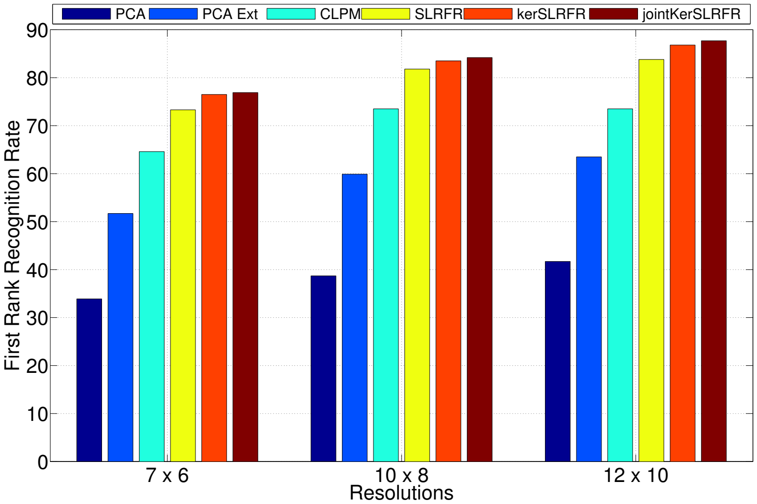


Figure 2.10: Recognition Rates for PIE data with probes at low resolutions

2.4.2.2 Observations

Figures 2.10, 2.11 and Table 2.2 show that the proposed method clearly outperforms previous algorithms. The proposed algorithms shows over 20% improvement over the MDS method [11] and 8% better than the CLPM method at rank one recognition rate, for

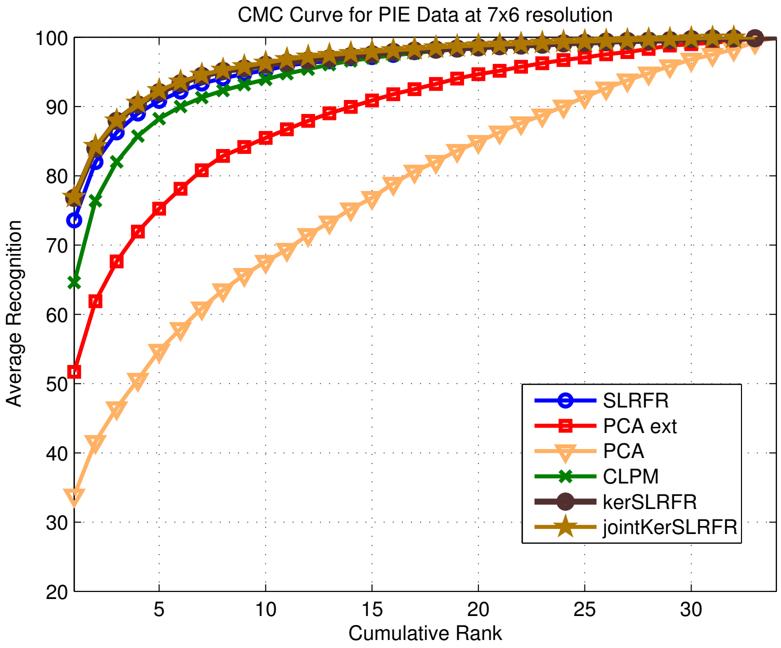


Figure 2.11: CMC (Cumulative Match Characteristic) Curves for PIE data with probes at 7×6 resolution

the probe resolution of 7×6. The kerSLRFR and jointKerSLRFR methods report better performance than VLR algorithm [149] at 7×6 resolution. Further, the CMC curves for SLRFR, kerSLRFR and jointKerSLRFR lie above the other methods for all the ranks, as shown in Figure 2.11. PCA using the extended gallery set also improves the performance over using a single gallery image.

2.4.3 AR Face dataset

We also tested the proposed algorithms on the AR Face dataset [69]. The AR face dataset consists of faces with varying illumination and expression conditions, captured in two sessions. We evaluated our algorithms on a set of 100 users. Images from the first session, seven for each subject,were used as training and gallery and the images from the second session, again seven per subject, were used for testing.

2.4.3.1 Implementation

To test our method and compare with existing metric-learning based methods [64] [11], we chose first 30 subjects from the first session as the training set. For the remaining 70 subjects, the experiment was done by choosing one gallery image per subject from the first session and taking the corresponding images from session 2 as probes. The procedure was repeated for all the 7 images in the session 1 and the final recognition rate was obtained by averaging over all the runs. The size of the HR images was fixed to 55×40. The LR images were obtained by smoothening followed by downsampling the HR images to 14×10. We also tested the performance of the CLPM [64] and PCA algorithms on the expanded gallery to get a fair comparison. Results from other algorithms are also tabulated.

2.4.3.2 Observations

Figure 2.12 shows the CMC curve for the first 5 ranks. Clearly, the proposed approaches outperform other methods. SLRFR gives better rank one performance than the CLPM algorithm, while kerSLRFR and jointKerSLRFR further increases the recognition over all the ranks. This further demonstrates that the proposed algorithms can also handle variations like expression change in the LR probe.

2.4.4 Outdoor Face Dataset

We also tested our method on a challenging outdoor face dataset. The database consists of face images of 18 individuals at different distances from camera. We chose a subset of 90 low resolution images, which were also corrupted with blur, illumination and pose variations. 5 high resolution, frontal and well-illuminated images were taken as the gallery set for each subject. The images were aligned using 5 manually selected facial points. Automatic alignment of LR faces using landmarks is a challenging problem by itself and we will explore in a separate work. The gallery resolution was fixed at 120×120

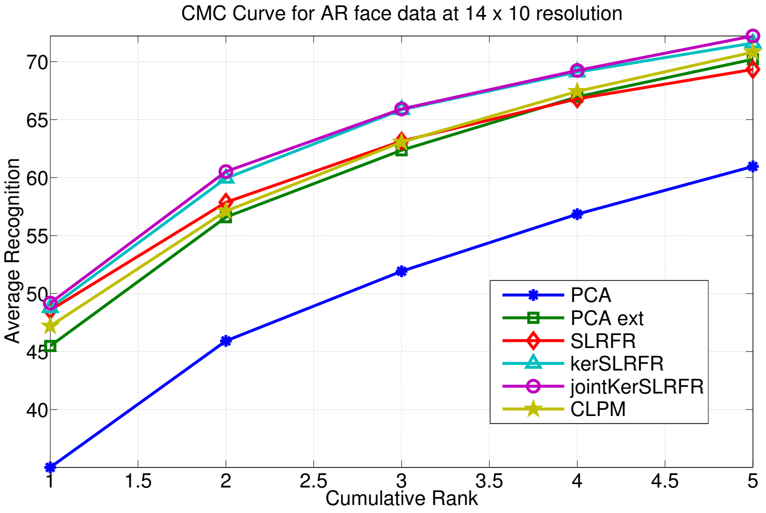


Figure 2.12: CMC Curves for AR face data with probes at 14×10 resolution

and the probe resolution at 20×20. Figure 2.13 shows some of the gallery images and the low quality probe images. The recognition rates for the dataset are shown in Table 2.3. We compare our method with the Regularized Discriminant Analysis (RDA) [36] and CLPM [64]. For the RDA comparison, we first used the PCA as a dimensionality reduction method to project the raw data onto an intermediate space, then we used the RDA to project the PCA coefficients onto a final feature space.

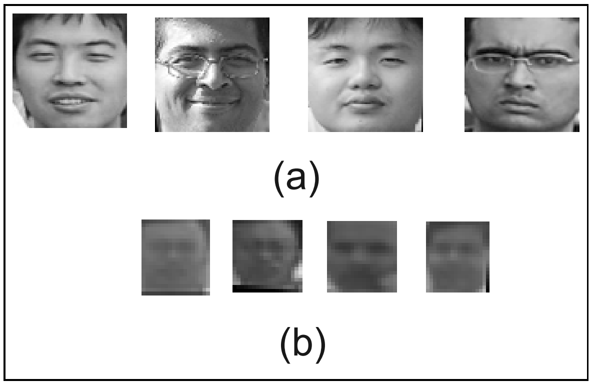


Figure 2.13: Example images from the outdoor face dataset (a) HR gallery images (b) LR probe images

Method	Recognition Rate
PCA	58.9%
reg LDA [36]	60%
CLPM [64]	16.7%
SLRFR	67.8%
kerSLRFR	$\mathbf{71}.1\%$
jointKerSLRFR	$\mathbf{71}.1\%$

Table 2.3: Performance for the Outdoor Face Dataset

2.4.4.1 Observations

It can be seen from the table that SLRFR outperforms other algorithms on this difficult outdoor face dataset. The kerSLRFR algorithm further improves the performance, however, the jointKerSLRFR doesn't improve it further. This may be because this is a challenging dataset containing variations other than LR, like pose, blur, etc. The CLPM algorithm performs rather poorly on this dataset, as it is unable to learn the challenging variations in the dataset.

2.5 Computational Efficiency

All the experiments were conducted using the 2.13GHz Intel Xeon processor on Matlab programming interface. The gallery extension step using relighting took an average of $2s$ per gallery image of size 48×40. The SLRFR method took on an average $0.07s$ to train each class, while classification of a probe image was done in an average of $0.1s$ at the resolution of 7×6. Similarly, kerSLRFR and jointKerSLRFR took $1s$ to train each class and $0.5s$ to classify at 7×6 resolution. Thus, the proposed algorithm is computationally efficient. Further, as the extended gallery can be used for all resolutions, it can be computed once and stored for a database.

2.6 Conclusions

We proposed an algorithm that can provide good accuracy for LR face images, even when only a single HR gallery image is provided per person. While the method avoids the complexity of previously proposed algorithms, it is also shown to provide state-of-the-art results when the LR probe face differs in illumination from the given gallery image. Further, we also show good results for a dataset with expression variations and a challenging outdoor face dataset. The idea of exploiting the information in a HR gallery image is novel and can be used to extend the limits of remote face recognition. We have also proposed a non-linear extension of the algorithm and a joint sparse coding framework for robust recognition at low resolutions. In future, we plan to extend our approach to handle variations like pose, alignment, etc which can affect the recognition at low resolutions. Discriminative framework for the proposed algorithms can also be explored as a future direction.

Chapter 3: Robust Feature-level Fusion

3.1 Introduction

Unimodal biometric systems rely on a single source of information such as a single iris or fingerprint or face for authentication [104]. Unfortunately these systems have to deal with some of the following inevitable problems [103]: (a) Noisy data: poor lighting on a user's face or occlusion are examples of noisy data. (b) Non-universality: the biometric system based on a single source of evidence may not be able to capture meaningful data from some users. For instance, an iris biometric system may extract incorrect texture patterns from the iris of certain users due to the presence of contact lenses. (c) Intra-class variations: in the case of fingerprint recognition, the presence of wrinkles due to wetness [59] can cause these variations. These types of variations often occur when a user incorrectly interacts with the sensor. (d) Spoof attack: hand signature forgery is an example of this type of attack. It has been observed that some of the limitations of unimodal biometric systems can be addressed by deploying multimodal biometric systems that essentially integrate the evidence presented by multiple sources of information such as iris, fingerprints and face. Such systems are less vulnerable to spoof attacks as it would be difficult for an imposter to simultaneously spoof multiple biometric traits of a genuine user. Due to sufficient population coverage, these systems are able to address the problem of non-universality.

Classification in multibiometric systems is done by fusing information from different biometric modalities. Information fusion can be done at different levels, broadly divided into feature-level, score-level and rank/decision-level fusion. Due to preservation of raw information, feature-level fusion can be more discriminative than score or decision-

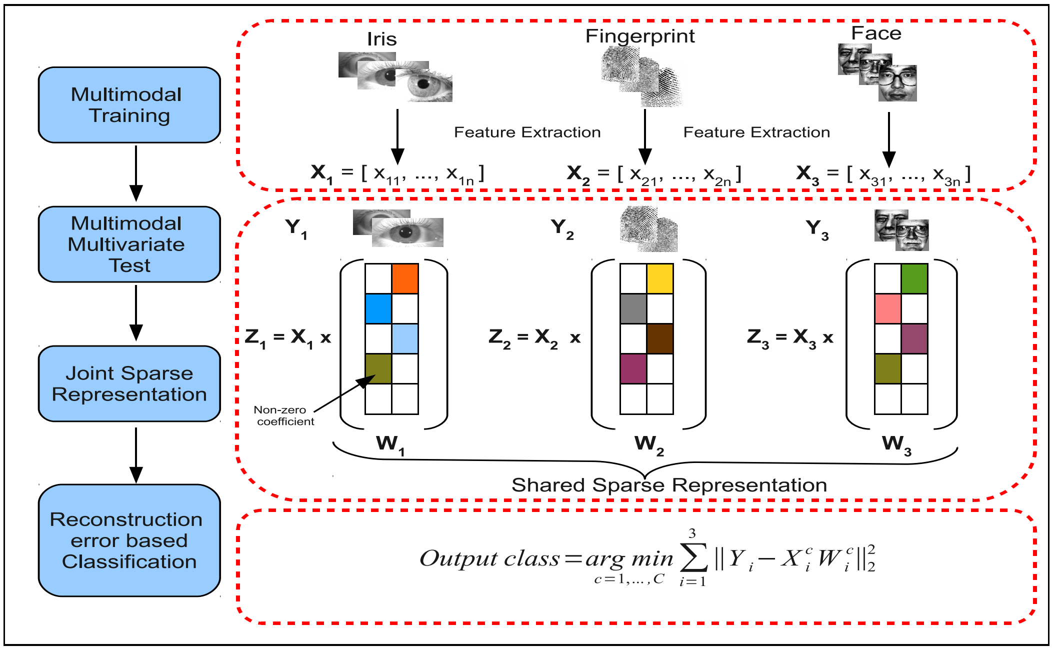

Multimodal Training
Multimodal Multivariate Test
Joint Sparse Representation
Reconstruction error based Classification
Iris
Fingerprint
Face
Feature Extraction
Feature Extraction
$X_1 = [x_{11}, \ldots, x_{1n}]$
$X_2 = [x_{21}, \ldots, x_{2n}]$
$X_3 = [x_{31}, \ldots, x_{3n}]$
Y_1
Y_2
Y_3
$Z_1 = X_1 \times$
$Z_2 = X_2 \times$
$Z_3 = X_3 \times$
Non-zero coefficient
W_1
W_2
W_3
Shared Sparse Representation
$Output\ class = \arg\min_{c=1,\ldots,C} \sum_{i=1}^{3} \| Y_i - X_i^c W_i^c \|_2^2$

Figure 3.1: Overview of our algorithm. The proposed algorithm represents the test data by a sparse linear combination of training data, while constraining the observations from different modalities of the test subject to share their sparse representations. Finally, classification is done by assigning the test data to the class with the lowest reconstruction error.

level fusion [56]. But, feature-level fusion methods are being explored in the biometric community only recently. This is because of the differences in features extracted from different sensors in terms of type and dimensions. Often features have large dimensions, and fusion becomes difficult at the feature level. The prevalent method is feature concatenation, which has been used for different multibiometric settings [99,105,148]. However, for high-dimensional feature vectors, simple feature concatenation may be inefficient and non-robust. A related work in the machine learning literature is Multiple Kernel Learning (MKL), which aims to integrate information from different features by learning a weighted combination of respective kernels. A detailed survey of MKL-based methods can be found in [38]. However, for multimodal systems, weight determination during testing is important, based on the quality of modalities. Also, a corrupted test sample from a modality must be rejected by the algorithm. Such a framework is not yet feasible in the MKL settings. Methods like [54, 126] try to exploit information from data from a different view to improve classifier performance. However, [54] being an unsupervised technique, is not suited for classification tasks, and [126] reduces to the MKL framework in a supervised setting. Similarly, SVM-2k [33] jointly learns SVM for two views, while maximizing the agreement between the projections of data from the two views. It is, however, not clear how this can be extended to multiple views, which is common in multimodal biometrics. A Fisher discriminant analysis based method has also been proposed for integrating multiple views in [27], but it is also similar to MKL with kernel Fisher discriminant analysis as the base learner [55].

In recent years, theories of Sparse Representation (SR) and Compressed Sensing (CS) have emerged as powerful tools for efficient processing of data in non-traditional ways [84]. This has led to a resurgence in interest in the principles of SR and CS for biometrics recognition [86]. Wright *et al.* [135] proposed the seminal sparse representation-based classification (SRC) algorithm for face recognition. It was shown that by exploiting the inherent sparsity of data, one can obtain improved recognition performance over traditional methods especially when data is contaminated by various artifacts such as illumination variations, disguise, occlusion and random pixel corruption. Pillai *et al.* extended this work for robust cancelable iris recognition in [93]. Nagesh and Li [75] presented an expression-invariant face recognition method using distributed CS and joint sparsity models. Patel *et al.* [88] proposed a dictionary-based method for face recognition under varying pose and illumination. A discriminative dictionary learning method for face recognition was also proposed by Zhang and Li [145]. For a survey of applications of SR and CS algorithms to biometric recognition, see [84], [86], [134], [130], [32] and the references therein.

Motivated by the success of SR in unimodal biometric recognition, we propose a joint sparsity-based algorithm for multimodal biometrics recognition. Figure 3.1 presents an overview of our framework. It is based on the well known regularized regression method, multi-task multi-variate Lasso [141], [72]. The proposed method imposes common sparsities both within each biometric modality and across different modalities. The idea of joint sparsity has been explored recently for image classification [142, 143] and segmentaion [22]. However our method is different from these previously proposed algorithms based on joint sparse representation for classification. For example, Yuan and Yan [142] proposed a multi-task sparse linear regression model for image classification. This method uses group sparsity to combine different features of an object for classification. Zhang *et al.* [143] proposed a joint dynamic sparse representation model for object recognition. Their essential goal was to recognize the same object viewed from multiple observations i.e., different poses. Our method is more general in that it can deal with both multi-modal as well as multi-variate sparse representations.

The proposed approach makes the following contributions:

- We present a robust feature level fusion algorithm for multibiometric recognition. Through the proposed joint sparse framework, we can easily handle unequal dimensions from different modalities by forcing the different features to interact through their sparse coefficients. Furthermore, the proposed algorithm can efficiently handle large dimensional feature vectors.

- We make the classification robust to occlusion and noise by introducing an error term in the optimization framework.

- The algorithm is easily generalizable to handle multiple test inputs from a modality.
- We introduce a quality measure for multimodal fusion based on the joint sparse representation.
- Lastly, we kernelize the algorithm to handle non-linearity in the data samples.

3.1.1 Organization

The chapter is organized as follows. In section 3.2, we describe the proposed sparsity-based multimodal recognition algorithm which is kernelized in section 3.4. The quality measure is described in 3.3. Experimental evaluations on a comprehensive multimodal dataset and a face database are described in section 3.5. Finally, in section 3.6, we discuss the computational complexity of the method. Concluding remarks are presented in section 3.7.

3.2 Joint sparsity-based multimodal biometrics recognition

Consider a multimodal C-class classification problem with D different biometric traits. Suppose there are $p = \sum_{j=1}^{C} p_j$ training samples in each biometric trait, where p_j is the number of training samples in class j. For each biometric trait $i = 1, \ldots, D$, we denote

$$\mathbf{X}^i = [\mathbf{X}_1^i, \mathbf{X}_2^i, \ldots, \mathbf{X}_C^i]$$

as an $n_i \times p$ dictionary of training samples consisting of C sub-dictionaries $\mathbf{X}_k^i$'s corresponding to C different classes. Each sub-dictionary

$$\mathbf{X}_j^i = [\mathbf{x}_{j,1}^i, \mathbf{x}_{j,2}^i, \ldots, \mathbf{x}_{j,p_j}^i] \in \mathbb{R}^{n_i \times p_j}$$

represents a set of training data from the ith modality labeled with the jth class. Elements of the dictionary are often referred to as atoms. In multimodal biometrics recognition problem, given test samples $\mathbf{Y}$, which consists of D different modalities $\{\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^D\}$ where each sample $\mathbf{Y}^i$ consists of d_i observations $\mathbf{Y}^i = [\mathbf{y}_1^i, \mathbf{y}_2^i, \ldots, \mathbf{y}_{d_i}^i] \in \mathbb{R}^{n_i \times d_i}$, the objective is to identify the class to which a test sample $\mathbf{Y}$ belongs to. Note that we do not constrain the number of samples per modality to be the same, as assumed in forming the training matrix. In what follows, we present a multimodal multivariate sparse representation-based algorithm for this problem [141], [72], [80].

3.2.1 Multimodal multivariate sparse representation

We propose to exploit the joint sparsity of coefficients from different biometric modalities to make a joint decision. To simplify this model, let us consider a bi-modal classification problem where the test sample $\mathbf{Y} = [\mathbf{Y}^1, \mathbf{Y}^2]$ consists of two different modalities such as iris and face. Suppose that $\mathbf{Y}^1$ belongs to the jth class. Then, it can be reconstructed by a linear combination of the atoms in the sub-dictionary $\mathbf{X}^1_j$. That is, $\mathbf{Y}^1 = \mathbf{X}^1\mathbf{\Gamma}^1 + \mathbf{N}^1$, where $\mathbf{\Gamma}^1$ is a sparse matrix with only p_j nonzero rows associated with the jth class and $\mathbf{N}^1$ is the noise matrix. Similarly, since $\mathbf{Y}^2$ represents the same subject, it belongs to the same class and can be represented by training samples in $\mathbf{X}^2_j$ with different set of coefficients $\mathbf{\Gamma}^2_j$. Thus, we can write $\mathbf{Y}^2 = \mathbf{X}^2\mathbf{\Gamma}^2 + \mathbf{N}^2$, where $\mathbf{\Gamma}^2$ is a sparse matrix that has the same sparsity pattern as $\mathbf{\Gamma}^1$. If we let $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2]$, then $\mathbf{\Gamma}$ is a sparse matrix with only p_j non-zero rows, as both $\mathbf{Y}^1$ and $\mathbf{Y}^2$ are represented by samples of jth class.

In the more general case where we have D modalities, if we denote $\{\mathbf{Y}^i\}_{i=1}^D$ as a set of D observations each consisting of d_i samples from each modality and let $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \ldots, \mathbf{\Gamma}^D] \in \mathbb{R}^{p \times d}$ be the matrix formed by concatenating the coefficient matrices with $d = \sum_{i=1}^D d_i$, then we can determine the row-sparse matrix $\mathbf{\Gamma}$ by solving the following ℓ_1/ℓ_q-regularized least square problem

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i\|_F^2 + \lambda\|\mathbf{\Gamma}\|_{1,q}, \tag{3.1}$$

where, λ is a positive parameter and q is set greater than 1 to make the optimization problem convex. Here, $\|\mathbf{\Gamma}\|_{1,q}$ is a norm defined as $\|\mathbf{\Gamma}\|_{1,q} = \sum_{k=1}^p \|\boldsymbol{\gamma}^k\|_q$ where $\boldsymbol{\gamma}^k$'s are the row vectors of $\mathbf{\Gamma}$ and $\|\mathbf{Y}\|_F$ is the Frobenius norm of the matrix $\mathbf{Y}$ defined as $\|\mathbf{Y}\|_F = \sqrt{\sum_{i,j} Y_{i,j}^2}$. The ℓ_1/ℓ_q regularization seeks a solution with sparse non-zero rows, hence, we get a representation consistent across all the modalities. Once $\hat{\mathbf{\Gamma}}$ is obtained, the class label associated with an observed vector is then declared as the one that produces the smallest approximation error,

$$\hat{j} = \arg\min_{j} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i \boldsymbol{\delta}_j^i(\mathbf{\Gamma}^i)\|_F^2, \tag{3.2}$$

where, $\boldsymbol{\delta}_j^i$ is the matrix indicator function defined by keeping rows corresponding to the jth class and setting all other rows equal to zero. Note that the optimization problem (3.1) reduces to the conventional Lasso [128] when $D = 1$ and $d = 1$. In the case, when $D = 1$ (3.1) is referred to as multivariate Lasso [141].

3.2.2 Robust multimodal multivariate sparse representation

In this section, we consider a more general problem where the data is contaminated by noise. In this case, the observation model can be modeled as

$$\mathbf{Y}^i = \mathbf{X}^i\mathbf{\Gamma}^i + \mathbf{Z}^i + \mathbf{N}^i, \quad i = 1, \ldots D, \tag{3.3}$$

where, $\mathbf{N}^i$ is a small dense additive noise and $\mathbf{Z}^i \in \mathbb{R}^{n_i \times d_i}$ is a matrix of background noise (occlusion) with arbitrarily large magnitude. One can assume that each $\mathbf{Z}^i$ is sparsely represented in some basis $\mathbf{B}^i \in \mathbb{R}^{n_i \times m_i}$. That is, $\mathbf{Z}^i = \mathbf{B}^i\mathbf{\Lambda}^i$ for some sparse matrices $\mathbf{\Lambda}^i \in \mathbb{R}^{m_i \times d_i}$. For simplicity, we assume $\mathbf{B}^i$ to be orthonormal. Hence, (3.3) can be rewritten as

$$\mathbf{Y}^i = \mathbf{X}^i\mathbf{\Gamma}^i + \mathbf{B}^i\mathbf{\Lambda}^i + \mathbf{N}^i, \quad i = 1, \ldots D, \tag{3.4}$$

With this model, one can simultaneously recover the coefficients $\mathbf{\Gamma}^i$ and $\mathbf{\Lambda}^i$ by taking advantage of the fact that $\mathbf{\Lambda}^i$ are sparse

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{\Lambda}} = \arg\min_{\mathbf{\Gamma},\mathbf{\Lambda}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{B}^i\mathbf{\Lambda}^i\|_F^2 + \lambda_1 \|\mathbf{\Gamma}\|_{1,q} + \lambda_2 \|\mathbf{\Lambda}\|_1, \tag{3.5}$$

where λ_1 and λ_2 are positive parameters and $\mathbf{\Lambda} = [\mathbf{\Lambda}^1, \mathbf{\Lambda}^2, \ldots, \mathbf{\Lambda}^D]$ is the sparse coefficient matrix corresponding to occlusion. The ℓ_1-norm of matrix $\mathbf{\Lambda}$ is defined as

$\|\mathbf{\Lambda}\|_1 = \sum_{i,j} |\Lambda_{i,j}|$. Note that the idea of exploiting the sparsity of occlusion term has been studied by Wright *et al.* [135] and Candes *et al.* [18].

Once $\mathbf{\Gamma}, \mathbf{\Lambda}$ are computed, the effect of occlusion can be removed by setting $\tilde{\mathbf{Y}}^i = \mathbf{Y}^i - \mathbf{B}^i\mathbf{\Lambda}^i$. One can then declare the class label associated to an observed vector as

$$\hat{j} = \arg\min_{j} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\boldsymbol{\delta}_j^i(\mathbf{\Gamma}^i) - \mathbf{B}^i\mathbf{\Lambda}^i\|_F^2. \tag{3.6}$$

3.2.3 Optimization algorithm

The optimization problem (3.5) is convex but difficult to solve due to the joint sparsity constraint. In this section, we present an approach based on the classical alternating direction method of multipliers (ADMM) [139], [1] to solve (3.5). Note that the optimization problem (3.1) can be solved by setting λ_2 equal to infinity. Let

$$\mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda}) = \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{B}^i\mathbf{\Lambda}^i\|_F^2.$$

Then, our goal is to solve the following optimization problem

$$\min_{\mathbf{\Gamma},\mathbf{\Lambda}} \mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda}) + \lambda_1\|\mathbf{\Gamma}\|_{1,q} + \lambda_2\|\mathbf{\Lambda}\|_1. \tag{3.7}$$

In ADMM the idea is to decouple $\mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda})$, $\|\mathbf{\Gamma}\|_{1,q}$ and $\|\mathbf{\Lambda}\|_1$ by introducing auxiliary variables to reformulate the problem into a constrained optimization problem

$$\begin{aligned} \min_{\mathbf{\Gamma},\mathbf{\Lambda},\mathbf{U},\mathbf{V}} \mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda}) + \lambda_1\|\mathbf{V}\|_{1,q} + \lambda_2\|\mathbf{U}\|_1 \quad \text{s. t.} \\ \mathbf{\Gamma} = \mathbf{V}, \mathbf{\Lambda} = \mathbf{U}. \end{aligned} \tag{3.8}$$

Since, (3.8) is an equally constrained problem, the Augmented Lagrangian method (ALM) [139] can be used to solve the problem. This can be done by minimizing the augmented Lagrangian function $f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{V}, \mathbf{U}; \mathbf{A}_\Lambda, \mathbf{A}_\Gamma)$ defined as

$$\begin{aligned} \mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda}) + \lambda_2\|\mathbf{U}\|_1 + \langle \mathbf{A}_\Lambda, \mathbf{\Lambda} - \mathbf{U}\rangle + \frac{\alpha_\Lambda}{2}\|\mathbf{\Lambda} - \mathbf{U}\|_F^2 + \\ \lambda_1\|\mathbf{V}\|_{1,q} + \langle \mathbf{A}_\Gamma, \mathbf{\Gamma} - \mathbf{V}\rangle + \frac{\alpha_\Gamma}{2}\|\mathbf{\Gamma} - \mathbf{V}\|_F^2, \end{aligned} \tag{3.9}$$

where $\mathbf{A}_\Lambda$ and $\mathbf{A}_\Gamma$ are the multipliers of the two linear constraints, and $\alpha_\Lambda, \alpha_\Gamma$ are the positive penalty parameters. The ALM algorithm solves $f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{V}, \mathbf{U}; \mathbf{A}_\Lambda, \mathbf{A}_\Gamma)$ with respect to $\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{U}$ and $\mathbf{V}$ jointly, keeping $\mathbf{A}_\Gamma$ and $\mathbf{A}_\Lambda$ fixed and then updating $\mathbf{A}_\Gamma$ and $\mathbf{A}_\Lambda$ keeping the remaining variables fixed. Due to the separable structure of the objective function $f_{\alpha_\Gamma,\alpha_\Lambda}$, one can further simplify the problem by minimizing $f_{\alpha_\Gamma,\alpha_\Lambda}$ with respect to variables $\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{U}$ and $\mathbf{V}$, separately. Different steps of the algorithm are given in Algorithm 1. In what follows, we describe each of the sub-optimization problems in detail.

```
Initialize: $\mathbf{\Gamma}_0, \mathbf{U}_0, \mathbf{V}_0, \mathbf{A}_{\Lambda,0}, \mathbf{A}_{\Gamma,0}, \alpha_\Gamma, \alpha_\Lambda$
While not converged do
  1. $\mathbf{\Gamma}_{t+1} = \arg\min_{\mathbf{\Gamma}} f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}, \mathbf{\Lambda}_t, \mathbf{U}_t, \mathbf{V}_t; \mathbf{A}_{\Gamma,t}, \mathbf{A}_{\Lambda,t})$
  2. $\mathbf{\Lambda}_{t+1} = \arg\min_{\mathbf{\Lambda}} f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}_{t+1}, \mathbf{\Lambda}, \mathbf{U}_t, \mathbf{V}_t; \mathbf{A}_{\Gamma,t}, \mathbf{A}_{\Lambda,t})$
  3. $\mathbf{U}_{t+1} = \arg\min_{\mathbf{U}} f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{U}, \mathbf{V}_t; \mathbf{A}_{\Gamma,t}, \mathbf{A}_{\Lambda,t})$
  4. $\mathbf{V}_{t+1} = \arg\min_{\mathbf{V}} f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}_{t+1}, \mathbf{\Lambda}_{t+1}, \mathbf{U}_{t+1}, \mathbf{V}; \mathbf{A}_{\Gamma,t}, \mathbf{A}_{\Lambda,t})$
  5. $\mathbf{A}_{\Gamma,t+1} \doteq \mathbf{A}_{\Gamma,t} + \alpha_\Gamma(\mathbf{\Gamma}_{t+1} - \mathbf{V}_{t+1})$
  6. $\mathbf{A}_{\Lambda,t+1} \doteq \mathbf{A}_{\Lambda,t} + \alpha_\Lambda(\mathbf{\Lambda}_{t+1} - \mathbf{U}_{t+1})$
```

Algorithm 1: Alternating Direction Method of Multipliers (ADMM).

3.2.3.1 Update step for $\mathbf{\Gamma}$

The first sub-optimization problem involves the minimization of $f_{\alpha_\Gamma,\alpha_\Lambda}(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{V}, \mathbf{U}; \mathbf{A}_\Lambda, \mathbf{A}_\Gamma)$ with respect to $\mathbf{\Gamma}$. It has the quadratic structure, which is easy to solve by setting the first-order derivative equal to zero. Furthermore, the loss function $\mathcal{C}(\mathbf{\Gamma}, \mathbf{\Lambda})$ is a sum of convex functions associated with sub-matrices $\mathbf{\Gamma}^i$, one can seek for $\mathbf{\Gamma}^i_{t+1}, \quad i = 1, \ldots, D$, which has the following solution

$$\mathbf{\Gamma}^i_{t+1} = (\mathbf{X}^{i^T}\mathbf{X}^i + \alpha_\Gamma \mathbf{I})^{-1}(\mathbf{X}^{i^T}(\mathbf{Y}^i - \mathbf{B}^i\mathbf{\Gamma}^i_t) + \alpha_\Gamma \mathbf{V}^i_t - \mathbf{A}^i_{\Gamma,t}),$$

where $\mathbf{I}$ is $p \times p$ identity matrix and $\mathbf{\Lambda}^i_t, \mathbf{\Gamma}^i_t$ and $\mathbf{A}^i_{\Gamma,t}$ are sub-matrices of $\mathbf{\Lambda}_t, \mathbf{\Gamma}_t$ and $\mathbf{A}_{\Gamma,t}$, respectively.

3.2.3.2 Update step for $\mathbf{\Lambda}$

The second sub-optimization problem is similar in nature, whose solution is given below

$$\mathbf{\Lambda}^i_{t+1} = (\mathbf{B}^{i^T}\mathbf{B}^i + \alpha_\Lambda \mathbf{I})^{-1}(\mathbf{B}^{i^T}(\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i_{t+1}) + \alpha_\Lambda \mathbf{U}^i_t - \mathbf{A}^i_{\Lambda,t}),$$

where $\mathbf{U}^i_t$ and $\mathbf{A}^i_{\Lambda,t}$ are sub-matrices of $\mathbf{U}_t$ and $\mathbf{A}_{\Lambda,t}$, respectively.

3.2.3.3 Update step for $\mathbf{U}$

The third sub-optimization problem is with respect to $\mathbf{U}$, which is the standard ℓ_1 minimization problem which can be recast as

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{\Lambda}_{t+1} + \alpha_\Lambda^{-1}\mathbf{A}_{\Lambda,t} - \mathbf{U}\|^2_F + \frac{\lambda_2}{\alpha_\Lambda}\|\mathbf{U}\|_1. \tag{3.10}$$

Equation (3.10) is the well-known shrinkage problem whose solution is given by

$$\mathbf{U}_{t+1} = \mathcal{S}\left(\mathbf{\Lambda}_{t+1} + \alpha_\Lambda^{-1}\mathbf{A}_{\Lambda,t}, \frac{\lambda_2}{\alpha_\Lambda}\right),$$

where $\mathcal{S}(a, b) = sgn(a)(|a| - b)$ for $|a| \geq b$ and zero otherwise.

3.2.3.4 Update step for $\mathbf{V}$

The final sub-optimization problem is with respect to $\mathbf{V}$ which can be reformulated as

$$\min_{\mathbf{V}} \frac{1}{2}\|\mathbf{\Gamma}_{t+1} + \alpha_\Gamma^{-1}\mathbf{A}_{\Gamma,t} - \mathbf{V}\|^2_F + \frac{\lambda_1}{\alpha_\Gamma}\|\mathbf{V}\|_{1,q}. \tag{3.11}$$

Due to the separable structure of (3.11), it can be solved by minimizing with respect to each row of $\mathbf{V}$ separately. Let $\boldsymbol{\gamma}_{i,t+1}$, $\mathbf{a}_{\Gamma,i,t}$ and $\mathbf{v}_{i,t+1}$ be rows of matrices $\mathbf{\Gamma}_{t+1}$, $\mathbf{A}_{\Gamma,t}$ and $\mathbf{V}_{t+1}$, respectively. Then for each $i = 1, \ldots, p$ we solve the following sub-problem

$$\mathbf{v}_{i,t+1} = \arg\min_{\mathbf{v}} \frac{1}{2}\|\mathbf{z} - \mathbf{v}\|^2_2 + \eta\|\mathbf{v}\|_q, \tag{3.12}$$

where $\mathbf{z} = \boldsymbol{\gamma}_{i,t+1} + \mathbf{a}_{\Gamma,i,t}\alpha_\Gamma^{-1}$ and $\eta = \frac{\lambda_1}{\alpha_\Gamma}$. One can derive the solution for (3.12) for any q. Here, we only focus on the case when $q = 2$. The solution of (3.12) has the following form

$$\mathbf{v}_{i,t+1} = \left(1 - \frac{\eta}{\|\mathbf{z}\|_2}\right)_+ \mathbf{z},$$

where $(\mathbf{v})_+$ is a vector with entries receiving values $\max(v_i, 0)$.

Our proposed Sparse Multimodal Biometrics Recognition (SMBR) method is summarized in Algorithm 2. We refer to the robust method that takes sparse error into account as SMBR-E (SMBR with error), and the initial case where it is not taken account as SMBR-WE (SMBR without error).

```
Input: Training samples $\{\mathbf{X_i}\}_{i=1}^D$, test sample $\{\mathbf{Y_i}\}_{i=1}^D$, Occlusion basis $\{\mathbf{B}\}_{i=1}^D$
Procedure: Obtain $\hat{\mathbf{\Gamma}}$ and $\hat{\mathbf{\Lambda}}$ by solving
    $\hat{\mathbf{\Gamma}}, \hat{\mathbf{\Lambda}} = \arg\min_{\mathbf{\Gamma},\mathbf{\Lambda}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{B}^i\mathbf{\Lambda}^i\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_{1,q} + \lambda_2\|\mathbf{\Lambda}\|_1$
Output: $\texttt{identity}(\mathbf{Y}) = \arg\min_j \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\boldsymbol{\delta}_j^i(\hat{\mathbf{\Gamma}}^i) - \mathbf{B}^i\hat{\mathbf{\Lambda}^i}\|_F^2$
```

Algorithm 2: Sparse Multimodal Biometrics Recognition (SMBR).

3.3 Quality based fusion

Ideally a fusion mechanism should give more weights to the more reliable modalities. Hence, the concept of quality is important in multimodal fusion. A quality measure based on sparse representation was introduced for faces in [135]. To decide whether a given test sample has good quality or not, its Sparsity Concentration Index (SCI) was calculated. Given a coefficient vector $\gamma \in \mathbb{R}^p$, the SCI is given as:

$$SCI(\boldsymbol{\gamma}) = \frac{\frac{C.\max_{j\in\{1,\cdots,C\}}\|\delta_j(\boldsymbol{\gamma})\|_1}{\|\boldsymbol{\gamma}\|_1} - 1}{C - 1},$$

where, δ_j is the indicator function keeping the coefficients corresponding to the j^{th} class and setting others to zero. SCI values close to 1 correspond to the case where the test sample can be represented well using the samples of a single class, hence is of high quality. On the other hand, samples with SCI close to 0 are not similar to any of the classes, and hence are of poor quality. This can be easily extended to the multimodal case using the joint sparse representation matrix $\hat{\boldsymbol{\Gamma}}$. In this case, we can define the quality, q_j^i for sample $\mathbf{y}_j^i$ as:

$$q_j^i = SCI(\hat{\boldsymbol{\Gamma}}_j^i),$$

where, $\hat{\boldsymbol{\Gamma}}_j^i$ is the j^{th} column of $\hat{\boldsymbol{\Gamma}}^i$. Given this quality measure, the classification rule (3.2) can be modified to include the quality measure.

$$\hat{j} = \arg\min_j \sum_{i=1}^{D} \sum_{k=1}^{d_i} q_k^i \|\mathbf{y}_k^i - \mathbf{X}^i \boldsymbol{\delta}_j(\boldsymbol{\Gamma}_k^i)\|_F^2, \tag{3.13}$$

where, $\boldsymbol{\delta}_j$ is the indicator function retaining the coefficients corresponding to j^{th} class.

3.4 Kernel space multimodal biometrics recognition

The class identities in the multibiometric dataset may not be linearly separable. Hence, we also extend the sparse multimodal fusion framework to kernel space. The kernel function, $\kappa : \mathbb{R}^n \times \mathbb{R}^n$, is defined as the inner product

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle,$$

where, ϕ is an implicit mapping projecting the vector $\mathbf{x}$ into a higher dimensional space.

3.4.1 Multivariate kernel sparse representation

Considering the general case of D modalities with $\{\mathbf{Y}^i\}_{i=1}^D$ as a set of d_i observations, the feature space representation can be written as:

$$\boldsymbol{\Phi}(\mathbf{Y}^i) = [\phi(\mathbf{y}_1^i), \phi(\mathbf{y}_2^i), ..., \phi(\mathbf{y}_d^i)].$$

Similarly, the dictionary of training samples for modality $i = 1, \cdots, D$ can be represented in feature space as

$$\boldsymbol{\Phi}(\mathbf{X}^i) = [\phi(\mathbf{X}_1^i), \phi(\mathbf{X}_2^i), \cdots, \phi(\mathbf{X}_C^i)].$$

As in joint linear space representation, we have:

$$\boldsymbol{\Phi}(\mathbf{Y}^i) = \boldsymbol{\Phi}(\mathbf{X}^i)\boldsymbol{\Gamma}^i,$$

where, $\mathbf{\Gamma}^i$ is the coefficient matrix associated with modality i. Incorporating information from all the sensors, we seek to solve the following optimization problem similar to the linear case:

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{\Phi}(\mathbf{Y}^i) - \mathbf{\Phi}(\mathbf{X}^i)\mathbf{\Gamma}^i\|_F^2 + \lambda\|\mathbf{\Gamma}\|_{1,q}, \tag{3.14}$$

where, $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \cdots, \mathbf{\Gamma}^D]$. It is clear that the information from all modalities is integrated via the shared sparsity pattern of the matrices $\{\mathbf{\Gamma}^i\}_{i=1}^{D}$. This can be reformulated in terms of kernel matrices as:

$$\begin{aligned}\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2}\sum_{i=1}^{D} \big(&\text{trace}(\mathbf{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}_i,\mathbf{X}_i}\mathbf{\Gamma}^i) \\ &-2\text{trace}(\mathbf{K}_{\mathbf{X}_i,\mathbf{Y}_i}\mathbf{\Gamma}^i)\big) + \lambda\|\mathbf{\Gamma}\|_{1,q},\end{aligned} \tag{3.15}$$

where, the kernel matrix $\mathbf{K}_{\mathbf{A},\mathbf{B}}$ is defined as:

$$\mathbf{K}_{\mathbf{A},\mathbf{B}}(i,j) = \langle \phi(\mathbf{a}_i), \phi(\mathbf{b}_j)\rangle, \tag{3.16}$$

$\mathbf{a}_i$ and $\mathbf{b}_j$ being i^{th} and j^{th} columns of $\mathbf{A}$ and $\mathbf{B}$ respectively.

3.4.2 Optimization Algorithm

Similar to the linear fusion method, we apply the ADMM to efficiently solve the problem for kernel fusion. This is done by introducing a new variable $\mathbf{V}$ and reformulating the problem (3.15) as:

$$\begin{aligned}\arg\min_{\mathbf{\Gamma},\mathbf{V}} \frac{1}{2}\sum_{i=1}^{D} &\big(\text{trace}(\mathbf{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i}\mathbf{\Gamma}^i) - 2\text{trace}(\mathbf{K}_{\mathbf{X}^i,\mathbf{Y}^i}\mathbf{\Gamma}^i)\big) \\ &+ \lambda\|\mathbf{V}\|_{1,q} \text{ s.t. } \mathbf{\Gamma} = \mathbf{V}.\end{aligned} \tag{3.17}$$

Rewriting the problem using the Lagrangian multiplier $\mathbf{P}_\Gamma$, the optimization problem becomes:

$$\begin{aligned}\arg\min_{\mathbf{\Gamma},\mathbf{V}} \frac{1}{2}\sum_{i=1}^{D} &\big(\text{trace}(\mathbf{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i}\mathbf{\Gamma}^i) - 2\text{trace}(\mathbf{K}_{\mathbf{X}^i,\mathbf{Y}^i}\mathbf{\Gamma}^i)\big) \\ &+ \lambda\|\mathbf{V}\|_{1,q} + \langle \mathbf{P}_\Gamma, \mathbf{\Gamma} - \mathbf{V}\rangle + \frac{\beta_\Gamma}{2}\|\mathbf{\Gamma} - \mathbf{V}\|_F^2,\end{aligned} \tag{3.18}$$

where, β_Γ is a positive penalty parameter. This upon re-arranging reduces to:

$$\begin{aligned}\arg\min_{\mathbf{\Gamma},\mathbf{V}} \frac{1}{2}\sum_{i=1}^{D} & \big(\text{trace}(\mathbf{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i}\mathbf{\Gamma}^i) - 2\text{trace}(\mathbf{K}_{\mathbf{X}^i,\mathbf{Y}^i}\mathbf{\Gamma}^i)\big) \\ & + \lambda\|\mathbf{V}\|_{1,q} + \frac{\beta_\Gamma}{2}\Big\|\mathbf{\Gamma} - \mathbf{V} + \frac{1}{\beta_\Gamma}\mathbf{P}_\Gamma\Big\|_F^2. \end{aligned} \tag{3.19}$$

Now, (3.19) can be solved in a similar way as the linear fusion problem in (3.5). The optimization method is summarized in Algorithm 3. It should be pointed out that each step has a simple closed-form expression.

```
Initialize: $\mathbf{\Gamma}_0, \mathbf{V}_0, \mathbf{B}_0, \beta_\Gamma$
While not converged do
    1. $\mathbf{\Gamma}_{t+1} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2}\sum_{i=1}^{D}\big(\text{trace}(\mathbf{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i}\mathbf{\Gamma}^i) - 2\text{trace}(\mathbf{K}_{\mathbf{X}^i,\mathbf{Y}^i}\mathbf{\Gamma}^i)\big) + \lambda\|\mathbf{V}_t\|_{1,q} + \frac{\beta_\Gamma}{2}\big\|\mathbf{\Gamma} - \mathbf{V_t} + \frac{1}{\beta_\Gamma}\mathbf{P}_{\Gamma,t}\big\|_F^2$
    2. $\mathbf{V}_{t+1} = \arg\min_{\mathbf{V}} \lambda\|\mathbf{V}\|_{1,q} + \frac{\beta_\Gamma}{2}\big\|\mathbf{\Gamma}_{t+1} - \mathbf{V} + \frac{1}{\beta_\Gamma}\mathbf{P}_{\Gamma,t}\big\|_F^2$
    3. $\mathbf{P}_{\Gamma,t+1} = \mathbf{P}_{\Gamma,t} + \beta_\Gamma(\mathbf{\Gamma}_{t+1} - \mathbf{V}_{t+1})$
```

Algorithm 3: Alternating Direction Method of Multipliers (ADMM) in kernel space.

3.4.2.1 Update steps for $\mathbf{\Gamma}_t$

$\mathbf{\Gamma}_{t+1}$ is obtained by updating each sub-matrix $\mathbf{\Gamma}_t^i$, $i = 1, \cdots, D$ as:

$$\mathbf{\Gamma}_t^i = (\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i} + \beta_\Gamma\mathbf{I})^{-1}(\mathbf{K}_{\mathbf{X}^i,\mathbf{Y}^i} + \beta_\Gamma\mathbf{V}_t^i - \mathbf{P}_{\Gamma,t}^i), \tag{3.20}$$

where, $\mathbf{I}$ is an identity matrix and $\mathbf{V}_t^i$, $\mathbf{P}_{\Gamma,t}^i$ are sub-matrices of $\mathbf{V}_t$ and $\mathbf{P}_{\Gamma,t}$ respectively.

3.4.2.2 Update steps for $\mathbf{V}_t$

The update equation for $\mathbf{V}_t$ is same as in the linear fusion case using (3.11) and (3.12), replacing $\mathbf{A}_{\Gamma,t}$ and α_Γ with $\mathbf{P}_{\Gamma,t}$ and β_Γ respectively.

3.4.3 Classification

Once $\boldsymbol{\Gamma}$ is obtained, classification can be done by assigning the class label as:

$$\hat{j} = \arg\min_{j} \sum_{i=1}^{D} \|\boldsymbol{\Phi}(\mathbf{Y}^i) - \boldsymbol{\Phi}(\mathbf{X}^i_j)\hat{\boldsymbol{\Gamma}}^i_j\|_F^2,$$

or in terms of kernel matrices as:

$$\hat{j} = \arg\min_{j} \sum_{i=1}^{D} \big(\text{trace}(\mathbf{K}_{\mathbf{YY}}) - 2\text{trace}(\hat{\boldsymbol{\Gamma}}_j^{i^T}\mathbf{K}_{\mathbf{X^i_jY}}\hat{\boldsymbol{\Gamma}}^i_j) \\ + \text{trace}(\hat{\boldsymbol{\Gamma}}_j^{i^T}\mathbf{K}_{\mathbf{X^i_jX^i_j}}\hat{\boldsymbol{\Gamma}}^i_j)\big). \qquad (3.21)$$

Here, $\mathbf{X}^i_j$ is the sub-dictionary associated with j^{th} class and $\hat{\boldsymbol{\Gamma}}^i_j$ is the coefficient matrix associated with this class.

The classification rule can be further extended to include the quality measure as in (3.13). But, we skip this step here, as we wish to study the effect of kernel representation and quality separately.

Multivariate Kernel Sparse Recognition (kerSMBR) algorithm is summarized in Algorithm 4:

Input: Training samples $\{\mathbf{X_i}\}_{i=1}^D$, test sample $\{\mathbf{Y_i}\}_{i=1}^D$

Procedure: Obtain $\hat{\boldsymbol{\Gamma}}$ by solving

$$\hat{\boldsymbol{\Gamma}} = \arg\min_{\boldsymbol{\Gamma}} \tfrac{1}{2}\textstyle\sum_{i=1}^{D} \big(\text{trace}(\boldsymbol{\Gamma}^{i^T}\mathbf{K}_{\mathbf{X}_i,\mathbf{X}_i}\boldsymbol{\Gamma}^i) - 2\text{trace}(\mathbf{K}_{\mathbf{X}_i,\mathbf{Y}_i}\boldsymbol{\Gamma}^i)\big) + \lambda\|\boldsymbol{\Gamma}\|_{1,q}$$

Output:

$\texttt{identity}(\mathbf{Y}) = \arg\min_j \sum_{i=1}^{D} \big(\text{trace}(\mathbf{K}_{\mathbf{YY}}) - 2\text{trace}(\hat{\boldsymbol{\Gamma}}_j^{i^T}\mathbf{K}_{\mathbf{X^i_jY}}\hat{\boldsymbol{\Gamma}}^i_j) + \text{trace}(\hat{\boldsymbol{\Gamma}}_j^{i^T}\mathbf{K}_{\mathbf{X^i_jX^i_j}}\hat{\boldsymbol{\Gamma}}^i_j))$

Algorithm 4: Kernel Sparse Multimodal Biometrics Recognition (kerSMBR).

3.5 Experiments

We evaluated our algorithm on two publicly available datasets - the WVU Multimodal dataset [108] and the AR face dataset [69] In the first experiment, we tested on the WVU dataset, which is one of the few publicly available datasets which allows fusion at image level. It is a challenging dataset consisting of samples from different biometric modalities for each subject.

In the second experiment, we show the applicability of the proposed approach to fusing information from *weak* biometrics extracted from face images. In particular, the periocular region has been shown to be a useful biometric [83]. Similarly, the nose region has also been explored as a biometric [74]. Sinha *et al* [127] have demonstrated that eyebrows are important for face recognition. However, each of these sub-regions may not be as discriminative as the whole face. The challenge for fusion algorithms is to be able to combine these weak modalities with a strong modality based on the whole face [65]. We demonstrate how our framework can be extended to address this problem. Further, we also show the effects of noise and occlusion on the performance of different algorithms. In all the experiments $\mathbf{B}_i$ was set to be identity for convenience, *i.e.*, we assume background noise to be sparse in the image domain.

3.5.1 WVU Multimodal Dataset

The WVU multimodal dataset is a comprehensive collection of different biometric modalities such as fingerprint, iris, palmprint, hand geometry and voice from subjects of different age, gender and ethnicity as described in Table 3.1. It is a challenging dataset as many of these samples are corrupted with blur, occlusion and sensor noise as shown in Figure 3.2. Out of these, we chose iris and fingerprint modalities for testing the proposed algorithms. In total, there are 2 iris (right and left iris) and 4 fingerprint modalities. Also, the evaluation was done on a subset of 219 subjects having samples in both modalities.

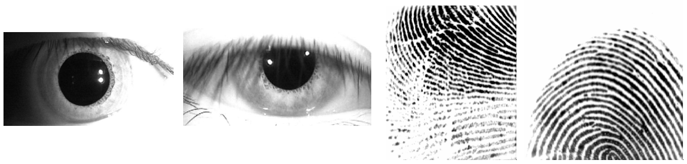


Figure 3.2: Examples of challenging images from the WVU Multimodal dataset. The images shown above suffer from various artifacts such as sensor noise, blur and occlusion.

Biometric Modality	**# of subjects**	**# of samples**
Iris	244	3099
Fingerprint	272	7219
Palm	263	683
Hand Geometry	217	3062
Voice	274	714

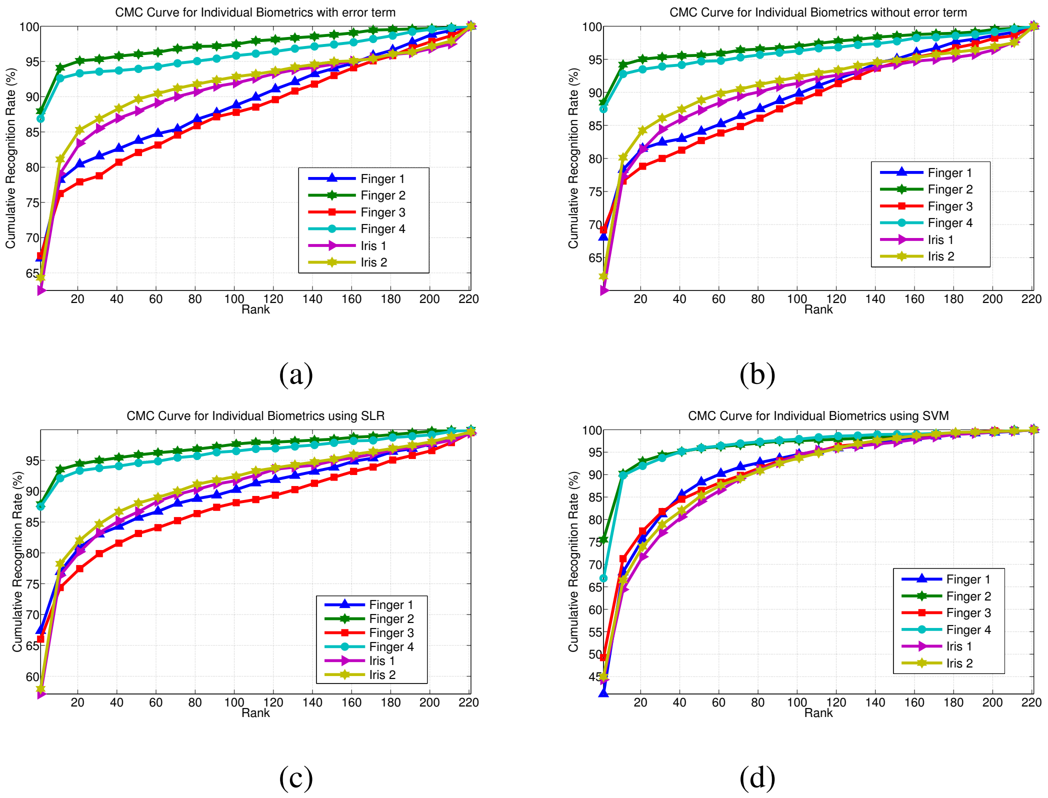
Table 3.1: WVU Biometric Data

3.5.1.1 Preprocessing

Robust pre-processing of images was done before feature extraction. Iris images were segmented using the method proposed in [94]. Following the segmentation step, 25×240 iris templates were generated by re-sampling using the publicly available code of Masek *et al.* [70]. Fingerprint images were enhanced using the filtering methods described in [107], and then the core point was detected from the enhanced images [50]. Features were then extracted around the detected core point.

3.5.1.2 Feature Extraction

Gabor features were extracted from the processed images as they have been shown to give good performance on both fingerprints [50] and iris [25]. For fingerprint samples, the processed images were convolved with Gabor filters at eight different orientations. Circular tessellations were extracted around the core point for all the filtered images similar to [50]. The tessellation consisted of 15 concentric bands, each of width 5 pixels and divided into 30 sectors. The mean values for each sector were concatenated to form the feature vector of size 3600×1. Features for iris images were formed by convolving the templates with a log-Gabor filter at a single scale, and vectorizing the template to give a 6000×1 dimensional feature.



(a) (b) (c) (d)

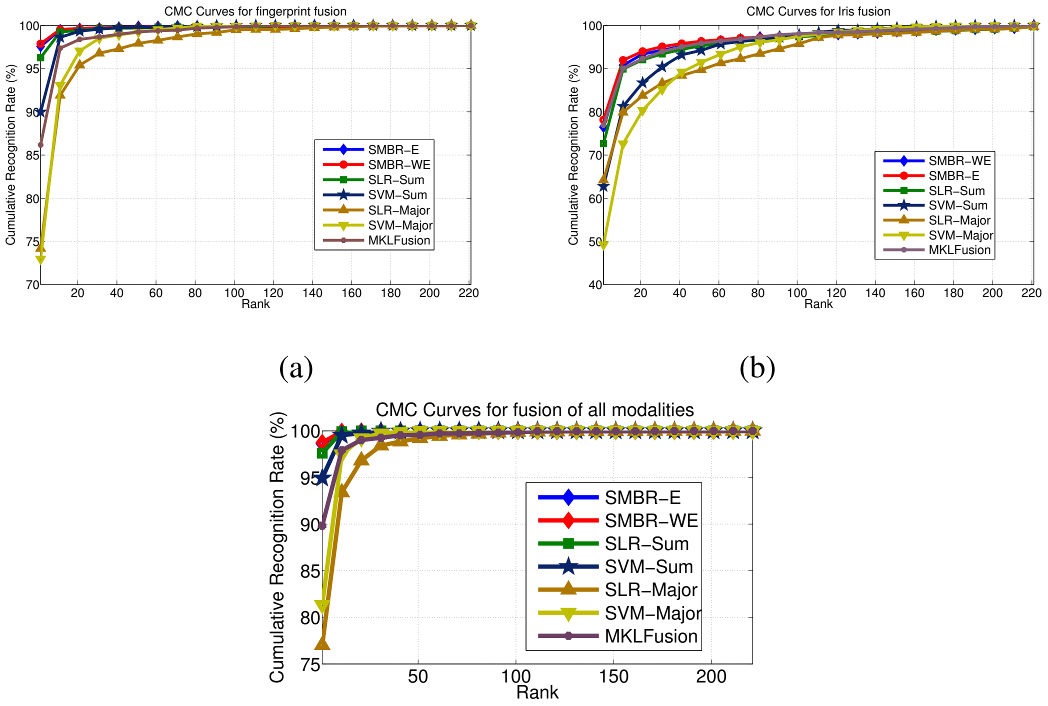
Figure 3.3: CMCs (Cumulative Match Curve) for individual modalities using (a) SMBR-E, (b) SMBR-WE, (c) SLR and (d) SVM methods on WVU Dataset.

	Finger 1	Finger 2	Finger 3	Finger 4	Iris 1	Iris 2
SMBR-WE	$\mathbf{68.1 \pm 1.1}$	$\mathbf{88.4 \pm 1.2}$	$\mathbf{69.2 \pm 1.5}$	$\mathbf{87.5 \pm 1.5}$	60.0 ± 1.5	62.1 ± 0.4
SMBR-E	67.1 ± 1.0	87.9 ± 0.8	67.4 ± 1.9	86.9 ± 1.5	$\mathbf{62.5 \pm 1.2}$	$\mathbf{64.3 \pm 1.0}$
SLR	67.4 ± 1.9	87.9 ± 1.3	66.0 ± 2.2	87.5 ± 1.3	57.1 ± 3.0	57.9 ± 2.7
SVM	41.1 ± 5.0	75.5 ± 2.2	49.2 ± 1.6	67.0 ± 8.3	44.3 ± 1.2	45.0 ± 2.9

Table 3.2: Rank one recognition performance on WVU dataset for individual modalities.

	SMBR-WE	SMBR-E	SLR-Sum	SLR-Major	SVM-Sum	SVM-Major	MKLFusion
4 Fingerprints	$\mathbf{97.9 \pm 0.4}$	97.6 ± 0.6	96.3 ± 0.8	74.2 ± 0.7	90.0 ± 2.2	73.0 ± 1.5	86.2 ± 1.2
2 Irises	76.5 ± 1.6	$\mathbf{78.2 \pm 1.2}$	72.7 ± 4.0	64.2 ± 2.7	62.8 ± 2.6	49.3 ± 2.0	76.8 ± 2.5
All modalities	$\mathbf{98.7 \pm 0.2}$	98.6 ± 0.5	97.6 ± 0.4	84.4 ± 0.9	94.9 ± 1.5	81.3 ± 1.7	89.8 ± 0.9

Table 3.3: Rank one recognition performance on WVU dataset for different fusion settings.



(a)

(b)

(c)

Figure 3.4: CMCs (Cumulative Match Curve) for multimodal fusion using (a) four fingerprints, (b) two irises and (c) all modalities on WVU dataset.

	SMBR-WE	SMBR-E	SLR-Sum	SLR-Major	SVM-Sum	SVM-Major
4 Fingerprints	$\mathbf{98.2 \pm 0.5}$	98.1 ± 0.5	97.5 ± 0.5	86.3 ± 0.6	93.6 ± 1.6	85.5 ± 0.9
2 Irises	76.9 ± 1.2	$\mathbf{78.8 \pm 1.7}$	74.1 ± 1.0	67.2 ± 2.4	64.3 ± 3.3	51.6 ± 2.0
All modalities	$\mathbf{98.8 \pm 0.4}$	98.6 ± 0.3	98.2 ± 0.2	93.8 ± 0.9	95.5 ± 1.5	93.3 ± 1.2

Table 3.4: Rank one recognition performance on WVU dataset using the proposed quality measure.

3.5.1.3 Experimental Set-up

The dataset was randomly divided into 4 training samples per class (1 sample here is 1 data sample each from 6 modalities) and the remaining 519 samples were used for testing. The recognition result was averaged over 5 runs. The proposed methods were compared with state-of-the-art classification methods such as sparse logistic regression (SLR) [58] and SVM [17]. As these methods cannot handle multiple modalities, we explored score-level and decision-level fusion methods for combining the results of individual modalities. For score-level fusion, the probability outputs for test sample of each modality, $\{\mathbf{y}_i\}_{i=1}^{6}$ were added together to give the final score vector. Classification was based upon the final score values. For decision-level fusion, the subject chosen by the maximum number of modalities was taken to be from the correct class. We further compared with an efficient multiclass implementation of MKL algorithm [96]. The proposed linear and kernel fusion techniques were tested separately and were compared with linear and kernel versions of SLR, SVM and MKL algorithms. We denote the score-level fusion of these methods as SLR-Sum and SVM-Sum, and the decision-level fusion as SLR-Major and SVM-Major. MKL based method is denoted as MKLFusion. We report the mean and standard deviation of rank one recognition rates for all the methods. We also show the Cumulative Match Curves (CMCs) for all the classifiers. The CMCs provide the performance measure for biometric recognition systems and has been shown to be equivalent to the ROC of the system [15].

Linear Fusion The recognition performances of SMBR-WE and SMBR-E was compared with linear SVM and linear SLR classification methods. The parameters λ_1 and λ_2 were set to 0.01 experimentally.

- *Comparsion of Methods:* Figure 3.3 and Table 3.2 show the performance on individual modalities. All the classifiers show a similar trend. The performance for all of them are lower on iris images and fingers 1 and 3. The proposed method show superior performance on all the modalities. Figure 3.4 and Table 3.3 show the recognition performance for different fusion settings. The proposed SMBR approach outperforms existing classification techniques. Further, the CMC curves of the proposed approaches lie above the other methods for all the fusion settings. Both SMBR-E and SMBR-WE have similar performance, though the latter seems to give a slightly better performance. This may be due to the penalty on the sparse error, though the error may not be sparse in the image domain. Further, sum-based fusion shows a superior performance over voting-based methods. MKL-based method shows good performance for iris fusion, but the performance drops for other two settings. This may be because by weighing kernels during training, it loses flexibility while testing when number of modalities increase.

- *Fusion with quality:* Clearly, different modalities have different levels of performance. Hence, we studied the effect of the proposed quality measure on the performance of different methods. For a consistent comparison, the quality values produced by SMBR-E method was used for all the algorithms. Table 3.4 shows the performance for the three fusion settings. The effect of including the quality measure can be studied by comparing with Table 3.3. Clearly, the recognition rate increases for all the methods across the fusion settings. Again SMBR-E and SMBR-WE give the best performances among all the methods.

- *Effect of joint sparsity:* We also studied the effect of joint sparsity constraint on the recognition performance. For this, SMBR-WE algorithm was run for different values of λ_1. Figure 3.10 shows the rank one recognition variation across λ_1 values for different fusion settings. All the curves show a sharp increase in performance around $\lambda_1 = 0$. Further, the increase is more for iris fusion, which shows around 5% improvement at $\lambda_1 = 0.005$ over $\lambda_1 = 0$. This shows that imposing joint sparsity constraint is important for fusion. Moreover, it helps in regulating fusion performance, when the reconstruction error alone is not sufficient to distinguish between different classes. The performance is then stable across λ_1 values, and starts decreasing slowly after reaching the optimum performance.

- *Variation with number of training samples:* We varied the number of training sam-

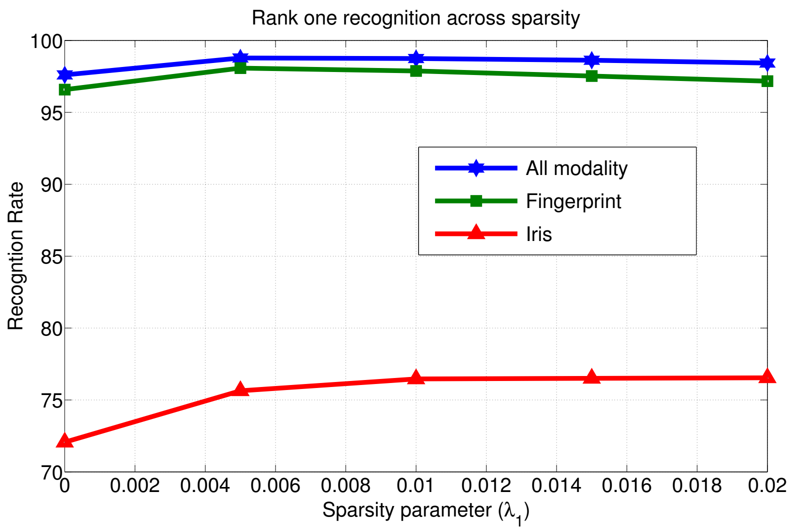


Figure 3.5: Variation of recognition performance with different values of sparsity constraint, λ_1.

ples and studied the effect on the proposed method along with SLR-Sum and MKL-Fusion. Figure 3.6 shows the variation for fusion of all the modalities. It can be seen that SMBR-WE and SMBR-E are stable across number of training samples, whereas the performance of SLR-Sum and MKLFusion based methods fall sharply. The fall in performance of SLR-Sum and MKLFusion can be attributed to the discriminative approaches of these methods, as well as score-based fusion, as the fusion further reduces the recognition performance when individual classifiers are not good.

- *Comparison with other score-based fusion methods:* Although sum-based fusion is a popular technique for score fusion, some other techniques have also been proposed. We evaluated the performance of likelihood-based fusion method proposed in [77]. The results are shown in Table 3.5. The method does not show good performance as it models score distribution as Gaussian Mixture Model. However, it is difficult to model score distribution due to large variations in data samples. The method is also affected by the curse of dimensionality.

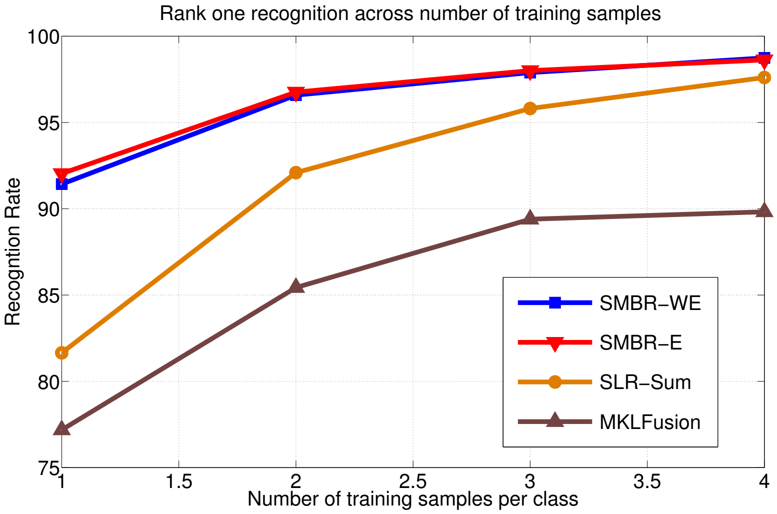


Figure 3.6: Variation of recognition performance with number of training samples.

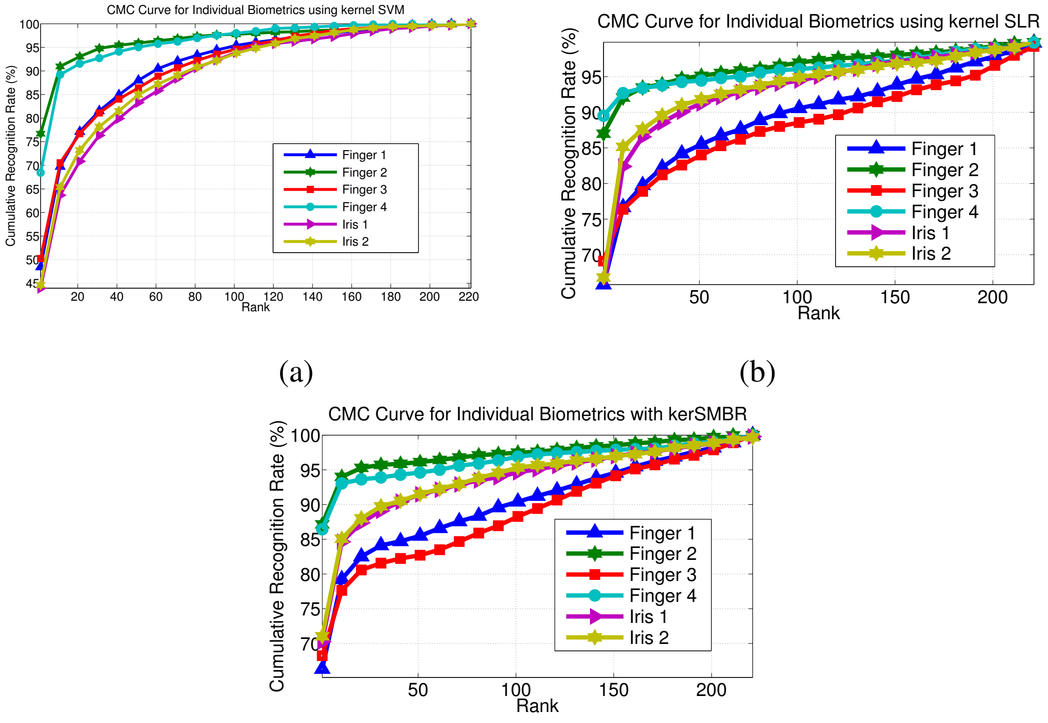
	2 irises	4 fingerprints	All modalities
SLR-Likelihood	66.6 ± 2.9	83.5 ± 2.5	75.1 ± 3.2
SVM-Likelihood	50.7 ± 2.4	31.9 ± 1.7	31.0 ± 3.4

Table 3.5: Rank one recognition performance with likelihood-based method [77] on WVU Dataset.

Kernel Fusion We further compared the performances of proposed kerSMBR with kernel SVM, kernel SLR and MKLFusion methods. In the experiments, we used Radial Basis Function (RBF) as the kernel, given as:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}\right),$$

σ being a parameter to control the width of the RBF. For MKLFusion, we gave linear, polynomial and RBF kernels as the base kernels for learning.



(a) (b)

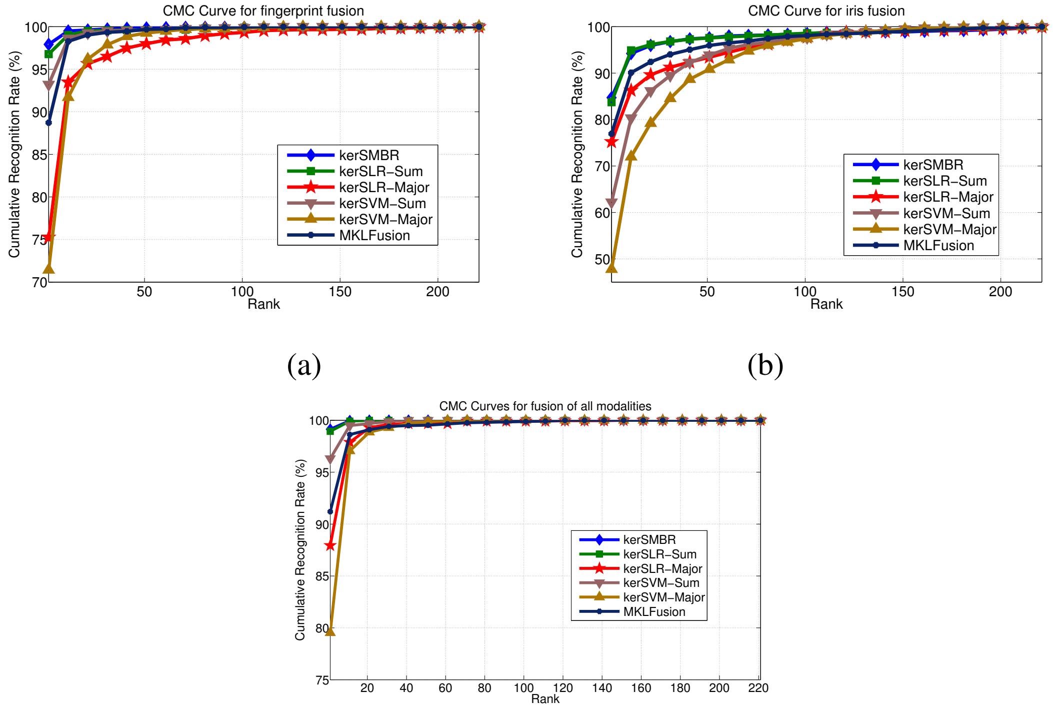
(c)

Figure 3.7: CMCs (Cumulative Match Curve) for individual modalities using (a) kernel SVM, (b) kernel SLR and (c) kerSMBR.

- *Hyperparameter tuning:* To fix the value of hyperparameter, σ, we iterated over different values of σ, $\{2^{-3}, 2^{-2}, \cdots, 2^{3}\}$ for one set of training and test split of the data. The value of σ giving the maximum performance was fixed for each modality, and the performance was averaged over a few iterations. λ and β_Γ were set to 0.01 and 0.01 respectively.

	Finger 1	Finger 2	Finger 3	Finger 4	Iris 1	Iris 2
kerSMBR	**66.3 ± 1.7**	**87.1 ± 1.0**	**69.1 ± 2.1**	86.4 ± 1.5	**70.3 ± 1.8**	**71.0 ± 1.6**
kerSLR	65.8 ± 1.8	86.9 ± 1.7	68.3 ± 2.0	**89.5 ± 1.6**	65.1 ± 1.7	66.8 ± 1.1
kerSVM	48.4 ± 5.4	76.7 ± 2.3	50.2 ± 1.9	68.4 ± 7.4	43.9 ± 1.1	44.6 ± 3.0

Table 3.6: Rank one recognition performance on WVU dataset for individual modalities using kernel methods.



(a)

(b)

(c)

Figure 3.8: CMCs (Cumulative Match Curve) for different fusion methods for (a) four fingerprints, (b) two irises and (c) all modalities.

	kerSMBR	kerSLR-Sum	kerSLR-Major	kerSVM-Sum	kerSVM-Major	MKLFusion
4 Fingerprints	**97.9 ± 0.3**	96.8 ± 0.7	75.2 ± 0.7	93.2 ± 1.2	71.4 ± 1.3	88.7 ± 0.9
2 Irises	**84.7 ± 1.7**	83.7 ± 1.8	75.2 ± 1.2	62.2 ± 2.8	47.8 ± 2.4	76.9 ± 2.4
All modalities	**99.1 ± 0.2**	98.9 ± 0.1	87.9 ± 0.6	96.3 ± 0.8	79.5 ± 1.6	91.2 ± 1.0

Table 3.7: Rank one recognition performance on WVU dataset for different fusion settings using kernel methods.

- *Comparison of methods:* Figure 3.7 and Table 3.6 show the performance of different methods on individual modalities, and Figure 3.8 and Table 3.7 on different fusion settings. Comparison of performance with linear fusion shows that the proposed kerSMBR significantly improves the performance on individual iris modalities as well as iris fusion. The performance on fingerprint modalities is similar, however the fusion of all 6 modalities (2 iris + 4 fingerprints) shows an improvement of 0.4%. kerSMBR also achieves the best accuracy among all the methods for different fusion settings. kerSLR scores better than kerSVM in all the cases, and it's accuracy is close to kerSMBR. The performance of kerSLR is better than the linear counterpart, however kerSVM does not show much improvement.

3.5.2 AR Face Dataset

The AR face dataset consists of faces with varying illumination, expression and occlusion conditions, captured in two sessions. We evaluated our algorithms on a set of 100 users. Images from the first session, 7 for each subject were used as training and the images from the second session, again 7 per subject, were used for testing. For testing the fusion algorithms, four weak modalities were extracted from the face images: left and right periocular, mouth and nose regions. This was done by applying rectangular masks as shown in Figure 3.9, and cropping out the respective regions. These, along with the whole face, were taken for fusion. Simple intensity values were used as features for all of them. The experimental set-up was similar to the previous section. The parameter values, λ_1 and λ_2 were set to 0.003 and 0.002 respectively. Furthermore, we also studied the effect of noise and occlusion on recognition performance.

- *Comparison of methods:* Table 3.8 shows the performance of different algorithms on the face dataset. Here, SR (sparse representation) shows the classification result using just the whole face. Block Sparse Method is a recent block sparsity based face recognition algorithm [32] and FDDL [140] is a state-of-the-art discriminative dictionaries based technique, but using only a single modality. Clearly, the

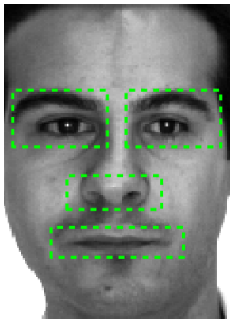


Figure 3.9: Face mask used to crop out different modalities.

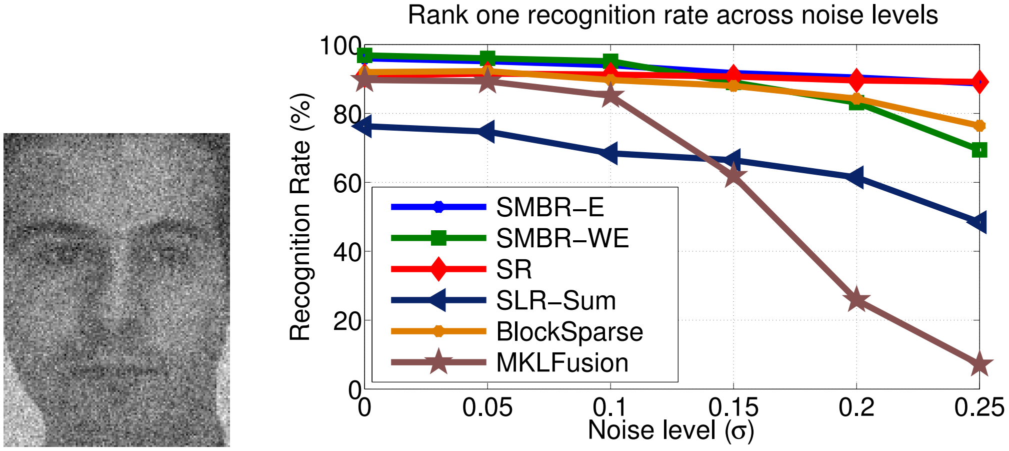


Figure 3.10: Effect of noise on rank one recognition performance for AR face dataset.

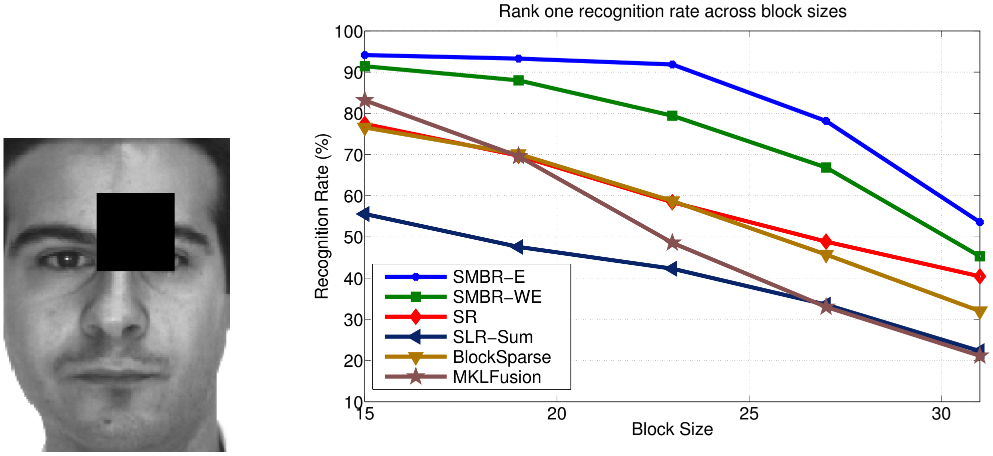


Figure 3.11: Effect of occlusion on rank one recognition performance for AR face dataset.

Method	Recognition Rate (%)	Method	Recognition Rate (%)
SMBR-WE	**96.9**	SVM-Sum	86.7
SMBR-E	96	SLR-Sum	77.9
SR	91	FDDL [140]	91.9
Block Sparse [32]	92.2	MKLFusion	89.7

Table 3.8: Rank one performance comparison of different methods on AR face dataset.

SMBR approach achieves about 4 % improvement over other techniques. Thus, robust classification using multiple modalities results in a significant improvement over the current benchmark. Further, a comparison with discriminative methods such as SLR and SVM shows that they perform poorly compared to the proposed method. This is because weak modalities are hard to discriminate, hence score-level fusion with strong modality does not improve performance. On the other hand, by appropriately weighing different modalities, MKLFusion achieves better result. However, by imposing reconstruction and joint sparsity simultaneously, the proposed method is able to achieve the best performance.

- *Effect of noise:* In this experiment, test images were corrupted with white Gaussian noise of increasing variance, σ^2. Comparisons are shown in Figure 3.10. It can be seen that both SMBR, SR and Block Sparse methods are stable with noise. The performance of other algorithms degrade sharply with noise level. This also highlights the problem with MKLFusion, as it is not robust to degradation during testing.

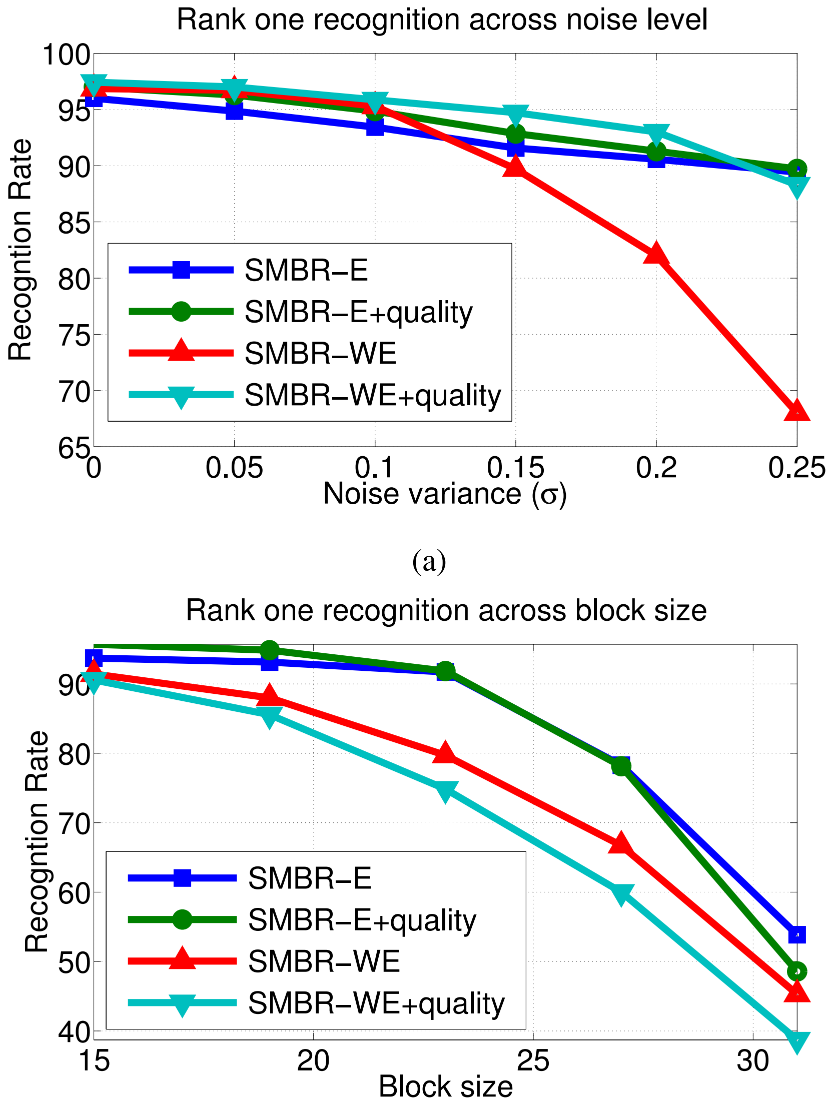
- *Effect of occlusion:* In this experiment, a randomly chosen block of the test image was occluded. The recognition performance was studied with increasing block size. Figure 3.11 shows the performance of various algorithms with block size. SMBR-E is the most stable among all the methods due to robust handling of error. Recognition rates for other methods fall sharply with increasing block size.

- *Recognition in spite of disguise:* We also performed experiment on the rest of the AR face dataset, occluded by sun-glass and scarves. Similar to the above experiment, 7 frontal non-occluded images per subject, from the first session, were used for training, and 12 occluded images per person, from both the sessions were used for testing. Again the proposed SMBR-WE and SMBR-E methods outperformed the other methods. SMBR-E method gave the best performance, improving by 17.7% over the Block Sparse method.

Method	Scarves	Sun-glass	Overall
SMBR-WE	**86.2**	36.0	61.1
SMBR-E	80.0	**75.0**	**77.5**
SR	45.3	52.3	48.8
Block Sparse [32]	65.8	53.8	59.8
SLR-Sum	72.2	39.6	55.9
SVM-Sum	13.8	42.5	28.1
MKLFusion	47.7	13.0	30.3

Table 3.9: Rank one performance comparison of different methods on images with disguise in AR face dataset.

- *Quality based fusion:* Quality determination is an important parameter in fusion here, as a strong modality is being combined with weak modalities. We studied the effect of quality measure introduced in Section 3.3. However, in this case we fix the quality for strong modality, *viz.* whole face to be 1, while for the weak modalities, the SCI values were taken. The recognition performance for SMBR-E and SMBR-WE across different noise and occlusion levels was studied. Figure 3.12 show the performance comparison with the unweighted methods. Using quality, the recognition performance for SMBR-WE goes up to 97.4 % from 96.9 %, whereas for SMBR-WE it increases to 97 % from 96 %. Similarly, results improve across different noise levels for both methods. However, SMBR-WE with quality shows worse performance as block size is increased. This may be because it does not handle sparse error, hence the quality values are not robust.



(a)

(b)

Figure 3.12: Effect of quality on recognition performance across (a) noise (b) random blocks on AR face dataset.

3.6 Computational Complexity

The proposed algorithms are computationally efficient. The main steps of the algorithms are the update steps for $\mathbf{\Gamma}$, $\mathbf{\Lambda}$, $\mathbf{U}$ and $\mathbf{V}$. For linear fusion, the update step for $\mathbf{\Gamma}$ involves computing $(\mathbf{X}^{i^T}\mathbf{X}^i + \alpha_\Gamma \mathbf{I})^{-1}$ and four matrix multiplications. The first term is constant across iterations and can be pre-computed. Matrix multiplication for two matrices of sizes $m \times n$ and $n \times p$ can be done in $\mathcal{O}(mnp)$ time. Hence, for a given training and test data, the computations are linear in feature dimension. Hence, large feature dimensions can be efficiently handled. Similarly, update step for $\mathbf{\Lambda}$ involves matrix multiplication $\mathbf{X}^i\mathbf{\Gamma}^i$. Update steps for $\mathbf{U}$ and $\mathbf{V}$ involves only scalar matrix computations and are very fast. Similarly in the kernel fusion, update for $\mathbf{\Gamma}$ involves calculating $(\mathbf{K}_{\mathbf{X}^i,\mathbf{X}^i} + \beta_\Gamma \mathbf{I})^{-1}$, which can be pre-computed. Other steps are similar to linear fusion. Classification step involves calculating the residual error for each class, and is efficient.

3.7 Conclusion

We proposed a novel joint sparsity-based feature level fusion algorithm for multimodal biometrics recognition. The algorithm is robust as it explicitly includes both noise and occlusion terms. An efficient algorithm based on alternative direction was proposed for solving the optimization problem. We also proposed a multimodal quality measure based on sparse representation. Further, the algorithm was kernelized to handle non-linear variations. Various experiments have shown that the method is robust and significantly improves the overall recognition accuracy.

Chapter 4: Coupled Projections for Adaptation of Dictionaries

4.1 Introduction

The study of sparse representation of signals and images has attracted tremendous interest in the last few years. Sparse representations of signals and images require learning an over-complete set of bases called a dictionary along with linear decomposition of signals and images as a combination of few atoms from the learned dictionary. Olshausen and Field [82] in their seminal work introduced the idea of learning dictionary from data instead of using off-the-shelf bases. Since then, data-driven dictionaries have been shown to work well for both image restoration [31] and classification tasks [135].

The efficiency of dictionaries in these wide range of applications can be attributed to the robust discriminant representations that they provide by adapting to the particular data samples. However, the learned dictionary may not be optimal if the target data has different distribution than the data used for training. These variations are commonplace in vision problems, and can happen due to changes in image sensor (web-cams vs SLRs), camera viewpoint, illumination conditions, etc. It has been shown that such changes can cause significant degradation in classifier performance [26]. Adapting dictionaries to new domains is a challenging task, and has only recently been explored in the vision literature. Yangqing *et al.* [53] considered a special case where corresponding samples from each domain were available, and learned a dictionary for each domain. More recently, Qiu *et al.* [95] proposed a method for adapting dictionaries for smoothly varying domains using regression. However, in practical applications, target domains are scarcely labeled, and domain shifts may result in abrupt feature changes (e.g., changes in resolution when comparing web-cams to DSLRs). Moreover, high dimensional features are often extracted for

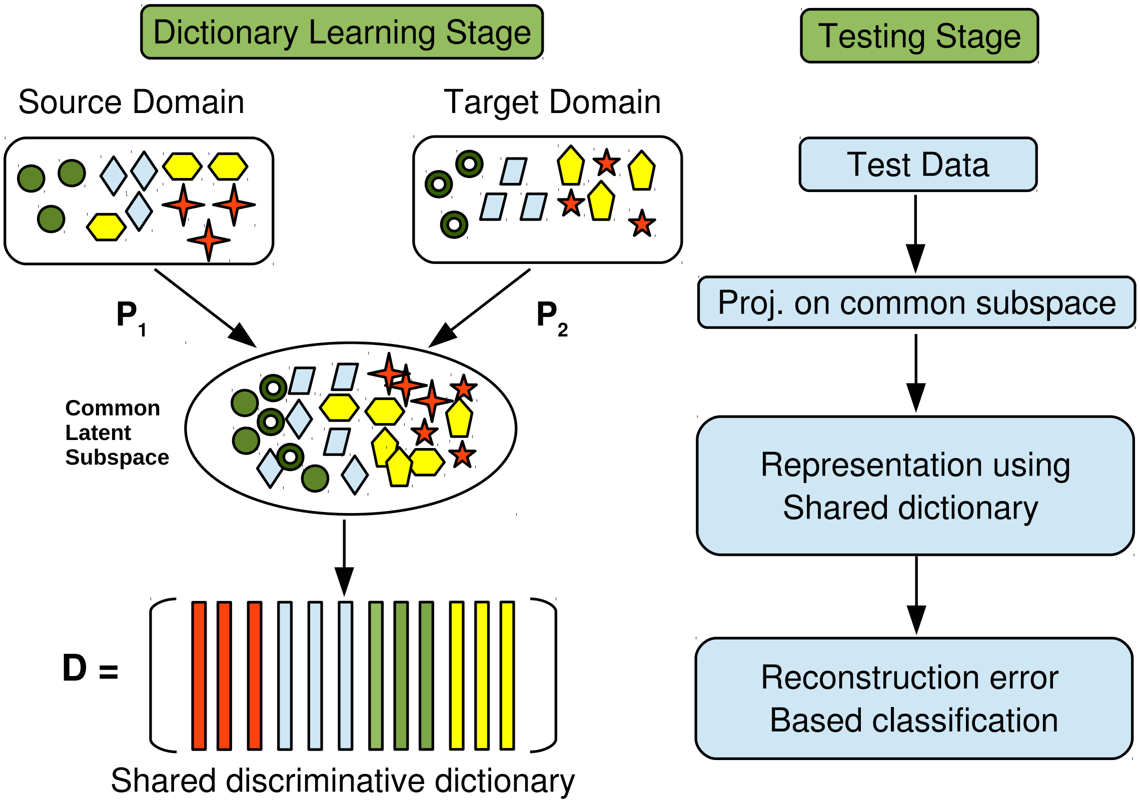


Figure 4.1: Overview of the proposed dictionary learning method.

object recognition. Hence learning a separate dictionary for each domain will have a severe space constraint, rendering it unfeasible for many practical applications. A subspace interpolation based method was proposed for adapting dictionaries in [81]. However, this method cannot be used for heterogeneous domain adaptation, where different features are extracted for different domains.

In view of the above challenges, we propose a robust method for learning a single dictionary to optimally represent both source and target data. As the features may not be correlated well in the original space, we project data from both the domains onto a common low-dimensional space, while maintaining the manifold structure of data. Simultaneously, we learn a compact dictionary which represents projected data from both the domains well. As the final objective is classification, we learn a class-wise discriminative dictionary. This joint optimization method offers several advantages in terms of generalizability and efficiency of the method. Firstly, learning separate projection matrix for each domain makes it easy to handle any changes in feature dimension and type in different domains. It also makes the algorithm conveniently extensible to handle multiple domains. Further, learning the dictionary in a low-dimensional space makes the algorithm faster, and irrelevant information in original features is discarded. Moreover, joint learning of dictionary and projections ensures that the common internal structure of data in both the domains is preserved, which can be represented well by sparse linear combinations of dictionary atoms.

An additional contribution of the work is an efficient optimization technique to solve this problem. Using kernel methods, the proposed algorithm can be easily made non-linear, and the resulting optimization problem has a few simple update steps. Further we extensively evaluate the method for different recognition scenarios and show that the proposed method is comparable with other recent algorithms for domain adaptation. We also demonstrate that the algorithm converges quickly and is efficient.

4.1.1 Chapter Organization

The chapter is organized in six sections. In Section 4.2, we describe some of the related works. The algorithm is formulated in Section 4.3, and the extension to non-linear case is described in Section 4.4. The classification scheme for the learned dictionary is described in Section 4.5. Experimental results are presented in Section 4.6, and the final concluding remarks are made in 4.7.

4.2 Related Work

In this section, we survey the recent domain adaptation works and the related sparse coding literature.

4.2.1 Domain Adaptation

The problem of adapting classifiers to new visual domains has recently gained importance in the vision community. Several approaches have been proposed for this problem, which can be broadly categorized into following categories:

4.2.1.1 Feature transform-based approaches

The idea of domain adaptation in vision community was introduced by Saenko *et al.* [109], in which a symmetric transformation between domains represented by the same features was learned. This was extended to general domain shifts in Kulis *et al.* [60] by learning an asymmetric transformation between domains. In [51], a transformation of source data onto target space was learnt, such that the joint representation is low-rank. Further, Baktashmotlagh *et al.* [7] proposed learning feature transformation for kernel mean matching between domains for adaptation. A subspace alignment-based method was also explored in [34].

4.2.1.2 Manifold interpolation-based approaches

Gopalan *et al* introduced the idea of interpolation between subspaces of different domains on the Grassmann manifold [41]. This was extended to learning a kernel distance between domains in [40]. A class-wise adaptation scheme based on parallel transport on manifold was introduced in [123].

4.2.1.3 Classifier transform-based approaches

Many methods have been proposed to adapt classifiers between domains for adaptation. A method for adapting SVMs across domains was proposed for concept detection in [138]. Similar methods based on transforming SVMs have been proposed in [29,30]. A multiple kernel learning-based approach for domain adaptation was proposed in [28]. Recently, a method for adaptation by reconstructing target classifiers using source classifiers was explored in [147].

4.2.1.4 Other approaches

A feature augmentation method was proposed in [66]. Gong *et al.* [39] described a method of choosing landmarks in the target domain for adaptation. An information theortic clustering-based adaptation approach was proposed in [122]. Recently, deep learning has also been used for domain adaptation [21, 24].

4.2.2 Sparse Coding

Here, we review some of the related works in sparse coding literature. Han *et al.* [46] suggested learning a shared embedding for different domains, along with a sparsity constraint on the representation. However, they assume pre-learned projections, which may not be optimal. In the dictionary learning literature, Yang *et al.* [136] and Wang *et al.* [132] proposed learning dictionary pairs for cross-modal synthesis. Similarly, methods for joint dimensionality reduction and sparse representation have also been proposed [37, 67, 78, 144]. Additional methods may be found within these references.

4.3 Problem Framework

The classical dictionary learning approach minimizes the representation error of the given set of data samples subject to a sparsity constraint [2]. Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{d \times N}$ be the data matrix. Then, the K-atoms dictionary, $\mathbf{D} \in \mathbb{R}^{d \times K}$, can be trained by solving the following optimization problem

$$\{\mathbf{D}^*, \mathbf{X}^*\} = \arg\min_{\mathbf{D},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ s.t. } \|\mathbf{x}_i\|_0 \leq T_0 \ \forall i,$$

where, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{K \times N}$ is the sparse representation of $\mathbf{Y}$ over $\mathbf{D}$, and T_0 is the sparsity level. Here, $\|.\|_0$-norm counts the number of nonzero elements in a vector and $\|.\|_F$ is the Frobenius norm of a matrix.

Now, consider a special case, where we have data from two domains, $\mathbf{Y_1} \in \mathbb{R}^{d_1 \times N_1}$ and $\mathbf{Y_2} \in \mathbb{R}^{d_2 \times N_2}$. We wish to learn a shared K-atoms dictionary, $\mathbf{D} \in \mathbb{R}^{d_f \times K}$ and mappings $\mathbf{P}_1 \in \mathbb{R}^{d_f \times d_1}$, $\mathbf{P}_2 \in \mathbb{R}^{d_f \times d_2}$ onto a common low-dimensional space, which will minimize the representation error in the projected space. Formally, we wish to minimize the following cost function:

$$\begin{aligned} \mathcal{C}_1(\mathbf{D}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{X_1}, \mathbf{X_2}) = \|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\mathbf{X_1}\|_F^2 + \\ \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\mathbf{X_2}\|_F^2, \end{aligned} \tag{4.1}$$

subject to sparsity constraints on $\mathbf{X_1}$ and $\mathbf{X_2}$. However, minimizing $\mathcal{C}_1(\mathbf{D}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{X_1}, \mathbf{X_2})$ will result in trivial solution as $\mathbf{P_i}$s can be set to $\mathbf{0}$. To overcome this, we regularize the solution space to get meaningful solutions.

4.3.1 Regularization

It will be desirable if the projections, while bringing the data from two domains to a shared subspace, do not lose too much information available in the original domains.

To facilitate this, we add a PCA-like regularization term which preserves energy in the original signal, given as:

$$\mathcal{C}_2(\mathbf{P_1},\mathbf{P_2}) = \|\mathbf{Y_1} - \mathbf{P_1^T P_1 Y_1}\|_F^2 + \|\mathbf{Y_2} - \mathbf{P_2^T P_2 Y_2}\|_F^2$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1,2. \tag{4.2}$$

It is easy to show after some algebraic manipulations that the costs $\mathcal{C}_1$ and $\mathcal{C}_2$, after ignoring the constant terms in $\mathbf{Y}$, can be written as:

$$\mathcal{C}_1(\mathbf{D},\tilde{\mathbf{P}},\tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2, \tag{4.3}$$

$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T), \tag{4.4}$$

where,

$$\tilde{\mathbf{P}} = [\mathbf{P_1}\ \mathbf{P_2}],\ \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_2} \end{pmatrix}, \text{and } \tilde{\mathbf{X}} = [\mathbf{X_1}\ \mathbf{X_2}].$$

Thus, the form of $\mathcal{C}_2$ is similar to the trace minimization problem [57]. Thus, the regularization approach can be generalized to different dimensionality reduction techniques. We describe some of the possible methods below:

1. **Manifold preserving regularization:** Let $\mathbf{W_1} \in \mathbb{R}^{N_1 \times N_1}$ and $\mathbf{W_2} \in \mathbb{R}^{N_2 \times N_2}$ be affinity matrices calculated from $\mathbf{Y_1}$ and $\mathbf{Y_2}$ using different methods in literature [8, 110]. The manifold preserving mapping can then be formulated as:

$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\sum_{i=1}^{2} \text{trace}(\mathbf{P_i Y_i})(\mathbf{I} - \mathbf{W_i})(\mathbf{I} - \mathbf{W_i^T})(\mathbf{P_i Y_i})^T$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1,2.$$

   Other possible manifold-based regularization approaches can also be explored [57].

2. **Discriminative regularization:** Let $\mathbf{H_{i,j}} = \mathbf{1}_{n_{i,j}}\mathbf{1}_{n_{i,j}}^T\ i = 1,2, j = 1,\cdots,C$ where, C is the number of classes in data and $n_{i,j}$ is the number of samples in class j for domain i and $\mathbf{1}_{n_{i,j}}$ is a column vector of length $n_{i,j}$. Define

$$\mathbf{H_i} = \text{diag}[\mathbf{H_{i,1}},\cdots,\mathbf{H_{i,C}}].$$

Then, discriminative LDA-like regularization can be formulated as in [57]:

$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\sum_{i=1}^{2} \text{trace}(\mathbf{P_i Y_i})(\mathbf{I} - \mathbf{H_i})(\mathbf{P_i Y_i})^T$$
$$\text{s.t. } (\mathbf{P_i Y_i})(\mathbf{P_i Y_i})^{\mathbf{T}} = \mathbf{I},\ i = 1, 2.$$

In this work, we focus on the PCA-like regularization (4.4), leaving the other approaches discussed above for future work. Hence, the overall objective function is given as:

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \underset{\mathbf{D},\tilde{\mathbf{P}},\tilde{\mathbf{X}}}{\arg\min}\ \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, 2 \text{ and } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j, \tag{4.5}$$

where, λ is a positive constant.

4.3.2 Multiple domains

The above formulation can be extended so that it can handle multiple domains. For the M domain problem, we simply construct matrices $\tilde{\mathbf{Y}}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}$ as:

$$\tilde{\mathbf{P}} = [\mathbf{P_1}, \cdots, \mathbf{P_M}]\,, \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y_M} \end{pmatrix},$$

and

$$\tilde{\mathbf{X}} = [\mathbf{X_1}, \cdots, \mathbf{X_M}].$$

With these definitions, (4.5) can be generalized to multiple domains as follows

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \underset{\mathbf{D},\tilde{\mathbf{P}},\tilde{\mathbf{X}}}{\arg\min}\ \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j. \tag{4.6}$$

4.3.3 Special case of $\mathbf{P_1} = \mathbf{P_2} = \cdots = \mathbf{P_M}$

For the special case of domain adaptation, where same features are extracted for all the domains such that $d_1 = d_2 = \cdots = d_M$, and the domain shift is not large (e.g. matching frontal faces to profile faces), the same projection matrix can be used for all the domains.

4.3.4 Discriminative Dictionary

The dictionary learned in (4.5) can reconstruct the two domains well, but it cannot discriminate between the data from different classes. Recent advances in learning discriminative dictionaries [97, 140] suggest that learning class-wise, mutually incoherent dictionaries works better for discrimination. To incorporate this into our approach, we write the dictionary $\mathbf{D}$ as $\mathbf{D} = [\mathbf{D_1}, \cdots, \mathbf{D_C}]$, where C is the total number of classes. We modify the cost function similar to [140], which encourages reconstruction samples of a given class by the dictionary of the corresponding class, and penalizes reconstruction by out-of-class dictionaries. The new cost function, $\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}})$ is given as:

$$\begin{aligned}\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \\ \nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2,\end{aligned} \tag{4.7}$$

where μ and ν are the weights given to the discriminative terms, and matrices $\tilde{\mathbf{X}}_{\mathbf{in}}$ and $\tilde{\mathbf{X}}_{\mathbf{out}}$ are given as:

$$\tilde{\mathbf{X}}_{\mathbf{in}}[i,j] = \begin{cases} \tilde{\mathbf{X}}[i,j], & \mathbf{D_i}, \tilde{\mathbf{Y}}_{\mathbf{j}} \in \text{ same class} \\ 0, & \text{otherwise,} \end{cases}$$

$$\tilde{\mathbf{X}}_{\mathbf{out}}[i,j] = \begin{cases} \tilde{\mathbf{X}}[i,j], & \mathbf{D_i}, \tilde{\mathbf{Y}}_{\mathbf{j}} \in \text{ different class} \\ 0, & \text{otherwise.} \end{cases}$$

The cost function is defined only for labeled data in both domains. Unlabeled data can be handled using semi-supervised approaches to dictionary learning [92]. However, we do not explore it further here. Also, note that we do not need to modify the forms of projection matrices, since they capture the overall domain shift, and hence are independent of class variations.

4.3.5 Optimization

The optimization problem (4.6) is non-convex in the variables $\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}$. Hence, we optimize the cost using alternate minimization strategy, where first $\tilde{\mathbf{P}}$ is updated, keeping $\mathbf{D}$, $\tilde{\mathbf{X}}$ fixed followed by updating $\mathbf{D}$ and $\tilde{\mathbf{X}}$, keeping $\tilde{\mathbf{P}}$ fixed.

- **Updating $\tilde{\mathbf{P}}$:** For fixed $\mathbf{D}, \tilde{\mathbf{X}}$, the optimization can be written as:

$$\begin{aligned}\tilde{\mathbf{P}}^* = \arg\min_{\tilde{\mathbf{P}}}\ &\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 \\ &-\lambda \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T) \text{ s.t. } \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I},\ i = 1, \cdots, M. \end{aligned} \tag{4.8}$$

  However, this is not a convex problem because of the orthonormality constraints on $\mathbf{P_i}$. Specifically, it involves optimization on the Stiefel manifold, hence, we solve it using the manifold optimization technique described in [133].

- **Updating $\mathbf{D}, \tilde{\mathbf{X}}$:** For fixed $\tilde{\mathbf{P}}$ the optimization problem can be written as:

$$\begin{aligned}\{\mathbf{D}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D},\tilde{\mathbf{X}}}\ &\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \\ &\mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2 \\ &\text{s.t. } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j. \end{aligned} \tag{4.9}$$

  This is discriminative dictionary learning problem, and we use the framework of [140] to update $\mathbf{D}, \tilde{\mathbf{X}}$. This can be easily generalized to utilize other dictionary learning algorithms as well.

The proposed Shared Discriminative Dictionary Learning (SDDL) algorithm is summarized in Algorithm 5.

Input: Data $\{\mathbf{Y_i}\}_{\mathbf{i=1}}^{M}$ and corresponding class labels $\{C_i\}_{i=1}^{M}$ for M domains, sparsity level T_0, dictionary size K and dimension d_f, parameter values μ, ν

Procedure:

1. *Initialize:* Initialize $\tilde{\mathbf{P}}$ such that $\mathbf{P_i}\mathbf{P_i} = \mathbf{I}\, \forall\, i = 1, \cdots, M$. For this, PCA of the data, $\mathbf{Y_i}$ can be used to initialize $\mathbf{P_i}$.

2. *Update step for* $\tilde{\mathbf{P}}$*:* Update $\tilde{\mathbf{P}}$ as:

$$\tilde{\mathbf{P}}^{*} = \underset{\tilde{\mathbf{P}}}{\arg\min}\ \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2$$
$$-\lambda \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T) \text{ s.t. } \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I},\ i = 1, \cdots, M$$

using Stiefel manifold optimization technique [133].

3. *Update step for* $\mathbf{D}, \tilde{\mathbf{X}}$*:* Learn common dictionary $\mathbf{D}$ and sparse code, $\tilde{\mathbf{X}}$ using discriminative dictionary learning algorithm such as FDDL [140]

$$\{\mathbf{D}^{*}, \tilde{\mathbf{X}}^{*}\} = \underset{\mathbf{D},\tilde{\mathbf{X}}}{\arg\min}\ \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 +$$
$$\nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2 \text{ s.t. } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j.$$

Output: Learned dictionary $\mathbf{D}$, projection matrices $\{\mathbf{P_i}\}_{\mathbf{i=1}}^{M}$.

Algorithm 5: Shared Domain-adapted Dictionary Learning (SDDL)

4.4 Non-linear extension

In many vision problems, projecting the original features may not be good enough due to non-linearity in data. This can be overcome by transforming the data into a high-dimensional feature space. Let $\mathbf{\Phi} : \mathbb{R}^n \rightarrow \mathcal{H}$ be a mapping to the reproducing kernel Hilbert space $\mathcal{H}$. The mapping $\boldsymbol{\mathcal{P}}_\mathbf{i}$ to the reduced space, can be characterized by a compact, linear operator, $\boldsymbol{\mathcal{P}}_\mathbf{i} : \mathcal{H} \rightarrow \mathbb{R}^d$. As the feature space can be infinite dimensional, the projection matrix $\boldsymbol{\mathcal{P}}_\mathbf{i}$ cannot be handled in this form. To make the kernelization of the algorithm possible, we use the following proposition:

Proposition 1: *There exists an optimal solution* $\mathbf{P}_\mathbf{1}^*, \cdots, \mathbf{P}_\mathbf{M}^*, \mathbf{D}^*$ *to equation (4.6), which has the following form:*

$$\mathbf{P}_\mathbf{i}^* = (\mathbf{Y}_\mathbf{i}\mathbf{A}_\mathbf{i})^\mathbf{T} \; \forall \, i = 1, \cdots, M, \tag{4.10}$$

$$\mathbf{D}^* = \tilde{\mathbf{P}}^* \tilde{\mathbf{Y}} \tilde{\mathbf{B}}, \tag{4.11}$$

where, $\tilde{\mathbf{P}}^* = [\mathbf{P}_\mathbf{1}^*, \cdots, \mathbf{P}_\mathbf{M}^*]$, *for some* $\mathbf{A}_\mathbf{i} \in \mathbb{R}^{N_i \times n}$ *and some* $\tilde{\mathbf{B}} \in \mathbb{R}^{\sum N_i \times K}$.
Proof: See Appendix B.

With this proposition, the cost functions can be written as:

$$\begin{aligned} \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) &= \|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2 + \\ \mu\|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})\|_F^2 &+ \nu\|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out}\|_F^2, \end{aligned} \tag{4.12}$$

$$\mathcal{C}_2(\tilde{\mathbf{A}}) = -\text{trace}((\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}})(\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}})^T), \tag{4.13}$$

where, $\tilde{\mathbf{K}} = \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{A}}^\mathbf{T} = [\mathbf{A}_\mathbf{1}^\mathbf{T}, \cdots, \mathbf{A}_\mathbf{M}^\mathbf{T}]$. The equality constraints now become:

$$\mathbf{P}_\mathbf{i}\mathbf{P}_\mathbf{i}^\mathbf{T} = \mathbf{A}_\mathbf{i}^\mathbf{T}\mathbf{K}_\mathbf{i}\mathbf{A}_\mathbf{i} = \mathbf{I}, \; \forall i \; = \; 1, \cdots, M, \tag{4.14}$$

where, $\mathbf{K}_\mathbf{i} = \mathbf{Y}_\mathbf{i}^\mathbf{T}\mathbf{Y}_\mathbf{i}$. The optimization problem now becomes:

$$\begin{aligned} \{\tilde{\mathbf{A}}^*, \tilde{\mathbf{B}}^*, \tilde{\mathbf{X}}^*\} &= \underset{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}}{\arg\min} \; \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) + \lambda\mathcal{C}_2(\tilde{\mathbf{A}}) \\ \text{s.t. } \mathbf{A}_\mathbf{i}^\mathbf{T}\mathbf{K}_\mathbf{i}\mathbf{A}_\mathbf{i} = \mathbf{I}, \; i &= 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j. \end{aligned} \tag{4.15}$$

This formulation allows joint update of $\mathbf{D}$ and $\mathbf{P}_\mathbf{i}$ via $\mathbf{A}_\mathbf{i}$.

Let $\boldsymbol{\mathcal{K}} = \langle \Phi(\tilde{\mathbf{Y}}), \Phi(\tilde{\mathbf{Y}}) \rangle_{\mathcal{H}}$. Then, it can be shown similar to proposition 1 that:

$$\boldsymbol{\mathcal{P}}_{\mathbf{i}}^{*} = \mathbf{A}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{Y})^{\mathbf{T}}; \mathbf{D}^{*} = \tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}}\tilde{\mathbf{B}}.$$

Thus, we get the cost functions as:

$$\begin{aligned} \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2 + \\ \mu\|\tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2 + \nu\|\tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}}\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2, \quad (4.16) \\ \mathcal{C}_2(\tilde{\mathbf{A}}) = -\text{trace}((\tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}})(\tilde{\mathbf{A}}^{\mathbf{T}}\boldsymbol{\mathcal{K}})^T) \quad (4.17) \end{aligned}$$

and the equality constraints as,

$$\mathbf{A}_{\mathbf{i}}^{\mathbf{T}}\boldsymbol{\mathcal{K}}_{\mathbf{i}}\mathbf{A}_{\mathbf{i}} = \mathbf{I} \quad \forall\, i = 1, \cdots, M,$$

where $\boldsymbol{\mathcal{K}}_i = \langle \Phi(\mathbf{Y}_{\mathbf{i}}), \Phi(\mathbf{Y}_{\mathbf{i}}) \rangle_{\mathcal{H}}$.

4.4.1 Update step for $\tilde{\mathbf{A}}$

Here we assume that $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed. Then, the optimization for $\tilde{\mathbf{A}}$ can be solved efficiently. We have the following proposition.

Proposition 2: *The optimal solution of equation (4.15) when $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed is:*

$$\tilde{\mathbf{A}}^{*} = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^{*}, \quad (4.18)$$

where, $\mathbf{V}$ and $\mathbf{S}$ come from the eigen decomposition of $\tilde{\mathbf{K}} = \mathbf{V}\mathbf{S}\mathbf{V}^{\mathbf{T}}$, and $\mathbf{G}^{*} \in \mathbb{R}^{\sum N_i \times n} = [\mathbf{G}_{\mathbf{1}}^{*\mathbf{T}}, \cdots, \mathbf{G}_{\mathbf{M}}^{*\mathbf{T}}]^T$ is the optimal solution of the following problem:

$$\begin{aligned} \{\mathbf{G}^{*}\} = \arg\min_{\mathbf{G}}\ \text{trace}[\mathbf{G}^{\mathbf{T}}\mathbf{H}\mathbf{G}] \\ \textit{s.t. } \mathbf{G}_{\mathbf{i}}^{\mathbf{T}}\mathbf{G}_{\mathbf{i}} = \mathbf{I}\,\forall\, i\ = 1, \cdots, M, \quad (4.19) \end{aligned}$$

where,

$$\begin{aligned} \mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}}) \\ (\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}}. \quad (4.20) \end{aligned}$$

Proof: See Appendix I.

Equation (4.19) is non-convex due to non-linear equality constraints. Specifically, due to the orthonormality condition on $\mathbf{G_i}$, it involves optimization on the Stiefel manifold. We solved this problem using the efficient approach presented in [133].

4.4.2 Update step for $\tilde{\mathbf{B}}, \tilde{\mathbf{X}}$

For a fixed $\tilde{\mathbf{A}}$, the problem becomes that of discriminative dictionary learning, with data as $\mathbf{Z} = \tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}$ and dictionary $\mathbf{D} = \tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}\tilde{\mathbf{B}}$. To jointly learn the dictionary, $\mathbf{D}$, and sparse code, $\tilde{\mathbf{X}}$, we use the framework of the discriminative dictionary learning approach presented in [140]. Once the dictionary, $\mathbf{D}$, is learned, we can update $\tilde{\mathbf{B}}$ as:

$$\tilde{\mathbf{B}} = \mathbf{Z}^{\dagger}\mathbf{D}, \tag{4.21}$$

where $\mathbf{Z}^{\dagger}$ is the pseudo-inverse of $\mathbf{Z}$ defined as $\mathbf{Z}^{\dagger} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$.

The proposed, Non-linear Shared Domain-adapted Dictionary Learning (kerSDDL) algorithm is summarized in Algorithm 6.

4.5 Classification

Given a test sample, $\mathbf{y_{te}}$ from domain k, we propose the following steps for classification, similar to [78].

4.5.1 Linear Classification

1. Compute the embedding of the sample in the common subspace, $\mathbf{z_{te}}$ using the projection, $\mathbf{P}^*_{\mathbf{k}}$.
$$\mathbf{z_{te}} = \mathbf{P_k}^*\mathbf{y_{te}}.$$

2. Compute the sparse coefficients, $\hat{\mathbf{x}}_{\mathbf{te}}$, of the embedded sample over dictionary $\mathbf{D}$

Input: Data $\{\mathbf{Y_i}\}_{\mathbf{i=1}}^{M}$ and corresponding class labels $\{C_i\}_{i=1}^{M}$ for M domains, sparsity level T_0, dictionary size K and dimension n, parameter values μ, ν

Procedure:

1. *Initialize:* Initialize $\tilde{\mathbf{A}}$ such that $\mathbf{A_i}\boldsymbol{\mathcal{K}}_\mathbf{i}\mathbf{A_i} = \mathbf{I}\ \forall\ i = 1, \cdots, M$. For this, find SVD of each kernel matrix, $\boldsymbol{\mathcal{K}}_\mathbf{i} = \mathbf{V_i}\mathbf{S_i}\mathbf{V_i^T}$. Set $\mathbf{A}_i$ as the matrix of eigen-vectors with top n eigen-values as columns.

2. *Update step for* $\tilde{\mathbf{B}}$*:* Learn common dictionary $\mathbf{D}$ with data as $\mathbf{Z} = \tilde{\mathbf{A}}^\mathbf{T}\boldsymbol{\mathcal{K}}$, and using discriminative dictionary learning algorithm as FDDL. Update $\tilde{\mathbf{B}}$ as:

$$\tilde{\mathbf{B}} = \mathbf{Z}^{\dagger}\mathbf{D}.$$

3. *Update step for* $\tilde{\mathbf{A}}$*:* Update $\tilde{\mathbf{A}}$ as:

$$\{\mathbf{G}^*\} = \arg\min_{\mathbf{G}}\ \text{trace}[\mathbf{G^T H G}]$$
$$\textit{s.t. } \mathbf{G_i^T G_i} = \mathbf{I}\ \forall\ i\ = 1, \cdots, M,$$

where, $\tilde{\mathbf{A}}^* = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^*$ and $\mathbf{H}$ is:

$$\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^\mathbf{T}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})$$
$$(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})^\mathbf{T} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})^\mathbf{T} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}}.$$

Output: Learned dictionary $\mathbf{D}$, projection matrices $\{\mathbf{A_i}\}_{\mathbf{i=1}}^{M}$.

Algorithm 6: Non-linear Shared Domain-adapted Dictionary Learning (kerSDDL)

using the OMP algorithm [89].

$$\mathbf{\hat{x}_{te}} = \arg\min_{\mathbf{x}} \ \|\mathbf{z_{te}} - \mathbf{Dx}\|_{\mathbf{F}}^{\mathbf{2}} \text{ s.t. } \|\mathbf{x}\|_0 \leq T_0.$$

3. Now, the sample can be assigned to class i, if the reconstruction using the class dictionary, $\mathbf{D_i}$ and the sparse code corresponding to the atoms of the dictionary, $\mathbf{\hat{x}_{te}^i}$ is minimum.

$$\text{Output class} = \arg\min_{i=1,\cdots,C} \ \|\mathbf{z_{te}} - \mathbf{D_i}\mathbf{\hat{x}_{te}^i}\|_{\mathbf{F}}^{\mathbf{2}}.$$

However, the reconstruction error may not be discriminative enough in the reduced space. So, we project the class-wise reconstruction, $\mathbf{D_i}\mathbf{\hat{x}_{te}^i}$ into the feature space, and assign the test sample to the class with the minimum error in the original feature space:

$$\text{Output class} = \arg\min_{i=1,\cdots,C} \ \|\mathbf{y_{te}} - \mathbf{P_k}^{*\mathbf{T}}\mathbf{D_i}\mathbf{\hat{x}_{te}^i}\|_{\mathbf{F}}^{\mathbf{2}}. \tag{4.22}$$

4.5.2 Non-linear classification

Here, we consider the general case of classifying mapping of the sample into kernel space, $\mathbf{\Phi}(\mathbf{y_{te}})$.

1. Compute the embedding of the sample in the common subspace, $\mathbf{z_{te}}$ using the projection, $\boldsymbol{\mathcal{P}}_{\mathbf{k}}^*$.

$$\mathbf{z_{te}} = \boldsymbol{\mathcal{P}}_{\mathbf{k}}^*\mathbf{\Phi}(\mathbf{y_{te}}) = \mathbf{A_k}\boldsymbol{\mathcal{K}}_{\mathbf{te}},$$

where, $\boldsymbol{\mathcal{K}}_{\mathbf{te}} = \langle \mathbf{\Phi}(\mathbf{Y_k}), \mathbf{\Phi}(\mathbf{y_{te}})\rangle$.

2. Compute the sparse coefficients, $\mathbf{\hat{x}_{te}}$, of the embedded sample over dictionary $\mathbf{D}$ using the OMP algorithm [89].

$$\mathbf{\hat{x}_{te}} = \arg\min_{\mathbf{x}} \ \|\mathbf{z_{te}} - \mathbf{Dx}\|_{\mathbf{F}}^{\mathbf{2}} \text{ s.t. } \|\mathbf{x}\|_0 \leq T_0.$$

3. Project the class-wise reconstruction, $\mathbf{D_i \hat{x}^i_{te}}$ into the feature space, and assign the test sample to the class with the minimum error in the original feature space:

$$
\begin{aligned}
\text{Output class} &= \underset{i=1,\cdots,C}{\arg\min}\ \|\boldsymbol{\Phi}(\mathbf{y_{te}}) - \boldsymbol{\mathcal{P}}^{*\mathbf{T}}_{\mathbf{k}}\mathbf{D_i}\mathbf{\hat{x}^i_{te}}\|^{\mathbf{2}}_{\mathbf{F}} \\
&= \underset{i=1,\cdots,C}{\arg\min}\ \boldsymbol{\kappa}_{\mathbf{te}} - \mathbf{2}\boldsymbol{\mathcal{K}}_{\mathbf{te}}\mathbf{A}^*_{\mathbf{k}}\mathbf{D_i} + \mathbf{\hat{x}^{iT}_{te}}\mathbf{D}^{\mathbf{T}}_{\mathbf{i}}\mathbf{A}^*_{\mathbf{k}}\boldsymbol{\mathcal{K}}_{\mathbf{k}}\mathbf{A}^*_{\mathbf{k}}\mathbf{D_i}\mathbf{\hat{x}^i_{te}},
\end{aligned}
$$

where $\boldsymbol{\kappa}_{\mathbf{te}} = \langle \boldsymbol{\Phi}(\mathbf{y_{te}}), \boldsymbol{\Phi}(\mathbf{y_{te}}) \rangle$.

4.6 Experiments

We conducted various experiments to ascertain the effectiveness of the proposed method. First, we demonstrate some synthesis and recognition results on the CMU Multi-PIE dataset for face recognition across pose and illumination variations. This also provides insights into our method through visual examples. Next we show the performance of our method on domain adaptation databases and compare it with existing adaptation algorithms.

4.6.1 CMU Multi-Pie Dataset

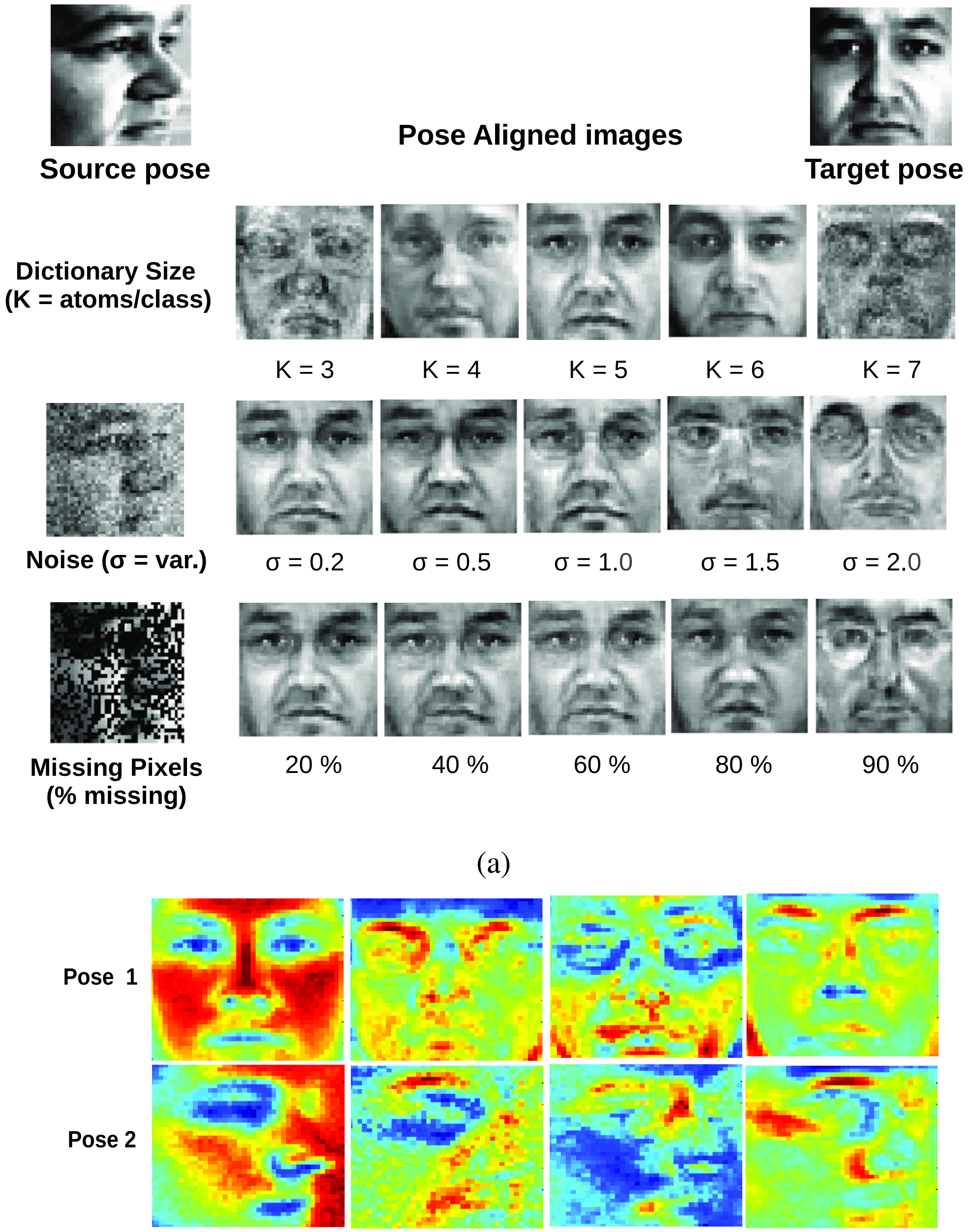
The Multi-PIE dataset [44] is a comprehensive face dataset of 337 subjects, having images taken across 15 poses, 20 illuminations, 6 expressions and 4 different sessions. For the purpose of our experiment, we used 129 subjects common to both Session 1 and 2. The experiment was done on 5 poses, ranging from frontal to 75^o. Frontal faces were taken as the source domain, while different off-frontal poses were taken as target domains. Dictionaries were trained using illuminations $\{1, 4, 7, 12, 17\}$ from the source and the target poses, in Session 1 per subject. All the illumination images from Session 2, for the target pose, were taken as probe images. The linear kernel was used for all the experiments.

4.6.1.1 Pose Alignment

First we consider the problem of pose alignment using the proposed dictionary learning framework. Pose alignment is challenging due to the highly non-linear changes induced by 3-D rotation of face. Images at the extreme pose of 60^{o} were taken as the target pose. A shared discriminative dictionary was learned using the approach described in Algorithm 5. Given the probe image, it was projected on the latent subspace and reconstructed using the dictionary. The reconstruction was back-projected onto the source pose domain, to give the aligned image. Figure 4.2(a) shows the synthesized images for various conditions. We can draw some useful insights about the method from this figure. Firstly, it can be seen that there is an optimal dictionary size, $K = 5$, where the best alignment is achieved. Further, by learning a discriminative dictionary, the identity of the subject is retained. For $K = 7$, the alignment is not good, as the learned dictionary is not able to successfully correlate the two domains when there are more atoms in the dictionary. Dictionary with $K = 3$ has higher reconstruction error, hence the result is not optimal. We chose $K = 5$ for additional experiments with noisy images. It can be seen that from rows 2 and 3 that the proposed method is robust even at high levels of noise and missing pixels. Moreover, de-noised and in-painted synthesized images are produced as shown in rows 2 and 3 of Figure 4.2(a), respectively. This shows the effectiveness of our method. Moreover, the learned projection matrices (Figure 4.2(b)) show that our method can learn the internal structure of the two domains. As a result, it is able to learn a robust common dictionary.

4.6.1.2 Recognition

We also conducted a recognition experiment using the set-up described above. Table 4.1 shows that our method compares favorably with some of the recently proposed multi-view recognition algorithms [112], and gives the best performance on average. The linear kernel was found to be giving better performance, hence, we do not report the results for kerSDDL. The dictionary learning algorithm, FDDL [140] is not optimal here



(a)

(b)

Figure 4.2: (a) Examples of pose-aligned images using the proposed method. Synthesis in various conditions demonstrate the robustness of the method. (b) First few components of the learned projection matrices for the two poses.

as it is not able to efficiently represent the non-linear changes introduced by the pose variation.

Method	Probe pose					Average
	15^o	30^o	45^o	60^o	75^o	
PCA	15.3	5.3	6.5	3.6	2.6	6.7
PLS [111]	39.3	40.5	41.6	41.1	38.7	40.2
LDA	98.0	94.2	91.7	84.9	79.0	89.5
CCA [111]	92.1	89.7	88.0	86.1	83.0	83.5
GMLDA [112]	**99.7**	**99.2**	98.6	94.9	95.4	97.6
FDDL [140]	96.8	90.6	94.4	91.4	90.5	92.7
SDDL	98.4	98.2	**98.9**	**99.1**	**98.8**	**98.7**

Table 4.1: Comparison of the proposed method with other algorithms for face recognition across pose.

4.6.2 Object Recognition

We now evaluate our method for object recognition. The experiments use the dataset which was introduced in [109]. The dataset consists of images from 3 sources: Amazon (consumer images from online merchant sites), DSLR (images by DSLR camera) and Webcam (low quality images from webcams). In addition, we also tested on the Caltech-256 dataset [43], taking it as the fourth domain. Figure 4.3 shows sample images from these datasets, and clearly highlights the differences between the domains. We follow 2 set-ups for testing the algorithm. In the first set-up, 10 common classes: BACKPACK, TOURING-BIKE, CALCULATOR, HEADPHONES, COMPUTER- KEYBOARD, LAPTOP-101, COMPUTER- MONITOR, COMPUTER-MOUSE, COFFEE- MUG, AND VIDEO- PROJECTOR, common to all the four datasets are used. In this case, there are a total of 2533 images. Each category has 8 to 151 images in a dataset. In the second set-up, we eval-

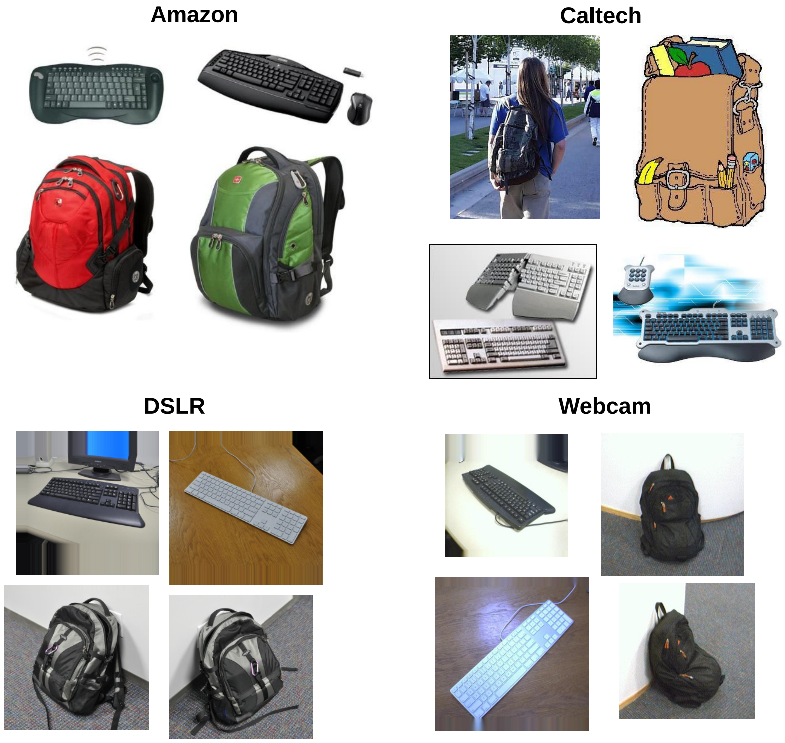


Figure 4.3: Example images from KEYBOARD and BACK-PACK categories in Caltech-256, Amazon, Webcam and DSLR. Caltech-256 and Amazon datasets have diverse images, Webcam and DSLR are similar datasets with mostly images from offices.

uate the methods for adaptation using multiple domains. In this case, we restrict to the first dataset, and test on all the 31 classes in it. For both the cases, we use 20 training samples per class for Amazon/Caltech, and 8 samples per class for DSLR/Webcam when used as source, and 3 training samples for all of them when used for target domain. The remaining data in the target domain is used for testing. The experiment is run 20 times for random train/test splits and the result is averaged over all the runs.

We demonstrate the effectiveness of the proposed method for the two cases: 1. same features extracted for all the domains, 2. different features extracted for different domains.

4.6.2.1 Adaptation with same features

First, we test the proposed algorithms for the case when the same feature is extracted for all the domains.

Feature Extraction: We used the 800-bin SURF features provided by [109] for the Amazon, DSLR and Webcam datasets. For the Caltech images, the SURF features were first extracted from the images of the Caltech data and a random subset of the Amazon dataset. The features obtained from the Amazon dataset were grouped into 800 clusters using the k-means algorithm. The cluster centers were then used to quantize the SURF features obtained from the Caltech data to form 800-bin histograms. The histograms were normalized and then used for classification.

Parameter Settings: We set $\mu = 4$ and $\nu = 30$. Dictionary size, $K = 4$ atoms per class and final dimension, $n = 60$ for the first set-up, for both SDDL and kerSDDL algorithms. For the second set-up, $K = 6$ atoms per class and $n = 90$ for SDDL and kerSDDL. For FDDL, the parameters, μ and ν are the same as SDDL, and we learn $K = 8$ atoms per class for the first set-up and $K = 10$ atoms per class for the second. The SDDL algorithm was trained using same projection matrix for all the domains as discussed in Section 4.3.3. We initialized the matrices as PCA of source, target data or both data taken together, and report the best performance among them. For kerSDDL method, we used the simple non-parametric histogram intersection kernel for reporting all the values. The projection matrix for kerSDDL was initialized as described in Algorithm 6. The FDDL dictionary was trained using both the source and the target domain features, as it was found to give the best results. Original histogram features were used for both the algorithms. Performance of the proposed SDDL method is compared to FDDL [140], and some recently proposed domain-adaptation algorithms [40–42, 51, 66, 81, 109].

1. **Results using single source:** Tables 4.2(a), 4.2(b) show a comparison of the results of different methods on eight source-target pairs. The proposed algorithms give the

(a) Performance comparison on single source four domains benchmark (C: caltech, A: amazon, D: dslr, W: webcam) for C → A, C → D, A → C, A → W source/target pairs

Methods	C → A	C → D	A → C	A → W
Metric [109]	33.7 ± 0.8	35.0 ± 1.1	27.3 ± 0.7	36.0 ± 1.0
SGF [41]	40.2 ± 0.7	36.6 ± 0.8	37.7 ± 0.5	37.9 ± 0.7
GFK [40]	46.1 ± 0.6	55.0 ± 0.9	39.6 ± 0.4	56.9 ± 1.0
HFA [66]	45.5 ± 0.9	51.9 ± 1.1	31.1 ± 0.6	58.6 ± 1.0
SID [81]	50 ± 0.5	57.1 ± 0.4	41.5 ± 0.8	57.8 ± 0.5
FDDL [140]	39.3 ± 2.9	55.0 ± 2.8	24.3 ± 2.2	50.4 ± 3.5
SDDL	**54.4 ± 2.2**	67.7 ± 4.0	**41.8 ± 2.2**	67.1 ± 3.2
kerSDDL	49.5 ± 2.6	**76.7 ± 3.9**	27.4 ± 2.4	**72.0 ± 4.8**

(b) Performance comparison on single source four domains benchmark (C: caltech, A: amazon, D: dslr, W: webcam) for W → C, W → A, D → A, D → W source/target pairs

Methods	W → C	W → A	D → A	D → W
Metric [109]	21.7 ± 0.5	32.3 ± 0.8	30.3 ± 0.8	55.6 ± 0.7
SGF [41]	29.2 ± 0.7	38.2 ± 0.6	39.2 ± 0.7	69.5 ± 0.9
GFK [40]	32.8 ± 0.1	46.2 ± 0.6	46.2 ± 0.6	80.2 ± 0.4
HFA [66]	31.1 ± 0.6	45.9 ± 0.7	45.8 ± 0.9	62.1 ± 0.7
SID [81]	40.6 ± 0.4	**51.5 ± 0.6**	50.3 ± 0.2	**87.8 ± 1.0**
FDDL [140]	22.9 ± 2.6	41.1 ± 2.6	36.7 ± 2.5	65.9 ± 4.9
SDDL	**41.5 ± 2.1**	48.2 ± 2.3	**50.6 ± 2.1**	86.4 ± 2.8
kerSDDL	29.7 ± 1.9	49.4 ± 2.1	48.9 ± 3.8	72.6 ± 2.1

(c) Performance comparison on multiple sources three domains benchmark

Source	Target	SGF*	SGF	RDALR	FDDL	SDDL	kerSDDL
		[42]	[41]	[51]	[140]		
dslr, amazon	webcam	**64.5 ± 0.3**	52 ± 2.5	36.9 ± 1.1	41.0 ± 2.4	53.6 ± 1.2	57.8 ± 2.4
amazon, webcam	dslr	51.3 ± 0.7	39 ± 1.1	31.2 ± 1.3	38.4 ± 3.4	55.8 ± 2.0	**56.7 ± 2.3**
webcam, dslr	amazon	**38.4 ± 1.0**	28 ± 0.8	20.9 ± 0.9	19.0 ± 1.2	23.8 ± 1.2	24.1 ± 1.6

Table 4.2: Comparison of the performance of the proposed method on the Amazon, Webcam, DSLR and Caltech datasets.

best performance for six domain pairs, and is the second best for two pairs. For Caltech-DSLR and Amazon-Webcam domain pairs, there is more than 15% improvement over the GFK [40] and SID [81] algorithms. Furthermore, a comparison with the FDDL algorithm shows that the learning framework of [140] is inefficient, when the test data comes from a different distribution than the data used for training. Both the SDDL and kerSDDL algorithms perform better than FDDL on all the pairs.

2. **Results using multiple sources:** As the proposed algorithm can also handle multiple domains, we also experimented with multiple source adaptation. Table 4.2 (c) shows the results for three possible combinations. The proposed methods outperforms the original SGF method [41] on two settings, and other methods for all the settings. However, [42] reports higher numbers on webcam and amazon as targets, using boosted classifiers. Similarly techniques can be explored for improving the proposed method as a future direction.

3. **Ease of adaptation:** A rank of domain (ROD) metric was introduced in [40] to measure the adaptability of different domains. It was shown that ROD correlates with the performance of adaptation algorithm. For example, Amazon-Webcam pair has higher ROD than DSLR-Webcam pair, hence, GFK performs worse on the former. However, for our case, we find that the recognition rates for these cases are 72.0 % and 72.6 %, respectively. This is the case because by learning projections along-with the common dictionary, we can achieve a better alignment of the datasets.

4. **Parameter Variations:** We also conducted experiments studying recognition performance under different input parameters. Figure 4.4 shows the result of different settings. The implications are briefly discussed below:

   (a) **Number of source images:** Here, we choose Amazon/Webcam domain pair, as it is ”difficult” to adapt. We increased the number of source images and studied the performance of SDDL and kerSDDL and compared it with FDDL.

It can be seen that while FDDL's performance decreases sharply with more source images, SDDL and kerSDDL methods show increase in the performance. Hence, by adapting the source to the target domain, our method can use the source information to increase the accuracy of target recognition, even when their distributions are very different.

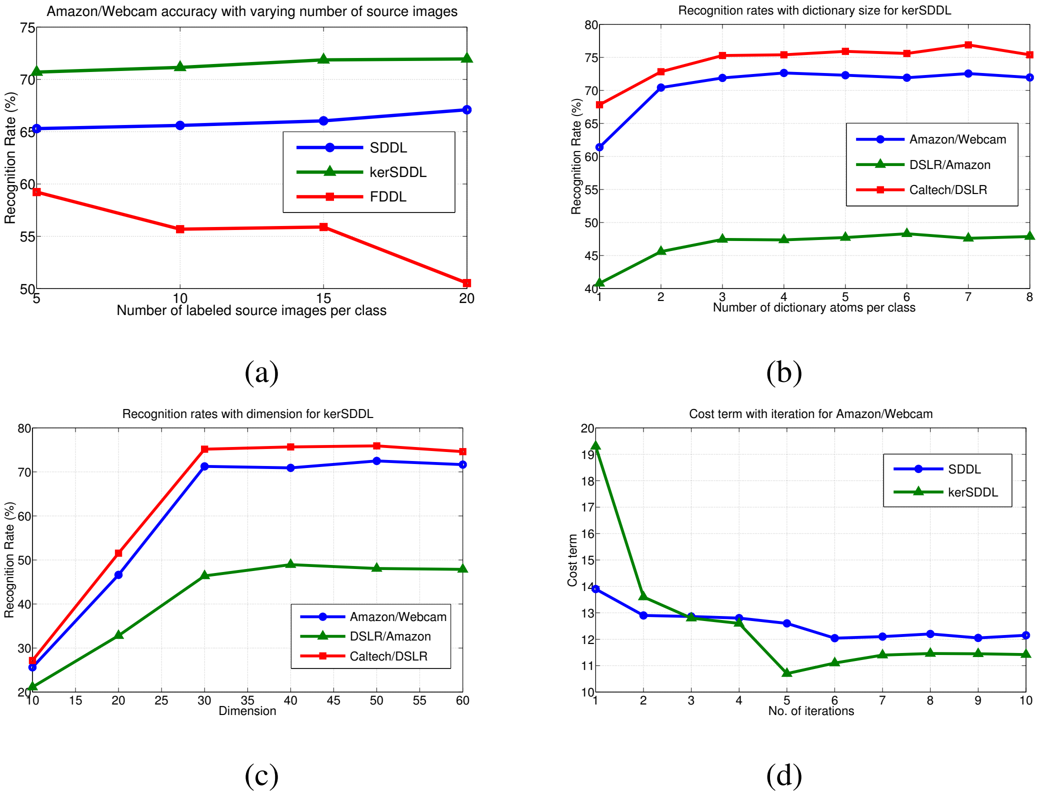
(b) **Dictionary size:** We varied the dictionary size for kerSDDL algorithm for different source-target pairs. All the domain pairs show an initial sharp increase in the performance, and then become almost flat after the dictionary size of 3 or 4. The flat region indicates that the alignment of the source and the target data is limited by the number of available target samples. But also, on a positive note, it can be seen that even a smaller dictionary can give the optimal performance.

(c) **Common subspace dimension:** Similar to the previous case, we get an initial sharp increase followed by a flat recognition curve. This shows that the method is effective even when the data is projected onto a low-dimensional space.

5. **Convergence:** Figure 4.4(d) shows the cost function with iteration for SDDL and kerSDDL algorithms. It can be seen that both the algorithms converge quickly in 5-6 iterations.

4.6.2.2 Adaptation with different features

The proposed methods can be generalized to cases when features of different types (like dimension) are extracted for different domains. Note that the original FDDL algorithm [140] cannot be used for such cases. Also some of the adaptation algorithms compared above cannot be generalized for such cases [40, 42, 51, 81]. We compare the proposed methods with recent heterogeneous adaptation methods [60, 66, 121, 131] and demonstrate their effectiveness.



(a) (b)

(c) (d)

Figure 4.4: Recognition performance under different: (a) number of source images, (b) dictionary size, (c) common subspace dimension. (d) Convergence of the proposed algorithms. Naming of domains is done as source/target.

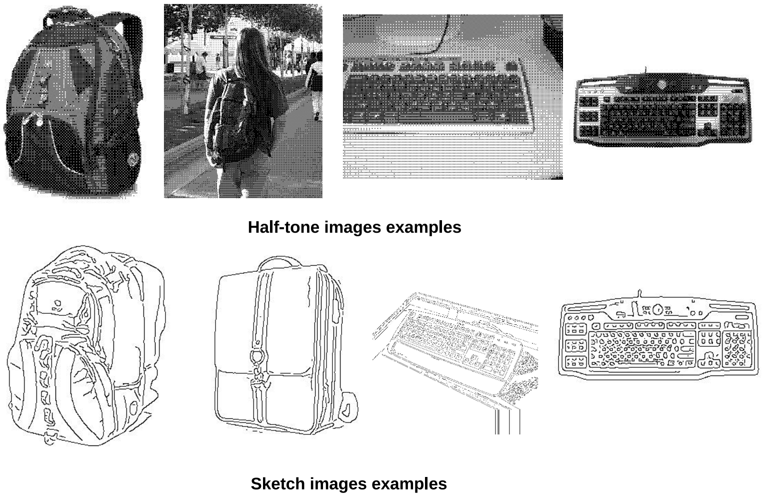


Figure 4.5: Example images from half-tone and sketch datasets.

Experiment Set-up: We restrict the evaluation to Amazon, DSLR and Webcam datasets, using all the 31 classes for evaluation. The train-test split was done as described in Section 4.6.2.1. The evaluation was done using three different experimental set-ups described as follows:

1. **DSLR-600 dataset:** We extracted 600-dimensional SURF features for the DSLR dataset as described in [60]. We present results for adaptation from the 800-dimensional SURF features extracted in Section 4.6.2.1 to the new features.

2. **Halftone and Sketch datasets:** To test the effectiveness of the proposed algorithms across different domain shifts, we created two new datasets by half-toning and edge detection from the original dataset. Figure 4.5 shows some of the images from these datasets. Half-toning images, which imitate the effect of jet-printing technology in the past, were generated using the dithering algorithm in [73]. Edge images are obtained by applying the Canny edge detector [19] with the threshold set to 0.07. We extracted 800-bin SURF features for both the datasets, following the same approach as for the original dataset.

Parameter Setting: We set $\mu = 4$ and $\nu = 30$. Dictionary size, $K = 4$ atoms per class and final dimension, $n = 90$ for all the set-ups, for both SDDL and kerSDDL algorithms. For the kerSDDL method, we used the non-parametric histogram intersection kernel for all the experiments. The projection matrix for the kerSDDL method was initialized as described in Algorithm 6. For SDDL, we initialized a separate projection matrix for each domain as described in Algorithm 5.

1. **DSLR-600 adaptation** Table 4.2(a) shows the comparison of the proposed methods for adaptation of 800-dimensinal SURF features to 600-dimensional SURF features from DSLR data. It can be seen that the kerSDDL method gives better performance than the recent state-of-art heterogeneous adaptation methods. The SDDL algorithm also performs on par with other algorithms.

2. **Half-tone and Sketch dataset adaptation** Tables 4.2(b), 4.2(c) show results for adaptation from original images to half-tone and sketch image datasets respectively.

The proposed algorithms are compared with [60] and nearest neighbor classification method. It can be seen that kerSDDL performs better than [60] for all the source-target pairs.

4.7 Conclusion

We presented a novel framework for adapting dictionaries to testing domains under arbitrary domain shifts. An efficient optimization method is presented. Furthermore, the method is kernelized so that it is robust and can deal with the non-linearity present in the data. The learned dictionary is compact and low-dimensional. To gain intuition into the working of the method, we demonstrated applications like pose alignment and pose-robust face recognition. We evaluated the proposed algorithms for different object recognition adaptations. Specifically, we showed that the methods can be used for cases like heterogeneous domain adaptation, where original dictionary learning framework cannot be applied. The proposed methods were compared with the recent domain adaptation algorithms, and the proposed methods were found to be better or comparable to the previous methods. Future works will include studying the effect of using unlabeled data while training, and other relevant problems like large-scale and online adaptation of dictionaries.

(a) Performance comparison on recognition across different features

Source	Target	Metric -asymm [60]	HeMap [121]	DAMA [131]	HFA [66]	SDDL	kerSDDL
amazon	dslr-600	53.1 ± 2.4	42.8 ± 2.4	53.3 ± 2.4	55.4 ± 2.8	50.4 ± 2.5	$\mathbf{61.5 \pm 3.6}$
webcam	dslr-600	53.0 ± 3.2	42.2 ± 2.6	53.2 ± 3.2	54.3 ± 3.7	49.4 ± 2.9	$\mathbf{58.3 \pm 2.6}$

(b) Performance comparison for adaptation to half-tone images

Methods	W → D-half	D → W-half	A → D-half	A → W-half
kNN	25.2 ± 2.6	35.2 ± 2.2	25.0 ± 2.0	34.0 ± 1.4
Metric-asymm [60]	38.8 ± 2.4	40.2 ± 2.0	33.8 ± 3.8	39.0 ± 2.2
SDDL	32.3 ± 1.7	36.4 ± 1.9	30.1 ± 2.0	34.7 ± 1.7
kerSDDL	$\mathbf{42.0 \pm 2.6}$	$\mathbf{43.0 \pm 2.3}$	$\mathbf{46.4 \pm 3.1}$	$\mathbf{51.0 \pm 2}$

(c) Performance comparison for adaptation to sketch images

Methods	W → D-sketch	D → W-sketch	A → D-sketch	A → W-sketch
kNN	31.4 ± 2.7	31.3 ± 1.7	32.1 ± 2.4	33.6 ± 2.7
Metric-asymm [60]	39.1 ± 2.7	35.0 ± 2.2	38.0 ± 2.8	37.3 ± 2.5
SDDL	35.8 ± 2.1	32.1 ± 1.8	33.8 ± 2.1	34.0 ± 1.8
kerSDDL	$\mathbf{41.5 \pm 2.6}$	$\mathbf{38.0 \pm 2.6}$	$\mathbf{42.1 \pm 2.4}$	$\mathbf{42.5 \pm 2.3}$

Table 4.3: Comparison of the performance of the proposed methods for performance on adaptation for DSLR-600, Half-tone and Sketch datasets.

Chapter 5: Analysis Sparse Coding

5.1 Introduction

Sparse representation-based data-driven models have become popular in vision and image processing communities. Olshausen and Field [82] in their seminal work introduced the idea of learning representation based on data itself rather than off-the-shelf bases. Since then sparse representation-based dictionaries have been widely used for image restoration and classification [2], [140], [68], [97], [145], [5], [115], [79], [84], [85]. Given a data matrix $\mathbf{Y} \in \mathbb{R}^{d \times N}$, whose columns represent d-dimensional signals, the basic formulation underlying these methods involves learning a K-atom synthesis dictionary $\mathbf{D}^* \in \mathbb{R}^{d \times K}$ and sparse code $\mathbf{X}^* \in \mathbb{R}^{K \times N}$, obtained as:

$$\{\mathbf{D}^*, \mathbf{X}^*\} = \underset{\mathbf{D},\mathbf{X}}{\arg\min} \; \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ s.t. } \|\mathbf{X}\|_0 \leq T_0$$

where, T_0 is the sparsity level. This is a non-convex problem and different schemes have been proposed for optimization, notably, K-SVD [2], matrix factorization [4] and gradient descent [5] techniques.

In recent years, an alternate analysis sparse coding (or co-sparse) model has also been examined [76]. Figure 5.1 presents a brief comparison of the two models. Previous works have shown that analysis model can yield richer feature representations and better results for image restoration [76]. However, to the best of our knowledge, the analysis framework has not been exploited yet for image classification tasks. In this work, we examine the application of the analysis model for recognition, and demonstrate that it can achieve comparable or better performance than synthesis models. Further, we show that the proposed approach can lead to a faster optimization at testing time, and the resulting

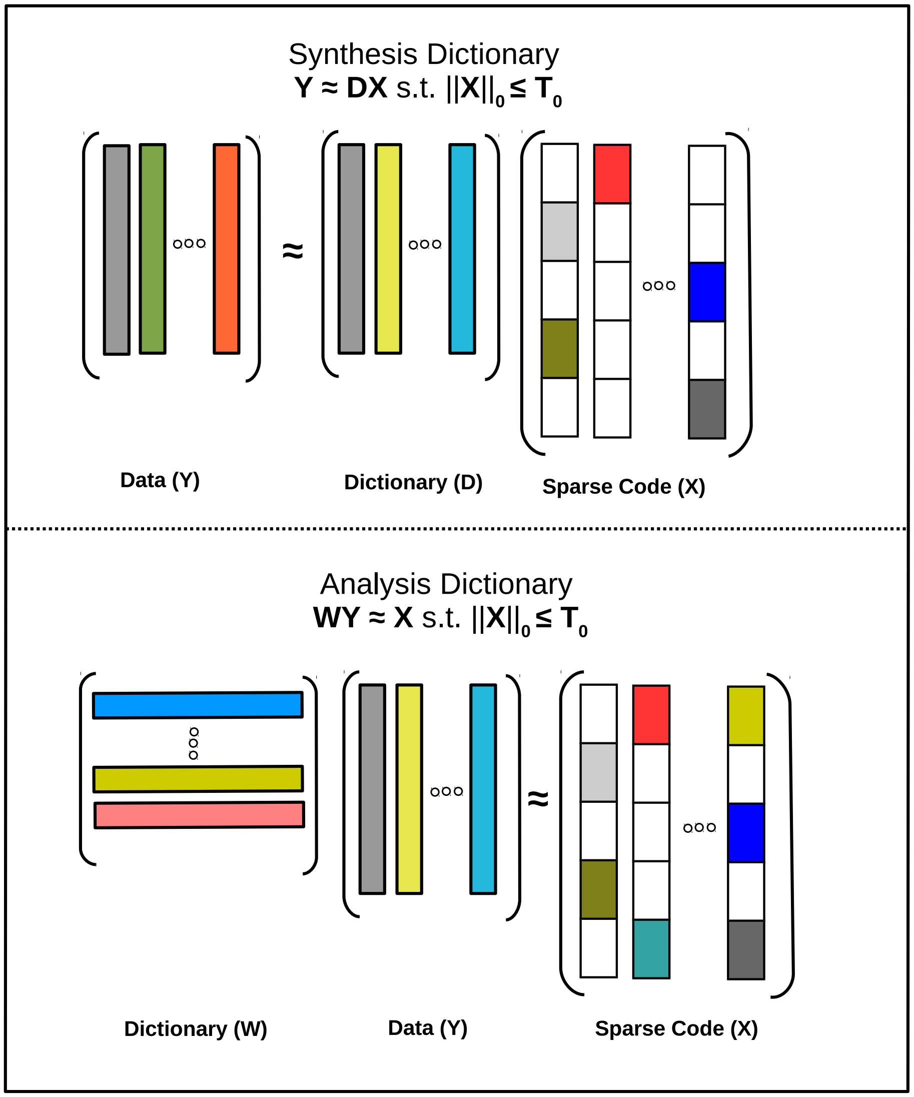


Figure 5.1: An overview of synthesis versus analysis models for sparse coding.

sparse codes are stable under noise and occlusion.

5.2 Organization

The chapter is organized in six sections. We review the related works in Section 5.3. The proposed formulation is described in Section 5.4 and the optimization scheme in Section 5.5. The classification procedure is described in Section 5.6 and experimental validations and results are presented in Section 5.7. Finally, we conclude the chapter in

Section 5.8.

5.3 Related Works

Analysis sparse coding models have only recently started receiving attention. A detailed analysis of analysis models was presented in [76]. An analysis K-SVD framework for learning the model was examined in [106]. Peleg *et al* [90] provided theoretical insights into the analysis model. Similarly, methods based on transform coding were proposed in [100, 101]. The idea behind transform coding is to learn transformation, instead of using off-the-shelf methods like DCT, FFT, etc, so that the resulting signal is sparse. These methods show similar performance as the previous analysis models, but have the added advantage of simpler gradient-based optimization and higher speed while testing. This work studies analysis model along the lines of transform coding method. However, we generalize it to different recognition scenarios.

5.4 Formulation

Given the data matrix, $\mathbf{Y} \in \mathbb{R}^{d\times N}$, whose columns represent d-dimensional training signals, in analysis dictionary framework [106], the objective is to learn $\mathbf{W} \in \mathbb{R}^{M\times d}$ which minimizes $\|\mathbf{WY}\|_0$. The optimization problem can be written as:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \ \|\mathbf{WY}\|_0 \text{ s.t. } \mathbf{W} \in \mathcal{A} \tag{5.1}$$

where, $\mathcal{A}$ is a set of constraints so that the problem is well regularized. However, the input samples can be noisy. In this case, the analysis model can be extended by expressing

$$\mathbf{Y} = \mathbf{X} + \mathbf{E}$$

where, $\mathbf{E}$ is noise and $\mathbf{WX}$ is sparse. This can be solved by the joint optimization problem:

$$\begin{aligned}\{\mathbf{W}^*, \mathbf{X}^*\} = \arg\min_{\mathbf{W},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{X}\|_F^2 \\ \text{s.t. } \|\mathbf{WX}\|_0 \leq T_0 \, , \mathbf{W} \in \mathcal{A}\end{aligned} \tag{5.2}$$

where, T_0 is the sparsity level. But, the transform coding framework [101] shows that handling the error in transformed domain as

$$\mathbf{WY} = \mathbf{X} + \mathbf{E}$$

is more general than (5.2). Hence, we solve the following optimization problem for analysis coding:

$$\begin{aligned} \{\mathbf{W}^*, \mathbf{X}^*\} = \underset{\mathbf{W},\mathbf{X}}{\arg\min} \; & \|\mathbf{WY} - \mathbf{X}\|_F^2 \\ \text{s.t. } & \|\mathbf{X}\|_0 \leq T_0 \, , \mathbf{W} \in \mathcal{A} \end{aligned} \tag{5.3}$$

To obtain a well-regularized solution, we constrain the set $\mathcal{A}$ to be matrices with row-wise norm to be unity. The unit norm condition is required to make the solution non-trivial. However, solving (5.3) with just these constraints may not lead to a well-conditioned solution. This is because the constraints presented above do not avoid the possibility of repeated rows or linearly dependent rows. To overcome these conditions, we add the following regularization terms to the criterion function:

$$R(\mathbf{W}) = \begin{cases} -\log(\det(\mathbf{W^TW})) & \text{if } m \geq d \\ -\log(\det(\mathbf{WW^T})) & \text{if } m < d \end{cases} \tag{5.4}$$

This regularization ensures that the learnt $\mathbf{W}$ has full column or row rank depending upon the matrix size. Further, the function is differentiable for cases where $\det(\mathbf{W^TW}) > 0$ or $\det(\mathbf{WW^T}) > 0$. Note that we consider both overcomplete and under-complete cases as both are common in recognition scenarios. Thus, the final optimization is given as:

$$\begin{aligned} \{\mathbf{W}^*, \mathbf{X}^*\} = \underset{\mathbf{W},\mathbf{X}}{\arg\min} \; & \|\mathbf{WY} - \mathbf{X}\|_F^2 + \lambda R(\mathbf{W}) \\ \text{s.t. } & \|\mathbf{w}_i\|_2 = 1 \; \forall \, i = 1, \cdots, M, \; \|\mathbf{X}\|_0 \leq T_0 \end{aligned} \tag{5.5}$$

where, $\mathbf{w}_i$ is the i^{th} row of dictionary matrix and $\lambda > 0$ is a hyperparameter. We now describe a strategy to solve the above optimization problem.

5.5 Optimization

The overall cost function is non-convex, however, we follow the strategy of alternate minimization to optimize the cost. This can be done in two steps:

- Update sparse code, $\mathbf{X}$: Fixing $\mathbf{W}$, the solution for $\mathbf{X}$ can be obtained by a simple thresholding. The optimal solution for $\mathbf{X}$ will be given by retaining the top T_0 coefficients in each column of $\mathbf{WY}$. We can also relaxed ℓ_0 constraint to ℓ_1 to make the problem convex. In this case, we can solve the following equivalent problem:

  $$\arg\min_{\mathbf{X}} \ \|\mathbf{WY} - \mathbf{X}\|_F^2 + \beta\|\mathbf{X}\|_1$$

  This can be solved by applying a soft thresholding scheme as follows:

  $$\mathbf{X}_{i,j} = \begin{cases} (\mathbf{WY})_{i,j} - \frac{\beta}{2} & \text{if } (\mathbf{WY})_{i,j} \geq \frac{\beta}{2} \\ (\mathbf{WY})_{i,j} + \frac{\beta}{2} & \text{if } (\mathbf{WY})_{i,j} < -\frac{\beta}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

- Update dictionary $\mathbf{W}$: Fixing $\mathbf{X}$, we now describe the update steps for $\mathbf{W}$. Even for a fixed $\mathbf{X}$, it is a non-convex problem. We solve the problem using conjugate gradient descent method [120] and then renormalizing the rows of $\mathbf{W}$ to unit norm. During the gradient descent, a small penalty of $\|\mathbf{W}\|_F^2$ can also be added to the cost term for stable solution [100]. The gradient of the function can be computed analytically and is given as:

  $$\nabla_{\mathbf{W}}(\|\mathbf{WY} - \mathbf{X}\|_F^2) = 2\mathbf{WYY^T} - 2\mathbf{YX^T} \tag{5.7}$$

  $$\nabla_{\mathbf{W}}(R(\mathbf{W})) = -2\mathbf{W}^{\dagger} \tag{5.8}$$

Thus, the optimization scheme is simple, and we found it to converge quickly during different experiments. A summary of the optimization scheme is given in Algorithm 7.

```
Input: Data $\mathbf{Y}$, sparsity level $T_0$, dictionary size $M$, $\lambda$, initial $\mathbf{W}$
Procedure:
Iterate till convergence,
    1. Update $\mathbf{X}$: Update $\mathbf{X}$ using thresholding method.
    2. Update $\mathbf{W}$: Perform conjugate gradient descent followed by renormalizing $\mathbf{W}$ row-wise.
Output: Analysis dictionary $\mathbf{W}$, sparse codes $\mathbf{X}$
```

Algorithm 7: Analysis Dictionary Learning (ADL)

5.5.1 Test Sparse Coding

At testing stage, given the test data $\mathbf{y_{te}}$ and trained dictionary $\mathbf{W_{tr}}$, the sparse code can be obtained by solving the optimization problem:

$$\mathbf{x}^*_{\mathbf{te}} = \arg\min_{\mathbf{x}} \ \|\mathbf{W_{tr}}\mathbf{y_{te}} - \mathbf{x}\|_F^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq T_0$$

This can be solved using the thresholding method described above. Hence, the encoding is efficient.

5.6 Classification

Given training samples from C classes $\{\mathbf{Y}_i\}_{i=1}^C$, we concatenate all the training samples to obtain a training matrix as follows

$$\mathbf{Y_{tr}} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_C] \in \mathbb{R}^{d\times N}.$$

We then apply Algorithm 7 to learn the analysis dictionary $\mathbf{W_{tr}}$. Note that we do not employ any discriminative cost while learning. Once $\mathbf{W_{tr}}$ is found, we apply (5.6) on the training data $\mathbf{Y_{tr}}$ and test data $\mathbf{Y_{te}}$ to obtain feature vectors $\mathbf{X_{tr}}$ and $\mathbf{X_{te}}$, respectively. Once the sparse codes are found, we train a Support Vector Machine (SVM) classifier on $\mathbf{X_{tr}}$ and test it on $\mathbf{X_{te}}$. The entire procedure for classification is summarized in Algorithm 8.

```
Input: Train Data $\mathbf{Y_{tr}}$, train label, $\ell_{\mathbf{tr}}$, test data $\mathbf{Y_{te}}$, $T_0$, $\lambda$, $M$.
Procedure:
1. Learn dictionary $\mathbf{W}$ from training data $\mathbf{Y_{tr}}$ and input parameters using Algorithm 7.
2. Obtain sparse codes $\mathbf{X_{tr}}$ and $\mathbf{X_{te}}$ using Eq. (5.6) and $\mathbf{W_{tr}}$.
3. Train SVM using $\mathbf{X_{tr}}$ and $\ell_{\mathbf{tr}}$ and test on $\mathbf{X_{te}}$.
Output: Test labels, $\ell_{\mathbf{te}}$
```

Algorithm 8: Classification using ADL.

5.7 Experiments

We conducted experiments on digit and face datasets to demonstrate the efficacy of the proposed method. We compare the proposed method with different synthesis based algorithms like SRC [135], K-SVD [2], discriminative K-SVD (DKSVD) [145], Fisher discriminant dictionary learning (FDDL) [140], supervised dictionary learning (SDL-G) [68] and incoherent dictionary learning [97]. Note that many of these algorithms use class-wise reconstruction error for classification. For a fair comparison, we report SVM-based classification for K-SVD [2] and FDDL [140] algorithms. The results for other methods are, however, reproduced as reported in literature.

5.7.1 USPS Digit Dataset

The USPS digit dataset [49] contains images of handwritten digits. The dataset is split into 7291 training and 2007 testing samples. We present results on recognition experiment as well as synthetic experiments to test robustness of the method to noise and missing pixels.

5.7.1.1 Convergence and Learnt Dictionary

Figure 5.2 shows the convergence of the optimization and learnt atoms of the dictionary. It can be seen that the cost converges smoothly. The output sparse codes also demonstrate that the learnt dictionary is meaningful, as there are few significant non-zero elements for each digit sample.

5.7.1.2 Overall Recognition

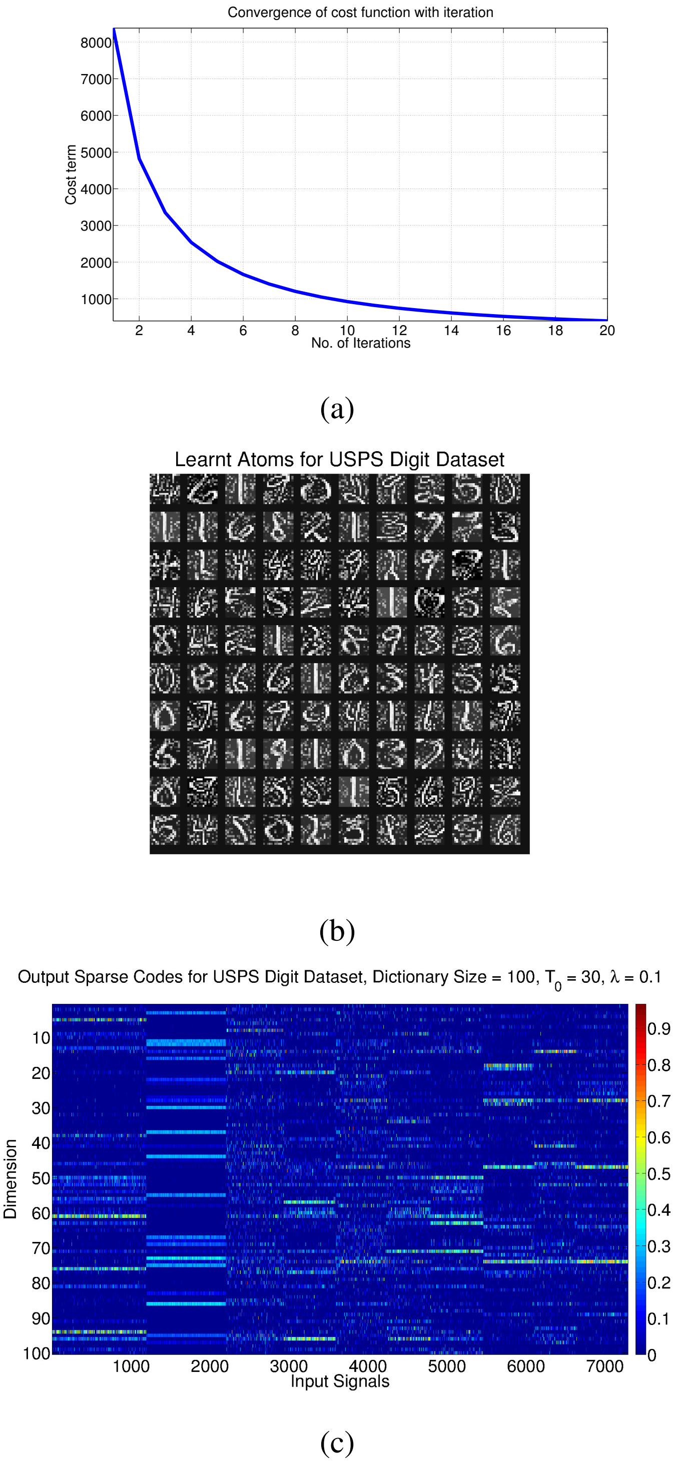
We then compared the recognition rate of proposed method with different synthesis dictionary-based algorithms. We trained an RBF-kernel based SVM classifier, tuning the parameters through cross-validation. The final result is reported for 900 atoms dictionary with $T_0 = 600, \lambda = 0.1$. It can be seen in Table 5.1 that the accuracy of the proposed method is comparable to other methods. In particular, the proposed method performs better than [68] and is comparable to [140] even though no discriminative cost has been used in training the method. Note that [97] uses reconstruction error for classification, hence, it is not directly comparable to the proposed method.

Method	**Recognition rate (%)**
ADL-SVM	94.5
KSVD-SVM [2]	92.1
FDDL-SVM [140]	94.7
SDL-G [68]	93.3
Ramirez *et al* [97]	96.0

Table 5.1: Recognition rates for USPS dataset.

5.7.1.3 Stability under noise and occlusion

We compare the stability of sparse codes generated by the proposed method to those generated by different synthesis coding methods, *viz.*, K-SVD [2] and FDDL [140]



(a)

(b)

(c)

Figure 5.2: (a) Convergence of the proposed analysis dictionary algorithm, (b) examples of the atoms learnt and (c) absolute value of output sparse codes produced by the algorithm.

under different distortions. In the first experiment, we added random Gaussian noise of increasing variance, and in the second experiment, we randomly set increasing percentage of pixels to zero. We compared the rank-one recognition rates of these methods using the NN-classifier. It can be seen from Figure 5.3 that the proposed method is more stable, esp. under addition of noise. Thus, analysis method are useful as often sparse codes are used as building blocks for recognition systems [14].

5.7.1.4 Encoding Speed

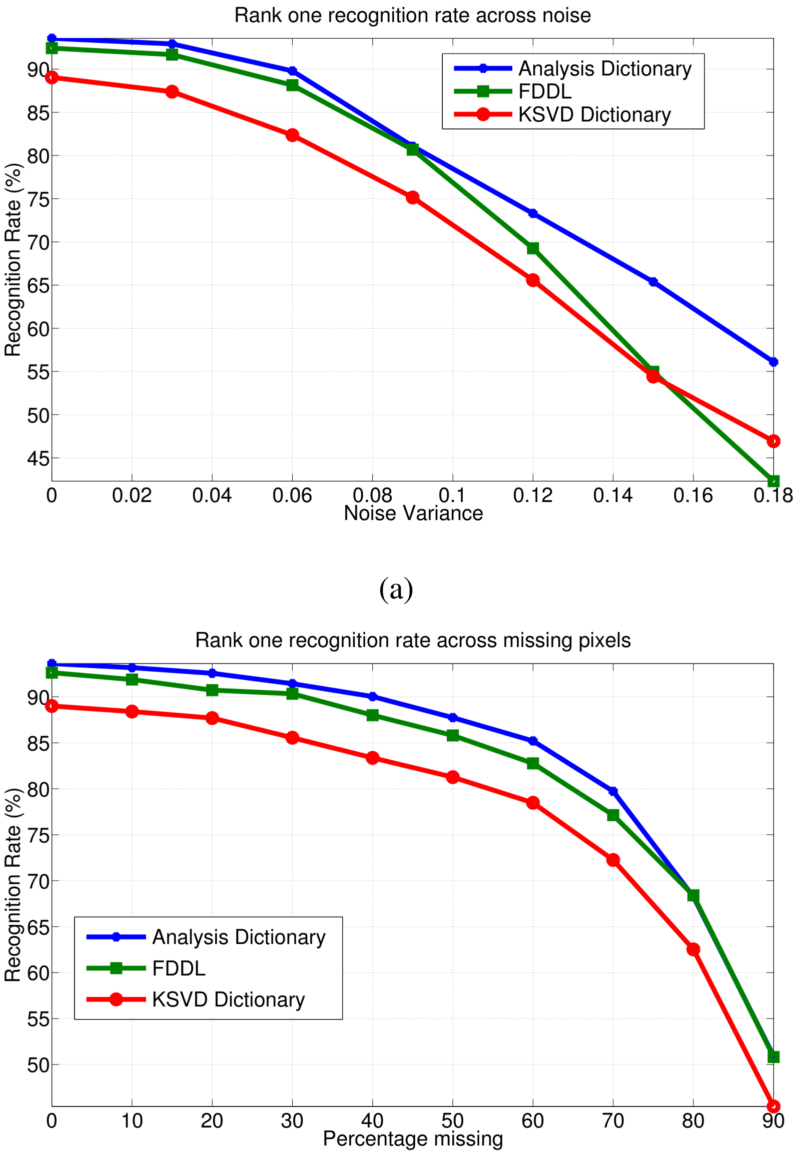
A significant advantage of the proposed approach over synthesis methods is the simple encoding scheme at test time. We compare the encoding time for the test images of the dataset with algorithms used in sparse coding in synthesis dictionaries, like OMP [89] and SPAMS [4]. Table 5.2 shows that the proposed ADL alogrithm is much faster than previous methods. All the tests were done on a 2.13 GHz Intel Xeon processor machine using Matlab programming interface.

Method	**Time (s)**
ADL	0.09
SPAMS [4]	0.15
OMP [89]	2.28

Table 5.2: Encoding speed for different methods for dictionary size $300, T_0 = 10$, number of samples = 2007.

5.7.2 AR Face Dataset

The AR face data set [69] consists of faces with varying illumination, expression, and occlusion conditions, captured in two sessions. We evaluated our algorithms on a set of 100 users. Images from the first session, seven for each subject, were used as training and the images from the second session, again seven per subject, were used for testing.



(a)

(b)

Figure 5.3: Stabiliy of different sparse coding algorithms under (a) noise, (b) missing pixels.

5.7.2.1 Recognition Comparison

Table 5.3 shows a comparison with different methods. The proposed method compares favorably with previously proposed synthesis sparse coding methods. Again it should be noted that SRC [135] uses reconstruction error for classification, and hence is not directly comparable. The proposed method however outperforms [145], which is a discriminative dictionary method.

Method	**Recognition rate (%)**
ADL-SVM	87.7
KSVD-SVM [2]	88.0
FDDL-SVM [140]	88.2
DKSVD [145]	85.4
SRC [135]	88.8

Table 5.3: Recognition rates for AR Face dataset.

5.7.2.2 Output Sparse Code

Figure 5.4 shows the output sparse codes for first 50 test samples. It can be seen that by exploiting the low-dimensional structure of face images, the proposed method is able to learn meaningful sparse codes.

5.8 Conclusion

We have demonstrated some applications of analysis sparse coding to image classification. The proposed approach compares favorably with previous synthesis sparse coding methods and is robust to noise and missing pixels. The method, further, has the advantage of simple encoding scheme at testing, thus, making it efficient.

In this chapter, we explored a basic formulation for analysis sparse coding. Future

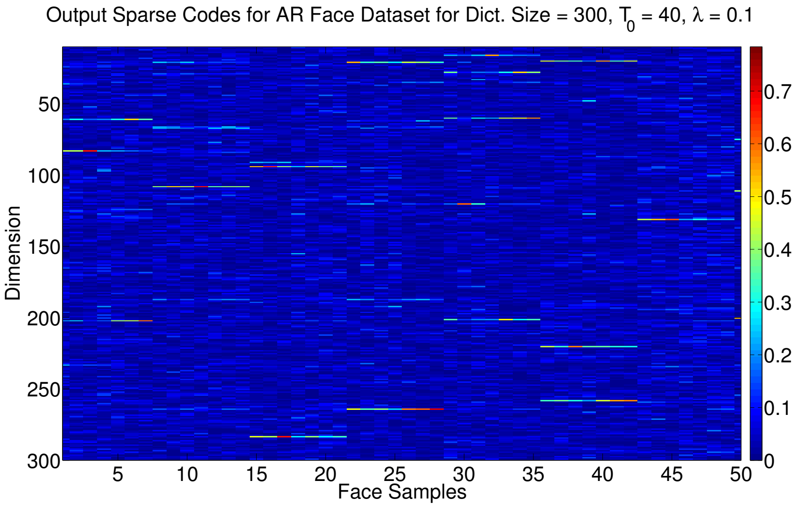


Figure 5.4: Output sparse codes produced by the proposed method on AR Face data.

directions include exploring discriminative methods as well as methods to handle to nonlinearity in data through kernel approaches. The method can also be extended for other vision tasks, like object detection, tracking, etc for which traditional sparse coding methods have been explored. The proposed method being efficient, looks promising for these applications that require both speed and accuracy.

Chapter 5: Summary and Future Directions

In this dissertation, we studied novel sparse coding approaches to different visual classification problems:

1. **Low resolution face recognition:** We studied the problem of face recognition at low resolutions. We proposed a synthesis-based approach for classifying the low resolution image, by exploiting 3D face models. A joint sparse coding framework, by sharing the sparse codes between high and low resolution training images, was described for robust recognition at low resolutions. We tested the method on different face datasets, and found the method to be superior than competing algorithms.

2. **Multimodal fusion:** We described a robust feature-level fusion method for multimodal biometric recognition. We extended the exisiting single modality sparse representation based classification scheme to multimodal fusion, using shared sparse codes across different modailities. Further we kernelized the algorithm, and proposed a quality measure to weigh different modalities at testing time. We demonstrated the effectiveness of proposed methods on a large multimodal dataset, fusion of weak modalities extracted from face image and robustness to noise and occlusion.

3. **Domain Adaptation** We considered the problem of adapting sparse representation, when the target data has distribution different from training. We described a technique which jointly learns projections of data in the two domains, and a latent dictionary which can succinctly represent both the domains in the projected low-dimensional space. The proposed method was efficient and performed on par or better than many competitive domain adaptation methods. We also showed the

application of the method to the challenging problem of heterogneous domain adaptation.

4. **Analysis Sparse Coding** Lastly, we described an analysis coding framework for image classification. We showed that the analysis coding framework gave similar performance as many of the synthesis sparse coding methods, while being much faster at test time.

Now, we describe possible future directions and extensions to the proposed methods.

1. **Robust Low Resolution Face Recognition** In chapter 2, we described a illumination-invariant low resolution face recognition algorithm. However, there can be other variations like noise, blur, pose, etc in the face images. Further, detection of faces at low resolutions itself is a big challenge. Also it is hard to align images at low resolutions. We will explore integrated approaches for detection and recognition at low resolutions, robust to alignment errors, blur and noise. We will also test the algorithm on more low resolution databases.

2. **Latent Sparse Fusion** As an extension of the robust feature-level fusion method, we propose to explore a latent space feature fusion method. The original method works in original feature space, and hence can be slow. Further, the original features may not be discriminative enough. We will explore a method of simultaneous projection onto a lower dimension space, while enforcing the joint sparsity criteria on the projections. Specifically, following the notation in Chapter 3, let $\{\mathbf{X}^i\}_{i=1}^D$ be the training data. We would like to learn projection matrices $\{\mathbf{P}^i\}_{i=1}^D$ for each modality, which reduce the feature dimension along-with maintaining joint sparsity property of projections:

$$\hat{\mathbf{\Gamma}}, \hat{\boldsymbol{P}}^i = \arg\min_{\mathbf{\Gamma}, \boldsymbol{P}^i} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{P}^i\mathbf{X}^i - \mathbf{P}^i\mathbf{X}^i\mathbf{\Gamma}^i\|_F^2 + \lambda\|\mathbf{\Gamma}\|_{1,q},$$
$$\text{s.t. } \mathbf{P}^i\mathbf{P}^{i,T} = I,\ diag(\mathbf{\Gamma}^i) = 0 \tag{5.1}$$

Here, the constraints of orthonormality of $\mathbf{P}^i$ and diagonal of $\mathbf{\Gamma}^i$ being zero help in avoiding the trivial solutions of null projection matrices and the test data being reconstructed by itself, respectively. Given the test data $\{\mathbf{Y}^i\}_{i=1}^{D}$, we use the projection matrices $\hat{\mathbf{P}}_i$ to project into lower dimension, and proceed as in Chapter 3. We will explore the effectiveness of this method for large scale multi-modal fusion, and speed and storage gains.

3. **Unsupervised Domain Adaptation** We plan to extend the proposed generalized domain-adaptive dictionaries to include unlabeled data in target domain. Other variations of the problem, like online adaptation can also be explored. We can also explore applications, like cross-modality matching problems. The problem here is using images captured by one sensor to recognize the test images captured by other sensors, like matching visible light images to infrared images, matching face images to sketches, etc.

4. **Efficient Feature Learning** We showed application of analysis sparse coding to efficient object recognition. We can extend the idea of analysis sparse coding for applications like detection, hierarchical feature learning and tracking to efficiently learn sparse codes. Traditional sparse coding approaches have been shown to give good performance for these tasks, however, they suffer due to slow sparse coding step. Analysis coding framework can be explored for these tasks to learn richer features as well as achieving efficiency in sparse coding.

Appendix A

Here, we will describe the kernel dictionary learning algorithm [129] and the framework for the proposed joint kernel dictionary learning algorithm (jointKerKSVD) as described in Chapter 2.

A1 Kernel Dictionary Learning

The optimization problem (2.9) can be solved in two stages.

A1.1 Sparse Coding

Here, $\mathbf{A}_i$ is kept fixed while searching for the optimal sparse code, $\mathbf{\Gamma}_i$. The cost term in (2.9) can be written as:

$$\|\boldsymbol{\phi}^L(\mathbf{X}_i^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\mathbf{\Gamma}_i\|_F^2 =$$

$$\sum_{j=1}^{m_i} \|\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_j\|_F^2,$$

where, $\boldsymbol{\gamma}_j$ is the sparse code for $\mathbf{x}_{i,j}^L$. Hence, the optimization problem can be broken up into m_i different sub-problems:

$$\arg\min_{\boldsymbol{\gamma}_j} \ \|\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_i\boldsymbol{\gamma}_j\|_F^2$$

$$\text{subject to} \|\boldsymbol{\gamma}_j\|_0 \leq T_0 \quad \forall\, j.$$

We can solve this using kernel orthogonal matching pursuit (KOMP). Let I_k denote the set of selected atoms at iteration k, $\hat{\mathbf{x}}_k$ denote the reconstruction of the signal, $\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L)$ using the selected atoms, $\mathbf{r}_k$ being the corresponding residue and $\boldsymbol{\gamma}_{j,k}$ the estimated sparse code at k^{th} iteration.

1. Start with $I_0 = \emptyset$, $\hat{\mathbf{x}}_k = 0$, $\boldsymbol{\gamma}_{j,k} = 0$.

2. Calculate the residue as:

$$\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L) = \boldsymbol{\phi}^L(\mathbf{X}_i^L)\hat{\mathbf{x}}_k + \mathbf{r}_k.$$

3. Project the residue on atoms not selected and add the atom with maximum projection value to I_k:

$$\begin{aligned}\tau_t &= (\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L) - \boldsymbol{\phi}^L(\mathbf{X}_i^L)\hat{\mathbf{x}}_k)^T(\mathbf{X}_i^L\mathbf{a}_t)\\ &= (\boldsymbol{\mathcal{K}}^L(\mathbf{x}_{i,j}^L, \mathbf{X}_i^L) - \hat{\mathbf{x}}_k^T\boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L, \mathbf{X}_i^L))\mathbf{a}_t,\ t \notin I_k. \qquad (1)\end{aligned}$$

   Update the set I_k as:

$$I_{k+1} = I_k \cup \underset{t \notin I_k}{\arg\max}\ |\tau_t|. \qquad (2)$$

4. Update the sparse code, $\boldsymbol{\gamma}_{k+1}$ and reconstruction, $\hat{\mathbf{x}}_{k+1}$ as:

$$\begin{aligned}\boldsymbol{\gamma}_{j,k+1} &= ((\boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_{I_{k+1}})^T(\boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_{I_{k+1}}))^{-1}\\ &\quad(\boldsymbol{\phi}^L(\mathbf{X}_i^L)\mathbf{A}_{I_{k+1}})^T\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L)\\ &\quad= (\mathbf{A}_{I_{k+1}}^T\boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L, \mathbf{X}_i^L)\mathbf{A}_{I_{k+1}})^{-1}\\ &\quad(\boldsymbol{\mathcal{K}}^L(\mathbf{x}_{i,j}^L, \mathbf{X}_i^L)\mathbf{A}_{I_{k+1}})^T, \qquad (3)\\ \hat{\mathbf{x}}_{k+1} &= \mathbf{A}_{I_{k+1}}\boldsymbol{\gamma}_{j,k+1}. \qquad (4)\end{aligned}$$

5. $k \leftarrow k+1$; Repeat steps 2-4 T_0 times.

A1.2 Dictionary update

Once the sparse codes are calculated, the dictionary $\mathbf{A}_i$ can be updated using kernel K-SVD or MOD methods as described in [129]. Here, we use the MOD to update the dictionary as follows:

$$\mathbf{A}_i = \mathbf{\Gamma}_i^T(\mathbf{\Gamma}_i\mathbf{\Gamma}_i^T)^{-1}.$$

The dictionary atoms are now normalized to unit norm in feature space:

$$\mathbf{A}_{i,j} = \frac{\mathbf{A}_{i,j}}{\sqrt{\mathbf{A}_{i,j}^T\boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L, \mathbf{X}_i^L)\mathbf{A}_{i,j}}},\ j = 1, \cdots, K.$$

A2 Joint kernel dictionary learning

The optimization problem (2.14) can be solved in a similar way as the kernel dictionary learning problem in two alterative steps:

A2.1 Sparse Coding

Here, we keep $\mathbf{A}_i^H$ and $\mathbf{A}_i^L$ fixed and learn the joint sparse code $\mathbf{\Gamma}_i$. The cost term in (2.15) can be written as:

$$\|\mathbf{\Phi}_1(\mathbf{X}_i^H, \mathbf{X}_i^L) - \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\tilde{\mathbf{A}}_i\mathbf{\Gamma}_i\|_F^2 =$$
$$\sum_{j=1}^{m_i} \|\mathbf{\Phi}_1(\mathbf{X}_{i,j}^H, \mathbf{X}_{i,j}^L) - \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\tilde{\mathbf{A}}_i\boldsymbol{\gamma}_j\|_F^2,$$

where, $\boldsymbol{\gamma}_j$ is the sparse code for $\mathbf{x}_{i,j}^L$. Thus, the optimization can be broken up into m_i sub-problems:

$$\arg\min_{\boldsymbol{\gamma}_j}\ \|\mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L) - \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\tilde{\mathbf{A}}_i\boldsymbol{\gamma}_j\|_F^2$$
$$\text{subject to}\|\boldsymbol{\gamma}_j\|_0 \leq T_0 \quad \forall\, j.$$

This is similar to the original kernel dictionary learning formulation, with the signal $\boldsymbol{\phi}^L(\mathbf{x}_{i,j}^L)$ replaced by $\mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L)$. Thus, the above problem can be solved using similar procedure as KOMP. Let I_k denote the set of selected atoms at iteration k, $\hat{\mathbf{x}}_k^{H,L}$ denote the reconstruction of the signal, $\mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L)$ using the selected atoms, $\mathbf{r}_k$ being the corresponding residue and $\boldsymbol{\gamma}_{j,k}$ the estimated sparse code at k^{th} iteration.

1. Start with $I_0 = \emptyset$, $\hat{\mathbf{x}}_k^{H,L} = 0$, $\boldsymbol{\gamma}_{j,k} = 0$.

2. Calculate the residue as:

$$\mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L) = \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\hat{\mathbf{x}}_k^{H,L} + \mathbf{r}_k.$$

3. Project the residue on atoms not selected and add the atom with maximum projection value to I_k:

$$\begin{aligned}\tau_t &= (\mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L) - \mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\hat{\mathbf{x}}_k^{H,L})^T \\ &\quad (\mathbf{\Phi}_2(\mathbf{X}_i^H, \mathbf{X}_i^L)\mathbf{a}_t) \\ &= (\boldsymbol{\mathcal{K}}^1 - (\hat{\mathbf{x}}_k^{H,L})^T \boldsymbol{\mathcal{K}}^2)\tilde{\mathbf{a}}_t,\ t \notin I_k, \end{aligned} \tag{5}$$

where,

$$\begin{aligned}\boldsymbol{\mathcal{K}}^1 &= \mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L)^T \mathbf{\Phi}_1(\mathbf{X}_{i,j}^H, \mathbf{X}_{i,j}^L) \\ &= \begin{bmatrix} \boldsymbol{\mathcal{K}}_H \\ \lambda \boldsymbol{\mathcal{K}}_L \end{bmatrix},\end{aligned}$$

and,

$$\begin{aligned}\boldsymbol{\mathcal{K}}^2 &= \mathbf{\Phi}_2(\mathbf{X}_{i,j}^H, \mathbf{x}_{i,j}^L)^T \mathbf{\Phi}_2(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L) \\ &= \begin{bmatrix} \boldsymbol{\mathcal{K}}_H & \mathbf{0} \\ \mathbf{0} & \lambda \boldsymbol{\mathcal{K}}_L \end{bmatrix}.\end{aligned}$$

Update the set I_k as:

$$I_{k+1} = I_k \cup \underset{t \notin I_k}{\arg\max}\ |\tau_t|.$$

4. Update the sparse code, $\boldsymbol{\gamma}_{j,k+1}$ and reconstruction, $\hat{\mathbf{x}}_{k+1}^{H,L}$ as:

$$\begin{aligned}\boldsymbol{\gamma}_{k+1} &= ((\mathbf{\Phi}_2(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L)\tilde{\mathbf{A}}_{I_{k+1}})^T (\mathbf{\Phi}_2(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L)\tilde{\mathbf{A}}_{I_{k+1}}))^{-1} \\ &\quad (\mathbf{\Phi}_2(\mathbf{X}_{i,j}^H, \mathbf{x}_{i,j}^L)\tilde{\mathbf{A}}_{I_{k+1}})^T \mathbf{\Phi}_1(\mathbf{x}_{i,j}^H, \mathbf{x}_{i,j}^L) \\ &= (\tilde{\mathbf{A}}_{I_{k+1}}^T \boldsymbol{\mathcal{K}}^2 \tilde{\mathbf{A}}_{I_{k+1}})^{-1} (\boldsymbol{\mathcal{K}}^1 \tilde{\mathbf{A}}_{I_{k+1}})^T, \end{aligned} \tag{6}$$

$$\hat{\mathbf{x}}_{k+1}^{H,L} = \tilde{\mathbf{A}}_{I_{k+1}} \boldsymbol{\gamma}_{j,k+1}. \tag{7}$$

5. $k \leftarrow k+1$; Repeat steps 2-4 T_0 times.

A2.2 Dictionary update

The dictionaries $\mathbf{A}_i^H$ and $\mathbf{A}_i^L$ can now be obtained using the MOD method as follows:

$$\mathbf{A}_i^H = \mathbf{\Gamma}_i^T(\mathbf{\Gamma}_i\mathbf{\Gamma}_i^T)^{-1},$$

$$\mathbf{A}_i^L = \mathbf{\Gamma}_i^T(\mathbf{\Gamma}_i\mathbf{\Gamma}_i^T)^{-1}.$$

Further the dictionary atoms are normalized to unit norm in feature space:

$$\mathbf{A}_{i,j}^H = \frac{\mathbf{A}_{i,j}^H}{\sqrt{(\mathbf{A}_{i,j}^H)^T\boldsymbol{\mathcal{K}}^H(\mathbf{X}_i^H,\mathbf{X}_i^H)\mathbf{A}_{i,j}^H}},\ j = 1,\cdots,K,$$

$$\mathbf{A}_{i,j}^L = \frac{\mathbf{A}_{i,j}^L}{\sqrt{(\mathbf{A}_{i,j}^L)^T\boldsymbol{\mathcal{K}}^L(\mathbf{X}_i^L,\mathbf{X}_i^L)\mathbf{A}_{i,j}^L}},\ j = 1,\cdots,K.$$

Appendix B

Here, we demonstrate proofs for Propositions 1 and 2 in Chapter 4.

The optimization problem (4.6) is given as:

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \underset{\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}}{\arg\min}\ \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j, \tag{1}$$

where,

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 +$$
$$\nu \|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2, \tag{2}$$
$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T). \tag{3}$$

Then the Proposition 1 is given as:

Proposition 1: *There exists an optimal solution* $\mathbf{P_1^*}, \cdots, \mathbf{P_M^*}, \mathbf{D}^*$ *to equation (4.6), which has the following form:*

$$\mathbf{P_i^*} = (\mathbf{Y_i A_i})^{\mathbf{T}}\ \forall\, i = 1, \cdots, M, \tag{4}$$
$$\mathbf{D}^* = \tilde{\mathbf{P}}^* \tilde{\mathbf{Y}} \tilde{\mathbf{B}}, \tag{5}$$

Proof:

Form for $\mathbf{D}^*$: First we will show the form for $\mathbf{D}^*$. We can decompose $\mathbf{D}^*$ into two orthogonal components as follows:

$$\mathbf{D}^* = \mathbf{D}_{\parallel} + \mathbf{D}_{\perp} \tag{6}$$

$$\text{where, } \mathbf{D}_{\parallel} = (\tilde{\mathbf{P}}\tilde{\mathbf{Y}})\tilde{\mathbf{B}},\ \mathbf{D}_{\perp}^{\mathbf{T}}(\tilde{\mathbf{P}}\tilde{\mathbf{Y}}) = \mathbf{0}, \tag{7}$$

for some $\mathbf{B} \in \mathbb{R}^{\sum_{i=1}^{M} N_i \times K}$. Substituting the value of $\mathbf{D}^*$ into the value of $\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}})$, we get for the three terms of $\mathcal{C}_1$, ignoring the multiplicative constants μ, ν:

$$\begin{aligned}
&\textbf{First Term} = \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}})^T(\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}))\\
&= \text{trace}(\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}} + \tilde{\mathbf{X}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}} +\\
&\tilde{\mathbf{X}}^{\mathbf{T}}\mathbf{D}_{\perp}^{\mathbf{T}}\mathbf{D}_{\perp}\tilde{\mathbf{X}})\\
&\geq \text{trace}(\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}} + \tilde{\mathbf{X}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}).
\end{aligned} \tag{8}$$

$$\begin{aligned}
&\textbf{Second Term} = \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}})^T(\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}))\\
&= \text{trace}(\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{in}} + \tilde{\mathbf{X}}_{\mathbf{in}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{in}} +\\
&\tilde{\mathbf{X}}_{\mathbf{in}}^{\mathbf{T}}\mathbf{D}_{\perp}^{\mathbf{T}}\mathbf{D}_{\perp}\tilde{\mathbf{X}}_{\mathbf{in}})\\
&\geq \text{trace}(\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{in}} + \tilde{\mathbf{X}}_{\mathbf{in}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{in}}).
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\textbf{Third Term} = \text{trace}(\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}}(\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}))\\
&= \text{trace}(\tilde{\mathbf{X}}_{\mathbf{out}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{out}} + \tilde{\mathbf{X}}_{\mathbf{out}}^{\mathbf{T}}\mathbf{D}_{\perp}^{\mathbf{T}}\mathbf{D}_{\perp}\tilde{\mathbf{X}}_{\mathbf{out}})\\
&\geq \text{trace}(\tilde{\mathbf{X}}_{\mathbf{out}}^{\mathbf{T}}\mathbf{D}_{\parallel}^{\mathbf{T}}\mathbf{D}_{\parallel}\tilde{\mathbf{X}}_{\mathbf{out}}).
\end{aligned} \tag{10}$$

The equality is reached when $\mathbf{D}_{\perp} = \mathbf{0}$. Hence, the form of $\mathbf{D}^*$ is:

$$\mathbf{D}^* = \tilde{\mathbf{P}}\tilde{\mathbf{Y}}\tilde{\mathbf{B}}.$$

Form for $\mathbf{P}_{\mathbf{i}}^*$: For each $i = 1, \cdots, M$, $\mathbf{P}_{\mathbf{i}}^*$ can be decomposed as:

$$\mathbf{P}_{\mathbf{i}}^* = \mathbf{P}_{\parallel,\mathbf{i}} + \mathbf{P}_{\perp,\mathbf{i}} \tag{11}$$

$$\text{where, } \mathbf{P}_{\parallel,\mathbf{i}} = (\mathbf{Y}_{\mathbf{i}}\mathbf{A}_{\mathbf{i}})^{\mathbf{T}}, \mathbf{P}_{\perp,\mathbf{i}}\mathbf{Y}_{\mathbf{i}} = \mathbf{0}. \tag{12}$$

Let $\tilde{\mathbf{P}}_{\|} = [\mathbf{P}_{\|,\mathbf{1}}, \cdots, \mathbf{P}_{\|,\mathbf{M}}]$ and $\tilde{\mathbf{P}}_{\perp} = [\mathbf{P}_{\perp,\mathbf{1}}, \cdots, \mathbf{P}_{\perp,\mathbf{M}}]$. Substituting the value for $\mathbf{D}^*$ into cost terms, we can write the terms of $\mathcal{C}_1$ as:

$$\begin{aligned}\textbf{First Term} &= \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_{\|}+\tilde{\mathbf{P}}_{\perp})\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}}\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}_{\|}^{\mathbf{T}}).\end{aligned} \tag{13}$$

$$\begin{aligned}\textbf{Second Term} &= \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_{\|}+\tilde{\mathbf{P}}_{\perp})\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}}\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}_{\|}^{\mathbf{T}}).\end{aligned} \tag{14}$$

$$\begin{aligned}\textbf{Third Term} &= \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_{\|}+\tilde{\mathbf{P}}_{\perp})\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}}\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}_{\|}^{\mathbf{T}}).\end{aligned} \tag{15}$$

The cost term, $\mathcal{C}_2$ can be written as:

$$\begin{aligned}&\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T)\\ &= -\text{trace}(((\tilde{\mathbf{P}}_{\|}+\tilde{\mathbf{P}}_{\perp})\tilde{\mathbf{Y}})((\tilde{\mathbf{P}}_{\|}+\tilde{\mathbf{P}}_{\perp})\tilde{\mathbf{Y}})^T)\\ &= -\text{trace}((\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}})^T).\end{aligned} \tag{16}$$

Putting all the terms together, the overall objective function becomes:

$$\begin{aligned}&\text{trace}(\tilde{\mathbf{P}}_{\|}\tilde{\mathbf{Y}}((\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}}+\mu(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\\ &(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}}+\nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}}-\lambda\mathbf{I})\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{P}}_{\|}^{\mathbf{T}})\\ &= \text{trace}(\tilde{\mathbf{A}}_{\mathbf{T}}\tilde{\mathbf{K}}((\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}}+\mu(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\\ &(\mathbf{I}-\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}}+\nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}}-\lambda\mathbf{I})\tilde{\mathbf{K}}\tilde{\mathbf{A}}).\end{aligned} \tag{17}$$

It can be seen that from (17), that the cost function is independent of $\mathbf{P}_{\perp,\mathbf{i}}$, hence it can be safely set to be $\mathbf{0}$. Hence,

$$\mathbf{P}_\mathbf{i}^* = (\mathbf{Y}_\mathbf{i}\mathbf{A}_\mathbf{i})^\mathbf{T}.$$

Updating $\tilde{\mathbf{A}}$

Using Proposition 1, optimization problem equation (4.6) becomes:

$$\begin{aligned}\{\tilde{\mathbf{A}}^*, \tilde{\mathbf{B}}^*, \mathbf{X}^*\} &= \arg\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}},\tilde{\mathbf{X}}} \ \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) + \lambda\mathcal{C}_2(\tilde{\mathbf{A}}) \\ \text{s.t. } \mathbf{A}_\mathbf{i}^\mathbf{T}\mathbf{K}_\mathbf{i}\mathbf{A}_\mathbf{i} = \mathbf{I},\ & i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j. \end{aligned} \tag{18}$$

Here, we assume that $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed. Then, the optimization for $\tilde{\mathbf{A}}$ can be solved efficiently. We have the following proposition.

Proposition 2: *The optimal solution of equation (4.15) when $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed is:*

$$\begin{aligned}\{\mathbf{G}^*\} &= \arg\min_{\mathbf{G}} \ \text{trace}[\mathbf{G}^\mathbf{T}\mathbf{H}\mathbf{G}] \\ \textit{s.t. } & \mathbf{G}_\mathbf{i}^\mathbf{T}\mathbf{G}_\mathbf{i} = \mathbf{I}\ \forall\ i\ = 1, \cdots, M,\end{aligned} \tag{19}$$

where,

$$\begin{aligned}\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^\mathbf{T}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in}) \\ (\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})^\mathbf{T} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})^\mathbf{T} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}}.\end{aligned} \tag{20}$$

Proof:

Let,

$$\tilde{\mathbf{K}} = \mathbf{V}\mathbf{S}\mathbf{V}^\mathbf{T},$$

$$\begin{aligned}&\tilde{\mathbf{H}} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^\mathbf{T}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T} + \\ &\mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{in})^\mathbf{T} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_\mathbf{out})^\mathbf{T} \\ &-\lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}},\end{aligned}$$

and

$$\mathbf{G} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}\tilde{\mathbf{A}}.$$

Substituting into (17), we get the required form of the optimization.

Bibliography

[1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20:681–695, March 2011.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[3] O. Arandjelovic and R. Cipolla. Face recognition from video using the generic shape-illumination manifold. *European Conference on Computer Vision*, pages IV: 27–40, 2006.

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[5] F. Bach, J. Mairal, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

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

[7] M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. *IEEE International Conference on Computer Vision*, pages 769–776, Dec 2013.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[9] S. Biswas, G. Aggarwal, and R. Chellappa. Robust estimation of albedo for illumination-invariant matching and shape recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):884–899, March 2009.

[10] S. Biswas, G. Aggarwal, P. J. Flynn, and K. W. Bowyer. Pose-robust recognition of low-resolution face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):3037–3049, 2013.

[11] S. Biswas, K. Bowyer, and P. Flynn. Multidimensional scaling for matching low-resolution facial images. *IEEE International Conference on Biometrics: Theory Applications and Systems*, pages 1–6, September 2010.

[12] S. Biswas, K. W. Bowyer, and P. J. Flynn. Multidimensional scaling for matching low-resolution face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2019–2030, Oct 2012.

[13] V. Blanz and T. Vetter. Face recognition based on fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1063–1074, 2003.

[14] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. *Advances in Neural Information Processing Systems*, pages 2115–2123, 2011.

[15] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior. The relation between the ROC curve and the CMC. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 15–20, 2005.

[16] M. J. Brooks and B. K. P. Horn. *Shape from Shading*. MIT Press, Cambridge, MA, 1989.

[17] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, June 1998.

[18] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58:1–37, May 2011.

[19] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986.

[20] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:275 – 282, June 2004.

[21] M. Chen, Z. Xu, F. Sha, and K. Q. Weinberger. Marginalized denoising autoencoders for domain adaptation. *International Conference on Machine Learning*, pages 767–774, 2012.

[22] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. *IEEE International Conference on Computer Vision*, pages 2439–2446, Barcelona, Spain, Nov. 2011.

[23] J.-Y. Choi, Y.-M. Ro, and K. Plataniotis. Color face recognition for degraded face images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 39(5):1217–1230, Oct 2009.

[24] S. Chopra, S. Balakrishnan, and R. Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. *ICML Workshop on Challenges in Representation Learning*, 2013.

[25] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, Jan. 2004.

[26] H. Daumé III. Frustratingly easy domain adaptation. *ACL*, 2007.

[27] T. Diethe, D. Hardoon, and J. Shawe-Taylor. Constructing nonlinear discriminants from multiple data views. *Machine Learning and Knowledge Discovery in Databases*, pages 328–343, 2010.

[28] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[29] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua. Domain adaptation from multiple sources via auxiliary classifiers. *International Conference on Machine Learning*, pages 289–296, 2009.

[30] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1338–1345, June 2012.

[31] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec 2006.

[32] E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1873–1879, Colorado Springs, USA, June 2011.

[33] J. Farquhar, H. Meng, S. Szedmak, D. Hardoon, and J. Shawe-taylor. Two view learning: SVM-2k, theory and practice. *Advances in Neural Information Processing Systems*, Vancouver, Dec. 2006.

[34] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, et al. Unsupervised visual domain adaptation using subspace alignment. *IEEE International Conference on Computer Vision*, 2013.

[35] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22:56–65, 2002.

[36] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[37] I. Gkioulekas and T. Zickler. Dimensionality reduction using the sparse linear model. *Advances in Neural Information Processing Systems*, pages 271–279, 2011.

[38] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[39] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. *International Conference on Machine Learning*, pages 222–230, 2013.

[40] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, June 2012.

[41] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. *IEEE International Conference on Computer Vision*, 2011.

[42] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shift by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[43] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[44] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image Vision Computing*, 28(5):807–813, 2010.

[45] B. Gunturk, A. Batur, Y. Altunbasak, I. Hayes, M.H., and R. Mersereau. Eigenface-domain super-resolution for face recognition. *Image Processing, IEEE Transactions on*, 12(5):597–606, May 2003.

[46] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang. Sparse unsupervised dimensionality reduction for multiple view data. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(10):1485–1496, Oct. 2012.

[47] P. Hennings-Yeomans, S. Baker, and B. Kumar. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[48] K. Hotta, T. Kurita, and T. Mishima. Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 70 – 75, April 1998.

[49] J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[50] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9:846–859, May 2000.

[51] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2175, June 2012.

[52] K. Jia and S. Gong. Multi-modal tensor face for simultaneous super-resolution and recognition. *IEEE International Conference on Computer Vision*, 2:1683 – 1690, October 2005.

[53] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. *Neural Information Processing Systems*, pages 982–990, 2010.

[54] S. Kakade and D. Foster. Multi-view regression via canonical correlation analysis. *Learning Theory*, pages 82–96, 2007.

[55] S. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel Fisher discriminant analysis. *23rd International Conference on Machine Learning*, pages 465–472, Pittsburgh, USA, June 2006.

[56] A. Klausner, A. Tengg, and B. Rinner. Vehicle classification on multi-sensor smart cameras using feature- and decision-fusion. *IEEE Conference on Distributed Smart Cameras*, pages 67–74, Vienna, Austria, Sept. 2007.

[57] E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011.

[58] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:957–968, June 2005.

[59] P. Krishnasamy, S. Belongie, and D. Kriegman. Wet fingerprint recognition: Challenges and opportunities. *International Joint Conference on Biometrics*, pages 1–7, Washington DC, USA, Oct. 2011.

[60] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1785–1792, June 2011.

[61] K.-C. Lee, J. Ho, and D. J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:684–698, 2005.

[62] S. Lee, J. Park, and S. Lee. Low resolution face recognition based on support vector data description. *Pattern Recognition*, 39(9):1809–1812, 2006.

[63] Z. Lei, S. Liao, A. Jain, and S. Li. Coupled discriminant analysis for heterogeneous face recognition. *IEEE Transactions on Information Forensics and Security*, 7(6):1707–1716, Dec 2012.

[64] B. Li, H. Chang, S. Shan, and X. Chen. Low-resolution face recognition via coupled locality preserving mappings. *IEEE Signal Processing Letters*, 17(1):20–23, January 2010.

[65] H. Li, K.-A. Toh, and L. Li. *Advanced Topics In Biometrics*. World Scientific Publishing Co. Pte. Ltd., 2012.

[66] W. Li, L. Duan, D. Xu, and I. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, June 2014.

[67] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, April 2012.

[68] J. Mairal, F. Bach, J. Ponce, A. Zisserman, and G. Sapiro. Supervised dictionary learning. *Advances in Neural Information Processing Systems*, 2008.

[69] A. M. Martinez and R. Benavente. The AR face database. Technical report, Ohio State University, 1998.

[70] L. Masek and P. Kovesi. MATLAB source code for biometric identification system based on iris patterns. Technical report, The University of Western Australia, 2003.

[71] G. Medioni, J. Choi, C.-H. Kuo, A. Choudhury, L. Zhang, and D. Fidaleo. Noncooperative persons identification at a distance with 3D face modeling. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, September 2007.

[72] L. Meier, S. V. D. Geer, and P. Bhlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B*, 70:53–71, Feb. 2008.

[73] V. Monga, N. Damera-Venkata, H. Rehman, and B. Evans. Halftoning matlab toolbox, 2005.

[74] A. Moorhouse, A. Evans, G. Atkinson, J. Sun, and M. Smith. The nose on your face may not be so plain: Using the nose as a biometric. *International Conference on Crime Detection and Prevention*, pages 1–6, London, UK, Dec. 2009.

[75] P. Nagesh and B. Li. A compressive sensing approach for expression-invariant face recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1518–1525, Miami, USA, June 2009.

[76] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. The co-sparse analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56, 2013.

[77] K. Nandakumar, Y. Chen, S. Dass, and A. Jain. Likelihood ratio-based biometric score fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:342–347, Feb. 2008.

[78] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Sparse embedding: A framework for sparsity promoting dimensionality reduction. *European Conference on Computer Vision*, pages 414–427, Oct. 2012.

[79] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, 2013.

[80] N. H. Nguyen, N. M. Nasrabadi, and T. D. Tran. Robust multi-sensor classification via joint sparse representation. *International Conference on Information Fusion*, pages 1–8, Chicago, USA, July 2011.

[81] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 692–699, June 2013.

[82] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[83] U. Park, R. Jillela, A. Ross, and A. Jain. Periocular biometrics in the visible spectrum. *IEEE Transactions on Information Forensics and Security*, 6:96–106, March 2011.

[84] V. M. Patel and R. Chellappa. Sparse representations, compressive sensing and dictionaries for pattern recognition. *Asian Conference on Pattern Recognition*, 2010.

[85] V. M. Patel and R. Chellappa. *Sparse representations and compressive sensing for imaging and vision*. SpringerBriefs, 2013.

[86] V. M. Patel, R. Chellappa, and M. Tistarelli. Sparse representations and random projections for robust and cancelable biometrics. *International Conference on Control, Automation, Robotics and Vision*, pages 1–6, Guangzhou, China, Dec. 2010.

[87] V. M. Patel, Y.-C. Chen, R. Chellappa, and P. J. Phillips. Dictionaries for image and video-based face recognition. *J. Opt. Soc. Am. A*, 31(5):1090–1103, May 2014.

[88] V. M. Patel, T. Wu, S. Biswas, P. Phillips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7:954–965, June 2012.

[89] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Asilomar Conference on Signals, Systems and Computers*, 1993.

[90] T. Peleg and M. Elad. Performance guarantees of the thresholding algorithm for the co-sparse analysis model. *IEEE Transactions on Information Theory*, 59(3):1832–1845, 2013.

[91] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:947–954, June 2005.

[92] J. Pillai, A. Shrivastava, V. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. *IEEE Conference on Image Processing*, Oct. 2012.

[93] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1877–1893, Sept. 2011.

[94] S. Pundlik, D. Woodard, and S. Birchfield. Non-ideal iris segmentation using graph cuts. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, Anchorage, USA, June 2008.

[95] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. *European Conference on Computer Vision*, 2012.

[96] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[97] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3508, June 2010.

[98] H. Rara, S. Elhabian, A. Ali, M. Miller, T. Starr, and A. Farag. Distant face recognition based on sparse-stereo reconstruction. *IEEE International Conference on Image Processing*, pages 4141–4144, November 2009.

[99] A. Rattani, D. Kisku, M. Bicego, and M. Tistarelli. Feature level fusion of face and fingerprint biometrics. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, Washington DC, USA, Sept. 2007.

[100] S. Ravishankar and Y. Bresler. Learning overcomplete sparsifying transforms for signal processing. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3088–3092, 2013.

[101] S. Ravishankar and Y. Bresler. Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, 61(5):1072–1086, 2013.

[102] C.-X. Ren, D.-Q. Dai, and H. Yan. Coupled kernel embedding for low-resolution face image recognition. *IEEE Transactions on Image Processing*, 21(8):3770–3783, Aug 2012.

[103] A. Ross and A. K. Jain. Multimodal biometrics: an overview. *Proc. European Signal Processing Conference*, pages 1221–1224, Vienna, Austria, Sept. 2004.

[104] A. Ross, K. Nandakumar, and A. K. Jain. *Handbook of Multibiometrics*. Springer, 2006.

[105] A. A. Ross and R. Govindarajan. Feature level fusion of hand and face biometrics. *Proc. of the SPIE*, 5779:196–204, Orlando, USA, Mar. 2005.

[106] R. Rubinstein, T. Peleg, and M. Elad. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2013.

[107] C. W. S. Chikkerur and V. Govindaraju. A systematic approach for feature extraction in fingerprint images. *International Conference on Bioinformatics and its Applications*, pages 344–350, Fort Lauderadale, USA, Dec. 2004.

[108] S. S. S. Crihalmeanu, A. Ross and L. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. *In Technical Report, WVU, Lane Department of Computer Science and Electrical Engineering*, 2007.

[109] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *European Conference on Computer Vision*, 6314:213–226, 2010.

[110] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, Dec 2003.

[111] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 2011.

[112] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167, June 2012.

[113] S. Shekhar, V. Patel, and R. Chellappa. Analysis sparse coding models for image-based classification. *IEEE International Conference on Image Processing*, 2014.

[114] S. Shekhar, V. Patel, N. Nasrabadi, and R. Chellappa. Joint sparse representation for robust multimodal biometrics recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):113–126, Jan 2014.

[115] S. Shekhar, V. M. Patel, and R. Chellappa. Synthesis-based recognition of low resolution faces. *International Joint Conference on Biometrics*, pages 1–6, Oct 2011.

[116] S. Shekhar, V. M. Patel, and R. Chellappa. Synthesis-based robust low resolution face recognition. *IEEE Transactions on Image Processing (under review)*, 2014.

[117] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Joint sparsity-based robust multimodal biometrics recognition. *ECCV Workshop on Information Fusion in Computer Vision for Concept Recognition (IFCVCR)*, Florence, Italy, Oct. 2012.

[118] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013.

[119] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Coupled projections for semi-supervised adaptation of dictionaries. *IEEE Transactions on Image Processing (under review)*, 2014.

[120] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[121] X. Shi, Q. Liu, W. Fan, P. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. *International Conference on Data Mining*, pages 1049–1054, Dec 2010.

[122] Y. Shi and F. Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *International Conference on Machine Learning*, pages 1079–1086, 2012.

[123] A. Shrivastava, S. Shekhar, and V. M. Patel. Unsupervised domain adaptation using parallel transport on grassmann manifold. *IEEE Winter conference on Applications of Computer Vision*, 2014.

[124] S. Siena, V. N. Boddeti, and B. V. Kumar. Coupled marginal Fisher analysis for low-resolution face recognition. *ECCV 2012: Workshops and Demonstrations*, pages 240–249, 2012.

[125] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):1615–1618, December 2003.

[126] V. Sindhwani and D. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. *25th International Conference on Machine learning*, pages 976–983, Helsinki, July 2008.

[127] 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*, 94:1948–1962, Nov. 2006.

[128] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58:267–288, 1996.

[129] H. Van Nguyen, V. Patel, N. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, Dec 2013.

[130] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma. Towards a practical face recognition system: Robust alignment and illumination via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:372–386, Feb. 2012.

[131] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. *International Joint Conference on Artificial Intelligence*, pages 1541–1546, 2011.

[132] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2216–2223, June 2012.

[133] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. Technical report, Rice University, 2010.

[134] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98:1031–1044, June 2010.

[135] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009.

[136] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, Aug. 2012.

[137] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8, June 2008.

[138] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. *International Conference on Multimedia*, pages 188–197, 2007.

[139] J. Yang and Y. Zhang. Alternating direction algorithms for l1 problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33:250–278, 2011.

[140] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. *IEEE International Conference on Computer Vision*, pages 543–550, Nov. 2011.

[141] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, Feb. 2006.

[142] X.-T. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3493–3500, San Fransisco, USA, June 2010.

[143] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang. Multi-observation visual recognition via joint dynamic sparse representation. *International Conference on Computer Vision*, pages 595–602, Barcelona, Spain, Nov. 2011.

[144] L. Zhang, M. Yang, Z. Feng, and D. Zhang. On the dimensionality reduction for sparse representation based face recognition. *International Conference on Pattern Recognition*, pages 1237–1240, Aug. 2010.

[145] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[146] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, Dec 2003.

[147] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan. Heterogeneous domain adaptation for multiple classes. *International Conference on Artificial Intelligence and Statistics*, pages 1095–1103, 2014.

[148] X. Zhou and B. Bhanu. Feature fusion of face and gait for human recognition at a distance in video. *International Conference on Pattern Recognition*, 4:529–532, Hong Kong, Aug. 2006.

[149] W. Zou and P. Yuen. Very low resolution face recognition problem. *IEEE Transactions on Image Processing*, 21(1):327–340, Jan 2012.