ABSTRACT

Title of dissertation: FACE RECOGNITION AND VERIFICATION IN UNCONSTRAINED ENVIRIONMENTS

Huimin Guo
Doctor of Philosophy, 2012

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

Face recognition has been a long standing problem in computer vision. General face recognition is challenging because of large appearance variability due to factors including pose, ambient lighting, expression, size of the face, age, and distance from the camera, etc. There are very accurate techniques to perform face recognition in controlled environments, especially when large numbers of samples are available for each face (individual). However, face identification under uncontrolled(unconstrained) environments or with limited training data is still an unsolved problem. There are two face recognition tasks: face identification (who is who in a probe face set, given a gallery face set) and face verification (same or not, given two faces). In this work, we study both face identification and verification in unconstrained environments.

Firstly, we propose a face verification framework that combines Partial Least Squares (PLS) and the One-Shot similarity model[1]. The idea is to describe a face with a large feature set combining shape, texture and color information. PLS

regression is applied to perform multi-channel feature weighting on this large feature set. Finally the PLS regression is used to compute the similarity score of an image pair by One-Shot learning (using a fixed negative set).

Secondly, we study face identification with image sets, where the gallery and probe are sets of face images of an individual. We model a face set by its covariance matrix (COV) which is a natural 2nd-order statistic of a sample set.By exploring an efficient metric for the SPD matrices, i.e., Log-Euclidean Distance (LED), we derive a kernel function that explicitly maps the covariance matrix from the Riemannian manifold to Euclidean space. Then, discriminative learning is performed on the COV manifold: the learning aims to maximize the between-class COV distance and minimize the within-class COV distance.

Sparse representation and dictionary learning have been widely used in face recognition, especially when large numbers of samples are available for each face (individual). Sparse coding is promising since it provides a more stable and discriminative face representation. In the last part of our work, we explore sparse coding and dictionary learning for face verification application. More specifically, in one approach, we apply sparse representations to face verification in two ways via a fix reference set as dictionary. In the other approach, we propose a dictionary learning framework with explicit pairwise constraints, which unifies the discriminative dictionary learning for pair matching (face verification) and classification (face recognition) problems.

Face Recognition and Verification in Unconstrained Environments

by

Huimin Guo

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
2012

Advisory Committee:
Professor Larry Davis, Chair/Advisor
Professor Rama Chellappa
Professor Min Wu
Professor David Jacobs
Professor Samir Khuller

Dedication

To my dear parents, *Xingqi Guo* and *Liqin Luo*.

Acknowledgments

First, I would like to thank my advisor Prof. Larry Davis for taking me as his student, and always being helpful and supportive over my Ph.D. student life. He provided lots of valuable guidance on the research work and also gave us independence to work on problems that we are interested in. It is really a pleasure to work with him. Secondly, I would like to thank Prof. Rama Chellappa for meaningful comments during my proposal exam and our project meetings. Also, I enjoyed his courses during my graduate studies very much, which also guided my journey in computer vision. I also thank Prof. David Jacobs, Prof. Min Wu and Prof. Samir Khuller for serving as my dissertation committee. Thanks to all of them all for taking time out of their busy schedules.

I also would like to thank those colleagues worked with me closely: William Schwartz, Ruiping Wang, Zhuolin Jiang, Jonghyun Choi. Without their generous help and collaboration, I couldn't move as far either on my research work or the projects. I really enjoyed working with them and learned a lot from them. I also thank lots of other group members for valuable discussions, Vlad, Arpit, Behjat, Brandyn.....

I thank lots of my friend in Maryland and in other parts of the world, your accompany made my Ph.D. exploration much more enjoyable: Stephen Chan, Qi Hu, Qiang Qiu, Jie Ni, Daozheng Chen, Yan Chen, Tao Wu, Ming Du, Yang Hu, Renting Xu, Xuan Liu, Zhan Shi, Shi Pu, Weisu Wang, Taotao Liu, Eric Luo... Thank you for always being there and sharing my ups and downs.

Finally, deepest thanks to my parents for bringing me up and their unconditional love to me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Face Recognition in Unconstrained Environments

During the past two decades, face recognition (FR) has received great attention and tremendous progress has been made[5, 6]. Face recognition research [5, 6] is driven by its variety of applications in areas such as public security, human computer interaction, and financial security. Face recognition mainly involves the following three tasks (see Figure 1.1): *identification* (*1:N matching* problem), *verification* (*1:1 matching* problem), *Watch list*. In the *identification* task, a probe image is matched against a set of labeled faces in a gallery set, and is identified as the person presenting the highest similarity score. In the *verification* task, given two face images, the goal is to decide whether these two images are of the same person or not. In the *watch list* task, the recognition system first determines if the identity of the query face image is in the watch list and, if yes, then identifies the individual. Usually, face recognition refers to the face identification task. We focus on the first two tasks.

There has been tremendous progress in face recognition. Under carefully or well controlled conditions high recognition rates can be obtained even when a large number of subjects is in the gallery [6, 5]. However, when this task is performed under uncontrolled conditions (unconstrained environments), such as uncontrolled (outdoor) lighting and changes in facial expressions, recognition rates decrease sig-

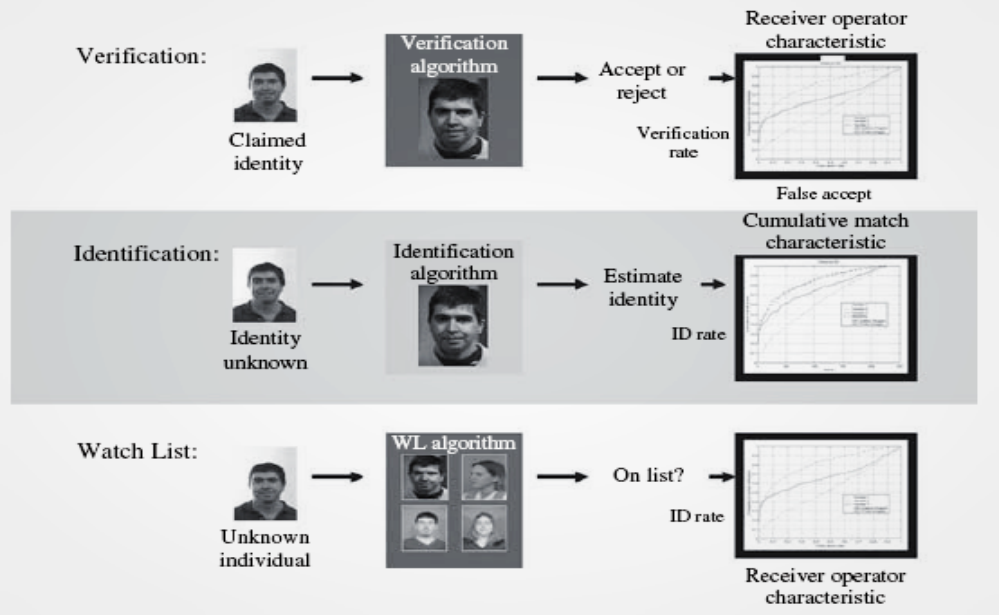


Figure 1.1: Three face recognition tasks: verification, identification, and watch list.

nificantly. Face appearances may change a lot when acquisition conditions are less constrained, making the recognition problem harder. Unconstrained environments include no restrictions over environmental conditions such as scale, pose, lighting, focus, resolution, facial expression, accessories, makeup, occlusions, background, and photographic quality, etc.

Another challenge is that most current face recognition algorithms perform well when several training images are available per subject; however they are not adequate for scenarios where a single sample per subject is available. Single-sample-size problem would make face recognition in unconstrained environments even more challenging. In real world applications, one training sample per subject presents advantages such as ease of collect galleries, low cost for storage and lower computa-

tional cost [7]. Thus, a robust face recognition (identification) system able to work with both single or small numbers of samples per subject is desirable.

1.2 Face Verification in Unconstrained Environments

In real world, face verification is more widely applicable and is also the foundation of the identification task. Since face verification is a binary classification problem on an input face pair, there are two major components of a verification approach: face representation and face matching. The extracted feature (descriptor) should be not only discriminative but also invariant to apparent changes and noise. The matching should be robust to variations from pose, expression, and occlusion, etc. These requirements make face verification a very challenging problem.

1.3 Face Recognition and Object Recognition with Image Sets

Face recognition based on image sets has recently attracted growing interest in the computer vision and pattern recognition community. This problem naturally arises in a wide range of applications including video surveillance, classification based on images from multi-view cameras and photo albums, and classification based on long term observations. In the task of face recognition from image sets, each set generally contains a large number of images (faces) that belong to the same person and cover large variations in the person's appearance due to camera pose changes, non-rigid deformations or different lighting conditions. While traditional recognition methods based on single-shot images have achieved a certain level of success under

constrained conditions, more robust face recognition can be expected by using sets as input rather than single images, especially in unconstrained environments. This is mainly because the image sets incorporates useful data variability information, which can be efficiently exploited under more realistic conditions with significantly larger variations.

There are many aspects that can be improved in the recognition of faces in unconstrained environments. In this work, we propose several methods for face recognition (identification either from single image or image sets) and face verification for unconstrained environments.

Chapter 2

Face Verification using Partial Least Squares One-Shot Model

2.1 Background

Previous research has shown that face recognition under well controlled acquisition conditions is relatively mature and provides high recognition rates even when a large number of subjects is in the gallery [6, 5]. However, when this task is performed under uncontrolled conditions (unconstrained environments), such as uncontrolled lighting and changes in facial expressions, recognition rates decrease significantly. Face appearances may change when acquisition conditions are uncontrolled, making the recognition problem harder. For example, there can be some extreme illuminations, expressions and out of focus images.

Recently, the Labeled Faces in the Wild (LFW)[2] dataset was released as a benchmark for the face verification (pair-matching) problem. The LFW images include considerable visual variations caused by, for example, lighting, pose, facial expression, partial occlusion, aging, scale, and misalignment. Figure 2.1 contains some examples of pairs of images from the same person that differ in lighting, pose, facial expression and partial occlusion. Face verification is a very challenging problem. Different from many classification problems where the specific class label of each image is given during training, only binary information such as same/different or relevant/irrlevant is provided for training data in applications such as face ver-

ification (given a *target* and a *query* image, determine whether they are from the same person), pair matching, image retrieval, etc. Typically, a discriminative similarity measure is learned through metric learning [8, 9, 10, 11] from pairs of training images labeled as 'same' or 'different'; this provides less specific information than known classes - category labels.

Since face verification is a binary classification problem of an input face pair, there are two major components of a verification approach: face representation and face matching. The extracted feature (descriptor) should be not only discriminative but also invariant to apparent changes and noise which are common in unconstrained environments. In order to reduce the problems associated with data collected under uncontrolled conditions, we consider a combination of low-level feature descriptors based on different clues (such approaches have provided significant improvements in object detection [12, 13] and recognition [14, 15]). Then, feature weighting is performed by Partial Least Squares (PLS), which handles very high-dimensional data presenting multicollinearity and works well very few samples are available [13, 16, 17, 18, 19].

Another important issue in face verification is learning an appropriate similarity measure which is robust to variations from pose, expression and occlusion. Most popular methods tailor the similarity measure to available training data by applying learning techniques [20]. In such methods, testing is performed using models (or similarity measures) learned beforehand. The other trend is to learn from one or very few training examples. Wolf et al. [21, 20] proposed the use of One-Shot Similarity (OSS) to learn discriminative models exclusive to vectors being compared, by

using a set of background samples. In our work, we use this One-Shot framework to learn models for feature vectors representing face samples on-the-fly. The prediction scores are computed from Partial Least Squares Regression. A down-side of employing one-shot scheme is the imbalance of the class distributions. However, studies have shown that data imbalance presents little influence on the performance of PLS modeling [22, 23]. Barker et al. [22] pointed out that PLS involves eigen-decomposition of the between-class scatter matrix solely, which only involves calculation of mean vectors of different classes. This does not depend on the number of samples in each class. In addition, Qu et al. [23] showed that the weight estimation performed by PLS helps it to extract favorable features for unbalanced classification.

There are several advantages of our method [24]: (1) It is unsupervised. No labeled training set is needed, either pair labels or identity information. All we need is a small unlabeled reference set. The discriminative models are learned online and exclusively for the pair being compared. (2) PLS has been shown, experimentally, to be robust to modest pose variations[25], expression, illumination, aging and other uncontrolled variations[26]. (3) There is almost no parameter tuning with PLS. The only parameter is the number of factors and it is not sensitive.

2.2 Related Work

There has been a significant amount of relevant works on face verification [14, 27, 28, 20, 29, 30, 31, 21, 32, 33]. Here we briefly review those state-of-the-art methods that have been evaluated on the LFW benchmark. At the end of this

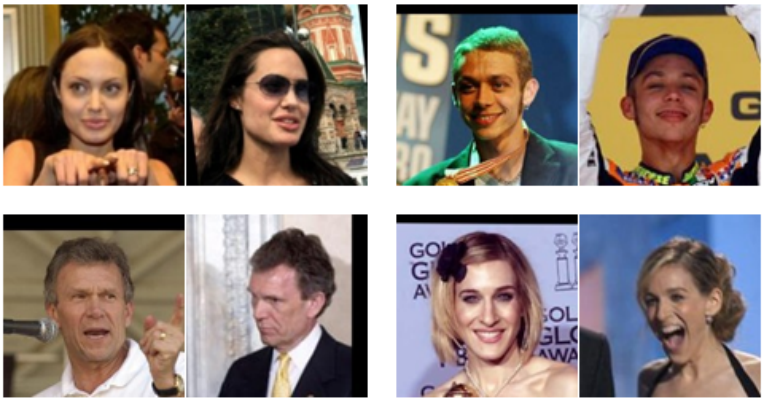


Figure 2.1: Examples of some images from the LFW dataset with variations in: Top row:(left) partial occlusion, (right)lighting; bottom row: (left) pose, (right) expression. Each corner shows a different subject. Note that each pair is from the same person.

section we also review some recent work using PLS in face identification[26, 25].

Some work focus principally on face descriptors [32, 31, 29]. Pinto[32] et al. combined variants of intensity and V1-Like models. The classification of face images was performed using large-scale multi kernel learning (MKL) associated with a support vector machine (SVM). In [31], an unsupervised learning-based encoding (LE) method was proposed to encode the micro-structures of a face with a single or a combination of multiple descriptors. In [29], Patterns of Oriented Edge Magnitudes (POEM) was introduced. The POEM feature is built by applying a self-similarity based structure on oriented magnitudes, calculated by accumulating a local histogram of gradient orientations over all pixels of image cells, centered on the pixel. Other works have employed metric learning[28, 34, 30] for learning similarity functions for verification. Guillaumin et al.[28] presented two methods for learning robust distance measures: (1) LDML: a logistic discriminant approach which learns the metric from a set of labeled image pairs and (2) MkNN: a nearest neighbour approach which computes the probability for two images belonging to the same class. In [30], a part based face representation (densely sampled overlapping image patches) is computed to enable elastic and partial matching. The distance metric is defined as each descriptor in one face is matched against its spatial neighborhood in the other face. [34] presented the Cosine Similarity Metric Learning as an alternative to Euclidean distance. The idea was to learn a transformation matrix by minimizing the cross-validation error with a gradient-based optimization algorithm.

The use of low-level feature descriptors has been an effective approach in face

recognition and face verification [14, 35, 36, 37, 38, 39]. SIFT and histogram of oriented gradients (HOG), which can be viewed as a quantized code of the facial gradients, are used in face recognition as effective descriptors [40, 41]. Local binary patterns (LBP) and Gabor filters are descriptors most widely used in face recognition. LBP is invariant to monotonic photometric change and can be efficiently extracted. Gabor features are characterized by spatial frequency, spatial locality, and orientational selectivity for coping with image variabilities such as illumination variations. There are several combinations or variations based on these LBP and Gabor descriptors [14, 35, 36, 37]. In addition, by varying a sampling radius, R, and combining the LBP images, a multiresolution representation based on LBP, called Multi-Scale Local Binary Patterns (MSLBP) [42] can be obtained. This representation has been suggested for texture classification and the results reported for this application show a better accuracy than that of the single scale LBP method. Recent research has focused on parameter learning with a HOG like template [43, 44]. Other LBP variants including Three-Patch based Local Binary Patterns (TPLBP), Four-Patch based Local Binary Patterns (FPLBP) [45] have been introduced for face recognition/verification.

Some of the best performing algorithms focus on the classifier design[27] and learning more discriminative models[21, 20, 46]. Kumar et al. [27] designed two methods: *attribute* classifiers, which are trained to recognize describable aspects of visual appearance, and *simile* classifiers, trained to recognize the similarity of faces, or regions of faces, with respect to specific reference people. Wolf et al.[21, 20, 46] proposed One-Shot Similarity [20] to learn discriminative models exclusive to vectors

being compared, by using a set of background samples. In [20], they used a random-patch based image representation with OSS as the similarity score and a SVM to classify. In [21], the OSS was extended to 'Two-Shot Similarity'(TSS). Also, the authors used the ranking of images most similar to a query image and employed these as a descriptor for that image. The best verification result was obtained by adding SVM based OSS and TSS to LDA. Yin et al. [47] used extra generic identities ('memory': containing multiple images with large intra-personal variation) as a bridge and the 'associate-predict' model to handle intra-personal variation. Most of the approaches mentioned above (especially the latter two categories) are supervised methods requiring a training set, which is referred to as the image-restricted setting in the LFW protocol. However, the training phase is burdensome and there are situations in which not providing training data is more practical. Some approaches design training-free face verification and are evaluated in the unsupervised setting on LFW dataset [48, 49]. In [48], the authors randomly selected 100 images from LFW as a reference set (without using label or pair-wise relationships of same or different) for the Borda count ranking between the Gabor Jet Descriptors. In another training-free approach, locally adaptive regression kernels (LARK)[49] were employed as visual descriptors, in conjunction with the matrix cosine similarity (MCS) measure.

2.2.1 Partial Least Squares and Face Recognition

Tackling face verification task with Partial Least Squares is motivated by one of our previous work [50] using Partial Least Squares regression to weight a combi-

nation of a large number of feature descriptors (with more than 70,000 descriptors) that capture different visual information in a one-against-all classification scheme with highly unbalanced class distributions with a single or very few samples in the positive class. The benefits are provided by combining an increasing number of feature descriptors weighted by Partial Least Squares to emphasize those that best discriminate among different subjects. The method is evaluated on the FRGC/FRVT dataset [51], and the FERET dataset [52]. Experiments show that the PLS [50, 53] based method outperforms current state-of-theart results, especially for recognizing faces acquired across varying conditions. In addition, it can also handle the problem of insufficient training data – experimental results show high performance when only a single sample per subject is available.

The partial least squares (PLS)-based approach outperforms state-of-art techniques in most of the comparisons involving standard face recognition datasets, particularly when the data is acquired under uncontrolled conditions, such as in experiment 4 of the FRGC dataset. In addition, the PLS-based method can also handle the problem of insufficient training data. Experiments on datasets with only a single sample per subject have shown high performance when the PLS-based algorithm is used.

More recently, in [25] the authors used PLS to linearly map images in differently modalities to a common linear subspace in which they are highly correlated. The work showed, in theory, that there exist linear projections of images taken in two modalities that map them to a space in which images of the same individual are very similar. In that work, PLS was shown to work well across modalities: high

resolution vs. low resolution, pose variation, and in comparing images to sketches.

2.3 Proposed Method

2.3.1 Overview of the Framework

The pipeline of our PLS One-Shot Model based face verification approach is presented in Figure 4.1. Firstly, a randomly selected set of images **A** (approximately 500 images from LFW) is set aside as background samples [21]. The images in this set is unlabeled and considered as 'negative' examples. It should not contain any images from individuals to be compared subsequently. All images in this set are aligned, cropped and their feature vectors are extracted and stored.

When a new pair of face images is presented, their two feature vectors are extracted and their similarity score is calculated by building PLS One-Shot Model using each of the vectors versus the set **A** and projecting the other vector to the other's model. The two projections provide the responses of the PLS regression models. The average of these two scores is used to measure the similarity of the image pair.

Finally, if the similarity score is above threshold, a match is declared; a non-match is declared, otherwise.

2.3.2 Feature Extraction

After cropping and resizing the faces, each sample is decomposed into overlapping blocks and then, a set of low-level feature descriptors is extracted from each

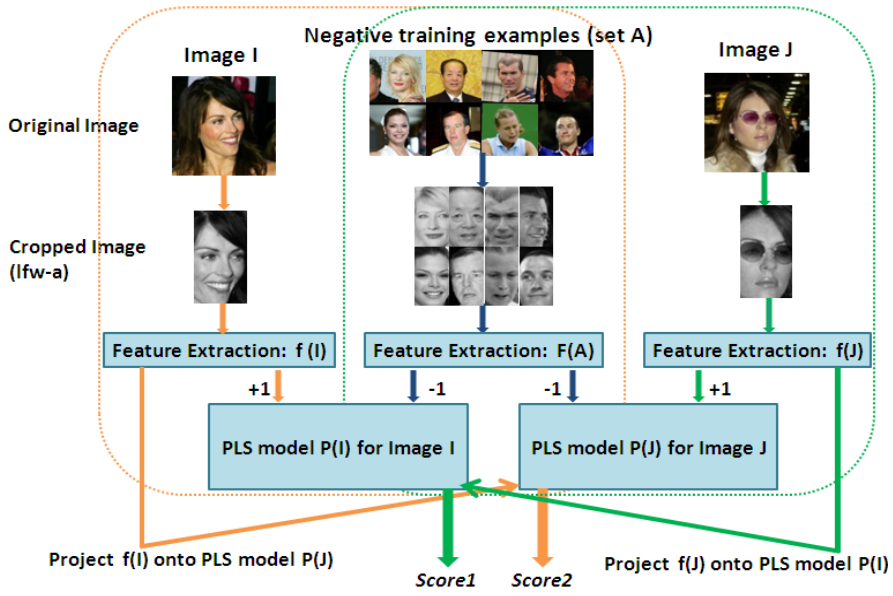


Figure 2.2: Our processing pipeline for face verification using the PLS One-Shot Model.

block. The feature extraction methods used capture information related to shape (histogram of oriented gradients (HOG) [54]), texture (captured by local binary patterns (LBP) [36] and multi-scale local binary patterns (MSLBP) [42]), color information (captured by averaging the intensities of pixels in a block, referred as to mean feature), and salient visual properties (captured by Gabor filters [55]) .

HOG captures edge or gradient structures that are characteristic of local shape. According to Dallal and Triggs [54], a consequence is a controllable degree of invariance to local geometric transformations, providing invariance, for example, to translations and rotations smaller than the local spatial or orientation bin size.

LBP characterizes the spatial structure of the local image texture and is invariant to monotonic transformations of the pixel gray values [36]. Its original version labels the pixels of an image by thresholding the 3×3 neighborhood with intensity g_p $(p = 0, 1, 2, ..., 7)$ with respect to its intensity of the center pixel g_c, then defines

$$S(g_p - g_c) = \begin{cases} 1, & g_p \geq g_c \\ 0, & g_p < g_c \end{cases} \tag{2.1}$$

Then, the LBP pattern of the image neighborhood is obtained by summing the corresponding thresholded values $S(g_p - g_c)$ weighted by a binomial factor of 2^p as

$$LBP = \sum_{p=0}^{7} S(g_p - g_c)2^p \tag{2.2}$$

Finally, a 256-bin histogram of the resulting labels is used as a feature descriptor for a patch of the image.

According to the size of the neighborhood employed, there are different versions of LBP. The 3x3 version described above is denoted LBP_8, due to the use

of 8 adjacent pixels spaced radially by 45°. LBP can also be employed in a multi-resolution framework by considering concentric circles of different radii, called MSLBP [42]. This method has not been widely used in face recognition.

In this work, in addition to the basic single scale LBP operator, we consider the MSLBP with setup $\text{LBP}_{8,2}$ (8 pixels on a circle whose radius is 2 pixels) and $\text{LBP}_{8,4}$ (8 pixels on a circle whose radius is 4 pixels). The two resulting histograms are simply concatenated and used as descriptors.

Gabor filters are widely used in object recognition since they capture a number of salient visual properties including spatial localization, orientation selectivity, and spatial frequency selectivity quite well [55]. They are robust to illumination variations since they detect amplitude-invariant spatial frequencies of pixel gray values. Gabor filters most commonly used in face recognition have the form:

$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} e^{(-\|k_{\mu,\nu}\|^2 \|z\|^2 / 2\sigma^2)} [e^{ik_{\mu\nu}z} - e^{-\sigma^2/2}] \tag{2.3}$$

where μ and ν define the orientation and scale of the Gabor kernels, $z = (x, y)$, $\|\cdot\|$ denotes the norm operator, and the wave vector is $k_{\mu,\nu} = k_\nu(cos\phi_\mu, sin\phi_\mu)$ where $k_\nu = k_{max}/f^\nu$ and $\phi_\mu = \pi\mu/8$ with k_{max} being the maximum frequency and f being the spacing factor between kernels in the frequency domain. In this work, we used $\sigma = 2\pi$, $k_{max}= \frac{\pi}{2}$, and $f = \sqrt{2}$.

The Gabor representation of a face is derived from convolving the gray-scale face image with the Gabor filters. Let $I(x, y)$ be the face image, its convolution with a Gabor filter is defined as follows

$$G_{\psi I}(x, y, \mu, \nu) = I(x, y) * \psi_{\mu\nu}(z) \tag{2.4}$$

where $*$ denotes the convolution operator. Five scales $\mu \in \{0, ..., 4\}$ and eight orientations $\nu \in \{0, ..., 7\}$ are used here, which results in 40 Gabor filters. For each Gabor filter, one magnitude is computed at each pixel position, resulting therefore in 40 descriptors per pixel. Then, the final feature vector is obtained by downsampling the Gabor features by a factor 4 (one per four rows and columns) in order to reduce the dimensionality of the feature vector to manageable sizes.

After the feature extraction process is performed for all blocks inside a cropped face, descriptors are concatenated creating a high-dimensional feature vector $\boldsymbol{v}$. This vector is used to describe the face.

2.3.3 Partial Least Squares Regression

Partial least squares is a method for modeling relations between sets of observed variables by means of latent variables. The basic idea of PLS is to construct new predictor variables, latent variables, as linear combinations of the original variables summarized in a matrix $\boldsymbol{X}$ of descriptor variables (features) and a vector $\boldsymbol{y}$ of response variables. Detailed description of the PLS method can be found in [16, 56, 57].

Let $\mathcal{X} \subset \mathbb{R}^m$ denote an m-dimensional feature space and let $\mathcal{Y} \subset \mathbb{R}$ be a scalar space representing the response variable. Let the number of samples be n. PLS decomposes a mean-centered matrix $\boldsymbol{X}_{n \times m}$ and mean-centered vector $\boldsymbol{y}_{n \times 1}$ into

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E}$$

$$\boldsymbol{y} = \boldsymbol{U}\boldsymbol{q}^T + \boldsymbol{f}$$

where $\boldsymbol{T}$ and $\boldsymbol{U}$ are $n \times p$ matrices containing p extracted latent vectors, the $(m \times p)$ matrix $\boldsymbol{P}$ and the $(1 \times p)$ vector $\boldsymbol{q}$ represent the loadings and the $n \times m$ matrix $\boldsymbol{E}$ and the $n \times 1$ vector $\boldsymbol{f}$ are the residuals. Using the nonlinear iterative partial least squares (NIPALS) algorithm [16], a set of weight vectors is constructed, stored in the matrix $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_p)$, such that

$$[cov(\boldsymbol{t}_i, \boldsymbol{u}_i)]^2 = \max_{|\boldsymbol{w}_i|=1} [cov(\boldsymbol{X}\boldsymbol{w}_i, \boldsymbol{y})]^2 \tag{2.5}$$

where $|\boldsymbol{w}_i|$ denotes the 2-norm of vector $\boldsymbol{w}_i$, $\boldsymbol{t}_i$ is the i-th column of matrix $\boldsymbol{T}$, $\boldsymbol{u}_i$ the i-th column of matrix $\boldsymbol{U}$, and $cov(\boldsymbol{t}_i, \boldsymbol{u}_i)$ is the sample covariance between latent vectors $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$. After extracting the latent vectors $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$, the matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$ are deflated by subtracting their rank-one approximations based on $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$. This process is repeated until the desired number of latent vectors has been extracted. Once the low dimensional representation of the data has been obtained by NIPALS, the regression coefficients $\boldsymbol{\beta}_{m \times 1}$ can estimated by

$$\boldsymbol{\beta} = \boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1}\boldsymbol{T}^T\boldsymbol{y}. \tag{2.6}$$

The regression response, y_v, for a feature vector $\boldsymbol{v}$ is obtained by

$$y_v = \overline{y} + \boldsymbol{\beta}^T\boldsymbol{v} \tag{2.7}$$

where $\overline{y}$ is the sample mean of $\boldsymbol{y}$.

It is important to point out that even though the number of weight vectors used to create the low dimensional representation of the data matrix $\boldsymbol{X}$ is p, Equation 2.7 shows that only a single dot product of a feature vector with the regression coefficients is needed to obtain the response of a PLS regression model – and it is

this response that is used to rank faces in a gallery. This characteristic makes the use of PLS particularly fast for finding matches for probe samples, in contrast to other methods where the number of dot product evaluations depends on the number of eigenvectors considered, which is quite large in general [58].

2.3.4 Face Verification with PLS One-Shot Model

To decide whether the images of two faces **I** and **J** are of the same individual or not, traditional methods for face verification use a large training set to learn models (or similarity measures) beforehand, and then employ this model to calculate the similarity of images **I** and **J**. In contrast, we learn the models for the images to be matched on-the-fly using the PLS One-Shot Model (PLS + OSS). The key idea behind the OSS[20] is to use a set **A** of *negative* training examples not containing images belonging to the people being compared. The details of OSS are described in [20]. (see Algorithm 1).

To perform face verification, we use the given training set as the fixed *negative* set (or reference set) **A**, see Figure 4.1. When a new pair of face images **I** and **J** is to be matched, they are cropped to the same size and the features are extracted. Then, a discriminative PLS Regression model is learned by taking **A** as the negative samples (with labels -1) and **I** to be the single positive sample (with label +1). Then, **J** is evaluated by this model to obtain a response (similarity score). This score gives a measure of how likely **J** shares the same label as **I** or belongs to the negative set (which means **J** might be very different from **I**). Symmetrically, we switch the roles

Algorithm 1: Computation of the symmetric One-Shot Similarity score for two vectors, I and J, given a set A of negative examples.

```
function One-Shot-Similarity(I, J, A)
    Model1 = train(I, A)
    Score1 = classify(J,Model1)
    Model2 = train(J, A)
    Score2 = classify(I, Model2)
    return 1/2(Score1 + Score2)
```

of **I** and **J** and execute the same procedure. The final similarity score for this pair is the average of the two scores.

2.4 Experimental on LFW Dataset

In this section, we evaluate our PLS One-Shot model based face verification on the LFW.

2.4.1 LFW Dataset

The Labeled Faces in the Wild (LFW)[2] dataset contains 13,233 face images labelled by the identity of the person. The faces show large variation in pose, expression, lighting, occlusion, and aging. There are three versions of the LFW dataset available: original, funneled and aligned. Wolf et al.[21] showed that the aligned version (lfw-a) is better than the funneled version in dealing with misalignment.

Therefore, we use the lfw-a n in all of our experiments.

The dataset comes with a division into 10 fully independent splits (folds) that can be used for cross validation [59]. Using only the given image pairs in the training set is referred to as the **image-restricted** paradigm; in this case, it is known whether an image pair belongs to the same person or not, while identity information is not used at all. The **unrestricted** paradigm refers to training methods that can use all available data, including the identity of the people in the images. Additionally, there is an **unsupervised** paradigm when there is no supervised information, such as in the form of same/not-same labels used. As an example of the unsupervised paradigm, in[48], the authors randomly selected 100 images from LFW for the Borda-count method that was used together with the Gabor descriptor. The 100 images were used simply as a reference set; their pair or identity information was not used.

In our evaluation, while performing each independent fold, we randomly choose 500 images from the training set (other 9 splits, 5400 image pairs) without using their pair information. The number 500 is chosen because experiments with several datasets show sufficiently good performance when the 'negative' set contains 300 to 1000 images. Then, these images are fixed as the 'negative' set (background samples) for this fold. According to the protocol, the 10 splits are mutually exclusive with respect to subject identities. Below, we present results using both the unsupervised and the image restricted paradigms.

Figure 2.3: Examples of face images with different croppings. Left to right: 80×148 (crop1), 80×110 (crop2), 80×64 (crop3).

2.4.2 Preprocessing

In our evaluation we consider three different crop regions: (1) centered face region with hair (2) centered face region without mouth (3) centered face region without mouth and hair. Figure 2.3 shows the three different croppings: 80×148 (crop1), 80×110 (crop2), 80×64 (crop3). There are pros and cons for each different cropping: removing some parts like the mouth or hair region could help alleviate effects due to expression and hat occlusion, while mouth/chin and hair style might also include some informative features. We tried these croppings and fused the three scores in a simple way (rely more on full region than the other two partial regions, fusion gives about 1% improvement from 'crop1 only', see Table 4.2):

$$finalscore = score(crop1) + 0.5 * score(crop2) + 0.5 * score(crop3) \tag{2.8}$$

For illumination normalization, our experiment found that the un-normalized images and images filtered by Difference of Gaussian give similar results with PLS One-Shot model. We report the results with DoG, as in [30, 31]. Since there

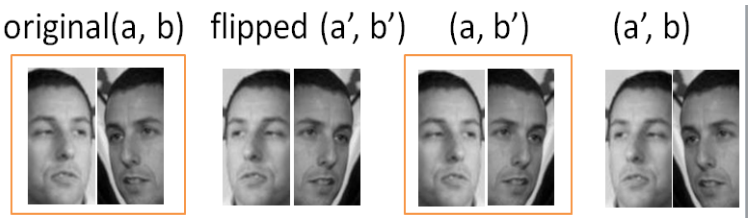


Figure 2.4: Examples of face image and its flipped image.

is significant pose variation within LFW, we additionally use the flipped (mirror) image. Figure 2.4 gives an example of an image pair and their flipped images. When comparing image pair **I** and **J**, we also compare **I** and the flipped image of **J**. Then the average of the two scores is taken as the final similarity score. We will show this simple flip step improves performance.

2.4.3 Experimental Setup

For HOG features, we use block sizes of 16×16 and 8×8 with strides of 4 and 4 pixels, respectively. For LBP features, we use block size of 16×16 with strides of 8 pixels. The Gabor features have 5 scales and 8 orientations, down sampled by a factor of 4. The PLS factor (number of latent vectors p) is set to 11.

For verification, given a pair of query image and target image, the goal is to correctly determine whether these two belong to the same subject. The well-known receiver operating characteristic (ROC) curve, which describes relations between false acceptance rates (FARs) and true acceptance rates (TARs), is used to evaluate the performance of verification algorithms. As the TAR increases, so does the FAR.

Therefore, one would expect an ideal verification framework to have TARs all equal to 1 for any FARs.

In prior work on the LFW benchmark, algorithms are typically evaluated by ROC curves and the classification accuracy (true positive rate) at the Equal Error Rate (EER). EER is the location on the ROC curve where the false positive rate and false negative rate are equal. We report our results in the form of ROC curves and the estimated mean classification accuracy and the standard error of the mean for the 10 cross-validation folds in View 2 of the dataset.

2.4.4 Comparison with the State-of-the-art Methods

As stated previously, we only uses a very small number of images from the training set as a reference set. No pair label or identity is used. Thus, we compare our method using the unsupervised paradigm. PLS One-Shot method outperforms other methods using the unsupervised paradigm by a large margin. At the same time, its performance is comparable to the best results using the image-restricted paradigm whose methods use pair information.

Comparison with Unsupervised paradigm. Table 4.2 shows the classification accuracy (at EER) of our method in comparison with other methods using the unsupervised paradigm. Figure 4.4 presents the ROC curve of our approach (pink line), along with the ROC curves of previous methods. As can be seen, PLS One-Shot outperforms the other methods by a very large margin. Similar to these methods, only a very small set from the LFW is used as a reference set. No other

Method	Classification accuracy
SD-MATCHES, aligned [48]	0.6410±0.0042
H-XS-40, aligned [48]	0.6945±0.0048
GJD-BC-100, aligned [48]	0.6847±0.0065
Our method (PLS + OSS, crop1 only)	**0.8418±0.0052**
Our method (PLS + OSS)	**0.8533±0.0038**
Our method (PLS + OSS, flip)	**0.8612±0.0047**

Table 2.1: Mean (± standard error) classification accuracy on the LFW dataset, Unsupervised Training benchmark using the PLS One-Shot Model, and the same model except the addition of the flipped image idea. The 'crop1 only' gives the result of the main cropping.

supervised information is used. Our PLS One-Shot model based face verification approach is simple and effective for this challenging real-world dataset.

Comparison with Image-Restricted paradigm. Since many state-of-the-art methods use the pair information and report their results using the Image-Restricted paradigm, we compare our results with them too. Table 5.2 shows the classification accuracy of our method in comparison with those methods with the Image-Restricted paradigm. Figure 5.4(b) contains the ROC curve of our approach (blue line), along with the ROC curves of previous methods with the Image-Restricted paradigm.

The results show that our approach is comparable with the state-of-the-art

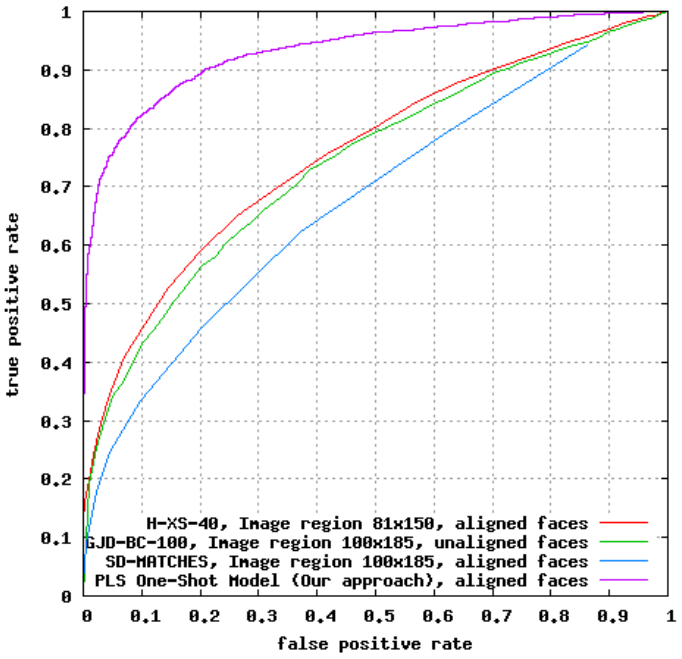


Figure 2.5: ROC curves for View 2 of the LFW dataset. Each point on the curve represents the score over the 10 folds (of false positive rate, true positive rate) for a fixed threshold. Unsupervised paradiam.

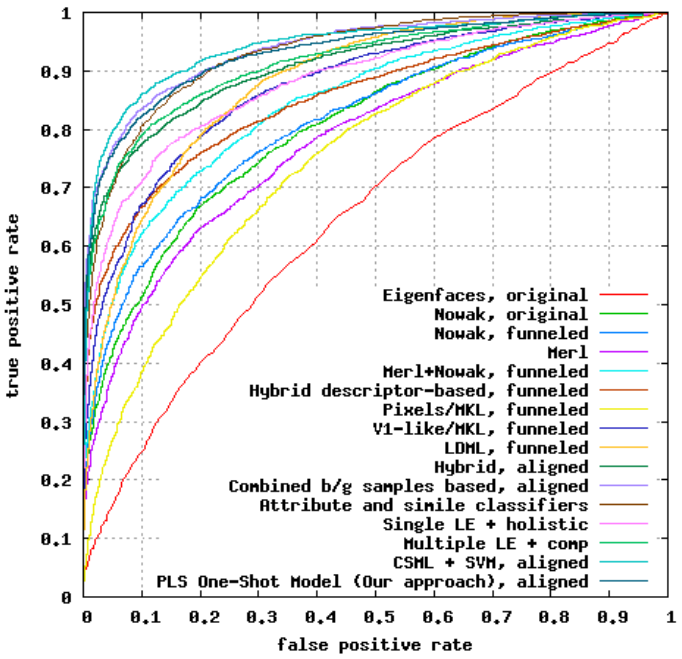


Figure 2.6: ROC curves for View 2 of the LFW dataset. Image-Restricted paradiam. The plots are gernerated from results reported on http://vis-www.cs.umass.edu.lfw.results.html.

Method	Classification accuracy
Eigenfaces, original[60]	0.6002±0.0079
Nowak, funneled [11]	0.7393±0.0049
MERL+Nowak, funneled [61]	0.7618 ±0.0058
Hybrid Descriptor, funneled [62]	0.7847±0.0051
Multi-Region Histograms [63]	0.7295±0.0055
V1-like/MKL [32]	0.7935±0.0055
LDML, funneled [28]	0.7927±0.0060
SVM + OSS [20]	0.7637±0.0065
POEM, aligned [29]	0.7542 ±0.0071
Hybrid, aligned [46]	0.8398±0.0035
Combined b/g samples based [21]	0.8683±0.0034
Attribute and Simile classifiers [27]	0.8529± 0.0123
Single LE + holistic, aligned [31]	0.8122±0.0053
Multiple LE + comp, aligned [31]	0.8445±0.0046
CSML + SVM, aligned [34]	0.8800 ±0.0037
Our method (PLS + OSS)	**0.8533±0.0038**
Our method (PLS + OSS, flip)	**0.8612±0.0047**

Table 2.2: Mean (± standard error) classification accuracy on the LFW dataset, compared to Image-Restricted Training benchmark using the PLS One-Shot Model.

methods on the LFW benchmark (we achieved 86.12% classification accuracy). On the LFW benchmark, Wolf's work[21] and the recently published CSML[34] have the best performance. Wolf's model has several layers and requires a large amount of training data. CSML uses the View 2 training data intensively to conduct their cross-validation error minimization based metric learning. Kumar[27] shows excellent result, marginally lower than ours. However, Kumar's work requires training high-level classifiers requiring a huge volume of images outside of the LFW dataset. The LE [31] method in the component-level relies on facial feature point detectors that have been trained with supervision. The method which is directly comparable is SVM+OSS [20].Overall, our approach achieves competitive accuracy without using any label information or local feature identification. Thus it could be easily generalized to other datasets.

2.5 Evaluations on GBU and BDCP Datasets

We also evaluated our PLS One-Shot Model based face verification algorithm on several other very challenging datasets collected under unconstrained environments.

For all the below experiments we use the same feature settings. For HOG features, we use block sizes of 16×16 and 8×8 with strides of 4 and 4 pixels, respectively. For LBP features, we use block size of 16×16 with strides of 8 pixels. The Gabor features have 5 scales and 8 orientations, down sampled by a factor of 4. The PLS factor (number of latent vectors p) is set to 11. The reference set is similar

to LFW, which contains around 500 images captured in the similar (not not too different) as the test set. The faces in GBU [64], BDCP are cropped and re-scaled to 128×160, 60×72respectively.

Sample faces from GBU dataset (ugly partition) are shown in Figure 2.9.

Figure 2.7: Sample face images from the ugly Partition from the GBU database.

Sample faces from BDCP dataset are shown in Figure 2.8.

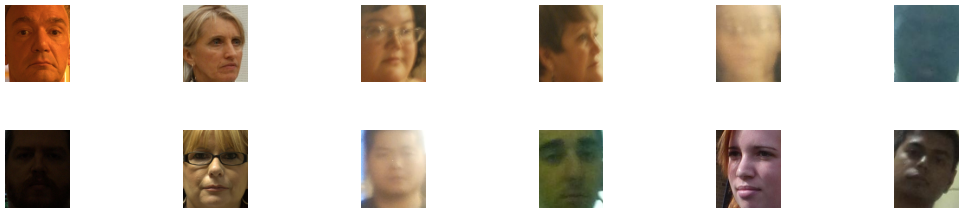


Figure 2.8: Sample face images from the ugly Partition from the BDCP database.

2.5.1 Evaluations on GBU dataset

The Good, the Bad, and the Ugly Face Challenge Problem [64] was created to encourage the development of algorithms that are robust to recognition across changes that occur in still frontal faces. The Ugly partition contains pairs of images considered difficult to recognize: allowing greater variability in pose, ambient lighting, expression, size of the face, and distance from the camera.

Figure 2.9 shows the ROC curve for the LRPCA-ocular baseline algorithm and PLS on the the very challenging Ugly partitions.

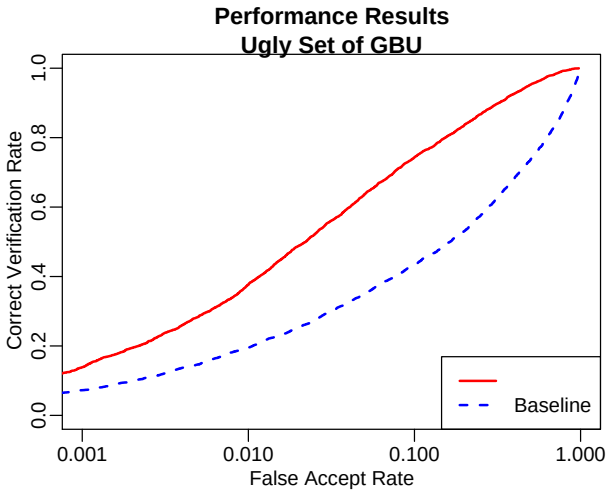


Figure 2.9: ROC plot for UGLY

2.5.2 Evaluations on BDCP

BDCP refers to BEST Development Challenging Problems. The target and query pairs are selected in the forms of (1) Frontal Near Field vs. Frontal; (2) Frontal Near Field vs. Frontal Far Field; (3) Frontal Near Field vs. NonFrontal Near Field; (4) Frontal Near Field vs. NonFrontal Far Field. Here, the near-field and far-field images are 6' and 15' from the camera respectively.

Figure 2.10 and Figure 2.11 shows the ROCs for the LRPCA-ocular baseline algorithm and PLS on the the four different masks. Again we can see that PLS outperforms the baseline at a large margin.

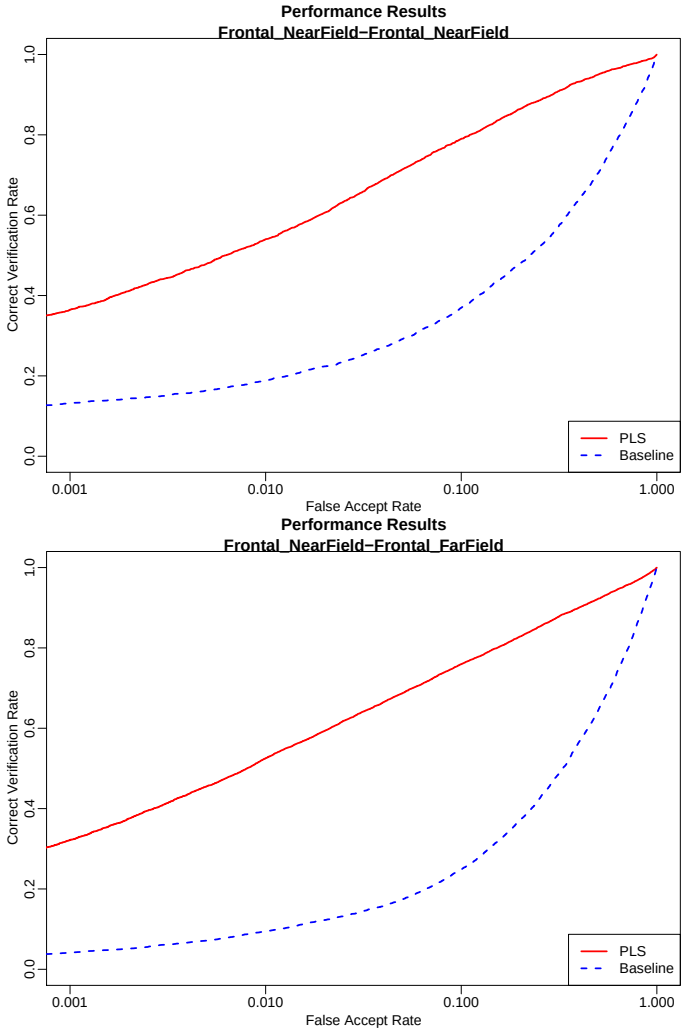


Figure 2.10: ROC plot comparison with baseline. (a) ROC plot for FNF VS. FNF.; (b) ROC plot for FNF VS. FFF.

2.6 Evaluation on the Maritime Dataset

2.6.1 Maritime Database

In order to study and develop more robust algorithms for unconstrained face recognition and verification, we have put together a remote face database in which a

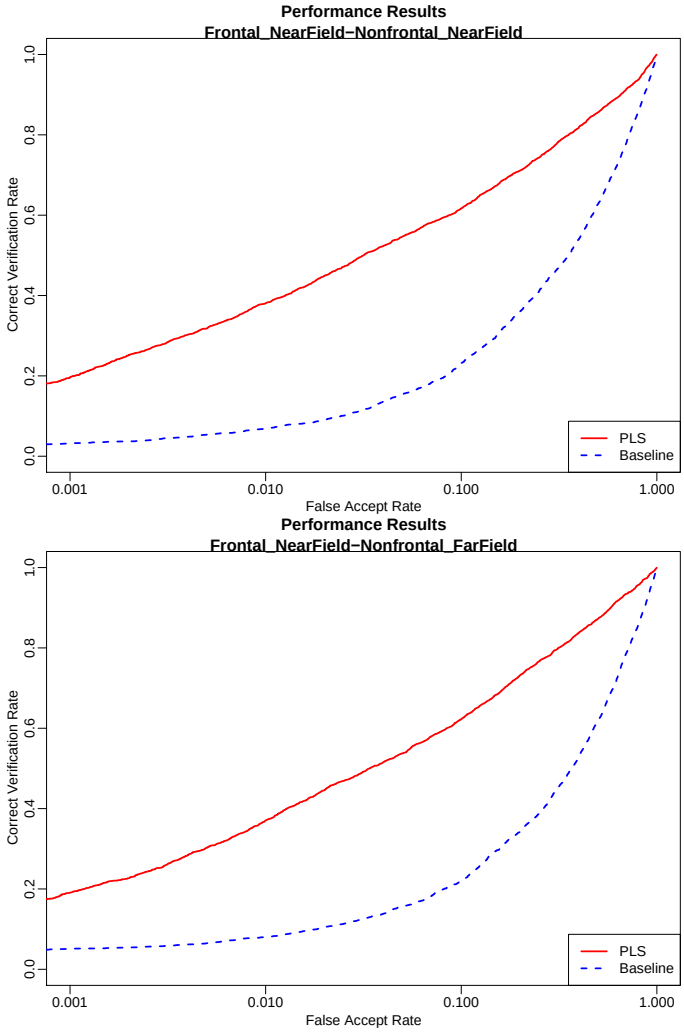


Figure 2.11: ROC plot comparison with baseline. (a) ROC plot for FNF VS. NFNF.; (b) ROC plot for FNF VS. NFF.

significant number of images are taken from long distances and under unconstrained outdoor environments [65]. The quality of the images differs in the following aspects: the illumination is not controlled and is often pretty bad in extreme conditions; there are pose variations and faces are also occluded as the subjects are not cooperative; finally, the effects of scattering and high magnification resulting from long distance contribute to the blurriness of face images. Figure 2.13 shows the sample scenarios

in the maritime domain. We manually cropped and organized the face images according to different illumination conditions, resolution, pose, blur or no-blur etc in a systematic way so that users can conveniently select the desired images for their experiments.

Due to uncontrolled data collection, there are numerous sources of variations in the captured face images. Further, these variations are usually co-present. For instance, a face image can have both blur and illumination variations, or pose variation and blur etc. To enable a systematic study of the effect of each variation, as well as not to exhaust all possible combinations of variations, we organized all the face images into several partitions based on the major variations we observe in the dataset. We describe the details of the partitions as follows. First, we picked about 90 images with very low resolution to form the *low resolution* folder. As only few images are fully frontal, we divided the remaining images into two folders *near frontal* (1166 images) and *non-frontal* (846 images). The *near frontal* folder contains images less than 10^o away from frontal, while *non-frontal* contains images with large pose variation. We do the following partitions on the *near frontal* folder. We first selected five clear, well illuminated images per subject to form the *clear* folder (This folder is later used as gallery for identification tasks and target for verification tasks). Then we selected images mainly with blur effect, images mainly with illumination variation, and images with both blur and illumination variations to form the *blur*, *illum* and *illum blur* folders, respectively. The partitions result in 75 images in the *blur* folder, 561 images in the *illum* folder and 128 images in the *illum blur* folder. Figure 2.12 shows the structure of partitions. Figure 2.14 shows

some representative images from different partitions.

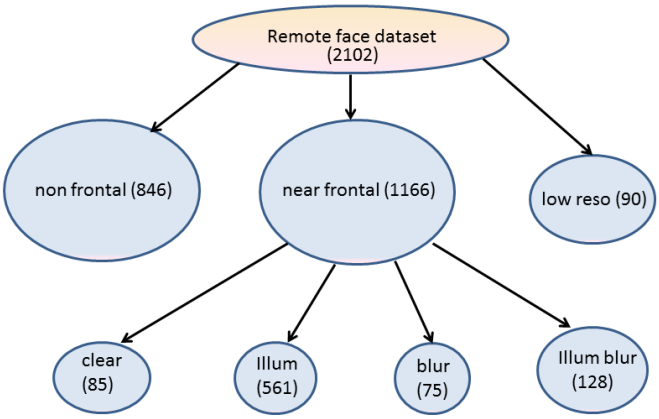


Figure 2.12: Structure of the database partitions with the number of images in each partition.

As can be seen from Figure 2.14(a), the captured images can be of very low resolution, with a typical resolution of 20 by 30 pixels. Moreover, low resolution images are often coupled with blurring effects. Also, large out-of-plane pose variations are observed as shown in Figure 2.14. Since the distance between the camera and subjects is large, high magnification blur can be seen from Figure 2.14(d). Furthermore, due to the motion between camera and subjects, some of the images also suffer from motion blur. Finally, in some of the images, we see the presence of both blur and poor illumination condition.

2.6.2 Experimental Setup

For HOG features, we use block sizes of 16×16 and 8×8 with strides of 4 and 4 pixels, respectively. For LBP features, we use block size of 16×16 with strides of

Figure 2.13: Typical images illustrating the different scenarios in the maritime domain.

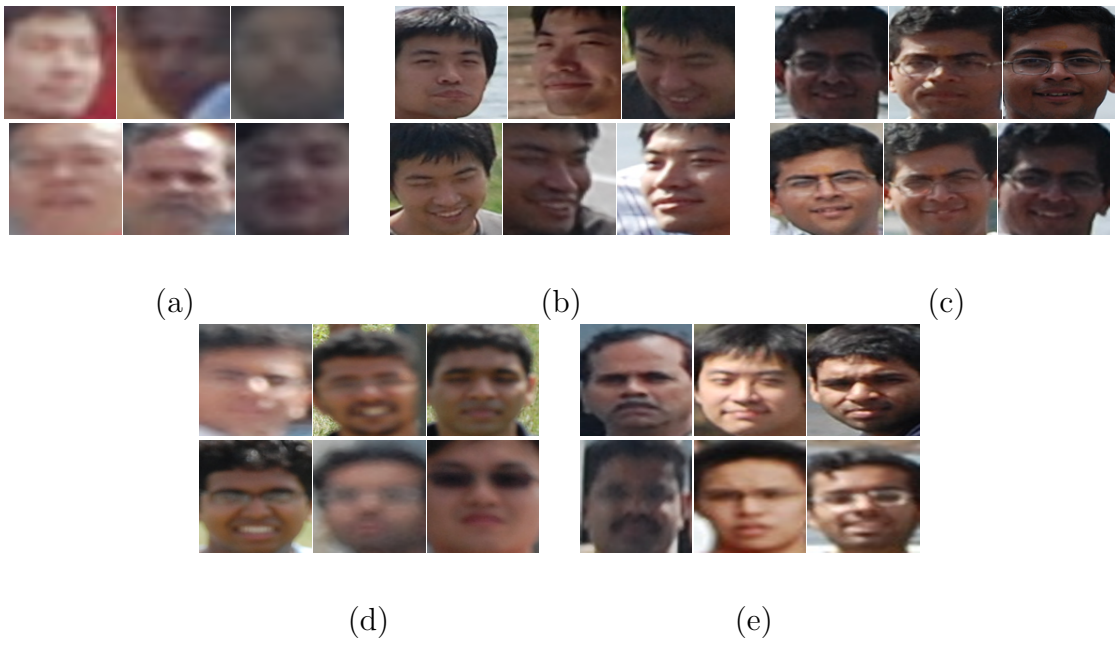


Figure 2.14: Sample cropped face images from different folders. (a) Sample images from the *low resolution* folder. (b) Sample images from the *non-frontal* folder. (c) Sample images from the *illum* folder. (d) Sample images from the *blur* folder. (e) Sample images from the *illum blur* folder (best viewed in color).

8 pixels. The Gabor features have 5 scales and 8 orientations, down sampled by a factor of 4. The PLS factor (number of latent vectors p) is set to 11. The reference set is similar to LFW, which contains around 500 images captured in the similar (not not too different) as the test set. The faces in GBU, BDCP are cropped and re-scaled to 128×160, 60×72respectively.

We use the images from the *clear* folder as the target, and the images from the *illum*, *blur*, *illum blur*, *low resolution* and *non-frontal* folders as query, respectively. Each algorithm is required to produce a similarity score of every two images in the dataset, which results in a $2,102 \times 2,102$ similarity matrix. We provide five binary mask matrices of ground truth for target/query match pairs:

1. *clear* vs. *illum*
2. *clear* vs. *blur*
3. *clear* vs. *illum blur*
4. *clear* vs. *low reso*
5. *clear* vs. *non-frontal*.

These masks are used to extract corresponding subset of the similarity matrix to compute the ROC curve in each scenario. In this way, the similarity score between a target image and a query image does not depend on other images in the target and query sets. Also redefining similarity which depends on the target or query set is not allowed.

We used 1000 images from FERET and LFW to form the training set. The same training set is used for all the algorithms evaluated. Note the verification procedure is a bit different from identification as follows:

Principal Component Analysis: We used 1000 training images to train a PCA model. Then, for each pair of test images, we project them onto the PCA model and the coefficients from the projection are used to compute the similarity.

Sparse representation: We use 1000 training images as a dictionary, and do sparse coding of each pair of images using this dictionary. The cosine similarity between the sparse codes is used to compare the two images.

Face Verification with PLS One-Shot Model: We used 1000 images from FERET and LFW to form the fixed reference set.

2.6.3 Results Analysis

The ROC curves corresponding to different methods are shown in Figure 2.15(a)-(e).

Again, we observe that, the sparse representation-based algorithm performs the worst in all cases. Sparse representation is an intuitively appealing method for rank-1 face recognition by containing multiple images per subject in the training dictionary. However, it is not straightforward to apply sparse representation-based algorithm for verification since verification is not a multiclass problem and one is given only a single image. Moreover, the identities in the testing stage (our database here) are always disjoint from the training set (FERET and LFW faces). Hence,

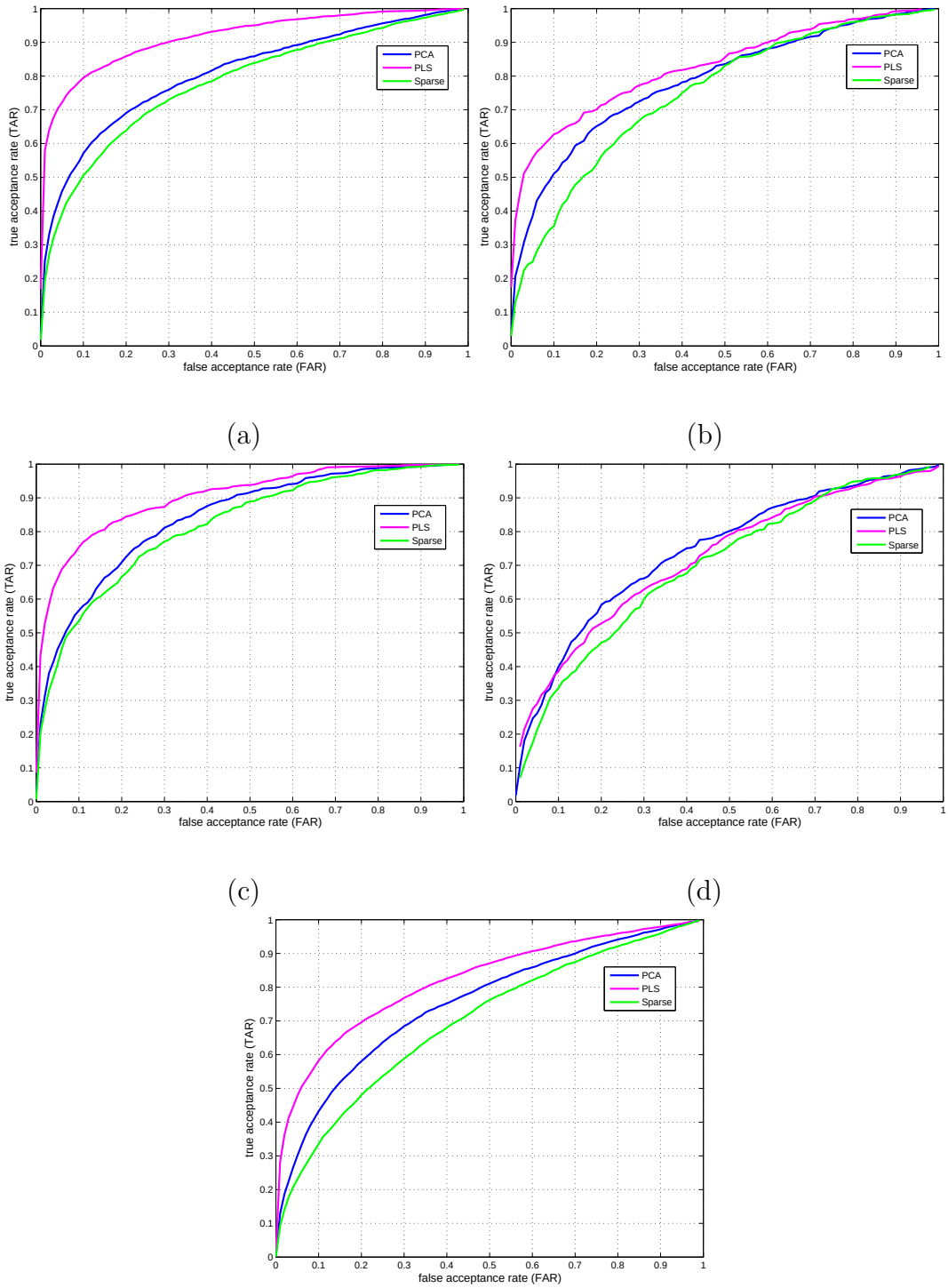


Figure 2.15: ROC plots for (a) *clear* vs. *illum* (b) *clear* vs. *blur* (d) *clear* vs. *illum blur* (d) *clear* vs. *low reso* (e) *clear* vs. *non-frontal*.

Table 2.3: Verification rates of the baseline, PLS and sparse representation based methods at FAR = 0.1.

	Methods		
Experiment	Baseline	Sparse representation	PLS
clear vs. *illum*	0.5407	0.5285	**0.7946**
clear vs. *illum blur*	0.5664	0.5351	**0.7541**
clear vs. *blur*	0.5107	0.3699	**0.6272**
clear vs. *low reso*	**0.3993**	0.2755	0.3859
clear vs. *nonfrontal*	0.4320	0.2970	**0.5820**

there is no guarantee that an image from a training set can be well reconstructed by the the atoms in the dictionary that is formed from the training images.

PLS One-Shot Model-based face verification achieves generally better results than the other two algorithms. This model is versatile: it performs multi-channel feature weighting on a rich feature set and the PLS regression response can be used efficiently to construct a similarity measure. The One-Shot learning builds discriminative models online exclusively to the pair being compared. The result on the *clear* vs. *low reso* experiment is unsatisfactory since the feature extraction does not contribute much in low resolution images. This was also observed in the identification experiment as well.

A summary of performance of the baseline, PLS and sparse representation-based algorithms at FAR = 0.1 is given in Table 2.15. The best rates for each

experiment are shown in bold letters.

2.7 Summary

We proposed a robust face verification approach based on PLS One-Shot model. This model is versatile - it performs multi-channel feature weighting on a rich feature set and the PLS regression response can be used efficiently to construct a similarity measure. The One-Shot learning builds discriminative models online exclusively to the pair being compared. A small set of unlabeled images used as the reference (negative) set is all that is needed. The approach was evaluated on the LFW benchmark and showed very comparable results to the state-of-the-art methods (image-restricted setting). When compared with other methods using the unsupervised setting, the proposed method outperformed them by a large margin. The verification results on the other three very challenging real world datasets (GBU, BDCP, Maritime) taken in unconstrained environments also demonstrate the robustness of our algorithm.

Chapter 3

Face Recognition from Sets of Images using Covariance Discriminative Learning

3.1 Face Recognition Based on Image Sets

Face recognition has traditionally been posed as the problem of identifying a face from a single image, and many methods assume that images are taken in controlled environments. However facial appearance changes dramatically under variations in pose, illumination, expression, etc., and images captured under controlled conditions may not suffice for reliable recognition under the more varied conditions that occur in real surveillance and video retrieval applications. In this Chapter, we focus on recognition problems using sets (multiple images) as input rather than single images.

Classification based on image sets has recently attracted growing interest in the computer vision and pattern recognition community [66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77]. This problem naturally arises in a wide range of applications including video surveillance, classification based on images from multi-view cameras and photo albums, and classification based on long term observations. In the task of image set classification, each set generally contains a large number of images (Figure 3.1) that belong to the same class and cover large variations in the objects

appearance due to camera pose changes, non-rigid deformations or different lighting conditions. While traditional recognition methods based on single-shot images have achieved a certain level of success under restricted conditions, more robust object recognition can be expected by using sets as input rather than single images. This is mainly because the image set incorporates useful data variability information, which can be efficiently exploited under more realistic conditions with significantly larger variations [66, 67, 68, 69].

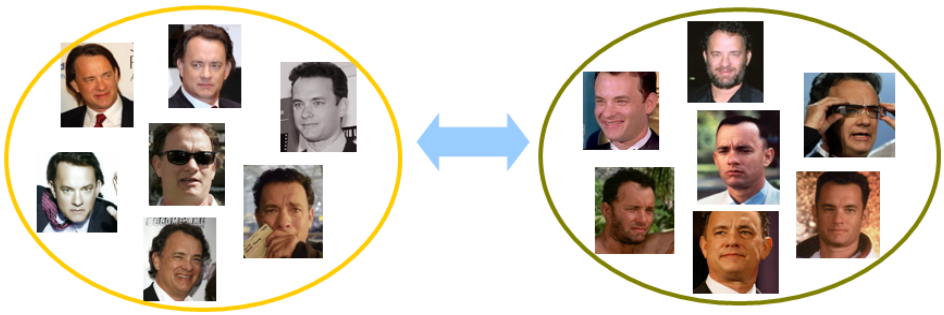


Figure 3.1: Image set classification for unconstrained face recognition.

Among the previous work, there is a category of video-based classification methods [70, 71] which focus on utilizing the temporal dynamic information between consecutive video frames. However, in the general scenario of image set classification, the images in a set are collected not necessarily from video sequences but possibly from multiple unordered observations - for example, in face recognition they could be images of a subject from photo albums or from surveillance systems where the subject might not face the camera all the time [68], so that images suitable for recognition are widely spaced in time and location. As this study mainly addresses

the general image set classification problem and does not explicitly exploit any assumption on data semantics, we will not compare our technique against methods using video dynamics.

3.2 Related Work on Image Set Classification

For image set classification, existing methods mainly focus on the key issues of how to model the image sets and how to measure their similarity. In most cases, the similarity function is defined specifically for certain image set modeling or representation methods. As far as set modeling is concerned, related approaches to image set classification broadly fall into two categories: model-based parametric methods and model-free nonparametric methods. Representative parametric methods include probabilistic models [72] and manifold density divergence [66]. They tend to represent each image set by a parametric distribution function and then measure the similarity between two distributions in terms of the Kullback-Leibler Divergence (KLD). In [72], face pattern variations are modeled by a relatively simplistic single Gaussian distribution in the face space. For more realistic and satisfactory modeling, Gaussian mixture models (GMM) were used in [66] instead. While these parametric methods have shown promising results in many applications, they typically need to solve a difficult parameter estimation problem and may have large performance fluctuations in cases where the training and novel test data sets have weak statistical correlations [68, 69].

In comparison, nonparametric methods typically relax the assumptions on

distributions of the set data, and try to model the image set in a more flexible manner. One class of prevalent methods is to use subspace learning techniques to account for the set data variability globally, following the pioneering work of is to use subspace learning techniques to account for the set data variability globally, following the pioneering work of [75]. These methods attempt to represent the image set either by a single linear subspace [68, 76, 77] or by a more sophisticated manifold in the form of a mixture of linear subspaces [67, 69]. To measure the subspace distance, the method of principal angle [78]is mainly exploited to capture the common data variation modes of two subspaces. Since they impose a uniform prior over data variations in different image sets, nonparametric methods have been shown to have many favorable properties [68, 69]. However, for appropriate manifold modeling, they typically require a large data set with dense sampling, while the linear subspace modeling does not well accommodate the case when the set is of small size but has large and complex data variations. As also indicated in [74], the linear subspace-based modeling has the limitation that it incorporates only relatively weak information (subspace angles) about the locations of the samples in the input feature space.

More recently, a new type of nonparametric methods [74], [79] based on matching the closest pair of points from two image sets has been introduced. In [73], a straightforward strategy is adopted to find the nearest actual sample images from the two sets without considering data variations across the set. In contrast, [74], [79] approximate the image set with a more theoretically principled affine subspace model and match the closest virtual points via a convex optimization. While intra-

class variations can be effectively handled, such methods are still susceptible to the presence of outliers and have relatively high computational cost [68, 73], due to their inherent single sample-based matching mechanism.

3.3 Overview of our approach

We propose [80] a novel Covariance Discriminative Learning (CDL) approach to image set classification. By representing each image set with its natural second-order statistic - covariance matrix - we formulate the problem as classifying points lying on a Riemannian manifold spanned by SPD matrices, i.e., nonsingular covariance matrices. Since classical learning algorithms cannot take points on the manifold as their direct input, we explore an efficient metric for the SPD matrices, i.e., Log-Euclidean distance (LED), and further derive a kernel function that explicitly maps the covariance matrix from the Riemannian manifold to a Euclidean space. Benefiting from this explicit kernel feature mapping, any learning method originally developed for vector spaces can be used, by taking either the Log-mapped covariance matrices as input to its linear formulation or the derived kernel function as input to its kernel formulation. A conceptual illustration of our approach is shown in Figure 3.2.

Here we exploit two representative methods - Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS), for their feasibility for our specific case where the number of samples (i.e., the number of image sets) is considerably smaller than the number of feature dimensions (i.e., the number of covariance matrix en-

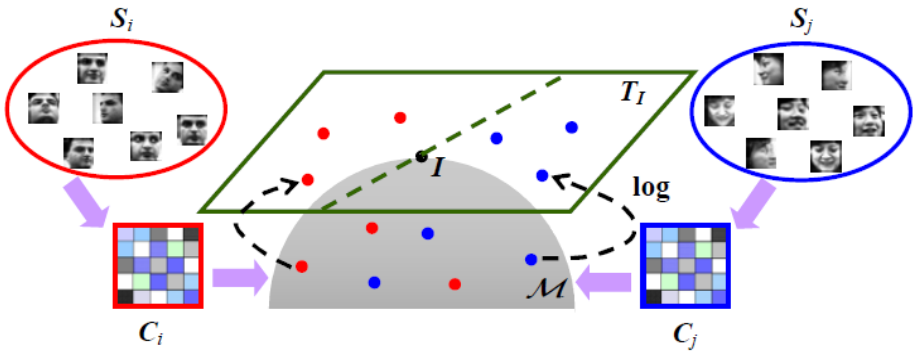


Figure 3.2: Conceptual illustration of the proposed CDL method. We model the image set S by its sample covariance matrix C, and formulate the problem as classifying points on the Riemannian manifold M. With the log map, traditional learning methods can be utilized in the tangent space T_I (which is a vector space) at the point of the identity matrix I on the manifold.

tries).

3.3.1 Set Modeling by Covariance Matrix

Let $S = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ be the data matrix of an image set with N samples, where $\mathbf{x}_i \in \mathbf{R}^{D\times 1}$ denotes the i-th sample with D -dimensional feature description. Here, the image intensity is used as the raw feature. We represent the image set with the $D \times D$ sample covariance matrix:

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \tag{3.1}$$

where $\overline{\mathbf{x}}$ is the mean of the image samples. The diagonal entries of the covariance matrix represent the variance of each individual feature, and the non-diagonal entries are their respective correlations.

While it is rather simple to derive and compute, there are several advantages to model the image set with its covariance matrix. As the raw second order statistic of a set of samples, the covariance matrix makes no assumption about the set data distribution, thus providing a natural representation for an image set with any number of samples and any type of features. The representation leads to an effective way to discriminate image sets of different classes by encoding the feature correlation information specific to each object class. Compared with previous single or mixture of linear subspace based methods [67, 69, 75, 77], the covariance matrix characterizes the structure of the set more realistically. In fact, the linear subspace is usually obtained by principal component analysis (PCA) of the image set, which reduces to eigen-decomposition of the covariance matrix. In this processing, the leading eigenvectors are typically extracted to serve as the subspace basis while the remaining eigenvalues are simply discarded. This makes the resulting subspace too loose to reflect the set distribution boundary. Compared to previous closest sample pair based methods [73, 74], the covariance matrix representation shows strong resistance to outliers, since it is a statistic of all samples and the noise corrupting samples are largely filtered out with an average filter during covariance computation.

Prior to our study here, covariance matrices have been used to characterize local regions within an image, named region covariance [81], and applied to several visual processing tasks such as object detection/recognition, object tracking and

texture classification [82, 83]. It should be noted that as a region descriptor, the covariance matrix in these works has several differences from the image set descriptor in this paper. For region covariance, each pixel of the image region is a sample, and sample features include pixel coordinate, intensity, and the first-order and second-order gradient. Since the number of pixels in the region is usually larger than the feature dimension, the region covariance matrix can generally be guaranteed to be nonsingular. However, in image set classification it is often the case that the number of images is less than the feature dimension, i.e. $N < D$. To avoid the singularity of the covariance matrix, a simple method is to add a small perturbation to the original covariance matrix: $\mathbf{C}^* = \mathbf{C} + \lambda\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In our experiments, λ is set to $10^{-3} \times trace(\mathbf{C})$ in case that a singular covariance is to be used for computing distance. Furthermore, when the covariance matrix is utilized as individual sample for learning algorithms, the number of samples (number of covariance matrices) is considerably smaller than the number of feature dimensions (number of covariance matrix entries) for our set covariance case, which is entirely opposite to the case of region covariance learning [81]. This will be the topic of the next section.

3.3.2 Covariance Discriminative Learning

It is well known in Riemannian geometry that the $d \times d$ SPD matrices (i.e., nonsingular covariance matrices) Sym_D^+ do not lie in a Euclidean space but on a Riemannian manifold. We naturally formulate the problem of image set classification

as classifying points lying on the Riemannian manifold spanned by SPD matrices. However, it is not trivial to learn a classifier on the manifold since classical learning algorithms are devoted to operating in vector space associated with Euclidean metrics and thus cannot take points on the manifold as their direct input. We next explore Riemannian metrics for covariance matrix by emphasizing the Log-Euclidean distance (LED) and then develop efficient learning algorithms associated with this metric.

3.3.2.1 Riemannian Metrics for Covariance Matrix

Here we introduce two different formulations of distance metric for Sym_D^+ that have been well established in the filed of Riemannian geometry. The first metric, affine-invariant distance (AID) [84], is defined in terms of the generalized eigenvalues of two covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$:

$$d_{AID}(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{i=1}^{D} ln^2 \lambda_i(\mathbf{C}_1, \mathbf{C}_2)} \tag{3.2}$$

where the eigenvalues $\lambda_i(\mathbf{C}_1, \mathbf{C}_2)$are computed from $|\lambda \mathbf{C}_1 - \mathbf{C}_2| = 0$. This metric is invariant under affine transformations and inversion, and has been mainly used as the distance measure for region covariance [81, 82, 83].

Another distance metric for Sym_D^+ is the Log-Euclidean distance (LED) [85] that results in classical Euclidean computations in the domain of matrix logarithms as follows:

$$d_{AID}(\mathbf{C}_1, \mathbf{C}_2) = \| \, log(\mathbf{C}_1^{-1/2}\mathbf{C}_2\mathbf{C}_1^{-1/2})) \, \|_F \tag{3.3}$$

where log is the ordinary matrix logarithm operator and $\| \cdot \|_F$ means the matrix Frobenius norm. Let $\mathbf{C} = \mathbf{U}\Sigma\mathbf{U}^T$ be the eigen-decomposition of a SPD matrix $\mathbf{C}$, its logarithm is a symmetric matrix and can be computed easily by

$$log(\mathbf{C}) = \mathbf{U} log(\Sigma)\mathbf{U}^T \tag{3.4}$$

where $log(\Sigma)$ is the diagonal matrix of the eigenvalue logarithms. The LED metric is particularly simple to use and avoids the computational limitations of the AID metric, while conserving excellent theoretical properties. Please refer to [85] for more detailed discussion on the similarities and differences between the two metrics.

The LED metric can be understood as projecting a point $\mathbf{C}$ on the Riemannian manifold M to a Euclidean space via the logarithm map:

$$\Psi_{log} : M \mapsto T_I, \mathbf{C} \to log(\mathbf{C}) \tag{3.5}$$

The image $\Psi_{log}(M)$ is the tangent space T_I at the point of identity matrix $\mathbf{I}$, which is a vector space spanned by the $d \times d$ symmetric matrices. The LED metric thus simply reduces to a Euclidean distance in $R^{d \times d}$. By computing inner product in the Euclidean space T_I, we actually derive a $Riemanniankernelfunction$ on the manifold M:

$$k_{log}(\mathbf{C}_1, \mathbf{C}_2) = tr[log(\mathbf{C}_1) \cdot log(\mathbf{C}_2)]. \tag{3.6}$$

It is easy to check that k_{log} is a symmetric real-valued function: $k_{log}(\mathbf{C}_i, \mathbf{C}_j) = k_{log}(\mathbf{C}_j, \mathbf{C}_i)$ for all $\mathbf{C}_i, \mathbf{C}_j \in M$. The positive definiteness of this function follows from the properties of Frobenius norm. For all $\mathbf{C}_1, ..., \mathbf{C}_n (\mathbf{C}_i \in M)$ and $b_1, ..., b_n (b_i) \in R$ for any $n \in R$, we have

$$\begin{aligned}\sum_{i,j} b_i b_j k_{log}(\mathbf{C}_i, \mathbf{C}_j) &= \sum_{i,j} tr[log(\mathbf{C}_i) \cdot log(\mathbf{C}_j)] \\ &= tr[(\sum_i b_i log(\mathbf{C}_i))^2] \\ &= \| \sum_i b_i log(\mathbf{C}_i) \|_F^2 \geq 0\end{aligned}$$

With these properties, the proposed Riemannian kernel in Eq 3.6 is proven to satisfy the Mercer's theorem [86]. It is interesting to note that traditional kernel functions (e.g., Gaussian kernel, polynomial kernel) are usually defined on a Euclidean space, and implicitly map the points from this Euclidean space to another higher dimensional Euclidean space, i.e., the so-called RKHS (reproducing kernel Hilbert space) feature space [86]. In contrast, our kernel function is defined on an unconventional Riemannian manifold and explicitly maps the points from the manifold to a Euclidean space through Eq. 3.5.

The explicit kernel feature mapping allows us to utilize any standard learning algorithm in vector space. We can either apply the linear formulation of the method to the Euclidean space T_I by taking the Log-mapped covariance matrices $log((C))$ as input, or apply its kernel formulation to the manifold M by taking the covariance

matrices (C) and the derived kernel function k_{log} as input. Let $D = d^2$, the $d \times d$ matrices are represented as D-dimensional sample vectors in these algorithms. As discussed in above section, for our set covariance learning, the number of samples is rather smaller than the number of feature dimensions, thus making the kernel formulation especially suited to this special case for the sake of efficiency. In the following, we explore two typical learning methods LDA and PLS for their feasibility, by focusing on their kernel formulations. The former learns a discriminant subspace and maps the samples to this subspace followed by Nearest Neighbor (NN) classification, while the latter directly learns a regression model between the observed samples and their corresponding class labels.

3.3.2.2 Learning with LDA with its Kernel Variant

Linear Discriminant Analysis (LDA) has proven to be an effective method for classification problems. Suppose we have a set of m samples $x_1, x_2, ..., x_m \in R^D$ belonging to c classes in the input data space. The kernel variant of LDA (KLDA) [87, 88] formulates the problem using the kernel trick as follows. Let $\phi : R^D \mapsto F$ be the feature map, an inner product can be defined on the feature space F with the kernel function as: $< \phi(x_i), \phi(x_j) >= k(x_i, x_j)$. Let $\mathbf{S}_b^\phi$, $\mathbf{S}_w^\phi$ and $\mathbf{S}_t^\phi$ denote the between-class, within-class and total scatter matrices in the feature space respectively, we have

$$\mathbf{S}_b^\phi = \sum_{k=1}^{c} m_k(\mu_\phi^{(k)} - \mu_\phi)(\mu_\phi^{(k)} - \mu_\phi)^T,$$
$$\mathbf{S}_w^\phi = \sum_{k=1}^{c}\sum_{i=1}^{m_k}(\phi(x_i^{(k)}) - \mu_\phi^{(k)})(\phi(x_i^{(k)}) - \mu_\phi^{(k)})^T, \tag{3.7}$$
$$\mathbf{S}_t^\phi = \mathbf{S}_b^\phi + \mathbf{S}_w^{phi} = \sum_{i=1}^{m}(\phi(x_i) - \mu_\phi)(\phi(x_i) - \mu_\phi)^T$$

where $\mu_\phi^{(k)}$ and μ_ϕ are the centroid of the k-th class and the global centroid, respectively in the feature space. m_k is the number of samples in the k-th class.

KLDA seeks the optimal discriminant direction $\mathbf{w}$ by solving the following optimization problem

$$\mathbf{w}_{opt} = argmax\frac{\mathbf{w}^T\mathbf{S}_b^\phi\mathbf{w}}{\mathbf{w}^T\mathbf{S}_w^\phi\mathbf{w}} = argmax\frac{\mathbf{w}^T\mathbf{S}_b^\phi\mathbf{w}}{\mathbf{w}^T\mathbf{S}_t^\phi\mathbf{w}} \tag{3.8}$$

By representer theorem, the optimal $\mathbf{w}$ in space F can be written as: $\mathbf{w}_{opt} = \sum_{i=1}^{m} a_i\phi(x_i)$. Let $\alpha = [a_1, ...a_m]^T$, it can be proven [87] that Eq. 3.3.2.2 is equivalent to:

$$\alpha_{opt} = argmax\frac{\alpha^T\mathbf{KWK}\alpha}{\alpha^T\mathbf{KK}\alpha} \tag{3.9}$$

The optimal α are given by the eigenvectors with respect to the largest eigenvalues of the following eigen-problem: $\mathbf{KWK}\alpha{=}\lambda\mathbf{KK}\alpha$, where $\mathbf{K}$ is the kernel Gram matrix: $K_{ij} = k(x_i, k_j)$ and $\mathbf{W}$ is defined as:

$$\mathbf{W}_{i,j} = \begin{cases} 1/m_k, & \text{if } \mathbf{x}_i, \mathbf{x}_j \text{ are both in the } k\text{-th class} \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

Each eigenvector α gives a direction vector $\mathbf{w}$ in the feature space. Grouping the maximum number $(c-1)$ of eigenvectors, we get $\mathbf{A} = [\alpha_1, ..., \alpha_{c-1}]$. For a data example $\mathbf{x} \in R^D$ in the input space, its $c-1$-dimensional projection in the discriminant subspace can be obtained by

$$\mathbf{z} = \mathbf{A}^T \mathbf{K}_t, where \mathbf{K}_t = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_m, \mathbf{x})]^T \tag{3.11}$$

For our set covariance learning, suppose we are given m gallery image sets $\mathbf{S}_i^g (i = 1, ..., m)$ with known class labels for training, and t probe image sets $\mathbf{S}_j^p (i = 1, ..., t)$ for testing. We first compute their corresponding covariance matrices $\mathbf{C}_i^g$, $\mathbf{C}_j^p$, and represent them by D-dimensional sample vectors. The training samples $\mathbf{C}_i^g$ and the proposed Riemannian kernel in Eq. 3.6 are then fed into KLDA to solve the optimization in Eq. 3.3.2.2. In the testing phase, both $\mathbf{C}_i^g$ and $\mathbf{C}_j^p$ are projected to the discriminant subspace through Eq. 3.3.2.2. NN classification in this c-1-dimension subspace is then conducted based on Euclidean distance.

3.3.2.3 Learning with PLS and its Kernel Variant

PLS was explained in Chapter 2. In the kernel formulation of PLS (KPLS) [89], we keep using the same notations as in KLDA for similicity. The basic idea of KPLS is to map the original X -space data into a RKHS feature space $\mathbf{F}$ with $\mathbf{R}^D \mapsto \mathbf{F}$, and

perform the kernel form of the NIPALS algorithm [16]. Let $\Phi = [\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_m)]^T$ be the feature matrix of the training points, then the kernel Gram matrix is written as $\mathbf{K}$=$\Phi\Phi^T$. Then the regression coefficients $\mathbf{B}$ in the feature space will have the form

$$\mathbf{B} = \Phi^T\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} \tag{3.12}$$

For a testing data sample $\mathbf{x} \in R^D$ in the X space, its KPLS prediction (class label) in the Y space can be obtained by

$$y_{predict} = [\phi(\mathbf{x})]^T\mathbf{B} = \mathbf{K}_t^T(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} \tag{3.13}$$

When applied to set covariance learning, we use the gallery image set $\mathbf{S}_i^g (i = 1, ..., m)$ and their associated class labels to learn the KPLS latent model. Specifically, all training covariance matrices $\mathbf{C}_i^g$, represented as D-dim sample vectors as in KLDA, are gathered to build the *predictor* matrix $\mathbf{X}_m \times D$. For each $\mathbf{C}_i^g$, we define its class membership indicator vector: $\mathbf{y}_i = [0, ..., 1, ..., 0]^T \in R^c$, where the k-th entry being 1 and all other entries being 0 indicates that $\mathbf{C}_i^g$ belongs to the k-th class. The *response* matrix $\mathbf{Y}_m \times c$ can then be easily constructed with y_i^T as its row vector. Taking our Riemannian kernel, KPLS is used to learn the regression model in Eq.(14). In the testing phase, given a probe image set $\mathbf{S}_j^p$ and the corresponding covariance matrix $\mathbf{C}_j^p$, its class membership indicator vector $\mathbf{y}_t$ can be computed from Eq.(15) by treating $\mathbf{C}_j^p$ as a testing example $\mathbf{x}_t$. The entry index with the

largest response in $\mathbf{y}_t$ then determines the class label of $\mathbf{S}_j^p$.

While PLS operates in a very different manner from LDA for classification, there in fact exists close theoretical connection between the two methods, as has been well studied in [57, 22]. In comparison to LDA, PLS has proven to be useful in situations where the number of observed variables (i.e., D) is much larger than the number of observations (i.e., m). This is just the case for our set covariance learning where $D = 160,000$ and $m < 150$. In addition, PLS is not limited by the $c - 1$ discrimination dimensions and may be more suitable in the situation of non-Gaussian class distributions in the feature space [57].

3.4 Experimental Results

3.4.1 Database

We test the proposed method on two visual classification tasks: face recognition with image sets and object categorization, and consider four datasets with different characteristics to ensure extensive evaluations. We give below a brief description of these databases.

Honda/UCSD [70]: This is the benchmark database for face recognition with image sets. It consists of 59 video sequences involving 20 different persons. Each video contains approximately 300-500 frames covering large variations in head pose and facial expression. We used a cascaded face detector [90] to collect faces in each video, and then resized each face to 20×20 gray image as [69, 74]. Histogram equalization was the only pre-processing used to eliminate lighting effects. Each

video generated an image set of faces.

CMU MoBo [91]: The MoBo (Motion of Body) database was originally collected for human identification from distance. There are 100 sequences of 25 different subjects. Each subject has 4 sequences captured in different walking situations: holding a ball, fast walking, slow walking, and walking on the incline. Each sequence has about 300 frames. Face image sets were constructed in the same way as above.

YouTube Celebrities [71]: This dataset was collected for face tracking and recognition in real world applications. It contains 1910 video clips of 47 celebrities, mostly actors/actresses and politicians, from YouTube. Each clip contains hundreds of frames, which are mostly low resolution and highly compressed. Compared with Honda and MoBo, this database is much more challenging as the videos exhibit very large variations in pose, illumination, expression, and other conditions. Face image sets were also constructed in the same way as above.

ETH-80 [92]: This is the benchmark database for object categorization. It contains images of 8 categories (apples, cars, cows, cups, dogs, horses, pears and tomatoes) with each category including 10 objects (e.g. 10 different dogs). Each object has 41 images of different views which form an image set. Object categorization is to classify an image set of an object into a known category (e.g. apple, car, etc.). 20×20 gray images were also used in our experiment.

Some examples of the the images for each dataset are shown in Figure. 3.3 (Honda), Figure. 3.4 (Mobo), Figure. 3.5 (Youtube), Figure. 3.6 (Eth-80), respectively.

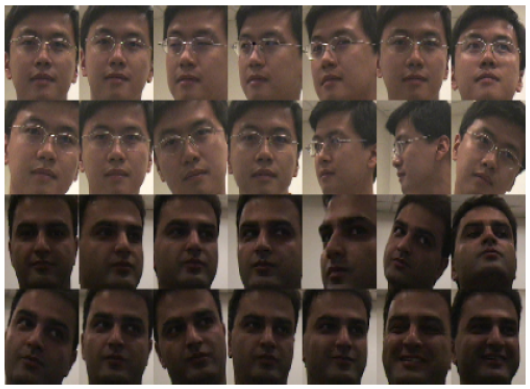


Figure 3.3: Some detected face images from videos of two subjects from the Honda/UCSD data set.

Figure 3.4: Some detected face images from videos of two subjects from the Mobo data set.

To allow comparison with the literature we followed the same protocol of [69, 74]. On all of the four datasets, we conducted ten-fold cross validation experiments, i.e., 10 randomly selected gallery/probe combinations, to report average identifica-

Figure 3.5: Some detected face images from videos of one subject from the Youtube data set.

Figure 3.6: Some sample sets of images (each column denotes one set) from the ETH-80 data set.

tion rates and standard deviations of different methods. Specifically, for both Honda and MoBo, each person had one image set as gallery and the rest sets for probe. For YouTube, in each fold, one person had 3 randomly chosen image sets for gallery and 6 for probe. For ETH-80, each category had 5 objects randomly chosen for gallery

and the other 5 objects for probe.

3.4.2 Comparative Methods and Settings

We compared the proposed approach with five representative image set classification methods in the literature. They include: Mutual Subspace Method (MSM) [75] and Discriminant Canonical Correlations (DCC) [68] which are based on linear subspace set modeling; Manifold-Manifold Distance (MMD) [69] and Manifold Discriminant Analysis (MDA) [93] which are based on nonlinear manifold modeling; Affine Hull based Image Set Distance (AHISD), Convex Hull Image Set Distance (CHISD) [74] and Sparse Approximated Nearest Point (SANP) [79] which are based on affine subspace modeling.

For fair comparison, the important parameters of each method were empirically tuned according to the recommendations in the original references as well as the source codes provided by the original authors. In MSM/ DCC/MMD, PCA was performed to learn the single or mixture of linear subspaces by preserving 95% of data energy. In MDA, the number of between-class NN local models and the subspace dimension were specified as [69]. For both AHISD and CHISD, we used their linear version and retained 95% energy by PCA. The error penalty in CHISD was set to C = 100 as [74]. For SANP, we adopted the same weight parameters as [79] for the convex optimization. Note that for the DCC learning on Honda and MoBo, the single training image set from each class was randomly divided into two subsets to construct the within-class sets, following the setting of [68, 74].

For our proposed framework, we tested two different combinations that include set modeling with covariance matrix (referred to as 'COV'), followed by discriminative learning with LDA or PLS (both using the kernel formulation). For covariance modeling, as stated in previous section to avoid the matrix singularity, regularization was applied to the original covariance matrix as: $\mathbf{C}^* = \mathbf{C} + \lambda\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In our experiments, λ is set to $10^{-3} \times trace(\mathbf{C})$. LDA/PLS utilized c-1 discriminant/latent dimensions. Since image sizes in all datasets are 20×20, the intensity feature dimension is d = 400 , and thus D = 160,000 . The number of gallery (training) image sets, m , is 20, 24, 141, 40 respectively for the four datasets. For the single sample per class learning with LDA on Honda and MoBo, the same strategy as that for DCC above was also adopted.

3.4.3 Results and Analysis

We summarize the recognition results of all methods on the four databases in Table 5.1. Each reported rate is an average over the ten folds of cross validation. Comparing the two COV based methods, we observe that PLS is better than LDA learning. Compared with other methods, the proposed COV+PLS delivers the highest rate on the benchmark databases (i.e. Honda/UCSD, YouTube, ETH-80) for 3 out of 4 all tasks compared with state-of-the-art methods.

In real-world applications, it is often the case that the image sets contain noisy data (i.e., images outside the category) and varying size. A desirable classification method should to be resistant to these challenges. Consider the face datasets Honda

Methods	Honda/UCSD	CMU MoBo	YouTube	ETH-80
MSM	0.925	0.852	0.611	0.878
MMD	0.971	0.902	0.629	0.863
DCC	0.980	0.881	0.648	0.905
MDA	1.000	0.943	0.653	0.890
AHISD	0.885	0.951	0.637	0.733
CHISD	0.905	0.940	0.663	0.735
SANP	0.936	0.963	0.684	0.755
COV+LDA (proposed)	0.980	0.867	0.675	0.945
COV+PLS (proposed)	**1.000**	**0.941**	**0.701**	**0.965**

Table 3.1: The mean recognition rates of different methods.

and MoBo for example. We experimentally study these two problems and evaluate the performance of different methods. Specifically, for the noisy set data problem, we follow [74] and conducted three experiments in which the gallery and/or the probe sets were systematically corrupted by adding one image from each of the other classes. The three cases are referred to as N_G (only gallery has noise), N_P (only probe has noise), and N_{G+P} (both). For the varying set size problem, we retained a certain number of samples from each image set (both gallery and probe) by uniform down-sampling and used the obtained subsets for classification. We tested four cases by extracting 200/100/50/25 samples, referred to as S_{200}, S_{100}, S_{50}, S_{25} respectively. In case a set contains fewer images than the specified number, the original set was used.

From the comparison results in Table 3.2 3.3 3.5, it can be seen that our proposed 'COV+PLS' shows high robustness against both challenges, with some slight performance drop. This can be mainly attributed to the advantages of using the covariance matrix as the set representation. For the noisy data case, the MSM/DCC/MMD/MDA seem more stable than the AHISD/CHISD/SANP since the former ones taking the set samples as a whole for subspace modeling and matching can alleviate the influence of noise samples to some extent. In contrast, based on matching the closest set points, the latter methods rely highly on the location of each individual sample and their model fitting can be heavily deteriorated by outliers. For the varying size case, we find that all other methods except AHISD/CHISD encounter problems to appropriately fit their models with decreased set size as expected. The unpredictable rate rise of AHISD/CHISD may also be explained by the

Honda/UCSD Dataset								
Methods	Clean	N_G	N_P	N_{G+P}	S_{200}	S_{100}	S_{50}	S_{25}
MSM	0.925	0.950	0.906	0.927	0.924	0.922	0.919	0.865
MMD	0.971	0.965	0.926	0.953	0.975	0.931	0.887	0.836
DCC	0.980	0.974	0.944	0.980	0.990	0.980	0.969	0.939
MDA	1.000	0.988	0.934	0.965	0.995	0.967	0.934	0.917
AHISD	0.885	0.890	0.808	0.805	0.913	0.915	0.939	0.964
CHISD	0.905	0.908	0.828	0.849	0.918	0.923	0.939	0.956
SANP	0.936	0.931	0.869	0.846	0.939	0.939	0.926	0.933
COV+PLS	**1.000**	**0.972**	**0.995**	**0.982**	**1.000**	**1.000**	**0.995**	**0.946**

Table 3.2: The mean recognition rates of different methods for Honda/UCSD Dataset.

fact that their model fitting is much sensitive to sample distribution.

Lastly, we compared the computational complexity of different methods with the benchmark Honda/UCSD dataset (m = 20) on a Pentium IV, 2.93 GHz PC. The time cost for each method is tabulated in Table 2. Training time is only needed by discriminant methods. Since kernel LDA/PLS learning in our method mainly involves the eigen-decomposition of mm kernel Gram matrix, they are very efficient. For testing, we report the classification time for matching one probe image set with the 20 gallery image sets. The superiority of our method can be clearly observed, especially over the three affine subspace based methods. As discussed in previous

CMU MoBo Dataset								
Methods	Clean	N_G	N_P	N_{G+P}	S_{200}	S_{100}	S_{50}	S_{25}
MSM	0.852	0.846	0.784	0.867	0.848	0.843	0.837	0.805
MMD	0.902	0.889	0.851	0.882	0.880	0.862	0.818	0.802
DCC	0.881	0.881	0.848	0.887	0.878	0.876	0.864	0.833
MDA	0.943	0.916	0.891	0.895	0.939	0.910	0.867	0.842
AHISD	0.951	0.927	0.835	0.748	0.956	0.967	0.954	0.941
CHISD	0.940	0.949	0.733	0.762	0.940	0.948	0.943	0.925
SANP	0.963	0.950	0.874	0.760	0.961	0.958	0.944	0.915
COV+PLS	**0.941**	**0.921**	**0.925**	**0.910**	**0.937**	**0.933**	**0.922**	**0.902**

Table 3.3: The mean recognition rates of different methods for CMU MoBo Dataset.

YouTube Dataset								
Methods	Clean	N_G	N_P	N_{G+P}	S_{200}	S_{100}	S_{50}	S_{25}
MSM	0.611	0.593	0.379	0.501	0.614	0.610	0.595	0.579
MMD	0.629	0.605	0.427	0.544	0.613	0.595	0.595	0.564
DCC	0.648	0.577	0.489	0.609	0.652	0.652	0.651	0.623
MDA	0.653	0.633	0.474	0.571	0.645	0.633	0.632	0.623
AHISD	0.637	0.609	0.418	0.327	0.645	0.669	0.683	0.679
CHISD	0.663	0.671	0.397	0.416	0.664	0.673	0.679	0.668
SANP	0.684	0.671	0.469	0.352	0.686	0.682	0.675	0.661
COV+PLS	**0.701**	**0.617**	**0.645**	**0.629**	**0.699**	**0.693**	**0.675**	**0.640**

Table 3.4: The mean recognition rates of different methods for YouTube Dataset.

	MSM	DCC	MMD	MDA	AHISD	CHISD	SANP	COV+PLS
Training	N/A	12.397	N/A	8.846	N/A	N/A	N/A	2.322
Testing	5.114	5.120	6.462	4.288	20.516	8.424	52.976	2.033

Table 3.5: Computation time (seconds) of different methods on Honda/UCSD for training and testing (classification of one image set).

section, the single sample-based matching mechanism and complex optimization procedure make these methods less appealing in terms of efficiency.

3.5 Summary

We have proposed an efficient image set classification method called Covariance Discriminative Learning (CDL). The method represents each image set with its covariance matrix and models the problem as classifying points on the Riemannian manifold spanned by nonsingular covariance matrices. We derived a novel Riemannian kernel function which successfully bridges the gap between traditional learning methods operating in vector spaces and the learning task on an unconventional manifold. We explored two typical methods, LDA and PLS, for learning, and demonstrated the advantages of PLS for our specific problem. The promising experimental results show the superiority of our method over the state-of-the-art in terms of accuracy and efficiency, as well as its robustness to the practical challenges of noisy set data and varying set size. For future work, we are exploring the incorporation of set mean information into covariance matrix modeling. We will also study

more robust estimator for the covariance matrix for more challenging problems with heavy noise.

Our methods are not limited to face images. They can also be used in other visual recognition problems where each example is represented by a set of images, and more generally in machine learning problems where the classes and test examples are represented by sets of feature vectors.

Chapter 4

Face Verification Using Sparse Representations

4.1 Sparse Representations and Face Recognition

Wright et al. [94] used sparse representations for face recognition by relating the problem of finding the most similar face to a noiseless signal reconstruction. Since then, many other researchers have developed methods for face recognition using sparse representations [95, 96, 97] and showed that such methods are robust to occlusion, expressions and disguise. The face identification problem is a multi-class problem naturally formulated by sparse coding since the goal of both problems is to obtain a noiseless signal reconstruction. To leverage the robustness of sparse coding for face verification, we formulate a sparse coding based face verification framework. It is, however, not trivial to extend the method to a binary classification problem of face verification.

In this Chapter, we propose [98] a sparse representation [94] based face verification method that is simple yet achieves good performance on the LFW dataset [2] without a training set (unsupervised) and in the image restricted training setting. Sparse coding [94] approximates a signal $\boldsymbol{y}$ by a linear combination of a few atoms from a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{y} \approx \boldsymbol{D}\boldsymbol{x}$, and leads to good performance in various vision applications. Sparse coding can extract stable and discriminative face representations under challenging variations. Our method measures two models of image

similarity via a dictionary (reference set). The intuition of the first model is very straightforward. Since sparse representations account for most or all information of a signal (a face) with a linear combination of a small number of elementary signals (reference set) called atoms, we would expect the sparse codes of two images from the same person to be similar. So, the similarity of the sparse codes can be a measure of similarity for the image pair. The other model measures the change of the sparse code of one image from the pair to be verified when the dictionary is expanded by adding the other image from the pair. Comparing the change of the sparse codes before and after adding the extra face image also provides a measure of similarity for the pair. We integrate these two models and the scores are fused by averaging or training an SVM.

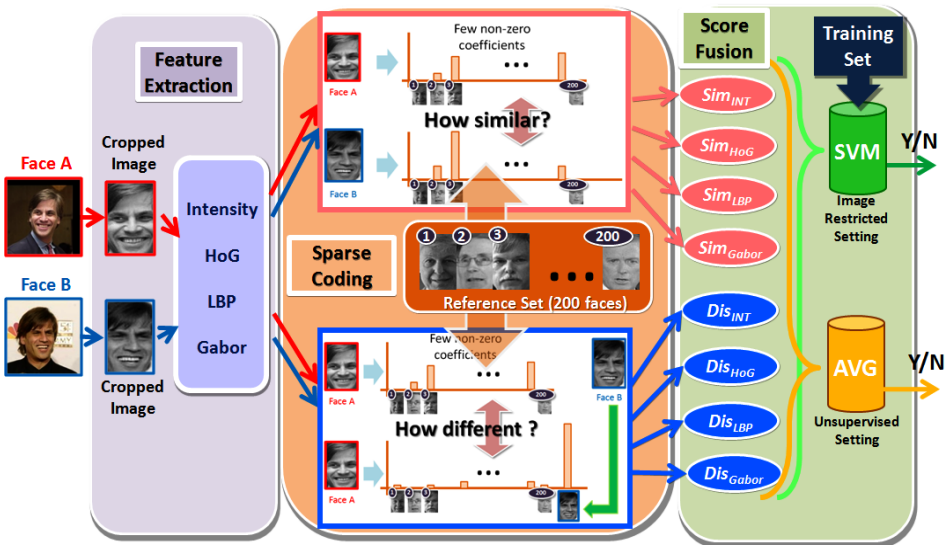


Figure 4.1: The proposed face verification framework based on sparse coding.

4.2 Proposed Method

4.2.1 Overview of the Framework

The main idea to convert a multi-class classification problem into a binary one is by utilizing a set of arbitrary face images as dummy classes. With the help of the dummy classes, called *reference set*, we formulate a binary classification problem using sparse representation.

Figure 4.1 illustrates the proposed method. Three steps are involved: feature extraction, sparse coding and score fusion.

In feature extraction, a pair of images, A and B, are cropped and re-scaled to a fixed size. Then, feature extraction is performed to obtain the intensity (INT), HoG [41], LBP [99], and Gabor [55] features as image descriptors.

In the sparse coding step, we exploit two methods to obtain the sparse representations for face A and B using the fixed *reference set* as a dictionary that contains a number of, say N, faces (the reference set is chosen from the training set in the LFW protocol, and the identities in the training set are disjoint from those in the testing stage). The first method (top half of the figure) directly measures the correlation of the generated sparse codes, which we refer to as the *similarity score*. The second method (bottom half) measures the difference of the two sparse codes of face A. One is obtained on the original dictionary and the other is on an augmented dictionary by adding B to the original dictionary. Then we do the same for face B, by adding A to the original dictionary. We refer to this as the *dissimilarity score*. We compute both the *similarity score* and *dissimilarity score* for each type of feature

descriptor. Sim_{INT}, Sim_{HoG}, Sim_{LBP}, and Sim_{Gabor} denote the similarity scores for each feature and Dis_{INT}, Dis_{HoG}, Dis_{LBP}, and Dis_{Gabor} denote the dissimilarity scores for each feature.

In the last stage, we fuse the eight scores obtained from different channels. We can either simply compute the average (AVG) of these eight scores in an unsupervised setting, or train an SVM to reduce the effect of overfitting to a particular score in a supervised setting.

4.2.2 Feature Extraction

After cropping and resizing the faces, each sample is decomposed into blocks and then a set of low-level feature descriptors is extracted from each block. The feature extraction methods capture information related to shape (histogram of oriented gradients (HOG)), texture (captured by local binary patterns (LBP)), color information (intensity) and salient visual properties (captured by Gabor filters).

4.2.3 Sparse Representation

A sparse representation-based face recognition algorithm was proposed in [94] and demonstrated to have high performance on the face identification task. Given a dictionary $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_N\}$ where $\boldsymbol{d}_i$ is the i-th dictionary atom (l_2-normalized) and a test sample $\boldsymbol{y}$, the sparse code of $\boldsymbol{y}$, $\hat{\boldsymbol{x}}$, can be obtained by solving the following l_1-minimization problem,

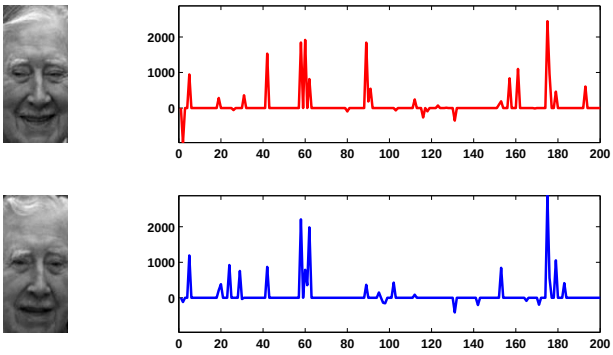
$$\hat{\boldsymbol{x}} = argmin_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \tag{4.1}$$

Sparse representation is an intuitively appealing method for face identification. The dictionary typically contains multiple face images for each person to be subsequently recognized. However, it is not straightforward to be directly applied to face verification since verification is not a multi-class problem that can be solved by choosing a few atoms from the dictionary. In face verification, a similarity measure is typically learned from pairs of training images labeled 'same' or 'different'. This provides less specific information than known identities - image labels.

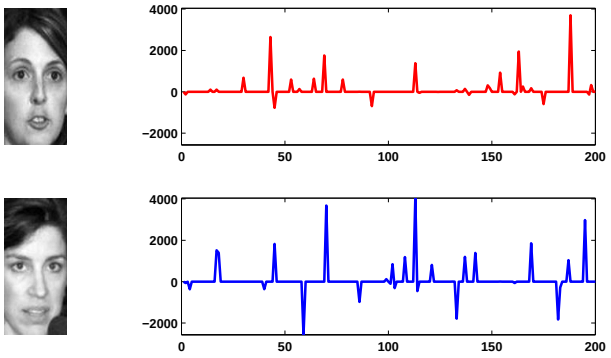
We instead use sparse representation for face verification problem in two different ways via a reference set, which we use as a dictionary: similarity score of two sparse codes and dissimilarity score of two sparse codes.

4.2.3.1 Similarity Score of Two Sparse Codes

A reference set is a set of images randomly selected from an image pool (e.g. training images) whose identities never appear in the test stage. We use the reference set as a dictionary $\boldsymbol{D}$ of size N to reconstruct the input image pairs (A,B). The feature vectors from the same individual are usually similar and more likely to have similar corresponding reconstructed signals by sparse coding, i.e. linear combination of dictionary atoms. We are interested in measuring the similarity of the sparse codes of A and B that approximates the similarity of the input images and let



(a) (b) SimScore = 0.7534



(c) (d) SimScore = -0.0038

Figure 4.2: An example of sparse codes (intensity feature) for 'similarity score' denoted by SimScore. (a) Original faces of a 'same' pair. (b) Sparse codes for the 'same' pair. (c) Original faces of a 'different' pair. (d) Sparse codes for the 'different' pair.

$$\begin{aligned} \hat{\boldsymbol{x}}_A^N &= argmin_{\boldsymbol{x}} \|\boldsymbol{y}_A - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \\ \hat{\boldsymbol{x}}_B^N &= argmin_{\boldsymbol{x}} \|\boldsymbol{y}_B - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \end{aligned} \tag{4.2}$$

be the sparse codes of A and B, respectively. Here, $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$ are feature vectors of input faces A and B respectively, $\boldsymbol{D}$ is the given dictionary, and γ is a penalty

weight on sparsity. We define the 'similarity score' of $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$, SimScore, by utilizing a similarity metric of $\hat{\boldsymbol{x}}_A^N$ and $\hat{\boldsymbol{x}}_B^N$,

$$\text{SimScore}(\boldsymbol{y}_A, \boldsymbol{y}_B) := \text{Similarity}(\hat{\boldsymbol{x}}_A^N, \hat{\boldsymbol{x}}_B^N) \tag{4.3}$$

We use the cosine similarity (CS) [34] as the similarity metric between two sparse codes. The CS of two vectors is defined as:

$$CS(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|} \tag{4.4}$$

Given a pair of feature vectors $(\boldsymbol{y}_A, \boldsymbol{y}_B)$, the 'similarity scores' (SimScore) of their sparse codes with the reference set from different feature channels are computed as:

$$\begin{aligned} Sim_{INT} &= CS(\hat{\boldsymbol{x}}_A^{N,INT}, \hat{\boldsymbol{x}}_B^{N,INT}) \\ Sim_{HoG} &= CS(\hat{\boldsymbol{x}}_A^{N,HoG}, \hat{\boldsymbol{x}}_B^{N,HoG}) \\ Sim_{LBP} &= CS(\hat{\boldsymbol{x}}_A^{N,LBP}, \hat{\boldsymbol{x}}_B^{N,LBP}) \\ Sim_{Gabor} &= CS(\hat{\boldsymbol{x}}_A^{N,Gabor}, \hat{\boldsymbol{x}}_B^{N,Gabor}) \end{aligned} \tag{4.5}$$

where $\hat{\boldsymbol{x}}_k^{N,feat}$ denotes the sparse code obtained from Eq. (4.2) using a dictionary with N atoms and $feat$ feature for face k, e.g., $\hat{\boldsymbol{x}}_A^{N,INT}$ represents the N dimensional sparse codes with respect to the N dictionary atoms computed from the intensity features of face A.

Figure 4.2-(a) and (b) show an example of a pair of faces from the same individual (with slight expression change) and their corresponding sparse codes gener-

ated from a dictionary with N=200 atoms using intensity(INT). Figure 4.2-(c) and (d) show a pair of faces from different individuals and their corresponding sparse codes generated from the intensity. It can be seen that sparse codes from the same individual (left) have much higher correlation (the responses to the 200 dictionary atoms have similar trend) than the sparse codes of the pair from different individuals (right).

4.2.3.2 Dissimilarity Score of Two Sparse Codes

Looking at only the similarity of the sparse codes is not making full use of the power of sparse coding. In face identification via sparse representation [94], the test face (probe) is represented as a sparse linear combination of the dictionary atoms. The coefficient of the most similar face in the dictionary to the test face is high while other coefficients are small or zero. We take advantage of this principle of the sparse coding in the following way.

For notation consistency, $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$ are feature vectors of input faces A and B respectively, $\boldsymbol{D}$ is the given original dictionary. We first compute the sparse coefficients of face A, $\hat{\boldsymbol{x}}_A^N$, using dictionary $\boldsymbol{D}$. Next, we add the l_2-normalized feature vector of face B, $\overline{\boldsymbol{y}_B}$, to the dictionary to construct a new augmented dictionary $\widetilde{\boldsymbol{D}}_B = [\boldsymbol{D}|\overline{\boldsymbol{y}_B}]$, of size $N+1$ and obtain another sparse code $\widetilde{\boldsymbol{x}}_A^{N+1}$ from the new dictionary $\widetilde{\boldsymbol{D}}_B$,

$$\begin{aligned}\hat{\boldsymbol{x}}_A^N &= argmin_{\boldsymbol{x}}\|\boldsymbol{y}_A - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \\ \widetilde{\boldsymbol{x}}_A^{N+1} &= argmin_{\boldsymbol{x}}\|\boldsymbol{y}_A - \widetilde{\boldsymbol{D}}_B\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1.\end{aligned} \tag{4.6}$$

Similarly, we can construct the augmented dictionary for face B, $\widetilde{\boldsymbol{D}}_A = [\boldsymbol{D}|\overline{\boldsymbol{y}_A}]$, by adding the l_2-normalized feature vector of face A to the original dictionary. Two sparse codes $\hat{\boldsymbol{x}}_B^N$ and $\widetilde{\boldsymbol{x}}_B^{N+1}$ are computed using the original dictionary $\boldsymbol{D}$ and the augmented dictionary $\widetilde{\boldsymbol{D}}_A$, respectively.

The motivation is, if two images are from the same individual, the $N+1$-th coefficient in the augmented dictionary will have a significantly high value and other coefficients will be diminished compared to the code obtained with the original dictionary. In contrast, when the two images are not from the same individual, the coefficients with respect to the original dictionary and the augmented dictionary do not significantly differ from each other. Thus, a higher dissimilarity of the two sparse codes obtained from the original dictionary and the augmented dictionary indicates a higher similarity of the pair being compared.

We compute the dissimilarity of the two sparse codes of face A before and after adding face B to the dictionary as follows,

$$\text{Dy}(\boldsymbol{y}_A) = 1 - \text{Similarity}(\hat{\boldsymbol{x}}_A^N, \widetilde{\boldsymbol{x}}_A^{N+1}(1{:}\text{N})) \tag{4.7}$$

Note that $\text{Dy}(\cdot)$ is defined on a single image in a given pair, whereas $\text{Similarity}(\cdot,\cdot)$ is defined with respect to two sparse codes. We can also obtain $\text{Dy}(\boldsymbol{y}_B)$, exchanging A and B. By averaging $\text{Dy}(\boldsymbol{y}_A)$ and $\text{Dy}(\boldsymbol{y}_B)$, we obtain the 'dissimilarity score' of

$\boldsymbol{y}_A$ and $\boldsymbol{y}_B$, DisScore,

$$\text{DisScore}(\boldsymbol{y}_A, \boldsymbol{y}_B) := \frac{\text{Dy}(\boldsymbol{y}_A) + \text{Dy}(\boldsymbol{y}_B)}{2} \tag{4.8}$$

The higher the score, the more similar the pair is.

Figure 4.3-(a) and (b) show a pair of faces from the same individual and their corresponding sparse codes before (red) and after (blue) adding the other to the dictionary. Figure 4.3-(c) and (d) show a pair of faces from different individuals and their corresponding sparse codes before and after adding the other to the dictionary. We can observe that the sparse codes from the same individual (left) shows significant difference in the first N atoms than the pair from different individuals (right).

As done for the similarity scores, we compute dissimilarity scores for four feature channels of intensity, HoG, LBP and Gabor to obtain Dis_{INT}, Dis_{HoG}, Dis_{LBP} and Dis_{Gabor}, respectively.

4.2.4 Score Fusion

Each feature descriptor and scoring method contains different discriminative power and should be aggregated in a reasonable way. According to [62, 46, 31, 21], combining multiple similarities from different descriptors usually boosts performance. We consider two simple approaches for fusing the eight scores (four feature channels $\times$ two scoring methods).

In the unsupervised setting, we simply average the eight scores from different

feature channels to obtain the final similarity score of the given pair. The averaging weighs every score equally. For the image restricted setting, we can fuse the scores by training a linear SVM to obtain more discriminative weights on each score using the given training set.

4.3 Experimental Results

We evaluate the proposed algorithm on the LFW dataset and compare the results with the state-of-the-art approaches. The dataset comes with a division of 10 splits/folds (disjoint subject identities) for cross validation with three paradigms of evaluation protocols: unsupervised, image-restricted, and image-unrestricted protocols [2]. In the **unsupervised** protocol, there is no training information of same/not-same labels. It is the most challenging due to lack of training samples. The **image-restricted** protocol refers to the setting of using only the restricted number of given image pairs for training. In this setting, it is known whether an image pair belongs to the same person or not, while identity information of each image is not provided. The **unrestricted** protocol refers to the training setting that can use all available data, including the identity of the people in the images that allows one to generate as many training pairs as possible. The latter two settings allow us to utilize available image pair information in the training set. In this paper, we only focus on the first two protocols. The aligned version, lfw-a, was used in all experiments.

In our evaluations, for each fold, we randomly choose N=200 images (one

image per individual) to construct a compact dictionary (reference set) from the training set without using their pair information. We have empirically tried varying dictionary size N from 200 to 500, and found that the size has only slight impact on the verification performance. For efficiency, we use a fixed size N=200 in the following experiments to report our result.

4.3.1 Experimental setup

To obtain a sparse solution to the least squares problem, we can choose either l_0 regularization or l_1 regularization in the least squares objective function (Eq.4.1). We choose the l_1 regularizer since it is hard to specify the number of nonzero coefficients, i.e., the hyper-parameter of the l_0 regularizer. We use the implementation of Lee et al. [100] due to its computational efficiency.

For the feature extraction step, we do not apply any photometric pre-processing. All the faces are cropped and rescaled to 80×148. For extracting HoG and LBP features, we divide each face into blocks of 20×20 size and extract 16-bin HoG feature and 59-bin uniform LBP feature for each block. For Gabor feature, we adopt five scales and eight orientations of the Gabor filters. The final Gabor feature vector is obtained by concatenating the responses at every five pixels in order to reduce the dimensionality of the feature vector to a manageable size.

4.3.2 Results from Different Feature Descriptors and Score Fusions

The performances of our method with individual feature and their fusion are shown in Table 4.1 (on fold 1 only). The first column shows the verification accuracy obtained by using the Euclidean distance of the original feature vector pairs as similarity measure. The second column shows the verification accuracy from the SimScore (Eq.4.3). The third column is from the DisScore (Eq.4.8). Both SimScore and DisScore for individual feature descriptors achieve significant improvements over the Euclidean distance. The 'Combined' scores are the results obtained by fusing the scores from all the four features by averaging (no training) or creating a vector of four scores and running an SVM on this vector. The **HybridSparse** scores are obtained by fusing the eight scores from both SimScore and DisScore. We can see that the **HybridSparse (Avg)**, obtained by simply averaging the eight scores with equal weight, achieves good verification accuracy (83.00%) and the **HybridSparse (SVM)** boosts the performance further to 84.67%. Generally, as we expect, score fusion can always achieve better result (as in [62, 46, 31, 21, 27, 47]) since there could be complimentary information across different scores.

4.3.3 Comparison with the State-of-the-art Methods

Comparison on the Unsupervised protocol Our method can be compared with other methods using the unsupervised protocol, since we simply sample a very small number of images from the training set for the reference set without using any pair labels of same/different or identity information. Table 4.2 shows the comparison

Table 4.1: Verification accuracy at Equal Error Rate on LFW dataset (fold 1 only) under different similarity measures.

Descriptor	Euclidean	SimScore	DisScore
Intensity	0.7133	0.7533	0.7633
HoG	0.6767	0.7733	0.7467
LBP	0.6700	0.7633	0.7667
Gabor	0.6933	0.7700	0.7533
Combined (Avg)	0.7067	0.8167	0.8033
Combined (SVM)	0.7267	0.8333	0.7967
HybridSparse (Avg)	N/A	**0.8300**	
HybridSparse (SVM)	N/A	**0.8467**	

Table 4.2: Mean (± standard error) verification accuracy on the LFW dataset (Unsupervised protocol).

Method	Accuracy
H-XS-40 [48]	0.6945±0.0048
GJD-BC-100 [48]	0.6847±0.0065
SD-MATCHES [48]	0.6410±0.0042
LARK [49]	0.7223±0.0049
HybridSparse (Avg)	**0.8377±0.0053**
HybridSparse (Avg, flip)	**0.8470±0.0047**

result at equal error rate. The 'flip' means that when comparing image pair A and B, we also compare A and the horizontally flipped image of B to reduce the effect of pose variation. Then the average of the two scores is taken as the final similarity score. Figure 4.4 presents the ROC curve of our approach (dotted red line), along with the ROC curves of previous methods. As shown, our approach significantly outperforms the other methods by a very large margin.

Comparison on the Image-Restricted protocol Table 5.2 shows the face verification accuracy of our method in comparison with state-of-the-art methods under the Image-Restricted protocol that allows using the training set with labels of same/different. Figure 5.4 shows the ROC curve of our approach (dotted red line), along with the ROC curves of selected recent state-of-the-art methods.

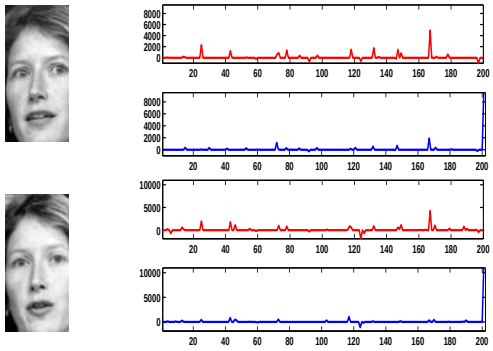
Table 4.3: Mean (± standard error) verification accuracy on the LFW dataset (Image-Restricted protocol). '$*$' denotes methods using outside training data.

Method	Accuracy
LDML, funneled [28]	0.7927±0.0060
POEM [29]	0.7542 ±0.0071
Hybrid [46]	0.8398±0.0035
Combined b/g samples based [21]	0.8683±0.0034
*Attribute and Simile classifiers [27]	0.8529± 0.0123
Single LE + holistic [31]	0.8122±0.0053
*Multiple LE + comp [31]	0.8445±0.0046
*Associate Predict [47]	0.9057 ±0.0056
LARK+OSS [49]	0.8512 ±0.0037
HybridSparse (SVM)	**0.8530 ±0.0040**
HybridSparse (SVM, flip)	**0.8624 ±0.0031**

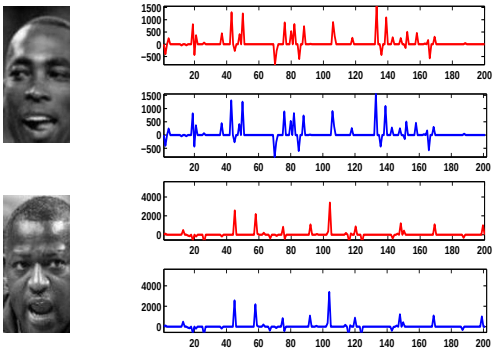
The results show that the verification accuracy of our approach is competitive to the state-of-the-art methods on the LFW benchmark in the challenging image-restricted protocol. It is worth noting that, methods marked by '$*$' (such as [27, 31, 47]) use training data outside of the LFW for facial point detection or pose/illumination classification and so on, which can have a significant impact on the verification accuracy, thus are not directly comparable to other methods including ours. Kumar et al. [27] achieves excellent results (however still marginally lower than ours) at the expense of an expensive training of high-level classifiers by incorporating a huge volume of images outside of the LFW dataset. The LE method [31] relies on facial feature point detectors. Predict-Associate [47] not only relies on facial feature point detectors, but also uses the Multi-PIE dataset with identities covering 7 poses and 4 illumination conditions. For other methods that we are in the same category with, [21] is the most comparable. Wolf et al. [21] also combines multiple descriptors, however, their method has complicated layers and leverages metric learning [47]. An additional disadvantage of this method is that it requires background samples (a fixed set of 'negative' examples) that have similar properties as the faces being compared. The background samples should not contain faces from any person who might subsequently appear in a pair to be compared. Overall, our simple approach achieves competitive accuracy without local feature detection or other additional information.

4.4 Conclusions and Future Work

We have proposed a novel approach for face verification using sparse coding in two different yet complimentary ways with a fixed reference set as a dictionary. The evaluation on the very challenging LFW dataset both under the unsupervised setting and image restricted training setting shows competitive results. We demonstrated that sparse coding can be a promising direction for face verification since it extracts more stable and discriminative face representation under challenging variations. In the next Chapter, we would explore pairwise dictionary learning for face verification applications.



(a) (b) DisScore = 0.2653



(c) (d) DisScore = 5.56×10^{-17}

Figure 4.3: An example of sparse codes (intensity feature) for 'dissimilarity score' denoted by DisScore. (a) Original faces of a 'same' pair. (b) Sparse codes with and without adding the other face to dictionary for the 'same' pair. (c) Original faces of a 'different' pair. (d) Sparse codes with and without adding the other face to dictionary for the 'different' pair. **Note that the range of horizontal axes of blue plots is [1,201] while that of red plots is [1,200] and the scales of vertical axes of two sparse codes for an image are consistent for comparison.** In the blue plots, one can observe the peak at 201 in (b) but not in (d). Also note that SimScore of pair (a) is 0.8551 and SimScore of pair (c) is 0.0471.

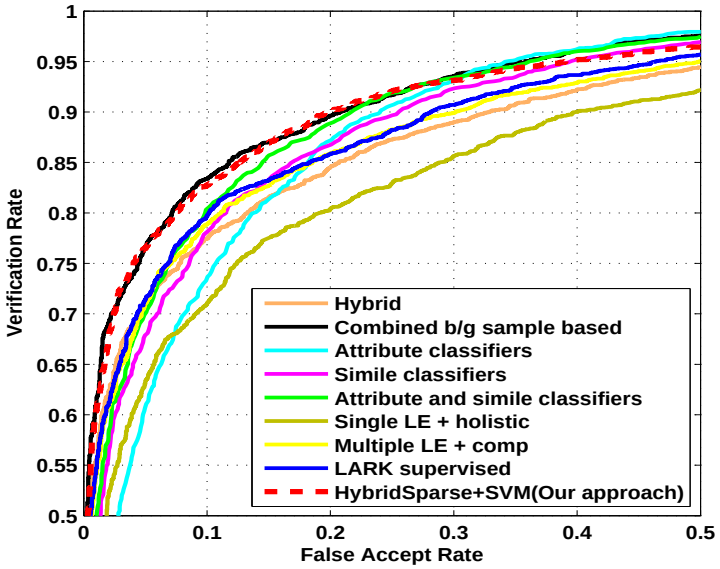


Figure 4.4: ROC curves on the LFW dataset (unsupervised protocol).

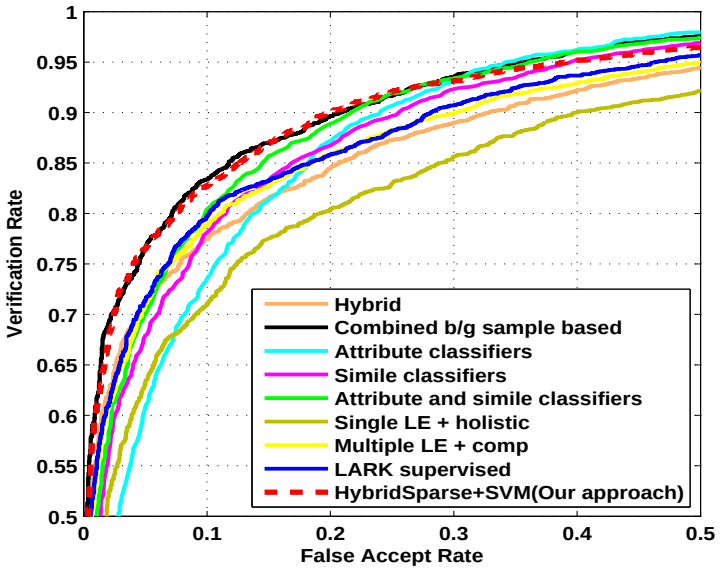


Figure 4.5: ROC curves on the LFW dataset (Image-Restricted protocol). Only shown with the selected **best** results that were recently reported for clarity.

Chapter 5

Discriminative Dictionary Learning with Pairwise Constraints

In the previous chapter, we apply sparse representations to face verification in two ways via a fix reference set as dictionary. As an extension work, in this Chapter, we propose a dictionary learning framework with explicit pairwise constraints, which unifies the discriminative dictionary learning for pair matching (face verification) and classification (face recognition) problems.

5.1 Introduction

Different from many classification problems where the specific class label of each image is given during training, only binary information such as same/different or relevant/irrlevant is provided for training data in applications such as face verification (given a *target* and a *query* image, determine whether they are from the same person), pair matching, image retrieval, etc. Typically, a discriminative similarity measure is learned through metric learning [8, 9, 10, 11] from pairs of training images labeled as 'same' or 'different'; this provides less specific information than known classes - category labels. In this paper, we propose a framework to learn a discriminative dictionary satisfying pairwise constraints. The learned dictionary is suitable for pair matching problems with the pairwise constraints from the binary similarity or dissimilarity information; in addition, it is also suitable for classification

problems given pairwise constraints about category information.

Sparse coding [101] approximates a signal y as a linear combination of a few atoms from a learned dictionary A, *i.e.*, $y = Ax$, and leads to good performance in numerous applications. The learned dictionary A is critical to performance. K-SVD [102] minimizes a reconstruction error to learn an over-complete dictionary. However, despite its many successful applications, K-SVD is not suitable for classification, where the dictionary should be not only representative, but also discriminative. Hence, some supervised dictionary learning approaches incorporate classification error into the objective function to construct a dictionary with discriminative power. However, such frameworks consider only discriminativeness in the classifier construction, but do not guarantee the discriminativeness in the sparse representations of input signals. The discriminative capability of a dictionary usually comes from category label information. We will show that considering the pair similarity/dissimilarity constraints without category labels during dictionary learning can also improve the discriminative power of a dictionary; no existing dictionary learning approach has fully explored this property. Our dictionary learning approach explicitly integrates pairwise constraints for sparse codes of input signals and a linear predictive classifier into one objective function. The learned dictionary encourages signals from the same class (or a similar pair) to have similar sparse codes, and signals from different classes (or a dissimilar pair) to have dissimilar sparse codes, illustrated in Figures 5.1 and 5.2. The similarity can be thresholded to yield a binary decision of same/different (face verification), or it can be used to find the most similar face in a gallery (face recognition). The main contributions of this Chapter

are:

- We present a dictionary learning framework with explicit pairwise constraints, which unifies the discriminative dictionary learning for pair matching and classification problems.

- Our framework furthermore integrates the pairwise constraints for sparse codes of input signals and a linear predictive classifier into the objective function for dictionary learning, which addresses the desirable properties of discriminativeness in the sparse representations of signals, and the discriminativeness in classifier construction.

- The objective function can be optimized via the efficient feature-sign search algorithm [103].

- Our approach is validated on various public face verification and recognition benchmarks.

5.1.1 Related Work on Face Verification and Dictionary Learning

As reviewed in previous Chapters, metric learning (ML) aims at learning a discriminative similarity measure between different images [8, 9, 10, 11]. An appropriate distance metric plays a very important role in many learning problems. Most work in metric learning, including LDML [8], MkNN [8], ITML [9], CSML [10], etc, relies on learning a Mahalanobis distance to map the feature space into a target space [11]. Less work, however, has been done for face verification using dictionary

learning with pairwise similarity and dissimilarity constraints on input training examples.

[101] used sparse representations for face recognition (1:N matching problem which finds a nearest neighbor of a given *probe* in a *gallery* face set) by relating the problem of finding the most similar face to noiseless signal reconstruction. Since then, many other researchers have developed methods for face recognition using sparse representations or dictionary learning [101, 104, 105, 106, 107, 97, 108]. Although many of these existing algorithms have been shown to perform well in classification (e.g. face recognition) applications, most of them do not explicitly deal with dictionary learning with pairwise constraints - when only binary information such as same/different or relevant/irrelevant is given in the training stage (e.g. face verification). Our dictionary learning framework is more general since it deals with face verification and face recognition problems simultaneously.

To enhance discrimination power, our dictionary learning framework explicitly integrates pairwise constraints for sparse codes of input signals and a linear predictive classifier into the objective function during training. Most previous approaches treat dictionary learning and classifier training as two separate processes, such as [109, 110, 111, 112, 113, 114]. In these approaches, a dictionary is typically learned first and then a classifier is trained based on it. There are also sophisticated approaches [115, 116, 117, 97] combining dictionary learning and classifier training in a mixed reconstructive and discriminative formulation. Our approach falls into this category. We learn a single dictionary and an optimal classifier jointly.

Laplacian Sparse Coding [118] explicitly introduces a locality preserving con-

straint among similar local features in the sparse coding step to preserve the consistence of the sparse codes. This is different since our approach is to learn a dictionary which encourages signals from a similar pair (or the same class) to have similar sparse codes. Furthermore, our approach integrates a linear predictive classifier into the objective function to learn the dictionary and the classifier simultaneously while [118] learns the dictionary and the classifier separately.

5.2 Sparse Coding and Dictionary Learning

In this section, we provide a brief review of sparse coding and dictionary learning.

5.2.1 Sparse Coding

Let $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, ...\boldsymbol{y}_N] \in \mathbb{R}^{n \times N}$ be the data matrix of N input signals, where $y_i \in \mathbb{R}^n$ denotes the i-th input signal with n-dimensional feature description. Given a dictionary $A = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_K] \in \mathbb{R}^{n \times K}$, where $\boldsymbol{a}_i$ is the i-th dictionary atom (l_2-normalized), sparse coding [101] with l_1 regularization computes the sparse representations $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N] \in \mathbb{R}^{K \times N}$ of the input signals Y, through solving the following l_1-minimization problem,

$$X^* = argmin_X \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma \|\boldsymbol{x}_i\|_1) \tag{5.1}$$

where constant γ is a sparsity constraint factor and the term $\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2$ denotes the reconstruction error. Each input signal $\boldsymbol{y}_i$ can be represented as a sparse linear combination of a few dictionary atoms. The feature-sign search algorithm [103] is an efficient algorithm that can be used to solve (1).

5.2.2 Dictionary Learning

The goal of dictionary learning is to find optimized dictionaries that provides a succinct representation for most statistically representative input signals. The learning procedure can be formulated as solving the following problem [103],

$$< A^*, X^* >= argmin_{A,X} \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma\|\boldsymbol{x}_i\|_1) \tag{5.2}$$

The optimization problem is convex in A (while holding X fixed) and convex in X (while holding A fixed), but not convex in both simultaneously. Usually, the above objective is iteratively optimized in a two stage manner, by alternatively optimizing with respect to A (bases) and X (coefficients) while holding the other fixed. The formulation (2) only focuses on minimizing the reconstruction error and does not consider the discriminative power of a dictionary for classification tasks. Hence, some supervised approaches [115, 116, 117, 97, 107] have been proposed to improve the discriminative power of dictionary, by integrating the category label information into the objective function of dictionary learning. However, most of them do not explicitly deal with dictionary learning with pairwise constraints.

5.3 Discriminative Dictionary Learning with Pairwise Constraints (DDL-PC)

In this section, we present our Discriminative Dictionary Learning with Pairwise Constraints algorithm which takes into account the relationships of each pair of learned sparse codes $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Here, the intuition is to encourage signals from a similar pair to have similar sparse codes. We subsequently focus on the effects

of adding a discriminative term, and a classification error term into the objective function in (5.2). We refer to them as DDL-PC1 and DDL-PC2, respectively.

5.3.1 DDL-PC1

To obtain discriminative sparse codes $\boldsymbol{x}$ with the pairwise constrained dictionary A, the objective function for dictionary construction is defined as:

$$\begin{aligned} < A^*, X^* > &= argmin_{A,X} \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma\|\boldsymbol{x}_i\|_1) + \frac{\beta}{2} \sum_{i,j=1}^{N} (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 M_{ij}) \\ &= argmin_{A,X} \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma\|\boldsymbol{x}_i\|_1) + \beta(Tr(X^T X D) - Tr(X^T X M)) \\ &= argmin_{A,X} \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma\|\boldsymbol{x}_i\|_1) + \beta(Tr(X^T X L)) \end{aligned} \tag{5.3}$$

where the constants γ and β control the relative contribution of the corresponding terms. The first term $\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2$ is the *reconstruction error term*, which evaluates the reconstruction error of the approximation to the input signals. The second term $\|\boldsymbol{x}_i\|_1$ is the *regularization term* for sparsity. The last term, which is new and proposed here, is the *discrimination term* called 'pairwise sparse code error' based on pairwise constraints which are encoded in matrix M. $D = diag\{d_1, ...d_N\}$ is a diagonal matrix whose diagonal elements are the sums of the row elements of M (see below), $d_i = \sum_{j=1}^{N} M_{ij}$. $L = D - M$ is the Laplacian matrix. Matrix M has different forms depending on the problems being considered. For example, in face verification, the relationship of a pair ($\boldsymbol{y}_i$, $\boldsymbol{y}_j$) is given as same/different. Thus, given the sets of 'same' and 'different' pairs $\mathcal{S}$ and $\mathcal{D}$, we define matrix M to encode the (dis)similarity information as

$$M_{ij} = \begin{cases} +1, & \text{if } (\boldsymbol{y}_i, \boldsymbol{y}_j) \in \mathcal{S} \\ -1, & \text{if } (\boldsymbol{y}_i, \boldsymbol{y}_j) \in \mathcal{D} \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

5.3.2 DDL-PC2

Although (5.3) can already be used for classification by defining M based on the pairwise similarity constraints with category labels (see Sec. 3.4), the classification error can be further included as an additional term in the objective function in (5.3). Here we use a linear predictive classifier $f(\boldsymbol{x}; W) = W\boldsymbol{x}$. The objective function for learning a pairwise constrained dictionary A with both reconstructive and discriminative power can then be defined as follows:

$$\begin{aligned} < A^*, X^*, W^* > = & \, argmin_{A,X,W} \sum_{i=1}^{N} (\|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \gamma \|\boldsymbol{x}_i\|_1) \\ & + \frac{\beta}{2} \sum_{i,j=1}^{N} (\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 M_{ij}) + \alpha \sum_{i=1}^{N} (\|\boldsymbol{h}_i - W\boldsymbol{x}_i\|_2^2 + \lambda \|W\|_2^2) \end{aligned} \tag{5.5}$$

The new term $\|\boldsymbol{h}_i - W\boldsymbol{x}_i\|_2^2 + \lambda\|W\|_2^2$, where $\|\boldsymbol{h}_i - W\boldsymbol{x}_i\|_2^2$ represents the classification error and $\|W\|_2^2$ is the regularization penalty term, supports learning an optimal linear predictive classifier. $\boldsymbol{h}_i = [0, 0, ...1...0, 0]^T \in \mathbb{R}^m$ (m: number of classes) is a label vector corresponding to an input signal $\boldsymbol{y}_i$, where the non-zero position indicates the class label of $\boldsymbol{y}_i$.

5.3.3 Optimization Procedure

In this section, we only describe the optimization procedure for DDL-PC2 since DDL-PC1 utilizes the same procedure except that $\alpha = 0$ in (5.6)(5.7)(5.8) and the classifier W update step is not considered during dictionary learning. Solving (5.5) is a challenging task because the objective function is not convex for A, X and W simultaneously; but fortunately, it is convex in one variable when the other two variables are fixed. In [103], (5.2) was solved by an efficient feature-sign search algorithm. Motivated by [103], we optimize A, X and W alternatively. Algorithm 1 presents the pseudocode of algorithm DDL-PC2.

5.3.3.1 Computing Sparse Codes X with Fixed A and W.

When A and W are fixed, we optimize $\boldsymbol{x}_i$ alternately and fix other $\boldsymbol{x}_j (j \neq i)$ for other signals. Optimizing (5.5) is equivalent to:

$$\min_{\boldsymbol{x}_i} \mathcal{L}(\boldsymbol{x}_i) + \gamma \|\boldsymbol{x}_i\|_1 \tag{5.6}$$

where $\mathcal{L}(\boldsymbol{x}_i) = \|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 + \beta(2\boldsymbol{x}_i^T(XL_i) - \boldsymbol{x}_i^T\boldsymbol{x}_i L_{ii}) + \alpha(\boldsymbol{x}_i^T W^T W \boldsymbol{x}_i - 2\boldsymbol{x}_i^T W^T \boldsymbol{h}_i)$, L_i is the i^{th} column of L and L_{ii} is the (i,i) element of L. (6) is exactly the problem that the feature-sign search algorithm in [103] solves. [103] iteratively searches for the coefficient sign vector $\boldsymbol{\theta}$ for $\boldsymbol{x}_i$, then (5.6) reduces to a standard, unconstrained quadratic optimization problem (QP). To compute the analytical solution, we calculate the gradient of $\mathcal{L}(\boldsymbol{x}_i)$ with respect to $\boldsymbol{x}_i$:

$$\frac{\partial \mathcal{L}(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} = 2A^T(A\boldsymbol{x}_i - \boldsymbol{y}_i) + 2\beta(XL_i) + 2\alpha(W^T W \boldsymbol{x}_i - W^T \boldsymbol{h}_i) + \gamma\boldsymbol{\theta} \tag{5.7}$$

Finally the analytic solution of $\boldsymbol{x}_i$ can be obtained when we have $\frac{\partial \mathcal{L}(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} = 0$:

$$\boldsymbol{x}_i^* = (A^T A + 2\beta L_{ii} I + 2\alpha W^T W)^{-1}(A^T \boldsymbol{y}_i + 2\alpha W^T \boldsymbol{h}_i - 2\beta \sum_{k \neq i} \boldsymbol{x}_k L_{ki} - \gamma \boldsymbol{\theta}) \quad (5.8)$$

In practice, a very small β is chosen to guarantee the Hessian matrix $(A^T A + 2\beta L_{ii} I)$ to be positive semidefinite, hence (3) is convex.

5.3.3.2 Updating Dictionary A with Fixed X and W.

Given X and W, we use the Lagrange dual in [103] to optimize the following objective function:

$$\min_A \sum_{i=1}^{N} \|\boldsymbol{y}_i - A\boldsymbol{x}_i\|_2^2 \quad s.t. \|\boldsymbol{a}_j\|_2^2 \leq c, \quad \forall j = 1...K. \quad (5.9)$$

The analytical solution of A can be computed as: $A^* = YX^T(XX^T + \Lambda)^{-1}$, where Λ is a diagonal matrix constructed from all the dual variables.

5.3.3.3 Updating Classifier W with Fixed X and A.

Given X and A, we employ the multivariate ridge regression model [116] to update W, with the quadratic loss and l_2 norm regularization:

$$\min_W \sum_{i=1}^{N} \|\boldsymbol{h}_i - W\boldsymbol{x}_i\|_2^2 + \lambda \|W\|_2^2, \quad (5.10)$$

which yields the following solution: $W^* = HX^T(XX^t + \lambda I)^{-1}$.

Algorithm 2: Discriminative Dictionary Learning with Pairwise Constraints-2 (DDL-PC2).

Input: input signals Y, Laplacian matrix L, label matrix H, regularization constant γ, β and α, iteration number $\hat{T}$

Output: learned dictionary A, classifier W and sparse code X.

Initialization: Compute initial A_0 via K-SVD, initial X_0, W_0 using (5.1), (5.10)

for $t = 1, 2, ..., \hat{T}$ **do**

Sparse Coding: compute sparse code X using (5.6);

Dictionary Update: update dictionary A using (5.9);

Classifier Update: update classifier W using (5.10).

end for

5.3.4 Matching Approach

5.3.4.1 Face Verification

In face verification or pair matching problems, a similarity measure is typically learned from pairs of training images labeled as 'same' or 'different'; this provides less specific information than known identities - image labels. Given a training set of pairs, we first construct matrix M with their pairwise relationships. For example, suppose three pairs of feature vectors are given - $(\boldsymbol{y}_1, \boldsymbol{y}_2)$ are features vectors from the same person, $(\boldsymbol{y}_3, \boldsymbol{y}_4)$ are also features vectors from the same person and $(\boldsymbol{y}_5, \boldsymbol{y}_6)$

are features vectors from different persons. Matrix M would then be:

$$M = \begin{bmatrix} & y1 & y2 & y3 & y4 & y5 & y6 \\ y1 & 0 & 1 & 0 & 0 & 0 & 0 \\ y2 & 1 & 0 & 0 & 0 & 0 & 0 \\ y3 & 0 & 0 & 0 & 1 & 0 & 0 \\ y4 & 0 & 0 & 1 & 0 & 0 & 0 \\ y5 & 0 & 0 & 0 & 0 & 0 & -1 \\ y6 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

With the given training set of pairs and the corresponding matrix M, an optimized discriminative dictionary A (initialized by K-SVD algorithm [102]) can be learned using DDL-PC1. Then, when a new test pair $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ comes in, we can compute the optimized sparse codes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with dictionary A by solving (5.1). Finally, the cosine similarity [10, 119] of the two sparse codes is used as the similarity metric between the image pair. This similarity is thresholded to yield a binary decision of same/different.

5.3.4.2 Face Recognition

In face recognition, class labels are given for each image in the training set. The pair relationships are derived from the category labels. If $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ belong to the same class, we define M_{ij} as 1; otherwise we set it to 0. Matrix M encoding the (dis)similarity information can be defined as

$$M_{ij} = \begin{cases} 1, & \text{if } (\boldsymbol{y}_i, \boldsymbol{y}_j) \in c_k, k = 1...m \\ 0, & \text{otherwise} \end{cases} \tag{5.11}$$

There are two ways to construct the classifier W here. For DDL-PC1, we obtain A and X first and then the matrix W is trained separately using (5.10). For DDL-PC2, we obtain A and W jointly using Algorithm 1.

Then, when a new test sample $\boldsymbol{y}_i$ comes in, we compute its sparse code $\boldsymbol{x}_i$ with respect to A by solving (5.1). Finally we simply use W to estimate a class label vector for $\boldsymbol{y}_i$: $l = W\boldsymbol{x}_i$, where $l \in \mathbb{R}^m$. The label of $\boldsymbol{y}_i$ is assigned as the index j where l_j is the largest element of l.

5.4 Experimental Results

We evaluate the proposed algorithm on the LFW dataset [2] for face verification task, and the Extended YaleB database [3] and AR face database [4] for face recognition task.

5.4.1 Face Verification Experiments

5.4.1.1 LFW Database

Again, we evaluated our approach on the the Labeled Faces in the Wild (LFW) dataset. In our evaluations, for each independent fold, we randomly choose 500 pairs of 'same' and 500 pairs of 'different' from the training set (other 9 splits, 5400 image pairs) to learn an optimal dictionary through DDL-PC1. The learned dictionary

consists of 510 atoms. γ is set to be 30 and β is set to be 0.1.

5.4.1.2 Experimental Setup

All the faces are cropped and rescaled to 80×148. According to [62, 46, 31, 21], combining multiple similarities from different descriptors usually boosts performance. In our experiments, the intensity, HoG, LBP, and Gabor features are used. Finally, the four scores for different features are fused by averaging (no training) or training SVM. For extracting HoG and LBP features, we divide the faces into blocks of 20×20 and extract the 16-bin HoG feature and the 59-bin uniform LBP feature for each block. For Gabor features, we adopt five scales and eight orientations of the Gabor filters. The final Gabor feature vector is obtained by concatenating the responses at every 10 pixels in order to reduce the dimensionality of the feature vector to manageable size.

Fig.5.3 shows some examples (5 'same' and 5 'different') of testing image pairs from the LFW dataset. The similarity scores obtained from KSVD dictionary learning and our DDL-PC1 are listed under each pair. As it shows, compared to KSVD, higher similarity scores for the 'same pairs' and lower similarity scores for 'different' pairs are obtained by our discriminative dictionary learning.

Table 5.1 summarizes the performances of our method with individual feature and their fusion. The first column shows the face verification accuracy (at equal error rate) obtained from using the Euclidean distance of the original feature vector pairs as similarity measure. The second column shows the accuracy from the dictionary

Table 5.1: Mean (± standard error) verification accuracy at equal error rate of different feature descriptors and their fused scores on LFW dataset. Euclidean, dictionaries learned by K-SVD and the proposed DDL-PC1 are compared.

Descriptor	Euclidean	K-SVD	DDL-PC1
Intensity	0.7140±0.0056	0.7424±0.0051	0.7870±0.0048
HoG	0.6803±0.0046	0.7524±0.0049	0.8030±0.0037
LBP	0.6763±0.0054	0.7433±0.0052	0.7876±0.0032
Gabor	0.6920±0.0041	0.7646±0.0047	0.7996±0.0052
Combined (Avg)	0.7013±0.0045	0.8056±0.0045	0.8410±0.0041
Combined (SVM)	0.7216±0.0047	0.8196±0.0036	0.8603±0.0033

learned by K-SVD (followed by the l_1 based sparse coding) and the third column shows those from the proposed DDL-PC1. The combined scores are the results from fusing the four scores for all features by averaging (no training) or training SVM. Clearly, DDL-PC1 works best in all situations comparing to 'Euclidean' and 'K-SVD'.

5.4.1.3 Comparison with the State-of-the-art Methods

Table 5.2 shows the face verification accuracy of our method compared with recent methods with the Image-Restricted protocol. The 'flip' means that when comparing image pair I and J, we also compare I and the horizontally flipped image of J to reduce the effects of pose variation. Then, the average of the two scores is taken as the final similarity score. Figure 5.4 contains the ROC curve of

Table 5.2: Mean (± standard error) verification accuracy on the LFW dataset, image-restricted protocol using the proposed DDL-PC1, and the same model except the addition of the 'flipped' image idea. '∗' denotes methods using outside training data.

Method	Accuracy
LDML [8]	0.7927±0.0060
Hybrid [46]	0.8398±0.0035
Combined b/g samples based [21]	0.8683±0.0034
*Attribute and Simile classifiers [27]	0.8529±0.0123
Single LE + holistic [31]	0.8122±0.0053
*Multiple LE + comp [31]	0.8445±0.0046
*Predict-Associate [47]	0.9057 ±0.0056
LARK + OSS [49]	0.8512 ±0.0037
DDL-PC1	**0.8603 ±0.0033**
DDL-PC1 (flip)	**0.8710 ±0.0035**

our approach (dotted red line), along with the ROC curves of selected recent state-of-the-art methods with the Image-Restricted protocol for presentation clarity.

The results show that the verification accuracy of our approach is comparable with the state-of-the-art methods on the LFW benchmark in the challenging image-restricted protocol. Moreover, the methods marked by '∗' use training data outside of LFW for facial point detection or pose/illumination classification and so on. Those can have a significant impact on verification accuracy, thus not di-

rectly comparable. Kumar [27] achieved excellent results, marginally lower than ours. However, the work of Kumar requires expensive training of high-level classifiers incorporating a huge volume of images outside of the LFW dataset. The LE method [31] relies on facial feature point detectors. Predict-Associate [47] not only relies on facial feature point detectors, but also uses the Multi-PIE dataset with identities covering 7 poses and 4 illumination conditions as prior knowledge. For other methods we are in the same category with, [21] is most comparable. Wolf [21] also combines multiple descriptors; their method adds up several layers of information and leverages metric learning [47]. Moreover, one disadvantage of Wolf's method is that it requires background samples (a fixed set of 'negative' examples) that have similar properties as the faces being compared and do not contain faces from any person who might subsequently appear in a pair to be compared. It learns models for each pair being compared on-the-fly, which might not be desirable in practical applications. Overall, our DDL-PC1 achieves competitive accuracy without local feature identification or any other additional information.

5.4.2 Face Recognition Experiments

5.4.2.1 Extended YaleB Database

The Extended YaleB database [3] contains 38 persons under 64 illumination conditions, $2,414$ frontal-face images. The original images are cropped to 192×168. We used the random face features [97, 101] to represent the face images. Following [97, 107], we project each face image into a 504-dimensional feature vector

using a random matrix of zero-mean normal distribution. Each row of the random matrix is l_2 normalized. We randomly sample 32 images per person for training and taking the rest as testing. We repeated 10 times such this sampling process and report their average as the recognition accuracy. The parameter γ is set to 20; β and α are set to 2.0 and λ is 1.0 here.

We fix the dictionary size of 570 atoms as in [97, 107] and evaluate our approach. We compare the recognition accuracy with K-SVD [102], D-KSVD [97], SRC [101], LLC [120] and recently proposed LC-KSVD [107]. We obtain the original implementations of LC-KSVD [1] from the authors [107]. A D-KSVD is implemented by eliminating the label consistent term in LC-KSVD. For SRC, we randomly select the average of dictionary size per person from each person and report the best result we achieved. For LLC, we perform the experiment with 30 local bases, which determines the sparsity of the LLC codes. The results are summarized in Table 5.3. Our approaches achieve better results than K-SVD, D-KSVD, SRC and LLC and are comparable to LC-KSVD.

We also evaluate our approach using random-face features and dictionary sizes 190, 380, 570 and 760. Then we compare the classification accuracy with state-of-art approaches including LC-KSVD, D-KSVD, K-SVD, SRC and LLC which use the same features and dictionary sizes. As shown in Figure 5.5, our approach has higher accuracy than K-SVD, D-KSVD, SRC and LLC, and is comparable to LC-KSVD.

[1] LC-KSVD here is the approach LC-KSVD2 in [107].

Table 5.3: Recognition results using random-face features on the Extended YaleB.

Method	K-SVD[102]	D-KSVD[97]	SRC[101]
Acc. (%)	90.5	94.1	88.6
LLC[120]	LC-KSVD[107]	**DDL-PC1**	**DDL-PC2**
82.3	95.0	**94.5**	**95.3**

5.4.2.2 AR Face Database

The AR face database [4] contains over $4,000$ color face images of 126 persons taken during two sessions, with 26 images per person. The main characteristic of the AR database is that it includes frontal views of faces with different facial expressions, lighting conditions and occlusion conditions. All the faces are cropped to 165×120. Following the standard evaluation protocol, we use a subset of the database consisting of $2,600$ images from 50 males and 50 females. For each person, we randomly select 20 images for training and the other six for testing. We report the results from the average of ten such random splits. Each face image is projected into the 540-dimensional feature vector with a randomly generated matrix as in [97, 107]. The feature descriptors used here are random face features. The parameter γ is set to be 30, β is 0.5, α and λ are 1.0.

We evaluate our approach with a dictionary of size 500 and compare with state-of-art approaches [102, 97, 101, 120, 107]. As shown in Table 5.4, both DDL-PC1 and DDL-PC2 obtain better results than K-SVD, D-KSVD, SRC, LLC and

Table 5.4: Recognition results using random face features on the AR face database.

Method	K-SVD[102]	D-KSVD[97]	SRC[101]
Acc. (%)	87.2	88.8	74.5
LLC[120]	LC-KSVD[107]	**DDL-PC1**	**DDL-PC2**
88.7	93.7	**94.0**	**96.0**

LC-KSVD. DDL-PC2 obtains a 2% improvement over DDL-PC1.

5.5 Conclusions

We presented a novel dictionary learning approach that tackles the pair matching and classification problem in a unified framework. We introduced a discriminative term called 'pairwise sparse code error' based on pairwise constraints and combined it with the classification error term to form the objective function of dictionary learning for better discriminating power. The objective function can be optimized by employing the efficient feature-sign search algorithm. The effectiveness of our approach was evaluated on both face verification and face recognition tasks. Experimental results on face verification demonstrated that our approach is competitive with existing techniques without using facial feature point detectors or other additional information. We also compared our approach with several recently proposed dictionary learning methods on two well-known face databases. Our approach can obtain comparable face recognition performance to state-of-art on both databases.

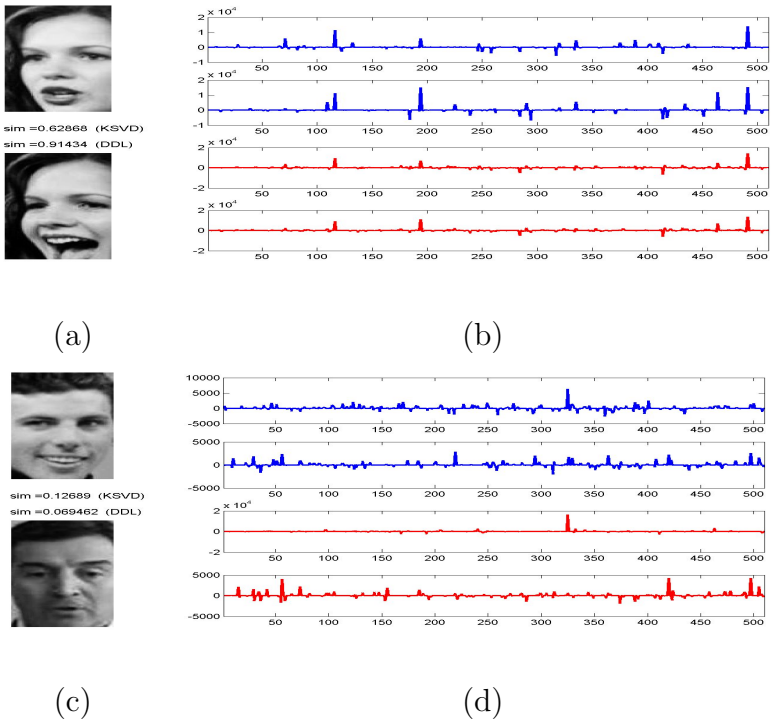
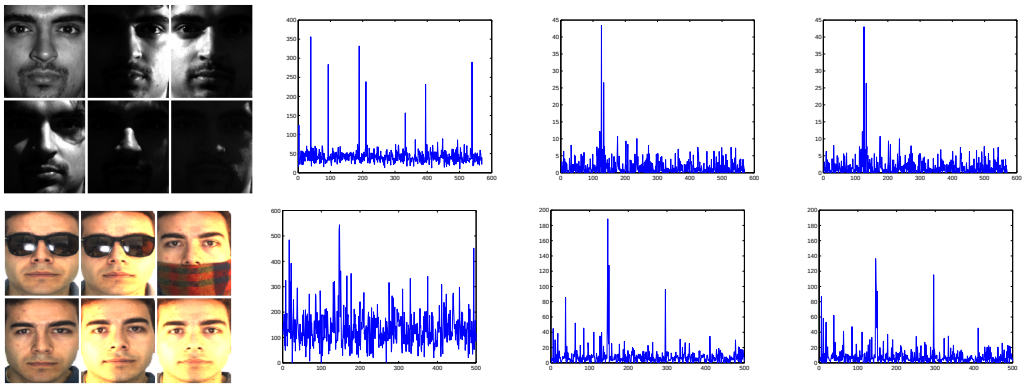


Figure 5.1: An example of sparse codes (HoG feature) and similarity scores obtained by K-SVD dictionary learning and our proposed discriminative dictionary learning with pairwise constraints. Image pairs are from test set 1 of the LFW [2] dataset. (a) Original faces of the 'same' pair and their similarity scores obtained by 'K-SVD' and 'DDL'. (b) Sparse codes for the 'same' pair obtained from 'K-SVD'(blue) and 'DDL'(red), respectively. (c) Original faces of a 'different' pair. (d) Sparse codes for the 'different' pair. It can be seen that our dictionary encourages a pair from 'same' person to have similar sparse codes while a pair from 'different' persons to have dissimilar sparse codes.



(a) Sample images (b) K-SVD (c) DDL-PC1(ours) (d) DDL-PC2(ours)

Figure 5.2: Examples of sparse codes using dictionaries learned by K-SVD and our approaches on the Extended YaleB [3] and AR [4] databases. X axis indicates the dimensions of sparse codes. Y axis indicates the average of absolute sparse codes for different testing images from the same class. The first and second row correspond to class 9 in Extended YaleB (32 images) and class 30 in AR database (6 images), respectively. The consistency of sparse codes of signals from the same class should have low entropy (*i.e.*, less high values) of these average sparse codes.

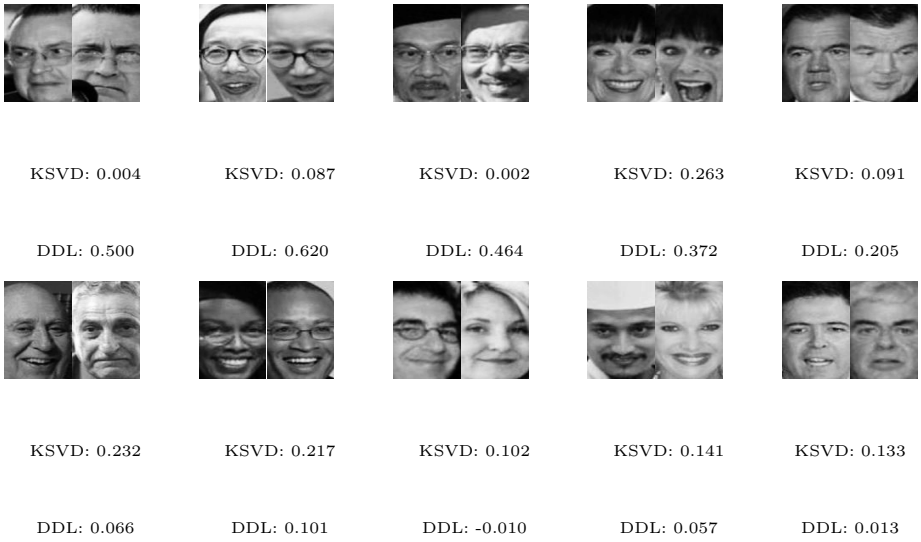


Figure 5.3: Examples of some image pairs from the LFW dataset and the similarity scores obtained from KSVD dictionary learning and proposed DDL-PC1 respectively. Top row: Five examples of '**same**' pairs; Bottom row: Five examples of '**different**' pairs.

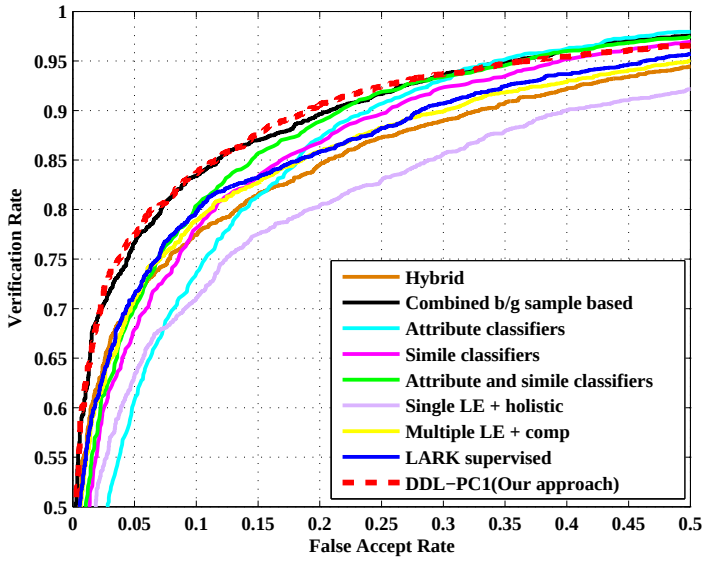


Figure 5.4: ROC curves for View 2 of the LFW dataset (Image-Restricted protocol). Only shown with the selected **best** results that recently reported for clarity.

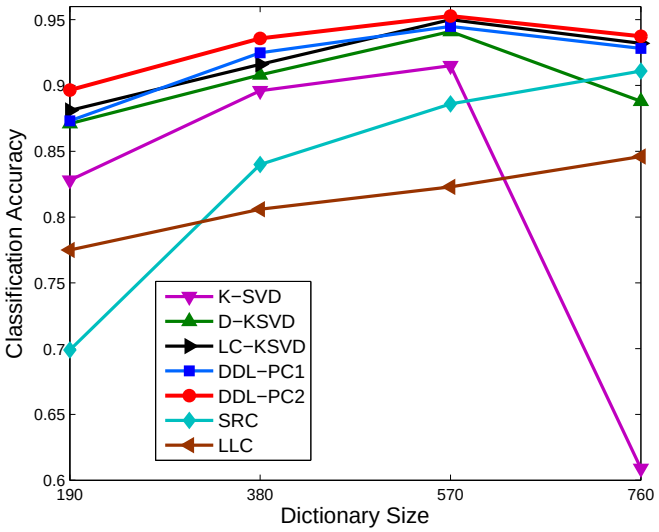


Figure 5.5: Recognition performance on the Extended YaleB with varying number of dictionary sizes.

Chapter 6

Conclusions

Face recognition has wide range of practical applications in access control, identification systems, surveillance, pervasive computing and social networks. Making it more realistic, such as our work presented here: face recognition and verification in unconstrained environments, is a very interesting yet challenging topic.

In this work, we only explored a few directions to address this problem:

(1) We propose a face verification framework that combines Partial Least Squares (PLS) and the One-Shot similarity model. The idea is to describe a face with a large feature set and use PLS regression is applied to perform multi-channel feature weighting. The verification results on the other three very challenging real world datasets (GBU, BDCP, Maritime) taken in unconstrained environments also demonstrate the robustness of our algorithm.

(2) We have proposed an efficient method for face recognition from image set called Covariance Discriminative Learning (CDL). The method represents each image set with its covariance matrix and models the problem as classifying points through kernel mapping on the Riemannian manifold. With superior accuracy, it is also very robust to the practical challenges of noisy set data and varying set size.

(3) We propose a face verification framework using sparse representations that integrates two ways of employing sparsity. The two ways of sparse coding are dif-

ferent yet complimentary. We exploit multiple scores using these two measures and fuse them by simple averaging for the situation where no training set is available (unsupervised) or by an SVM when a training set is given.

(4) We present a dictionary learning framework with explicit pairwise constraints, which unifies the discriminative dictionary learning for pair matching and classification problems. Our approach is validated on various public face verification and recognition benchmarks.

Potential directions and future works extending this dissertation has been discussed in the last section of each chapter.

Bibliography

[1] H. Guo, W. R. Schwartz, and L. S. Davis. Face Verification using Large Feature Sets and One Shot Similarity. In *International Joint Conference on Biometrics*, 2011.

[2] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[3] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[4] A. M. Martinez and R. Benavente. The AR Face Database. Technical report, June 1998.

[5] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.

[6] A.S. Tolba, A.H. El-Baz, and A.A. El-Harby. Face recognition: A literature review. *International Journal of Signal Processing*, 2:88–103, 2006.

[7] X. Tan, S. Chen, Z. Zhou, and F. Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39:1725–1745, 2006.

[8] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification, 2009. *ICCV*.

[9] Jason Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[10] Hieu V. Nguyen and Li Bai. Cosine similarity metric learning for face verification, 2010. *ACCV*.

[11] Eric Nowak. Learning visual similarity measures for comparing never seen objects. In *Proc. IEEE CVPR*, 2007.

[12] Bo Wu and R. Nevatia. Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In *CVPR'08*, pages 1–8, June 2008.

[13] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *ICCV'09*, 2009.

[14] Xiaoyang Tan and Bill Triggs. Fusing Gabor and LBP feature sets for kernel-based face recognition. In *AMFG'07*, pages 235–249, 2007.

[15] Lior Wolf, Tal Hassner, and Yaniv Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2010.

[16] H. Wold. Partial least squares. In S. Kotz and N.L. Johnson, editors, *Encyclopedia of Statistical Sciences*, volume 6, pages 581–591. Wiley, New York, 1985.

[17] W. R. Schwartz and L. S. Davis. Learning discriminative appearance-based models using partial least squares. In *SIBGRAPI'09*, 2009.

[18] C. Dhanjal, S.R. Gunn, and J. Shawe-Taylor. Efficient sparse kernel feature extraction based on partial least squares. *IEEE Transaction on PAMI*, 31(8):1347 –1361, aug. 2009.

[19] Quan sen Sun, Zhong Jin, Pheng ann Heng, and De shen Xia. A novel feature fusion method based on partial least squares regression. In *The third International Conference on. Advances in Pattern Recognition*, volume 1, pages 268–277, 2005.

[20] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *ICCV'09*, Sept. 2009.

[21] Lior Wolf, Tal Hassner, and Yaniv Taigman. Similarity scores based on background samples. In *ACCV'09*, 2009.

[22] Matthew Barker and William Rayens. Partial Least Squares for Discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

[23] Hai-Ni Qu, Guo-Zheng Li, and Wei-Sheng Xu. An Asymmetric Classifier Based on Partial Least Squares. *Pattern Recognition*, 43(10):3448–3457, 2010.

[24] H. Guo, W. R. Schwartz, and L. S. Davis. Face Verification using Large Feature Sets and One Shot Similarity. In *International Joint Conference on Biometrics*, 2011.

[25] Abhishek Sharma and David Jacobs. Pls based multi-modal face recognition. In *CVPR'11*, 2011.

[26] W.R. Schwartz, H.M Guo, and L.S. Davis. A robust and scalable approach to face identification. In *ECCV'10*, volume 5, pages 476–489, Sept. 2010.

[27] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV'09*, 2009.

[28] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV'09*, pages 498 –505, 2009.

[29] Ngoc-Son Vu and Alice Caplier. Face recognition with patterns of oriented edge magnitudes. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 313–326, 2010.

[30] Gang Hua and Amir Akbarzadeh. A Robust Elastic and Partial Matching Metric for Face Recognition. In *International Conference on Computer Vision (ICCV)*, 2009.

[31] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning-based descriptor. In *CVPR'10*, pages 2707–2714, 2010.

[32] N. Pinto, J.J. DiCarlo, and D.D. Cox. How far can you get with a modern face recognition test set using only simple features? In *CVPR'09*, volume 0, pages 2591–2598, Los Alamitos, CA, USA, 2009.

[33] Michael D. Lieberman, Sima Taheri, Huimin Guo, Fatemeh Mirrashed, Inbal Yahav, Aleks Aris, and Ben Shneiderman. Visual exploration across biomedical databases. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8:536–550, 2011.

[34] Hieu V. Nguyen and Li Bai. Cosine Similarity Metric Learning for Face Verification. In *ACCV*. LNCS, 2010.

[35] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In *International Conference on Computer Vision*, pages 786–791, 2005.

[36] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *ECCV'04*, 35, pages 469–481, 2004.

[37] B. Zhang, S.G. Shan, X.L. Chen, and W. Gao. Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition. *IEEE Transactions on Image Processing*, 16:57–68, 2007.

[38] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19:1635–1650, 2010.

[39] J. Choi, H. Guo, W. R. Schwartz, and L. S. Davis. A Complementary Local Feature Descriptor for Face Identification. In *IEEE Workshop on Applications of Computer Vision*, pages 121–128, 2012.

[40] Jun Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and Bao-Liang Lu. Person-Specific SIFT Features for Face Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 593–596, 2007.

[41] Alberto Albiol, David Monzo, Antoine Martin, Jorge Sastre, and Antonio Albiol. Face Recognition Using HOG-EBGM. *Pattern Recognition Letters*, 29(10):1537–1543, 2008.

[42] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[43] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face Recognition with Learning-based descriptor. In *CVPR*, 2010.

[44] N. Pinto and D. Cox. Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 2011.

[45] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV. (2008) (b) Similarity Scores based on Background Samples*, 2008.

[46] Yaniv Taigman, Lior Wolf, and Tal Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference (BMVC)*, 2009.

[47] Qi Yin, Xiaoou Tang, and Jian Sun. An associate-predict model for face recognition. In *CVPR*, 2011.

[48] Javier Ruiz-del Solar, Rodrigo Verschae, and Mauricio Correa. Recognition of faces in unconstrained environments: a comparative study. *EURASIP J. Adv. Signal Process*, 2009:1:1–1:19, January 2009.

[49] Hae Jong Seo and Peyman Milanfar. Face verification using the lark representation. In *IEEE Transactions on Information Forensics and Security*, 2011.

[50] W. R. Schwartz, H. Guo, and L. S. Davis. A robust and scalable approach to face identification. In *Proceedings of the 11th European conference on Computer vision*, pages 476–489, Berlin, Heidelberg, 2010.

[51] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 947–9546, San Diego, CA, June 2005.

[52] P.J. Phillips, H. Wechsler, J. Huang, and P.J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16:295–306, 1998.

[53] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis. Face Identification Using Large Feature Sets. *IEEE Transactions on Image Processing*, 21(4):2245–2255, 2012.

[54] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR'05*, pages 886–893, 2005.

[55] J. G. Daugman. Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters. *Journal of the Optical Society of America A*, 2:1160–1169, 1985.

[56] Lars Elden. Partial least-squares vs. Lanczos bidiagonalization–I: analysis of a projection method for multiple regression. *Computational Statistics & Data Analysis*, 46(1):11 – 31, 2004.

[57] R. Rosipal and N. Kramer. Overview and recent advances in partial least squares. *Lecture Notes in Computer Science*, 3940:34–51, 2006.

[58] K. Delac, M. Grgic, and S. Grgic. Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology*, 15(5):252–260, 2005.

[59] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[60] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[61] Gary B. Huang, Michael J. Jones, and Eric Learned Miller. LFW Results Using a Combined Nowak Plus MERL Recognizer. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille France, 2008.

[62] Lior Wolf, Tal Hassner, and Yaniv Taigman. Y.: Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV. (2008)*, 2008.

[63] Conrad Sanderson and Brian C. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. In *Proceedings of the Third International Conference on Advances in Biometrics*, ICB '09, pages 199–208, Berlin, Heidelberg, 2009. Springer-Verlag.

[64] P. Jonathon Phillips, J. Ross Beveridge, Bruce A. Draper, Geof H. Givens, Alice J. O'Toole, David S. Bolme, Joseph P. Dunlop, Yui Man Lui, Hassan Sahibzada, and Samuel Weimer. An introduction to the good, the bad, and the ugly face recognition challenge problem. In *FG*, pages 346–353. IEEE, 2011.

[65] R. Chellappa. Annual progress report: Muri on remote multi-modal biometrics for maritime domain. *University of Maryland, College Park, MD, Technical Report*, 2009.

[66] Ognjen Arandjelovic;, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell. Face recognition with image sets using manifold density divergence. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:581–588, 2005.

[67] Wei Fan and Dit-Yan Yeung. Locally linear models on face appearance manifolds with application to dual-subspace based classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1384–1390, 2006.

[68] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1005–1018, 2007.

[69] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[70] Jeffrey Ho, Ming hsuan Yang, and David Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 313–320, 2003.

[71] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[72] Gregory Shakhnarovich, John W. Fisher, and Trevor Darrell. Face recognition from long-term observations. In *In Proc. IEEE European Conference on Computer Vision*, pages 851–868, 2002.

[73] Shin'ichi Satoh. Comparative evaluation of face sequence matching for content-based video access, 2000.

[74] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, June, 2010*, pages 2567–2573, San Francisco, CA, Etats-Unis, June 2011.

[75] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:318, 1998.

[76] Lior Wolf, Amnon Shashua, and Donald Geman. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4:2003, 2003.

[77] Jihun Hamm and Daniel D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning, 2008.

[78] Harold Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377, 1936.

[79] Yiqun Hu and Ajmal S. Mian and Robyn Owens. Sparse Approximated Nearest Points for Image Set Classification. In *Proceedings of the 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'11)*, pages 121–128, Colorado Springs, June 2011.

[80] Ruiping Wang, Huimin Guo, L.S. Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2496 –2503, june 2012.

[81] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *In Proc. 9th European Conf. on Computer Vision*, pages 589–600, 2006.

[82] Oncel Tuzel, Fatih Porikli, and Peter Meer. Human detection via classification on riemannian manifolds. In *IN PROC. OF THE IEEE CONF. ON COMPUTER VISION AND PATTERN RECOGNITION*, pages 1–8, 2007.

[83] Yanwei Pang, Yuan Yuan, and Xuelong Li. Gabor-based region covariance matrices for face recognition. *IEEE Trans. Circuits Syst. Video Techn.*, 18(7):989–993, 2008.

[84] Wolfgang F?rstner, Boudewijn Moonen, Fdpdq Gdq, and Carl Friedrich Gauss. A metric for covariance matrices, 1999.

[85] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Analysis Applications*.

[86] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[87] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach, 2000.

[88] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007.

[89] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Primal space sparse kernel partial least squares regression for large scale problems. In *In Proceedings of International Joint Conference on Neural Networks (IJCNN*, pages 561–566, 2004.

[90] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[91] Ralph Gross and Jianbo Shi. The cmu motion of body (mobo) database. Technical report, Robotics Institute, CMU, 2001.

[92] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR03*, pages 409–415, 2003.

[93] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:429–436, 2009.

[94] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Trans. PAMI*, 31(2):210–227, 2009.

[95] Ivor Wai-Hung Tsang Shenghua Gao and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. In *ECCV'10*.

[96] Z. Jiang, Z. Lin, and L. S. Davis. Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *CVPR*, 2011.

[97] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.

[98] Huimin Guo, Ruiping Wang, Jonghyun Choi, and L.S. Davis. Face verification using sparse representations. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 37 –44, june 2012.

[99] T. Ahonen, A. Hadid, and M. Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE T. PAMI*, 28(12):2037–2041, 2006.

[100] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808, 2007.

[101] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31, February 2009.

[102] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionries for sparse representation. *IEEE Trans. on Signal Processing*, 54(1):4311–4322, 2006.

[103] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms, 2007. *NIPS*.

[104] P. Nagesh and Baoxin Li. A compressive sensing approach for expression-invariant face recognition, 2009. *CVPR*.

[105] Meng Yang and Lei Zhang. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary, 2010. *ECCV*.

[106] Ivor Wai-Hung Tsang Shenghua Gao and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition, 2010. *ECCV*.

[107] Zhuolin Jiang, Zhe Lin, and L.S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd, june 2011. *CVPR*.

[108] Meng Yang, Lei Zhang 0006, Xiangchu Feng, and David Zhang. Fisher discrimination dictionary learning for sparse representation., 2011. *ICCV*.

[109] K. Huang and S. Aviyente. Sparse representation for signal classification, 2007. *NIPS*.

[110] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition, 2010. *CVPR*.

[111] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification, 2007. *Conf. on Uncertainty in AI*.

[112] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edage detection and image interpretation, 2008. *ECCV*.

[113] W. Zhang, A. Surve, X. Fern, and T. Dietterich. Learning non-redundant codebooks for classifying complex objects, 2009. *ICML*.

[114] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaryies, 2007. *IMA Preprint 2213*.

[115] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding, 2010. *CVPR*.

[116] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition, 2008. *CVPR*.

[117] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning, 2009. *NIPS*.

[118] Shenghua Gao, Ivor W. Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, 2010.

[119] Shuicheng Yan, Huan Wang, Xiaoou Tang, and T. Huang. Exploring feature descritors for face recognition, 2007. *ICASSP*.

[120] J. Wang, J. Yang, K. Yu, F. Lv, T. huang, and Y. Gong. Locality-constrained linear coding for image classification, 2010. *CVPR*.