Feature extraction and classification for image analysis

Liu, Nan

2009

Liu, N. (2009). Feature extraction and classification for image analysis. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/20614>

<https://doi.org/10.32657/10356/20614>

Feature Extraction and Classification for Image Analysis

Liu Nan

**School of Electrical & Electronic Engineering
NANYANG TECHNOLOGICAL UNIVERSITY**

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Liu Nan

To my parents for their encouragement.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Wang Han, for his professional guidance and advice throughout my research studies towards the Ph.D. degree. Dr. Wang's invaluable support and continuous encouragement greatly increased my confidence and made the whole research process thoroughly enjoyable. I also learned a lot from his analytic skills and insightful ways of looking at problems. I am deeply grateful to Nanyang Technological University for providing me research scholarship. Thanks also go to all the members of the Intelligent Robotics Lab in Nanyang Technological University for the nice atmosphere.

This work is dedicated to my parents who have provided me with the ideal environment to grow up in. Their persistent emotional support always gives me the strength to pass through all kinds of difficulties. I would like to extend my special thanks to those faithful friends, for the kind help they have ever offered to me and every great time we have shared together.

Abstract

Pattern recognition techniques have been widely used in a variety of scientific disciplines including computer vision, image understanding, biology and so on. Although many methods present satisfactory performances for image analysis, they still have several weak points and thus leave a lot of room for further improvements. For example, the linear discriminant analysis (LDA) algorithm is able to extract discriminative features, but the small sample size (SSS) problem limits its application scope.

In this thesis, several feature extraction and learning algorithms are proposed to improve the classification performance in image analysis. In the first proposal, the multiple Trace feature (MTF) is constructed as a novel pattern representation by integrating several Trace transforms where genetic algorithms (GAs) serve as the information fusion tool. Moreover, a novel fitness function is proposed for GAs by combining the bootstrap aggregating algorithm and the cross-validation scheme. As a result, the GAs-based iterative learning process is able to deal with the overfitting problem by using the new fitness.

Since conventional feature extraction algorithms are independent of classification performance, we proposed two performance-driven subspace learning methods as the extensions of the principal component analysis (PCA) and the

kernel PCA (KPCA). In both methods, the genetic algorithms are employed to pursue optimal subspaces by means of a set of class-dependent weights.

Nevertheless, discriminative representations of the original patterns are not necessarily able to achieve satisfactory testing results. Therefore, two novel classifiers are presented, which are derived from the extreme learning machine (ELM) and the evolutionary ELM (E-ELM). These proposed algorithms utilize a cross-validation based training structure to tackle the overfitting problem so that they can generalize well on unseen data.

For each proposed method, extensive experiments are designed to evaluate its effectiveness in various real-world problems including face recognition, object recognition and a number of image understanding tasks. Comparative studies with several state-of-the-art techniques show that the methods developed in this thesis are capable of enhancing the generalization ability for image analysis.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	5
1.3 Contributions of the Thesis	6
1.4 Outlines of the Thesis	8
2 Background	10
2.1 Feature Extraction	11
2.1.1 Principal Component Analysis and Its Variants	12
2.1.2 Linear Discriminant Analysis and Its Variants	15
2.1.3 Manifold Learning Methods	16

2.1.4	Other Feature Extraction Algorithms	18
2.2	Pattern Classification	19
2.2.1	Classifiers	20
2.2.2	Classifier Combination	21
3	Pattern Analysis with Trace Transform	23
3.1	Introduction	23
3.2	Image Representation Using Single Trace Transform	25
3.3	The Proposed Multiple Trace Feature	28
3.3.1	Dimensionality Reduction for Trace Transform	28
3.3.2	Genetic Algorithms-Based Multiple Trace Feature	31
3.3.3	Cross-Validation Plus Bagging: New Fitness Function	35
3.4	Experiments	37
3.4.1	Database	37
3.4.2	Experimental Results	38
3.4.3	Computational Complexity	40
3.5	Summary	43
4	Weighted Principal Components Extraction	44
4.1	Introduction	44
4.2	The Proposed EWPCA Algorithm	49
4.2.1	Weighted Principal Component Analysis	49
4.2.2	Optimal Weights Selection with Genetic Algorithms	54
4.3	The Proposed EWKPCA Algorithm	63

4.3.1	Data Manipulation in High Dimensional Space	63
4.3.2	Weighted Kernel Construction	65
4.3.3	Estimation of Weights	69
4.4	Experiments	71
4.4.1	Face Recognition	72
4.4.2	UCI Image Data Sets	83
4.4.3	Object Categorization	88
4.5	Discussion	90
4.6	Summary	93
5	Classification with Modified Extreme Learning Machine	95
5.1	Introduction	95
5.2	Preliminaries	97
5.2.1	Extreme Learning Machine	97
5.2.2	Evolutionary Extreme Learning Machine	99
5.3	The Proposed Ensemble Based ELM	100
5.4	The Proposed Cross-Validation Based E-ELM	102
5.5	Experiments	104
5.6	Summary	108
6	Conclusion and Future Work	109
6.1	Conclusion	109
6.2	Recommendations for Future Research	111

<i>Contents</i>	vii
Author's Publications	112
Bibliography	114

List of Figures

1.1	General architecture of a typical pattern recognition system. . .	2
3.1	(a) Sample image from ORL face database; (b) the corresponding Trace transform of the image in (a) with functional T_1 in Table 3.1; (c) definition of the parameters of an image tracing line. . .	26
3.2	Feature extraction for single Trace transform.	28
3.3	Eight Trace transforms of the image in Figure 3.1(a). The Trace functionals are T_1 to T_8 in Table 3.1.	32
3.4	The experimental results on combo database with different dimensionality reduction methods: (a) PCA based methods; (b) LDA based methods.	41
4.1	The architecture of the proposed framework.	48
4.2	The visual structure of the feature set $\mathbf{X}$, where $\mathbf{x}_k = [\mathbf{x}_k(1), \mathbf{x}_k(2), \dots, \mathbf{x}_k(M)]^T$. In this example, each face image belongs to one of c persons.	51
4.3	The mechanism of chromosome coding in genetic algorithms. . .	54
4.4	Evolutionary procedure for optimal weights selection.	59

4.5	Two-dimensional mappings of a subset of UCI Optdigits data set. (a) PCA and (b) EWPCA.	62
4.6	Four face databases used in the experiments: (a) GTFD, (b) ORL, (c) UMIST, and (d) Yale.	74
4.7	Results on ORL database. (a) LDA, PCA, and EWPCA; (b) LDA, KPCA, and EWKPCA.	77
4.8	Recognition rate on GTFD data set using EWPCA with different population sizes where feature dimension is five.	78
4.9	Experimental results with the 2DLDA algorithm on two face databases: (a) ORL and (b) GTFD.	80
4.10	Experimental results by combining the state-of-the-art LDA-based methods with EWPCA on ORL and GTFD databases: (a) DLDA-based methods on ORL, (b) DLDA-based methods on GTFD.	81
4.11	Experimental results by combining the state-of-the-art LDA-based methods with EWPCA on ORL and GTFD databases: (a) RLDA-based methods on ORL, (b) RLDA-based methods on GTFD.	82
4.12	Recorded variations on different parameters of EWPCA during evolution where κ is 0.02. The results are based on a subset of UCI Optdigits data set.	85
4.13	Sample images in COIL-20 database. In this example, 6 out of 72 images of the same object are selected to show within-class and between-class differences. As seen from these pictures, large pose variation is observed in each category.	88
4.14	Recognition rate on COIL-20 database where dimensionality of the feature vector is three.	89

5.1	The architecture of the EN-ELM algorithm.	101
5.2	Results on the ORL database using EN-ELM method: (a) Classification results with different number of folds for cross-validation where $\tilde{N}$ is 100; (b) testing results with different number of hidden nodes where R is 10.	107

List of Tables

3.1	List of the Trace functionals used in this study.	27
3.2	Recognition rates on the combo face database.	40
4.1	Recognition rate on face databases.	75
4.2	Comparisons with the state-of-the-art methods in handling the SSS problem on ORL and GTFD face databases.	79
4.3	Summary of UCI data sets. All data sets have been partitioned into two parts for training and testing.	83
4.4	Recognition rate on UCI databases.	84
4.5	Recognition rate with the EWPCA algorithm on a subset of UCI optdigits database where the feature dimension is two. The class separability R_{BW} of the original data set is 1.0549, and the testing accuracy using features extracted by PCA is 82.14%.	86
4.6	Recognition rate on COIL-20 database where six samples per class are used for training.	90
5.1	Data sets used in the experiments.	105
5.2	Comparisons of the experimental results on three face databases and two UCI image data sets.	106

Chapter 1

Introduction

1.1 Motivation

With rapid advancement of computer and database technologies, understanding and mining useful information from huge amount of data attract numerous efforts from the areas of databases, machine learning, and statistics [1]. Pattern recognition is the study of how computers sense the environment, learn from stored patterns of interest, and make decisions to categorize unseen data. Recognizing patterns is an easy task to human beings, whereas it is difficult for machines to accomplish. Nevertheless, since computers have several advantages on processing speed and data storage compared with human, many pattern recognition techniques have been proposed and applied to a variety of scientific disciplines including computer vision, image understanding, speech recognition, computational biology and so on. Image analysis is one of the most studied problems in pattern recognition, which has been widely used in many applications such as face detection and recognition, fingerprint authentication, and

Figure 1.1: General architecture of a typical pattern recognition system.

gait recognition. In this thesis, we concentrate on investigating pattern analysis techniques and their applications to visual learning where the inputs of the system are digital images.

In general, the representations of patterns in image analysis can be pixels, curves, or features from statistical models. Having the representations at hand, correlation, distance measure, or discriminant functions can be applied to distinguish among different patterns [2]. Usually, a learning process on training samples is required, and the prediction of unknown patterns is accomplished by using the parameters learned from the training phase. Figure 1.1 depicts the general architecture of a typical pattern recognition system. Depending on the nature of the application, operations such as segmentation, noise removal, and normalization, are required as preprocessing on the raw data. Subsequently, feature extraction/selection techniques are deployed to reduce the feature dimensionality such that redundant information can be removed. Finally, classifiers are applied for prediction on testing data. This thesis emphasizes the stages of feature extraction and classification.

Template matching [3] is one of the earliest approaches to image analysis, in which geometric features (e.g., circle, corner) are utilized and the similarity between two patterns is evaluated in terms of a correlation. Meanwhile the statistical approach in pattern recognition has been intensively studied and successfully implemented in numerous commercial products. In the statisti-

cal framework, each pattern is represented in terms of d measurements and is viewed as a point in a d -dimensional space. Jain et al. [2] pointed out that the probability of misclassification does not increase with the increment of feature number as long as the class-conditional densities are completely known. They also emphasized that when the number of training samples is much larger than the dimensionality of features, the added features will bring the danger to degrade the generalization performance of the classifier. This phenomenon is a result of the curse of dimensionality [4]. Unfortunately, all the commonly used classifiers suffer from this problem. It is suggested to increase the ratio of the number of training samples to the feature dimensionality to avoid degrading classification accuracy [5]. Moreover, limited yet discriminative features are able to alleviate the burden of classifier design. Apparently, dimensionality reduction is essential to pattern recognition, particularly to image analysis where feature dimension is usually very high.

In the past few years, numerous algorithms have been proposed for dimensionality reduction, among which the principal component analysis (PCA) [6] and the linear discriminant analysis (LDA) [7] are the most prominent. Due to its unsupervised nature, PCA cannot guarantee satisfactory classification results. On the contrary, LDA takes into account class information for extracting features, thus it generally promises to enhance the classification ability. However, LDA suffers from the small sample size (SSS) problem [8] which limits its usage in many applications of image analysis. Moreover, because both PCA and LDA are based on linear mapping whilst most real-world data sets are nonlinearly distributed, it is necessary to seek nonlinear formulations for them. The kernel trick [9] is usually adopted for extending linear method to nonlinear one, which results in the kernel PCA (KPCA) [10] and the kernel Fisher discrimi-

nant analysis (KFDA) [11] algorithms. In spite of these developments, effective feature extraction is still far from being a solved problem in pattern recognition.

In addition to applying powerful dimensionality reduction methods, selection of appropriate pattern representation is also crucial in extracting discriminative features. As a result, the discrete cosine transform [12], the Fourier transform [13], and recently developed Trace transform [14] are employed to create alternative representations for the original images. These successful implementations motivate us to discover novel pattern representations so that the overall generalization performance can be improved.

To accomplish the task of recognition, besides seeking discriminative representation for the original pattern, choosing a suitable classifier plays an important role in both the training and testing phases. During the learning stage, the classification rule is formed by collecting knowledge from training samples, then the well established classifier is applied to categorize unseen testing data. In supervised learning, classifiers always suffer from overtraining which may degrade the generalization performance. In other words, although small training errors are obtained in the training phase, the testing result might be unsatisfactory. Furthermore, it is observed that the sets of patterns misclassified by different classifiers would not necessarily overlap which suggests that combining the outputs of various classifiers has potential to offer better prediction results. Therefore, the ensemble-based decision making strategy [15] is possible to be adopted for constructing reliable image analysis systems.

1.2 Objectives

In the past few decades, many feature extraction and classification methods have been developed for image analysis. However, since the volumes of image data are tremendous and the dimensions of data are very high in many applications, the performances of the systems are far from satisfactory. Therefore, the primary objectives of this thesis are to improve several state-of-the-art feature extraction algorithms and classification strategies to enhance the generalization abilities for a variety of applications in image analysis. Details are given as follows.

- The Trace transform [14] is a recently developed tool to provide an alternative representation for the original image. It has been successfully applied to face recognition [16], [17] and object categorization [14]. All these works utilize a single transform to extract features for recognition. It should be noted that various Trace transforms can be obtained by applying different Trace functionals on a single image. Consequently, diversified characteristics of the image can be discovered. In this scenario, the first objective is to investigate the possibility to fuse multiple Trace transforms together instead of using a single transform to provide a novel image representation.
- Due to the lack of class information, both PCA and KPCA may not be able to extract discriminative features. Meanwhile LDA considers the category property, but it suffers from the SSS problem which limits its usage in many real-world applications. In addition, these subspace methods are not directly related to the classification accuracy. Therefore, the second objective is to propose feature extraction algorithms by associating PCA

and KPCA with LDA such that the SSS problem can be avoided and the generalization ability can be enhanced.

- The extreme learning machine (ELM) [18] is considered as an efficient learning algorithm for single-hidden layer feedforward neural network. Although ELM is fast and presents good generalization performance, it still leaves enough room for improvement. For this classifier, the overfitting problem is inevitable. As the cross-validation scheme is a widely adopted approach to tackle the overfitting problem, the third objective is to discuss its feasibility in working with ELM to generate reliable and efficient classifiers.

1.3 Contributions of the Thesis

The major contributions of this thesis are elaborated as follows:

- A novel image representation using multiple Trace transforms is proposed. Compared to single Trace transform, multiple transforms based representation is possible to characterize various aspects of the original image as diversified features can be extracted by using different Trace functionals. The integration of multiple transforms is accomplished by the genetic algorithms. The evolutionary procedure is guided by the classification accuracy based fitness in the training phase. Moreover, a bootstrap aggregating (Bagging) based fitness is introduced. The rationale behind this improvement is that the use of bootstrap replicates is able to alleviate the overtraining problem and therefore enhance the generalization ability.
- The evolutionary weighted principal component analysis (EWPCA) and

the evolutionary weighted kernel principal component analysis (EWKPCA) algorithms are proposed for feature extraction. Within the above methods, an accuracy-orientated strategy is adopted, where the genetic algorithms play a key role in the entire learning process. EWPCA and EWKPCA extend PCA and KPCA algorithms to be supervised in terms of a modified Fisher criterion which involves in the feature extraction as the discrimination term in the fitness function (another term is the validation accuracy in the training set). This criterion is capable of handling the SSS problem as its calculation is based on geometric distances whereas conventional Fisher criterion suffers from the issue on singular within-class scatter matrix. Furthermore, since the evolutionary optimization process for feature extraction is guided by the validation accuracy in the training set, satisfactory testing results can be expected.

- To have an efficient classifier for image analysis, an ensemble based ELM (EN-ELM) algorithm is proposed, in which the cross-validation scheme is used for training to relieve the problem of overfitting and an ensemble of learners is created for decision making. During the creation of the decision ensemble, the parameters of ELM are adaptively updated according to particular criterions (e.g., validation accuracy), and consequently, the predictors in the ensemble become more robust with respect to classification. After constructing the ensemble, half of the most powerful predictors are selected for making reliable predictions. In addition to EN-ELM, an improved method on evolutionary ELM (E-ELM) is presented by integrating the cross-validation strategy into the training process. The original E-ELM method depends heavily on the validation set (by default, the testing set is equally partitioned into one validation set to facilitate train-

ing and one set for real testing). However, there are no sufficient samples for validation in many image analysis applications. Consequently, we propose the method that does the validation within the training set rather than using an extra data set. Moreover, the cost function in the E-ELM algorithm is also modified so as to fit the task of classification.

1.4 Outlines of the Thesis

This thesis is organized into six chapters, among which previously mentioned major contributions are elaborated in three individual chapters.

Chapter 2 gives an overview on feature extraction and classification for image analysis. It presents several representative feature extraction algorithms and their variants to provide an up-to-date survey for the area of pattern recognition. This chapter also introduces some popular classification methods and classifier combination rules for pattern analysis.

Chapter 3 presents the multiple Trace feature (MTF) as a new image representation which is obtained by taking advantage of the various available Trace transforms generated from a single image. In the process of MTF construction, genetic algorithms serve as the information fusion tool. Moreover, a new type of fitness function based on both the cross-validation scheme and the Bagging algorithm is presented to guide the evolving process.

Chapter 4 proposes an accuracy-driven learning method, from which two novel feature extractors EWPCA and EWKPCA are derived. Within this method, the class-dependent weights are introduced to represent the differences among the samples from various categories, and the genetic algorithms are im-

plemented to seek for suitable weights in terms of optimizing the fitness function. Being different from most evolution-based pattern recognition systems where the learning accuracy is considered as the sole component of fitness, a modified Fisher criterion is derived from the weights as an additional term to express the class separability.

Chapter 5 introduces two novel classifiers based on ELM and E-ELM. As ELM minimizes the training error on the entire training data set and E-ELM calculates the error on a validation set, ELM is subject to overfitting and E-ELM requires extra samples for validation. Therefore, the cross-validation scheme is proposed to be embedded into the training phase so as to relieve the overtraining problem and an extra validation set is not necessarily needed. As a result, one improvement on ELM and an extension of E-ELM are presented in this chapter.

Chapter 6 concludes this thesis and gives several promising directions for the future works.

Chapter 2

Background

As shown in Figure 1.1, a pattern recognition system usually consists of three steps, of which feature extraction and classification are of great importance in determining its generalization performance. The feature extraction step preserves discriminative features by discarding irrelevant attributes, and the classification step is a decision making process with respect to assigning class labels to the testing patterns. Before extracting features and categorizing the samples into different classes, it is essential to seek appropriate representations for the patterns to enable a successful classification. A “good” representation is expected to reveal the structural relationships among the components and is able to express the model of the pattern [7].

In image analysis, geometric features are widely used and can be categorized into low-level features (without shape information) and high-level features (with shape information) [3]. Edges and corners are considered as important low-level feature descriptions, and a number of methods have been proposed for extracting them such as the Canny edge detector [19] and the SUSAN corner detector

[20], etc. Moreover, the scale-invariant feature transform (SIFT) algorithm [21] was proposed to detect and describe local features in images. SIFT is able to find distinctive keypoints that are invariant to location, scale and rotation etc. Lines and circles are typical high-level features as these shapes could be the components of a pattern. Hough transform [22] and its variants [23] are commonly employed for finding shapes in the images.

Even though there are numerous methods for extracting geometric features, we attempt to limit the scope of this thesis to statistical pattern recognition such that the state-of-the-art feature extraction and classification algorithms can be implemented for image analysis (We assume that preprocessing on images such as denoising, enhancement, segmentation etc have been completed). In result, statistical features instead of geometric features like points, curves, or shapes, are extracted to represent the original images. In the statistical formulation, each pattern is represented as a d -dimensional feature vector where the original dimensionality is determined by the total number of pixels in the image. The statistical representation is also widely adopted for applications such as financial data analysis and bioinformatics other than image analysis. In the remaining of this chapter, a number of feature extraction algorithms are introduced. In addition, the issues related to classification are briefed.

2.1 Feature Extraction

In most image analysis systems, the input images are presented in high dimension. In the case of lack of samples, the curse of dimensionality arises and the performance of classifier has the risk of being weakened. Therefore, selecting

or extracting features to form a low-dimensional representation is a necessary procedure, because it will not only facilitate the decision making but also alleviate the computational cost. The techniques for dimensionality reduction can be roughly divided into two categories, feature extraction [2] and feature selection [1], depending on whether or not the original features are transformed to new features. The distinct difference between them is that the feature subset obtained by feature selection methods may retain the important information for understanding physical meaning of the features, while new features generated through linear or nonlinear combination with the help of feature extraction techniques cannot be clearly interpreted in physics.

When browsing current advancement of dimensionality reduction, it is observed that feature extraction attracts more attentions than feature selection because of its superior ability in picking up discriminative features for decision making. The feature extraction methodologies are conventionally categorized as linear methods and nonlinear methods. In this section, we alternatively present the literature review based on several representative algorithms to show a complete picture of this area.

2.1.1 Principal Component Analysis and Its Variants

Principal component analysis (PCA), also known as Karhunen-Loeve transform (KLT), is by far one of the most widely used linear feature extraction methods. Because of its easy-to-use and satisfactory performance, PCA has been applied to a variety of problems, such as face recognition [24], electrocardiogram (ECG) signal processing [25], and microarray data analysis [26]. Since PCA has some drawbacks, numerous researches have been conducted for improving it.

In many applications, it is a common case that parts of the data are not available. To solve this problem, Skočaj et al. [27] proposed a weighted and robust learning subspace method that could enable compensating for the missing data and build a consistent representation for the original data. The rationale behind this method is replacing the standard energy function with a robust weighted energy term. Moreover, their technique is found to be closely related to De la Torre and Black's work [28] while it can be accomplished in a simple and fast manner.

Yang et al. [29] looked at feature extraction for image representation from a two-dimensional point of view. They developed a technique called two-dimensional PCA (2DPCA) which is based on 2D image matrices rather than 1D feature vector. To use the conventional PCA, 2D images must be transformed into 1D vector so that the covariance matrix can be constructed. Because the formulated image vectors are usually in high dimension, the decomposition of such a large matrix becomes difficult. The 2DPCA method directly constructs the covariance matrix from 2D images to facilitate the matrix evaluation and reduce the processing time. This algorithm is specifically designed for image representation, and thus it is not applicable to the data that cannot be expressed in two-dimensional format.

Probabilistic formulation for PCA is another interesting topic in extending the standard PCA. Moghaddam and Pentland [30] proposed a framework for probabilistic visual learning, and Tipping et al. [31] presented a probabilistic PCA (PPCA) algorithm. Furthermore, the dual probabilistic PCA (DPPCA) model was introduced by Lawrence [32] to interpret PCA probabilistically. The advantage of the DPPCA model is that the nonlinear mapping from the em-

bedded space can be easily fulfilled through Gaussian processes.

Since the covariance matrix manipulation is time consuming in PCA, Kumar and Negi [33] introduced the SubXPCA method to alleviate the computational burden and meanwhile achieve superior generalization performance. They also proved that PCA and SubPCA [34] are two special cases of SubXPCA. Both SubPCA and SubXPCA start feature extraction with partitioning each sample to several sub-patterns. The difference between them is that SubPCA neglects the correlations among local features while SubXPCA retains the relationship to summarize the variance.

In [35], Vidal et al. proposed an algebro-geometric approach called generalized principal component analysis (GPCA) to represent the subspaces with a set of homogeneous polynomials where the degree is set as the number of subspaces. They argued that these polynomials can be estimated linearly from data if the number of subspaces is known, consequently the task of subspace segmentation can be considered as classifying one point per subspace. Although GPCA benefits from solid mathematical foundation, it unlikely performs well when the number and the dimensions of the subspaces are large.

Since the aforementioned improvements on PCA are linear methods whilst many real-world data are nonlinearly distributed, it is hard to exploit the essential structure of the data. Kernel PCA (KPCA) [10] is the most prominent nonlinear extension on PCA by mapping the original data to the Hilbert space by means of the kernel trick. Besides utilizing the kernel trick, the strategies of mapping the input data to a number of subspaces by applying a set of nonlinear functions were proposed where the functions are determined either by neural networks [36] or through evolutionary process [37].

2.1.2 Linear Discriminant Analysis and Its Variants

Although the principal components found by PCA are useful for representing data in a compact style, they may not be helpful for discriminating between the patterns from different classes. Linear discriminant analysis (LDA) [7], [38] is a typical supervised algorithm that extracts features by maximizing the between-class scatter matrix and minimizing the within-class scatter matrix. LDA and its variants have been extensively used for pattern recognition [39], especially face recognition [38], [40].

In general, LDA-based methods increase the class separability so that discriminative features can be obtained. Unfortunately, the small sample size (SSS) problem [8] is one of the difficulties faced by LDA. This problem was intensively studied in [41]. To find the solutions for the SSS problem, an early attempt [38] is to implement PCA before applying LDA. Zhao et al. [42] used a small perturbation to the within-class scatter matrix to avoid the singularity. Furthermore, Null space LDA (NLDA) [43] was proposed as another solution and later a Dual-space LDA algorithm [44] was presented as an improvement of NLDA to take full advantage of the whole space. Recently, a new LDA-based algorithm called GA-Fisher [45] was introduced where genetic algorithms are used to automatically select the eigenvectors for LDA. In Chapter 4, we also attempt to solve the SSS problem by combining the merits of LDA and PCA with a novel criterion.

To extend LDA to handle nonlinearly distributed data, Mika et al. [11] proposed the kernel Fisher discriminant analysis (KFDA) method by adopting the kernel trick. In [46], another kernel-based algorithm, the generalized discriminant analysis (GDA) was introduced to deal with nonlinear discrimina-

tive analysis. GDA uses an underlying theory close to support vector machines to map the input vectors into a high-dimensional feature space such that the classification problem can be solved in a linear way.

Since pattern recognition systems usually tackle the K -class problems with $K > 2$, the Fisher criterion is suboptimal with respect to classification rate [47]. Loog et al. [47] therefore derived an approximate pairwise accuracy criterion (aPAC) by introducing weights to the between-class scatter matrix and proved that aPAC-based approach outperformed LDA in many situations. Furthermore, Tang et al. [40] introduced a relevance weighted linear dimensionality reduction (WLDR) to reduce the influence of outlier classes. The weights are derived either from the class separability or through an evolution process. Based on the idea of WLDR, two novel weighted versions of PCA and KPCA are proposed in Chapter 4, where the weights are selected with the genetic algorithms by means of optimizing a modified Fisher criterion.

2.1.3 Manifold Learning Methods

Recently, some manifold learning methods were developed as nonlinear tools for finding the inherent nonlinear structure hidden in the observe space. Unlike kernel-based dimensionality reduction algorithms, manifold learning schemes do not have to map the original data to high dimensions so that computational cost is reduced. In general, these methods can be categorized as two types: global embedding and local embedding. ISOMAP [48] proposed by Tenenbaum et al., is a typical global technique that preserves global structure in the embedding space, whereas Laplacian Eigenmaps [49] and locally linear embedding (LLE) [50] are classified as local methods. Among the family of manifold learning

methods, LLE [51] receives the most attentions in the past few years.

Since Roweis and Saul [50] introduced LLE for nonlinear dimensionality reduction in 2000, many efforts have been done to improve the conventional LLE algorithm. Right after the introduction of LLE and ISOMAP, Donoho and Grimes [52] proposed a Hessian LLE algorithm (HLLE) which is modelled on the original LLE [51]. HLLE is developed to discover a parametrization for data lying on a manifold embedded in high-dimensional Euclidean space.

As LLE is an unsupervised learning algorithm in which low-dimensional, neighborhood-preserving embeddings are computed from high-dimensional inputs, its generalization ability might be poor. Furthermore, its unsupervised nature assumes that the input data reside on a single continuous manifold. However, the multiple-class classification problems are not the case. Li et al. [53] addressed the difficulties in implementing LLE for pattern recognition, and proposed a supervised version of LLE, namely locally linear discriminant embedding (LLDE). In LLDE, a vector translation and distance rescaling model is constructed to increase the discrimination of features. Another supervised extension of LLE, discriminant LLE (DLLE) [54], was developed to preserve local geometric properties within each class and enforce the separability among different classes. It is observed that by incorporating the class information, LLE-based methods are capable of handling recognition problems well.

In [55], Mekuz et al. introduced a framework of subspace manifold learning with sample weights and argued that the usage of weights can not only control the effect of outliers, but also tune the learning of the manifold. Moreover, an incremental LLE method (ILLE) [56] was proposed and demonstrated that it is particularly useful when the data is not uniformly distributed and the

local linearity constraints do not hold. Inspired by the ideas of robust PCA [28], Chang and Yeung [57] presented a robust extension of LLE to tackle with outliers in the data.

2.1.4 Other Feature Extraction Algorithms

Besides the above mentioned popular dimensionality reduction algorithms, there are various other techniques for extracting features to reflect the intrinsic characteristics of the original data. Independent component analysis (ICA) is a computational method for extracting components among which the statistical dependence is minimized. ICA and its enhanced model have been successfully applied to face recognition [58], [59]. Liu and Wechsler [60] presented an independent Gabor features (IGFs) method and evaluated its effectiveness on face images. Moreover, Bach and Jordan [61] extended ICA from linear to nonlinear by means of the kernel method.

The Fukunaga-Koontz transform (FKT) [62] is another methodology for feature extraction. It has been verified intensively for face recognition [63], [64]. Huo [62] analyzed FKT from a statistical point of view and proved that FKT is the optimal low-rank approximation to an optimal classifier under certain conditions. In [64], the standard linear FKT is extended to be able to work in higher dimensional (kernel) subspaces. Recently, Zhang and Sim [65] brought a fresh perspective to LDA via FKT. In their works, a new criterion derived from the Bhattacharyya distance was proposed to handle the multiple discriminant analysis (MDA), which was claimed to be superior to the Fisher criterion.

Although a number of feature extraction methods have been reviewed so far, there are more algorithms to be introduced such as the evolutionary pur-

suit (EP) [66], the constrained maximum variance mapping (CMVM) [67], and the Tensor representation-based discriminant analysis [68]. Furthermore, transformations can be used to transfer the images into new domains so as to extract features. Existing transformation-based systems usually employ techniques such as the discrete cosine transform (DCT) [69], [70], the Fourier transform [13], the Gabor wavelets [71], the Radon transform [72] and the Trace transform [17]. As there are limited studies on the applications of Trace transform, we will investigate the possibility of extracting discriminative features from multiple transforms. Generally speaking, the dimensionality reduction methods attempt to extract discriminative yet compact features from the original images to improve the generalization ability. Therefore, it is always necessary to look for powerful feature extractors so as to achieve satisfactory testing performance.

2.2 Pattern Classification

Once a proper representation is found for the original image pattern, a classifier can be designed to learn the intrinsic characteristics from training samples and predict labels for testing instances. In practice, the choice of a classifier depends on the requirements of the applications. For example, if the system is required to respond fast to the inputs, the classifier should not be too complex. In the rest of this section, several popular classifiers are reviewed and some rules for classifier combination are highlighted.

2.2.1 Classifiers

Serving as one of the major tasks in image understanding, classifier design has a long history. Therefore, numerous classifiers have been proposed. Judging a classifier usually depends on evaluating its generalization ability that refers to the classifier's performance in categorizing unseen patterns [2]. Since the same classifier may have various performances on different applications, it is suggested to analyze the needs of the system before choosing a proper classifier.

The brief introduction to the classifiers starts with describing several widely used methods when pattern recognition is in an early stage of development. As one of the simplest approaches, the k -nearest neighbor algorithm (k -NN) [73] has been demonstrated to have satisfactory classification performances on a wide range of applications. Due to its nonparametric nature, k -NN can be used for any classification problems even if the distributions of the feature values are not Gaussian [74]. As a parametric method, the naive Bayes classifier [7] makes decisions with strong independence assumptions, that is, the presence of a particular feature of a class is unrelated to the presence of any other feature. In practice, a small amount of training data is usually enough for the naive Bayes classifier to categorize patterns. In data mining and machine learning, decision tree is another practical method for inductive inference [75]. The C4.5 algorithm [76] developed by Quinlan has been one of the most famous methods for generating a decision tree.

Support vector machines (SVMs) [77], [78], a set of related supervised learning methods based on a novel statistical learning theory [79], have been proposed for effective classification and regression. Schölkopf and Smola [77] presented a thorough study on SVMs. So far, SVMs have been successfully implemented

for face recognition [80], [81], [82], and computational biology [83], etc. Furthermore, due to their popularity in pattern recognition, a lot of softwares have been developed, among which the LIBSVM package [84] is a prominent one.

Taking advantage of its capability to learn complex nonlinear input-output relationships, neural network (NN) [7] has been actively used to do predictions with the trained network. The feed-forward network is a commonly adopted network topology for pattern recognition [85]. Regarding its disadvantages, the extreme learning machine (ELM) [18] was introduced and has been further improved and validated [86], [87], [88]. Compared with the conventional gradient-based learning approaches, ELM has a fast learning process and meanwhile retains good generalization ability. In Chapter 5, two classifiers are proposed by extending the ELM and E-ELM algorithms.

2.2.2 Classifier Combination

In image analysis systems, there may have different feature sets or different classification methods. Therefore, we can obtain a set of classifiers (an ensemble) and attempt to combine them to improve the overall generalization ability. Different combining schemes can result in diversified outputs. Jain et al. [2] brought us a concise yet informative introduction to the classifier combination. Polikar [15] comprehensively reviewed the area of multiple classifier system (ensemble system) for decision making.

The outputs of a predictor can be either class labels or class-specific continuous values (the degrees of support given to those classes), and this thesis focuses on the label-based combination rules. Majority voting [89] is one of the most commonly used combining strategies. This rule seeks the class that

receives the highest number of votes and assigns it to the predicted label for the testing pattern. Since each classifier in an ensemble does not necessarily contribute to the final decision equally, Littlestone and Warmuth [90] proposed weighing each classifier to make individual predictors discriminative. However, if the performances of the classifiers are not known as prior knowledge, it is not easy to determine the weights for the classifiers in the ensemble. Several strategies can be employed to evaluate a classifier such as testing it on a separate validation set, or even on the training set [15].

To obtain satisfactory testing results, adequate and representative training samples are of great importance to guarantee a successful learning. However, applications such as face recognition [8] and microarray data analysis [88], suffer from the lack of training data. A plausible solution is to use resampling techniques to create a series of partially overlapped random subsets from the original data and subsequently train the classifiers. Then the aforementioned combination rules can be applied to do an effective decision making on unseen data. The bootstrap resampling is a simple yet powerful tool on which two famous ensemble methods, namely bootstrap aggregating (Bagging) [91] and Boosting [92], were developed. In [93], an insightful survey was given to the bootstrap-inspired techniques for computational intelligence. Bagging uses an intuitive and simple structure to improve the overall performance. It generates a set of bootstrap replicas of the training data from which the classifiers are trained, and combines the outputs of these predictors on testing patterns with majority voting. As a sophisticated extension to Boosting, AdaBoost [94] employs weighted majority voting to make decision with hypotheses obtained from weak learners, where the weights are calculated through iterative learning.

Chapter 3

Pattern Analysis with Trace Transform

3.1 Introduction

In pattern classification, people are not always satisfied with the quantity of training samples, especially in those applications related to human face authentication, where only few images are available per person. Therefore, numerous feature extraction methods are proposed to select as many discriminative features as possible for recognition. However, those instances are limited, and no matter how powerful the features and the classifiers are, the increase on the generalization performance has an upper bound.

In recent years, a number of researches have been conducted to investigate the possibility of extracting features from new image representations, such as the frequency spaces obtained by performing the discrete cosine transform [12] and the fourier transform [13]. Trace transform [14] provides an alternative

representation of the original image. Given different Trace functionals, various data sets are created in which differences among transforms of the same image can be explored. The geometrically invariant *triple features* and affine invariant features [14], [95], [96], [97] are extracted from Trace transform for object representation and recognition. Regarding the high computational burden in generating Trace transforms, Fahmy et al. [16], [98] proposed accelerating Trace transform-based recognition with FPGA hardware. Besides generating invariant *triple features*, Trace transform offers an alternative representation of a 2D grey object [96]. Therefore, depending on the choice of functional, it is possible to make explicit differences among 2D transforms of the same image which are implicitly presented in the original representation [17]. This is the reason that we would like to investigate the potential usage of the Trace transform for face recognition. Some works [17] have been done in recognizing face images by using Trace transform. The authors convert the Trace transform to binary edge map by setting some thresholds, and the efforts are put on the shape matching of the features extracted from a single transform. As those transforms obtained from different Trace functionals are handled separately, our proposed method focuses on utilizing multiple Trace transforms together and fusing them to construct discriminatory features which are able to improve the overall generalization performance.

In this chapter, multiple Trace feature (MTF), a novel image representation constructed through evolutionary process, is proposed. Moreover, a new type of fitness function based on both cross-validation [31] and Bagging [91] is proposed to guide the evolving procedure to achieve better testing performance. Algorithmic details and supporting results on performance evaluations are described in the remaining sections.

3.2 Image Representation Using Single Trace Transform

The Trace transform [14] is a new approach for image processing from which invariant features under transformations such as rotation, translation and scaling can be extracted. For clarification, two coordinate systems are defined, the original C_1 and the distorted C_2 , where C_2 is obtained from C_1 by means of rotation angle $-\theta$, scaling ν and translation parameter $(-s_0 \cos \psi_0, -s_0 \sin \psi_0)$. It is shown that straight lines will still appear as straight lines after the transformations between two coordinate systems [14]. Therefore, a line parameterized by (ϕ, p, t) has following parameters in the old coordinate system C_1 ,

$$\phi_{old} = \phi - \theta \quad (3.1)$$

$$p_{old} = \nu[p - s_0 \cos(\psi_0 - \phi)] \quad (3.2)$$

$$t_{old} = \nu[t - s_0 \sin(\psi_0 - \phi)] \quad (3.3)$$

Assume scanning an image with multiple lines as shown in Figure 3.1, the Trace transform T is generated in which p is the distance of the line from image center and ϕ is used to indicate the orientation of the normal to the line. Consequently, the Trace transform has the following mathematical formulation,

$$g(F; C_1; \phi, p) = T(F(C_1; \phi, p, t)) \quad (3.4)$$

where $F(C_1; \phi, p, t)$ means the values of the image function along the chosen line. In other words, assuming that o is the center of the image, a number

Figure 3.1: (a) Sample image from ORL face database; (b) the corresponding Trace transform of the image in (a) with functional T_1 in Table 3.1; (c) definition of the parameters of an image tracing line.

of tracing lines defined by ϕ and p are used to scan the image and then the Trace functional is applied on each line for calculation, where $\phi \in (0, 2\pi]$ and p is determined by the image dimension. In summary, the generation of Trace transform is a standard procedure which can be applied to any image in the same manner. It can be seen in Figure 3.1(b) that the resultant representation is a two-dimensional function of variables ϕ and p , and can be interpreted as a new 2D image. Since the *triple features* [14] are obtained by applying three functionals on the original image, discriminative information may be removed during calculations, we therefore make use of the Trace transform of the image

Table 3.1: List of the Trace functionals used in this study.

T_t	Trace Functionals $T(f(t))$ and descriptions
T_1	$[\int_0^\infty f(t) ^p dt]^q, p = 4, q = 1/p$
T_2	$\int_0^\infty f(t)' dt$, one dimensional numerical gradient of t , $f(t)' = [(t_2 - t_1), (t_3 - t_2), \dots, (t_n - t_{n-1})]$
T_3	$\text{median}_t\{f(t), f(t)' \}$
T_4	$\int_0^\infty \left \frac{d}{dt} \mathcal{M}\{f(t)\} \right dt$, $\mathcal{M}$ is a median filtering operator, using a local window of length 3, and differentiation means taking the difference of successive samples
T_5	$\text{median}_t\{r^p f(t), f(t)' ^{1/2}\}$, $p = 0.5, r = l - c , l = 1, 2, \dots, n, c = \text{median}_l\{l, f(t)\}$
T_6	$\int_{c^*}^\infty r^p f(t) dt$, $p = 0.5, r = l - c , l = 1, 2, \dots, n, c = \text{median}_l\{l, f(t)\}$, c^* signifies the nearest integer to c
T_7	$\left \int_{c^*+1}^\infty e^{ik \log(r)} r^p f(t) dt \right $, $p = 0, k = 3, r = l - c , l = 1, 2, \dots, n, c = \text{median}_l\{l, f(t)^{1/2} \}$, c^* signifies the nearest integer to c
T_8	$\int_c^\infty r^p f(t) dt$, $p = 0.5, r = l - c , l = 1, 2, \dots, n, c = \frac{1}{S} \int_0^\infty r f(t) dt, S = \int_0^\infty f(t) dt$

for facial feature extraction. As the resultant transform depends on the functional that we use, different Trace transforms can be produced from an image using different functionals. So far there are 22 available Trace functionals [17], and some of them share similar mathematical formulations. Therefore, in our implementations only eight diversified functionals shown in Table 3.1 are used. The designation $\text{median}_a\{a, b\}$ means the weighted median of sequence a with weights in the sequence b . For example, $\text{median}\{\{2, 8, 1, 6\}, \{2, 2, 3, 2\}\}$ indicates the median of numbers 2, 8, 1 and 6 with corresponding weights 2, 2, 3, and 2. This means the standard median of numbers 2, 2, 8, 8, 1, 1, 1, 6, 6, i.e.,

Figure 3.2: Feature extraction for single Trace transform.

the median of the ranked sequence 1, 1, 1, 2, 2, 6, 6, 8, 8. Thus the result is 2.

3.3 The Proposed Multiple Trace Feature

3.3.1 Dimensionality Reduction for Trace Transform

Given that Trace functional T_t is implemented to the original image, $g^t(F; C_1, \phi, p)$ is generated where t is the index of Trace functional. In order to extract features for further processing such as classification, the 2D image g^t is divided into ϕ vectors to form 1D string as shown in (3.5).

$$\mathbf{x}^t = [g^t(F; C_1, 1, p)^T, \dots, g^t(F; C_1, \phi, p)^T]^T \quad (3.5)$$

However, the vector dimension is fairly large (e.g., $\sim 50,000$) so that redundant or irrelevant information may exist. Thus, dimensionality reduction algorithms are required to decrease the feature complexity. To simplify the description, the superscript t is omitted thereafter within this section.

The feature extractors PCA and LDA are implemented to remove redundancy within the Trace transforms. PCA generates a set of orthonormal basis

vectors, known as principal components, that maximize the scatter of all the projected samples. Assume that there are N patterns and each has M observations in the feature set, we have the data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$. By substituting the grand mean $\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$, the data set becomes

$$\tilde{\mathbf{X}} = [\mathbf{x}_1 - \bar{\mathbf{X}}, \mathbf{x}_2 - \bar{\mathbf{X}}, \dots, \mathbf{x}_N - \bar{\mathbf{X}}] \quad (3.6)$$

Then the covariance matrix of the data set is subsequently defined as

$$\Sigma_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{X}})(\mathbf{x}_i - \bar{\mathbf{X}})^T = \frac{1}{N} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T \quad (3.7)$$

Having $\Sigma_{\mathbf{x}}$, the eigenvector and eigenvalue matrices $\mathbf{U}$, $\mathbf{\Lambda}$ are computed as $\mathbf{\Lambda} \mathbf{U} = \Sigma_{\mathbf{x}} \mathbf{U}$, where $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of covariance matrix $\Sigma_{\mathbf{x}}$, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]$. Assume that the eigenvalues are sorted in decreasing order, the first m leading eigenvectors define the feature space $\tilde{\mathbf{U}} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$, to which the samples will be projected. Considering any feature vector $\mathbf{x}_k \in \mathbf{X}$, the principal component vector $\mathbf{y}_k$ is an orthogonal projection of data $\mathbf{x}_k$, described by $\mathbf{y}_k = \tilde{\mathbf{U}}^T \mathbf{x}_k$, that is,

$$\mathbf{y}_k = [\mathbf{u}_1^T \mathbf{x}_k, \mathbf{u}_2^T \mathbf{x}_k, \dots, \mathbf{u}_m^T \mathbf{x}_k]^T \quad (3.8)$$

$\mathbf{y}_k$ captures the most expressive features of $\mathbf{x}_k$ by projecting the original data onto the subspace. However, PCA tends to maximize the scatter of all projected samples, which may not be appropriate for recognition as the differences among patterns are not able to be distinguished.

Although PCA is efficient for dimensionality reduction, in the context of classification, PCA paradigm does not provide any information for class dis-

crimination. Therefore, LDA [7] is proposed to embed the between-class and within-class information for dimensionality reduction. We assume that each pattern $\mathbf{x}_k$ in set $\mathbf{X}$ belongs to one of c classes $\{l_1, l_2, \dots, l_c\}$. LDA is used to find an optimal subspace in which the ratio of the between-class scatter and the within-class scatter is maximized. Let the between-class scatter matrix be defined as

$$\mathbf{S}_B = \sum_{i=1}^c N^i (\bar{\mathbf{X}}^i - \bar{\mathbf{X}})(\bar{\mathbf{X}}^i - \bar{\mathbf{X}})^T \quad (3.9)$$

and the within-class matrix be defined as

$$\mathbf{S}_W = \sum_{i=1}^c \sum_{\mathbf{x}_k \in l_i} (\mathbf{x}_k - \bar{\mathbf{X}}^i)(\mathbf{x}_k - \bar{\mathbf{X}}^i)^T \quad (3.10)$$

where $\bar{\mathbf{X}} = \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k$ is the grand mean of the data set, and $\bar{\mathbf{X}}^i$ is the mean of class l_i . N^i is the number of samples in class l_i . If $\mathbf{S}_W$ is nonsingular, the optimal subspace Ψ_{opt} is chosen as the matrix with orthonormal columns which maximizes the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix of the projected samples,

$$\begin{aligned} \Psi_{\text{opt}} &= \arg \max_{\Psi} \frac{|\Psi^T \mathbf{S}_B \Psi|}{|\Psi^T \mathbf{S}_W \Psi|} \\ &= [\psi_1, \psi_2, \dots, \psi_m] \end{aligned} \quad (3.11)$$

Ψ_{opt} is a set of generalized eigenvectors of $\mathbf{S}_B$ and $\mathbf{S}_W$ corresponding to m largest generalized eigenvalues λ_i , i.e., $\mathbf{S}_B \Psi_i = \lambda_i \mathbf{S}_W \Psi_i$, and $i = 1, 2, \dots, m$. Note that $\text{rank}(\mathbf{S}_B) \leq \min\{m, c-1\}$, thus there are at most $c-1$ nonzero generalized eigenvalues, and the upper bound on m is $c-1$ [7]. The feature vector $\mathbf{y}_k$ for any query variable $\mathbf{x}_k$ can be calculated as $\mathbf{y}_k = \Psi_{\text{opt}}^T \mathbf{x}_k$. In comparing with the features extracted by performing PCA, $\Psi_{\text{opt}}^T \mathbf{x}_k$ may achieve better classification

results because of the integration of class information.

3.3.2 Genetic Algorithms-Based Multiple Trace Feature

Single Trace transform can be used instead of the original image for classification [14], [17], but there is no prior knowledge on how efficient a specific functional T_t is. In other words, the Trace transform using one functional may perform well in application A while unsatisfactory in accomplishing task B. In face recognition, specific Trace functional T_t could be found during a training phase in terms of successfully suppressing variations in pose, illumination and expression [17]. Although the low-dimensional vector $\mathbf{y}^t$ captures the most expressive features, its high variance does not necessarily lead to good discriminatory ability unless the corresponding distribution is multimodal and the modes correspond to the classes to be discriminated [31]. An advantage of the Trace transform is its capability of generating several transforms by implementing different functionals. Each Trace transform may reveal diverse aspect of the original image by means of the inherent properties of the functionals. Figure 3.3 shows the diversity of Trace transforms for the same original image. Rather than selecting the “best” Trace transform, we attempt to investigate the possibilities of combining various transforms to synthesize a new image representation for image analysis. Therefore, we propose a novel image representation by fusing multiple Trace feature vectors. To clarify the terminology, the new representation is called the multiple Trace feature. Given a single image, a series of feature vectors $\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^8$ are generated by implementing Trace functionals $T_1, T_2, \dots, T_8$. Suppose

Figure 3.3: Eight Trace transforms of the image in Figure 3.1(a). The Trace functionals are T_1 to T_8 in Table 3.1.

that the feature dimension is m , we have

$$\begin{cases} \mathbf{y}^1 = [\mathbf{y}^1(1), \mathbf{y}^1(2), \dots, \mathbf{y}^1(m)]^T \\ \mathbf{y}^2 = [\mathbf{y}^2(1), \mathbf{y}^2(2), \dots, \mathbf{y}^2(m)]^T \\ \vdots \\ \mathbf{y}^8 = [\mathbf{y}^8(1), \mathbf{y}^8(2), \dots, \mathbf{y}^8(m)]^T \end{cases} \quad (3.12)$$

According to the proposed feature fusion mechanism, the new representation is

$$\mathbf{y} = [\mathbf{y}^{I_1}(1), \mathbf{y}^{I_2}(2), \dots, \mathbf{y}^{I_m}(m)]^T \quad (3.13)$$

where $I_i \in \{1, 2, \dots, 8\}$ and $i = 1, 2, \dots, m$. Therefore, 8^m possible new features are created. Obviously, a huge number of feature combinations are there, from which we should select proper features for generating satisfactory classification performance. Hence, the problem can be considered as a task of discovering global optimum. It is a challenge to tackle such a complex task, as most optimization algorithms may get into local minimum during the convergence.

As the genetic algorithms (GAs) [99] are a class of robust optimization method and have been used extensively in pattern recognition [66], we naturally want to take the merit of GAs to solve the above problems. Genetic algorithms start the search by randomly generating a set of scalars to form the initial population. Then the GAs make choices of evolution via genetic operators selection, crossover, and mutation. The fitness plays a dominant role in controlling the convergence to global optimum.

The chromosome in the population is coded as a set of m (feature dimension) scalars $\alpha_1, \alpha_2, \dots, \alpha_m$ where $\alpha_i \in (0, 8)$ and $i = 1, 2, \dots, m$. The scalars can be

interpreted as:

$$\left\{ \begin{array}{l} \text{if } \alpha_i \in (0, 1], \quad t = 1 \Rightarrow \mathbf{y}(i) = \mathbf{y}^1(i) \\ \text{if } \alpha_i \in (1, 2], \quad t = 2 \Rightarrow \mathbf{y}(i) = \mathbf{y}^2(i) \\ \quad \quad \quad \vdots \\ \text{if } \alpha_i \in (7, 8), \quad t = 8 \Rightarrow \mathbf{y}(i) = \mathbf{y}^8(i) \end{array} \right. \quad (3.14)$$

Since eight Trace functionals are used, the upper bound of α_i is 8. In general, a set of N_p (population size) chromosomes are randomly generated to constitute the initial population. In order to get a trade-off between the computational cost and the diversity of the population, we use a population size of 100 in experimental validation. Then, for each generation k , $\mathbf{P}(k)$ indicates the population of individuals (chromosomes). During each generation, individuals are evaluated by calculating their fitness values, based on which the chromosome selection is done probabilistically for creating a mating pool $\tilde{\mathbf{P}}(k)$. The fitness related selection mechanism can ensure that fitter chromosomes have higher probability of survival. Classification accuracy (CA) is usually employed as the fitness function to guide the search in the field of evolutionary pattern recognition [66]. In order to avoid overtraining which may degrade the overall generalization performance, K -fold cross-validation scheme is applied for measuring the fitness values. Thus the fitness function $\zeta(\mathcal{F})$ is computed as $\frac{1}{K} \sum_{i=1}^K \text{CA}_i$, where CA_i is the recognition rate in one of K validation sets of the training data.

After evaluating the fitness, GAs move the step to create a new generation $\mathbf{P}(k+1)$. Firstly, $(1 - p_c)N_p$ members of $\tilde{\mathbf{P}}(k)$ with top fitness values are probabilistically selected to be part of $\mathbf{P}(k+1)$. Then, crossover operator is applied to the remaining individuals in $\tilde{\mathbf{P}}(k)$. The offsprings after crossover

together with the selected chromosomes form the new generation $\mathbf{P}(k + 1)$. Finally, $p_m \cdot N_p$ candidates in the new generation undergo mutation so as to introduce diversity into the population to prevent a premature convergence where p_c and p_m are the crossover and mutation probabilities, respectively. The evolutionary process will be repeated several times until the termination criteria is achieved. In our work, 200 generations are set as the criterion to stop the evolution. Once the training process is finished, the scalars encoded in the optimal individual of the last generation are used for testing.

3.3.3 Cross-Validation Plus Bagging: New Fitness Function

Bagging predictor [91] generates multiple models using bootstrap samples, and votes for class prediction in classification task. By implementing such scheme, classification accuracy and stability are improved, and overfitting can be avoided due to reduced variance. Therefore, we propose incorporating the Bagging scheme to enhance cross-validation based fitness function so that higher recognition rate can be achieved.

Let L denote the training set, in calculating fitness using K -fold cross-validation, approximately $\frac{(K-1)N}{K}$ instances in L_{trn} are used for training and the remaining $\frac{N}{K}$ samples in L_{val} are used for validation during the evolutionary procedure. As the number of instances in L_{val} is small in face recognition, high classification accuracy could be achieved in the evolving stage, while poor performance is probably observed in real testing. In addition, as there are usually limited samples per class in face databases, the entire sets of samples from some classes may be assigned into either L_{trn} or L_{val} , which will result in an

unsuccessful training. To investigate possible solutions for the above difficulties, Bagging method is combined with cross-validation scheme to facilitate the training in which validation is done on L rather than L_{val} . In result, patterns from all classes can involve in both training and validation and contribute to the learning process. Furthermore, bootstrap replicates of L_{trn} are used so as to avoid possible overtraining which is brought by big overlaps between L and L_{trn} (Note that L but not L_{val} is used for validation in the training process). Technical details are described as follows. Given data set $L_{\text{trn}} = \{(\mathbf{y}_n, l_n)\}$, $n = 1, 2, \dots, \frac{(K-1)N}{K}$ with labels $l_n \in \{l_1, l_2, \dots, l_c\}$, $h(\mathbf{y}, L_{\text{trn}})$ is defined as the predictor, i.e., classifier. Suppose B bootstrap replicates are to be created, we repeat the following process B times. Firstly, we take a bootstrap sample L_{trn}^b by randomly drawing p_b percent of L_{trn} with replacement, where $b = 1, 2, \dots, B$. Then we call the learner $h_b(\mathbf{y}, L_{\text{trn}}^b)$ to receive the hypothesis and add h_b to the ensemble. So far we have B hypotheses in the ensemble. The next action is to make a final decision using plurality majority voting. For any test instance $\mathbf{y}' \in L$, a series of predictions $h_1, h_2, \dots, h_B$ are available. Then the vote given to class l_j by predictor h_b is defined as

$$\mathcal{D}_{b,j} = \begin{cases} 1 & \text{if } h_b \text{ picks class } l_j \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

After obtaining the total votes received by each class, we select the class with highest vote as the predicted label using (3.16).

$$h(\mathbf{y}', L_{\text{trn}}) = \arg \max_{j=1}^c \sum_{b=1}^B \mathcal{D}_{b,j}, \quad j = 1, 2, \dots, c \quad (3.16)$$

Since there are K sets of L_{trn} , the above training procedures are repeated K

times, and the averaged value of all classification accuracies is used as the new fitness. After pre-defined generations, a set of scalars α_1^{opt} , α_2^{opt} , ..., α_m^{opt} are selected, from which optimal feature vector $\mathbf{y}^{\text{opt}}$ can be extracted. Finally, we apply the genetic algorithms optimized outputs to the original training set L for learning and the testing set for experimental validation.

3.4 Experiments

3.4.1 Database

The proposed multiple Trace feature is evaluated with a combo face data set consisting of 555 instances for training and 575 examples for testing. The data set is a combination of ORL [100], Yale [38], and UMIST [101] databases in which large variations of illuminations and poses are presented.

- The ORL database contains 400 images of 40 individuals under a high degree of variability in expression and pose. In ORL database, each person has ten different images. There are variations such as glasses or no glasses and open or close eyes. All the images are taken under a dark background largely frontal faces with slight movements, and there are also some variations in scale. In the experiments, we select 40 persons each with 5 samples randomly as the training set, and the remaining 200 samples are used to form the testing set.
- The UMIST face database consists of 564 images of 20 people. Each covering a range of poses from profile to frontal views. Subjects cover a range of race, sex, appearance. In the experiments, 280 images are used

for training, and the remaining images are used for testing.

- The Yale database contains 165 frontal face images covering 15 individuals taken under 11 different conditions. Each individual has different facial expressions (happy, sad, winking, sleepy, and surprised), illumination conditions (center-light, left-light, right-light), etc. The training set for the experiments consists of 75 randomly selected images (15 persons with each 5 images), and the remaining 90 images are included into the testing set.

In the experimental implementation, eight sets of Trace transforms are obtained by applying the Trace functionals in Table 3.1, from which the multiple Trace feature can be extracted as a new pattern representation.

3.4.2 Experimental Results

Having obtained the face data set, another essential task is to choose a proper classifier. Among numerous powerful predictors, a widely used instance-based method, k -nearest neighbor algorithm (k -NN) [31], is selected. Because this method simply stores the training examples rather than constructs an explicit description of the target function, the emphasis of evaluation can be put onto the proposed features. Given the training set $L = \{(\mathbf{y}_n, l_n)\}$ where $n = 1, 2, \dots, N$ and a query feature vector $\mathbf{y}'$, k -NN algorithm firstly calculates the standard Euclidean distances between test sample and stored training examples, then categorizes $\mathbf{y}'$ to the class having most instances among the k neighbors. The

distances $d(\mathbf{y}_n, \mathbf{y}')$ are computed as

$$d(\mathbf{y}_n, \mathbf{y}') = \sqrt{\sum_{i=1}^m (\mathbf{y}_n(i) - \mathbf{y}'(i))^2} \quad (3.17)$$

The general classification process using k -NN is given as follows.

- Training phase: k -NN does not construct explicit description of target function but simply stores the training examples $(\mathbf{y}_1, l_1), (\mathbf{y}_2, l_2), \dots, (\mathbf{y}_N, l_N)$.
- Testing phase: Given the query pattern $\mathbf{y}'$, k instances in the training set are selected as nearest neighbors in terms of Euclidean distances. The pattern $\mathbf{y}'$ is assigned to the class having most instances among the k neighbors.

Furthermore, in the evolutionary system, 10-fold cross-validation is adopted for fitness estimation and the values of p_c and p_m are set as 0.65 and 0.004.

In the experiments, the proposed multiple Trace feature is compared with other facial features extracted from PCA and LDA to evaluate its effectiveness. The classification accuracies shown in Table 3.2 reveal that MTF outperforms other features in all dimensions. In this table, MTF_P^c and MTF_L^c use the fitness function calculated from cross-validation, in which P and L represent dimensionality reduction methods PCA and LDA, respectively. MTF_P^b and MTF_L^b implement the Bagging-based fitness function and lead to higher accuracies in most feature dimensions compared with the cross-validation based strategy. Moreover, the experimental results in Table 3.2 are also visualized in Figure 3.4 for clear depiction. It is noted that the improvement is significant when feature dimension is low. With the increase of m (feature dimension), the classification

accuracy will reach to a high level, and thus the room of further improvement is limited. In this study we attempt to demonstrate the effectiveness of using multiple Trace transforms with eight functionals only, therefore it is possible to obtain better generalization performance if more functionals are employed.

Table 3.2: Recognition rates on the combo face database.

m	PCA	LDA	MTF _P ^c	MTF _L ^c	MTF _P ^b	MTF _L ^b
70	0.9148	0.9078	0.9165	0.9270	0.9200	0.9322
65	0.9130	0.9078	0.9217	0.9339	0.9252	0.9252
60	0.9113	0.9078	0.9235	0.9304	0.9165	0.9339
55	0.9009	0.9096	0.9183	0.9304	0.9252	0.9304
50	0.9078	0.9026	0.9252	0.9304	0.9183	0.9443
45	0.8974	0.9113	0.9235	0.9391	0.9322	0.9322
40	0.8974	0.9078	0.9252	0.9304	0.9304	0.9357
35	0.8974	0.9148	0.9252	0.9426	0.9287	0.9443
30	0.9043	0.9183	0.9217	0.9252	0.9217	0.9322
25	0.9009	0.9130	0.9113	0.9270	0.9130	0.9426
20	0.8748	0.8922	0.9200	0.9217	0.9165	0.9339
15	0.8643	0.8957	0.9026	0.9165	0.9078	0.9235
10	0.8157	0.8122	0.8852	0.9026	0.9061	0.9130
05	0.6296	0.6696	0.8087	0.8487	0.8452	0.8539

3.4.3 Computational Complexity

The analysis of the computational complexity for the proposed algorithm is straightforward. The whole procedure can be divided into several components for discussion because the generating of Trace transform and the subsequent feature extraction are independent from the evolutionary process. Assume that n_p , n_ϕ , n_t are the number of samples of parameter p , ϕ , and t , respectively. The complexity of creating Trace transforms is calculated as $O(\frac{N_T C_T n_t n_p n_\phi}{2})$ [14], where C_T is the number of operations per sample for each Trace functional and N_T is the total number of functionals. Next, in the N_T training sets comprising

Figure 3.4: The experimental results on combo database with different dimensionality reduction methods: (a) PCA based methods; (b) LDA based methods.

N instances each, $O(n_\phi N_T N)$ operations are needed for the conversion from 2D Trace transform to a string in the training set, and $O(N^3 N_T)$ computations are required for implementing PCA [31].

The above operations are done prior to the training process. Because k -

NN classifier is implemented for prediction in the evolutionary procedure, it is important to know the computational cost of k -NN which will affect the complexity of training. In general, the computational complexity of k -NN classifier is $O(\text{number of training sample} \times \text{feature dimension})$. Recall the parameters in genetic algorithms, N_p , p_c , p_m are population size, crossover probability and mutation probability. In addition, N_G is defined as the number of generations. The calculation of complexity follows the framework of Salomon's works [102]. The bit coding used in our work means that the scalars $\alpha_1, \alpha_2, \dots, \alpha_m$ are encoded by q bits. Thus, each chromosome is a bit string of length $l = mq$ on which the genetic strategy is applied. As a result, a complexity of $O(l \ln l)$ is derived [102]. As K -fold cross-validation scheme is used to estimate fitness of candidate, approximately $\frac{(K-1)N}{K}$ samples are used to train the k -NN classifier. Consequently, the overall complexity of the evolving process is $O((l \ln l + N_p K \frac{(K-1)N}{K} m) N_G)$ where $O(\frac{(K-1)N}{K} m)$ is the complexity of a single classifier. The computational complexity of the system can be computed as

$$\begin{aligned} \text{Complexity} = O\left(\frac{N_T C_T n_t n_p n_\phi}{2} + n_\phi N_T N + N^3 N_T + \right. \\ \left. N_G l \ln l + N_G N_p (K-1) N m\right) \end{aligned} \quad (3.18)$$

The values of parameters used in this chapter are specific for this application. In the experiments, $N_T = 8$ Trace functionals are applied to $N = 555$ training samples. The parameters C_T , n_t , n_p , n_ϕ are set as 10, 100, 72, and 360, respectively. n_p varies according to different image dimensions. As the genetic algorithms use 10-bit coding for chromosome, q is 10. Other settings can be found in previous sections. The algorithm was implemented in MATLAB and run on a Pentium 4 3.2GHz processor. It took around 20 hours for Trace trans-

form generation and several hours for training to discover the optimal scalars. Testing on a single face image finished within seconds.

In the face recognition system proposed by Srisuk et al. [17], it took about 100 hours for training with Pentium 4 processor. Since their algorithms seek for optimal parameters through reinforcement learning to extract shapes, the training relies on 2D Trace transforms. In our proposal, the Trace transforms are converted to low-dimensional vectors before training, thus the operations on 2D transforms are not involved in the learning process. Therefore, it is reasonable that the computing time is saved when feature vectors with reduced dimension rather than two-dimensional Trace transforms are used for training.

3.5 Summary

In this chapter, various Trace transforms are fused by the genetic algorithms to generate the multiple Trace feature for face recognition. Our contribution is the proposal of the integration of Trace transforms into the framework of evolutionary computation. In addition to cross-validation based fitness function, bootstrap aggregating algorithm is employed to enhance the estimation of fitness such that the overtraining problem might be alleviated and the generalization ability is possible to be enhanced. Comprehensive experimental results have been presented to demonstrate the robustness and discriminatory power of MTF. Ideally, the proposed feature has a wide range of applications for image analysis, not just limited to face recognition.

Chapter 4

Weighted Principal Components Extraction

4.1 Introduction

In many applications of pattern recognition and information retrieval, the system performances depend heavily on particular choice of the features. The process of feature extraction should involve the derivation of the salient features by reducing the redundant information of the data and providing enhanced discriminatory power [66]. Jain et al. [2] pointed out that the probability of misclassification does not increase with the increment of feature dimension as long as the class-conditional densities are completely known. If the dimensionality of the feature vector is much larger than the number of training samples, the system performance may be poor because of the redundancy within feature vectors. Therefore, increasing the ratio of the number of training samples to the feature dimension is suggested such that the degradation on the classification

accuracy can be avoided [5]. Moreover, limited yet salient features can alleviate the burden of classifier design. As a consequence, a number of algorithms [1], [4] were proposed in the past few years for feature selection and extraction so that compact representations can be obtained to reflect the intrinsic characteristics of the original data.

The feature extraction algorithms are broadly categorized into supervised methods and unsupervised methods depending on whether or not the class information is deployed. Among unsupervised approaches, principal component analysis (PCA) [7], [6] is one representative technique and has been widely used. Due to its simplicity in both theory and implementation, PCA is commonly used for face recognition [24], ECG signal processing [25] and many other applications. In image analysis, undesirable artifacts, noise, occlusions may exist, and one solution is to compensate for pixels with weights. As a result, weighted and robust learning [28] and robust PCA [27] were proposed, where the weights are estimated through the optimization algorithms which minimize the reconstruction errors. To fit the data well, several extensions of the conventional PCA such as two-dimensional PCA [29], probabilistic PCA [31], have been developed. In addition, Kumar et al. [33] proposed a technique to alleviate the computational complexity for PCA. So far, the aforementioned PCA-based approaches promise to have good results only if the samples are linearly separable. However, real-world data like face patterns are nonlinearly distributed. Since linear feature extractors are not always suitable for expressing data, nonlinear dimensionality reduction techniques are required to handle the nonlinearity of the data, among which kernel PCA (KPCA) [10] is the most prominent one. By implementing the kernel trick, KPCA maps the data into a high dimensional feature space in which the original samples are possible to be linearly separable so that the

classification performance can be improved. Instead of utilizing the kernel trick, a few other nonlinear versions of PCA were also developed to tackle nonlinear systems [36], [37].

To enhance the classification ability, numerous supervised dimensionality reduction methods have been proposed, which take into account class information for extracting discriminative features [53]. Linear discriminant analysis (LDA) [7], [38] is a typical supervised algorithm, which extracts features by maximizing the between-class scatter matrix and minimizing the within-class scatter matrix. As a nonlinear extension using kernel trick, the kernel Fisher discriminant analysis (KFDA) [11] was proposed to handle nonlinearly distributed data. Nevertheless, although LDA-based methods are able to enhance the class separability to obtain discriminative features, they still suffer from various difficulties like the small sample size (SSS) problem [8], i.e., the LDA technique is not applicable when the number of training samples is smaller than the feature dimension as the within-class scatter matrix is singular. Wang [103] pointed out that the features leading to large class separability do not have to result in a high predictive accuracy. Moreover, the assumption for LDA to find the optimal subspace is that the samples from different classes should be distributed as homoscedastic Gaussian model [104], which cannot be satisfied in most real-world problems because of the lack of samples and/or existence of outliers [40]. As a consequence, Tang et al. [40] introduced the relevance weights to tackle the outliers and improve the overall performance where the weights are determined either by numerical calculation or through evolution strategy.

It is commonly accepted that PCA, LDA, and most of their extensions are carried out as preprocessing step before classification, while pattern recognition

systems seek high predictive accuracy no matter what the features are and in which way they are generated. Since the integration of dimensionality reduction and classification in the training phase is a feasible solution to reduce prediction risk on unseen data, a number of algorithms are proposed, in which evolution strategies are commonly employed as the information fusion tools. Representative works include evolutionary pursuit [66], evolutionary discriminant analysis [105], relevance weighted LDA [40], and GA-Fisher [45]. In addition to the genetic algorithms (GAs) based systems, particle swarm optimization (PSO) was also proposed to be coupled with random subspace method for feature extraction [106]. In the past few years, evolution strategies were discussed extensively for feature selection [107], [74], and classification [108].

In general, by designing the evolutionary procedures carefully, researchers attempt to extract discriminative features to represent the original patterns, and meanwhile intend to look for a trade-off between minimizing the training error and maximizing the testing accuracy. It should be mentioned that some previous works contribute to single feature extraction technique only. Therefore, a lot of room is left for further investigations on the feasibility of fusing two or more algorithms because each feature extractor has its own advantages and disadvantages.

In this chapter, an accuracy-driven algorithm shown in Figure 4.1 is proposed for building two novel feature extractors based on PCA and KPCA, in which a modified Fisher criterion is employed to express the class separability. Within this strategy, class-dependent weights are introduced to represent differences among the samples from various categories, and genetic algorithms are implemented to seek for suitable weights in terms of optimizing the fitness

Figure 4.1: The architecture of the proposed framework.

function. As a result, evolutionary weighted PCA (EWPCA) and evolutionary weighted KPCA (EWKPCA) are proposed as novel supervised techniques for dimensionality reduction.

The proposed architecture shared by both EWPCA and EWKPCA is summarized as follows. Firstly, a population of individuals (candidate solutions) that encode the weights are randomly initialized. Secondly, the features are extracted by projecting the original data onto the the weighted subspaces, and subsequently, are categorized by the classifiers to evaluate the fitness. The third action is to evolve the initial population by means of genetic operators to find the most fit individual. Finally, the weights encoded in the optimal individual are used to extract features for the samples in both training and testing sets. In the proposed algorithm, the fitness function acts dominantly in determining the “optimum”. Unlike most evolution-based pattern recognition systems where the learning accuracy is considered as the sole component of fitness, we create the modified Fisher criterion as an additional term to express the class separation. By utilizing this new fitness, the extracted features could present enhanced discriminatory power in terms of achieving small training error and

large class separability. Therefore, good generalization performance is possible to be obtained when both terms in the fitness are well balanced.

In summary, an advantage of the proposed algorithms is that the class information is incorporated into PCA and KPCA by means of the modified Fisher criterion, so that conventional principal component-based subspace techniques are extended to be supervised. Another virtue is that the modified Fisher criterion will be always available regardless of the singularity of the within-class scatter $\mathbf{S}_W$ as the calculation of the criterion is not based on matrix manipulation. To demonstrate the effectiveness of the proposed feature extractors for classification, EWPCA and EWKPCA are implemented and evaluated in a diverse spectrum of applications such as object recognition, face authentication, and UCI image data sets.

4.2 The Proposed EWPCA Algorithm

As pointed out by Jain et al. [2], in the case that the number of training samples is small relative to the dimensionality of feature vector, the added features will degrade the generalization performance of the classifier. Apparently, dimensionality reduction on the features is essential to recognition. Two commonly used linear feature extractors, PCA and LDA, have been described in Chapter 3.

4.2.1 Weighted Principal Component Analysis

In order to obtain the best possible performance, prior knowledge about a problem ought to be incorporated into the training phase [77]. If there is no additional knowledge at hand, we should rely on the information embedded in the

training samples [27]. As an unsupervised dimensionality reduction algorithm, conventional PCA does not take account of characteristics of training samples. Hence, PCA treats all the samples equally regardless of their diverse contributions to generalization.

PCA is a linear feature extractor whilst it is suspected that most data sets are nonlinearly separable. Apparently, features extracted by PCA may not retain the information of the original samples well. Hiden et al. [37] claimed that a common procedure to overcome the above problem is to “linearize” the data with suitable transformations prior to analysis. In our proposal, weights are simply employed instead of complex nonlinear mapping functions for compensating the original data to facilitate improving classification accuracy. By introducing randomly generated weights that later are adaptively tuned, the proposed weighted PCA (WPCA) is able to incorporate category properties into the subspace from which discriminative features can be extracted. One weighted version of PCA [27] adds weights to each pixel and individual image so that the learned models can fit the training data well. However, in the context of pattern recognition, generalization performance may be poor as subtle changes on weights will affect the outputs dramatically. Moreover, optimizing such huge number of weights is also a challenge. Comprehensive descriptions of the proposed WPCA algorithm are as follows.

Assume that there are N patterns and each has M observations in the feature set shown in Figure 4.2 (face images are used for illustration, and M is the total of number of pixels in images), we have the data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$. It is noted that each pattern $\mathbf{x}_k$ in $\mathbf{X}$ belongs to one of c classes $\{l_1, l_2, \dots, l_c\}$. We define the weight vector as $[\gamma_1, \gamma_2, \dots, \gamma_N]$, and assign each variable to one of c

Figure 4.2: The visual structure of the feature set $\mathbf{X}$, where $\mathbf{x}_k = [\mathbf{x}_k(1), \mathbf{x}_k(2), \dots, \mathbf{x}_k(M)]^T$. In this example, each face image belongs to one of c persons.

values $\{\mu_1, \mu_2, \dots, \mu_c\}$ according to their class properties.

$$\gamma_i = \mu_j \quad \text{if } \mathbf{x}_i \in j\text{th class} \quad (4.1)$$

where $i = 1, \dots, N$ and $j = 1, \dots, c$. The weights are observed to be class-dependent so that they can stand for the dissimilarities among different classes.

By substituting the grand mean $\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ and multiplying γ_i , a weighted data set $\mathbf{X}_W$ is created from the original data set.

$$\mathbf{X}_W = [\gamma_1 \mathbf{x}_1 - \gamma_1 \bar{\mathbf{X}}, \dots, \gamma_N \mathbf{x}_N - \gamma_N \bar{\mathbf{X}}] \quad (4.2)$$

Subsequently, the covariance matrix of the weighted data set is calculated as

follows,

$$\begin{aligned}
\Sigma_{\mathbf{x}_w} &= \frac{1}{N} \mathbf{X}_w \mathbf{X}_w^T \\
&= \frac{1}{N} \sum_{i=1}^N (\gamma_i \mathbf{x}_i - \gamma_i \bar{\mathbf{X}})(\gamma_i \mathbf{x}_i - \gamma_i \bar{\mathbf{X}})^T \\
&= \frac{1}{N} \sum_{i=1}^N \gamma_i^2 (\mathbf{x}_i - \bar{\mathbf{X}})(\mathbf{x}_i - \bar{\mathbf{X}})^T
\end{aligned} \tag{4.3}$$

Define $\delta_i = \gamma_i^2$, the covariance matrix $\Sigma_{\mathbf{x}_w}$ can be alternatively represented as

$$\begin{aligned}
\Sigma_{\mathbf{x}_w} &= \frac{1}{N} \sum_{i=1}^N \delta_i (\mathbf{x}_i - \bar{\mathbf{X}})(\mathbf{x}_i - \bar{\mathbf{X}})^T \\
&= \frac{1}{N} \sum_{i=1}^N f(\delta_i, \Sigma_{\mathbf{x}_i})
\end{aligned} \tag{4.4}$$

where weight δ_i indicates how important the corresponding pattern contributes to the covariance matrix. $f(a, b)$ is a function that describes the relationship between weight δ_i and the covariance matrix of pattern $\mathbf{x}_i$. In this study, $f(a, b)$ is the scalar multiplication operator, that is $f(a, b) = ab$. Therefore, the formulation of the covariance matrix shares similar format with the relevance weighted within-class covariance matrix in [40], and the weights are usually estimated based on relationships among classes. Then decomposition is applied to the covariance matrix to calculate its eigenvalues and corresponding eigenvectors as

$$\Lambda \mathbf{U}_w = \Sigma_{\mathbf{x}_w} \mathbf{U}_w \tag{4.5}$$

where $\mathbf{U}_w = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]$ are the orthonormal eigenvectors and Λ is the diagonal matrix containing the eigenvalues of the covariance matrix $\Sigma_{\mathbf{x}_w}$. The vectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m$ corresponding to the m th largest eigenvalues, form a

weighted feature space $\tilde{\mathbf{U}}_W = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$, where $m \leq N$. By projecting pattern $\mathbf{x}_k$ onto the weighted feature space $\tilde{\mathbf{U}}_W$, the m -dimensional feature vector can be computed as

$$\mathbf{y}_k = \tilde{\mathbf{U}}_W^T (\mathbf{x}_k - \bar{\mathbf{X}}) \quad (4.6)$$

A feature vector calculated using (4.6) is used to represent the original sample, in which $\mathbf{y}_k$ is able to capture the most dominant features of the original data.

The reconstruction of the variable in data set $\mathbf{X}$ is given straightforwardly as $\hat{\mathbf{x}}_k = \tilde{\mathbf{U}}_W \mathbf{y}_k + \bar{\mathbf{X}}$. Then the reconstruction error ε on the training data is calculated as $\sum_{k=1}^N \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2$. From the statistical point of view, small errors are expected so that patterns in reduced dimensionality can be reconstructed with minimal loss. In a practical recognition system, small reconstruction error has not shown to be closely correlated with high recognition accuracy, that is less significant features may appear dominant in discriminating among different classes. As a consequence, minimizing the reconstruction error is not necessary to be a target of WPCA for extracting expressive features. Therefore, the determination of weights can be considered as an optimization problem where recognition rate is anticipated to be maximized. Moreover, the modified Fisher criterion with respect to the class information, is embedded into the process of feature extraction to make WPCA supervised so that high predictive accuracy can be obtained. From (4.1), it is known that the number of weights is determined by c which is unlikely to be a small number in many real-world applications. Consequently, traditional numerical optimization methods like gradient descent strategy may not approximate the target function well as there are too many variables to be optimized.

Figure 4.3: The mechanism of chromosome coding in genetic algorithms.

4.2.2 Optimal Weights Selection with Genetic Algorithms

In the community of evolutionary computation, genetic algorithms [99] are a class of optimization procedures inspired by the biological mechanisms of reproduction. In our work, GAs are used as the search algorithm to discover the most fitting weights for WPCA, where the optimality is defined with respect to classification accuracy and the class separability. Given appropriate weights, it is possible to enhance both the training performance and the generalization ability. In the rest of this section, we describe the representation of chromosome, and subsequently introduce the customized fitness function. Finally, the entire evolving process is illustrated with real-world data where three basic genetic operators (selection, crossover, and mutation) guide the optimization.

We employ a simple encoding scheme where the chromosome is defined as a set of weights $\mu_1, \mu_2, \dots, \mu_c$. In the chromosome, 10 bits (resolution) are used to represent each variable. Each chromosome represents a combination of the weights, from which $\mathbf{y}_k$ in (4.6) are calculated. Figure 4.3 depicts the components of each individual (chromosome) in the population.

In order to initialize the population we need: (a) The number of bits in a solution candidate, which is depending on the required precision; (b) total number of solution candidates (population size). In general, the initial population is generated randomly. Because the population size influences the computation

cost and the diversity of individuals, it is usually heuristically determined to give a trade-off between efficiency and variety.

The chromosome selection for the next generation is done on the basis of fitness. In most evolution-based feature extraction methods, the training accuracy is employed as the fitness measure [66], [40], [74]. Taking advantage of Liu and Wechsler's work [66], the fitness function $\zeta(\mathcal{F})$ consisting of two terms is defined as

$$\begin{aligned}\zeta(\mathcal{F}) &= \zeta_{ca}(\mathcal{F}) + \kappa\zeta_d(\mathcal{F}) \\ &= CA + \kappa R_{BW}\end{aligned}\tag{4.7}$$

where $\zeta_{ca}(\mathcal{F})$ is the performance accuracy term, and $\zeta_d(\mathcal{F})$ is the discrimination term. In the fitness function, CA represents how good the extracted features are in terms of classification accuracy, and R_{BW} is the modified Fisher criterion used for indicating the class separability. Since better classification requires large class separability but the latter does not necessarily lead to the first, a positive constant κ is introduced to determine the importance of the second term relative to the first one in the fitness function. Maximizing the fitness function will lead to high classification accuracy and large class separability. By combining those two terms with a proper κ , GAs can discover more discriminative principal components than standard PCA. Therefore, good generalization performance might be obtained on testing data sets. In pattern recognition, because the classification accuracy is in a dominant position for evaluating the effectiveness of the feature extraction algorithm [66], κ is empirically chosen such that CA can contribute more to the fitness than the discriminant term does.

To determine the performance accuracy term $\zeta_{ca}(\mathcal{F})$ of the fitness, the over-

all accuracy in training phase could be used. In early studies of evolutionary dimensionality reduction, the accuracy on testing set was used as the fitness function [109]. However, as emphasized by Sun et al. [107], the enrollment of the test set is not appropriate in determining fitness because bias can be introduced to the classification phase. In many recently developed evolution-based pattern recognition systems, a separate validation set is used to overcome the overtraining problem [74], [86], [107]. Generally speaking, the data set is partitioned into three disjoint sets: training, validation (tuning), and testing. During the learning procedure, the training and validation sets are used to train classifiers and provide feedbacks to the evolutionary algorithms, respectively. Then the testing set is employed to evaluate the effectiveness of learning with best tuned parameters determined through validation. Although an extra validation set can prohibit overfitting, in many real-world applications, training samples are not enough to form a validation set. In such scenario, it is not a wise choice for sacrificing training data to create validation set as the decrease of learning data may substantially degrade the performance in generalization. Therefore, we implement the cross-validation scheme to estimate CA for the fitness. By doing so, the accuracy and stability of testing are likely to be improved, and the overfitting problem may be avoided due to the reduced variance. In detail, let L denote the training set, in calculating CA using K -fold cross-validation, approximately $\frac{(K-1)N}{K}$ instances composing $L_{\text{trn}} = \{(\mathbf{y}_n, l_n)\}$, $n = 1, 2, \dots, \frac{(K-1)N}{K}$ with labels $l_n \in \{l_1, l_2, \dots, l_c\}$, are used for training and the remaining $\frac{N}{K}$ samples constitute the corresponding L_{val} for validation. $h(\mathbf{y}, L_{\text{trn}})$ is defined as the predictor (classifier) across training, validation, and testing. Since there are K sets of L_{trn} , the training procedures are repeated K times, and the mean value of all validation accuracies calculated as $\text{CA} = \frac{1}{K} \sum_{k=1}^K \text{CA}_k$ is used for fitness.

Because there are no overlaps between each pair of L_{trn} and L_{val} , high classification accuracy can be achieved in the evolving stage and good performance is probably observed in testing.

The second component of the fitness function (4.7) is the discrimination term $\zeta_d(\mathcal{F})$. Since PCA treats the between-class and within-class variations equally, the Fisher criterion [38] may be a choice as the class separability in the fitness function. Nevertheless, $\mathbf{S}_W$ can be singular due to the small sample size (SSS) problem [110], that is, the data are from a high-dimensional space, and the dimensionality exceeds the sample size in general. Thus a distance-based modified Fisher criterion R_{BW} is proposed to indicate the class separability. We define the between-class difference as D_B and within-class difference as D_W . As described in the definition of the fitness function, in order to improve the generalization ability, R_{BW} should be maximized during the evolving process. In this new criterion, rather than using scatter matrices like $\mathbf{S}_B$ and $\mathbf{S}_W$ in standard LDA, D_B and D_W are calculated to present the class separation such that decomposition of matrix is not necessary and thus the singularity of $\mathbf{S}_W$ will be no longer a issue. To provide a clear description in calculating R_{BW} , $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ is employed to represent the weighted data set with $\mathbf{z}_k = \gamma_k \mathbf{x}_k - \gamma_k \bar{\mathbf{X}}$ where $\mathbf{z}_k$ belongs to one of c classes $\{l_1, l_2, \dots, l_c\}$. Suppose that the differences between any two feature vectors $\mathbf{h}_i$ and $\mathbf{h}_j$ are described as the cost of matching given by the χ^2 test statistic [17], [7], and $t \in [1, M]$ is the dimensionality of the feature vector,

$$\chi^2(\mathbf{h}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{t=1}^M \frac{[\mathbf{h}_i(t) - \mathbf{h}_j(t)]^2}{\mathbf{h}_i(t) + \mathbf{h}_j(t)} \quad (4.8)$$

The between-class difference can be calculated as

$$D_B = \sum_{i=1}^c N^i |\chi^2(\bar{\mathbf{Z}}^i, \bar{\mathbf{Z}})| \quad (4.9)$$

and the within-class variation is

$$D_W = \sum_{i=1}^c \sum_{\mathbf{z}_k \in l_i} |\chi^2(\mathbf{z}_k, \bar{\mathbf{Z}}^i)| \quad (4.10)$$

where $\bar{\mathbf{Z}}^i$ is the mean of weighted samples of class Φ_i , and $\bar{\mathbf{Z}}$ is the mean of the entire data set. N^i is defined as the number of class l_i . In order to increase the discrimination of the weighted data set, we want to maximize the modified Fisher criterion R_{BW} which is defined as:

$$R_{BW} = \frac{D_B}{D_W} = \frac{\sum_{i=1}^c N^i |\chi^2(\bar{\mathbf{Z}}^i, \bar{\mathbf{Z}})|}{\sum_{i=1}^c \sum_{\mathbf{z}_k \in l_i} |\chi^2(\mathbf{z}_k, \bar{\mathbf{Z}}^i)|} \quad (4.11)$$

From (4.6), we notice that the dimensionality of the original data is reduced by discarding those principal components which do not contribute significantly to the overall variation. However, the information used for distinguishing among different patterns will be lost. Next, we will show how to extract statistically dominant and discriminative features through the genetic algorithms-based evolutionary process.

Three genetic operators, selection, crossover, and mutation, are used to guide the search to pick out the most fitting individual. After initializing the population, these genetic operators are applied repeatedly, from which the elite chromosome is obtained once the termination criterion is met. In this work, the criterion to stop the evolution is that a pre-defined number of generations N_G has been completed. The individual having the maximum fitness in the

Evolutionary Process for Optimal Weights Selection

Parameters of genetic algorithms:

- $\mathbf{P}(k)$: Population of N_p individuals at generation k , where individual is represented as the weight vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_c]$.
- N_p : Population size.
- p_c : Crossover probability.
- p_m : Mutation probability.

Evolving procedure:

1. Set $k = 0$ and initialize the population $\mathbf{P}(k)$ at random. Each chromosome is a sequence of binary codes representing weights $\mu_1, \mu_2, \dots, \mu_c$. The population size N_p should be set to give a trade-off between the computational cost and the diversity of the population.
 2. Evaluate the fitness function $\zeta(\mathcal{F})$ defined in (4.7) for all individuals in population $\mathbf{P}(k)$.
 3. If the termination criteria is met, the evolving procedure should be stopped and switched to 6). Otherwise, do steps 4) and 5).
 4. Increase k with 1. Selection, crossover and mutation on $\mathbf{P}(k)$ are then implemented to create new generation $\mathbf{P}(k + 1)$.
 - Firstly, a mating pool $\tilde{\mathbf{P}}(k)$ of size N_p is generated from population $\mathbf{P}(k)$ via stochastic universal selection.
 - Then, $\frac{p_c \cdot N_p}{2}$ pairs of members in the mating pool are randomly selected, on which crossover operator is applied.
 - Finally, $p_m \cdot N_p$ candidates after crossover are subject to mutation and the outputs form the new generation population $\mathbf{P}(k + 1)$.
 5. Go back to step 2) and re-evaluate the fitness.
 6. Give the outputs such as the best individual (chromosome) and its fitness value. Then the optimal weights are obtained, based on which the features can be extracted using WPCA algorithm.
-

Figure 4.4: Evolutionary procedure for optimal weights selection.

population of the last generation provides the optimal weights for WPCA. The evolving process is illustrated in Figure 4.4.

According to the results of fitness evaluation, the reproduction process is implemented if the termination criterion is not achieved. The selection operator chooses the chromosomes from the current generation into a mating pool by taking into consideration the fitness values of the individuals. The chromosomes with higher fitness values will likely appear more times in the mating pool. As one of popular selection techniques, the roulette wheel is widely used. However, it may not yield the expected performance if the population size is small. Therefore, the stochastic universal sampling [99] method is adopted.

Based on the probabilistic selection of individuals, the crossover that exchanges information between two parent chromosomes for generating two offsprings is carried out on the mating pool. There are three basic types of crossovers: one point crossover, two point crossover, and uniform crossover. Differing from the other two schemes, in uniform crossover, each gene of the offspring is selected randomly from the corresponding genes of the parents. In the proposed WPCA algorithm, because how the weights correlate to each other is unknown, we choose uniform crossover to produce offsprings for the next generation.

Subsequently, the mutation operator is applied to the offsprings with the mutation probability p_m . The mutation on the chromosomes will introduce a degree of diversity to the population to prevent a premature convergence and help to sample unexplored regions of the search space. As mutation usually plays a minor role in the evolutionary procedure, p_m is chosen as a small value.

After pre-defined generations, a set of weights $\mu_1^{\text{opt}}, \mu_2^{\text{opt}}, \dots, \mu_c^{\text{opt}}$ are selected,

from which the weighted feature space is constructed. Then we project the samples in both training and testing sets onto the GAs-optimized subspace to extract features.

To show the effectiveness of the EWPCA algorithm, a subset of UCI optical recognition of handwritten digits set (Optdigits) is used for illustration. The data set contains randomly selected 500 training samples and 896 testing patterns of digit 1, 3, 5, 7, 8. EWPCA starts its evolution process with an initialized population of 50 individuals. Guided by the fitness function, GAs finish their searches for optimal weights after 50 generations. Then the weights encoded in the best chromosome are used to construct a new subspace. It is presented in the results that the testing accuracies and their corresponding R_{BW} with features obtained by PCA and EWPCA are 82.14%, 1.0549, and 87.17%, 1.7946, respectively. Apparently, EWPCA provides more discriminative features than PCA in terms of achieving higher predictive accuracy. Moreover, the class separability is maximized through the evolving process. Figure 4.5 depicts the two-dimensional mappings of 500 training samples obtained by implementing PCA and EWPCA. It can be observed in Figure 4.5(a) that the projection by PCA cannot clearly distinguish digit 1 and digit 7, whereas the clusters of the patterns in Figure 4.5(b) are well separated with the help of EWPCA. The results and figures reveal that although the introduced weights are not able to create a compact within-class scatter, they no doubt enhance the class separability and present good performance in both training and testing.

Figure 4.5: Two-dimensional mappings of a subset of UCI Optdigits data set. (a) PCA and (b) EWPCA.

4.3 The Proposed EWKPCA Algorithm

The EWPCA algorithm takes advantage of linear feature extraction algorithms PCA and LDA whilst most data sets to be tackled in pattern recognition are nonlinear. Therefore, we expect to extend the EWPCA from linear to nonlinear by means of the kernel trick [9]. In this section, the weighted kernel PCA (WKPCA) is proposed. The key issue to be addressed in WKPCA is where and how the weights are incorporated. Next, we describe the basic idea of nonlinear mapping, and subsequently elaborate in the proposed WKPCA algorithm. GAS are again employed as the optimization tool for weights selection.

4.3.1 Data Manipulation in High Dimensional Space

First of all, we investigate the procedure of KPCA for nonlinear mapping. Let us consider a feature space $\mathbf{F}$ related to the input domain by a map

$$\Phi : \mathbf{X} \rightarrow \mathbf{F}, \mathbf{x} \mapsto \Phi(\mathbf{x}) \quad (4.12)$$

In this chapter, RBF (Gaussian) kernel function is employed for computation, and it is formulated as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (4.13)$$

Assuming that the input data are zero centered, i.e., $\sum_{i=1}^N \Phi(\mathbf{x}_i) = 0$, the covariance matrix has the same formulation as PCA

$$\Sigma^\Phi = \frac{1}{N} \sum_{j=1}^N \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)^\top \quad (4.14)$$

The next action is to find eigenvalues $\Lambda \geq 0$ and nonzero eigenvectors $\mathbf{U}$ so that $\Lambda \mathbf{U} = \Sigma^\Phi \mathbf{U}$ is satisfied. Because all solutions $\mathbf{U}$ with $\lambda \neq 0$ lie in the span of $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$, we can instead consider that

$$\Lambda \langle \Phi(\mathbf{x}_n), \mathbf{U} \rangle = \langle \Phi(\mathbf{x}_n), \Sigma^\Phi \mathbf{U} \rangle \quad (4.15)$$

where $n = 1, \dots, N$ and there exist coefficients α_i such that $\mathbf{U} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i)$. Consequently, we have

$$\begin{aligned} & \Lambda \sum_{i=1}^N \alpha_i \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_i) \rangle \\ &= \frac{1}{N} \sum_{i=1}^N \alpha_i \left\langle \Phi(\mathbf{x}_n), \sum_{j=1}^N \Phi(\mathbf{x}_j) \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i) \rangle \right\rangle \end{aligned} \quad (4.16)$$

By defining an $N \times N$ Gram matrix as

$$K_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \quad (4.17)$$

we obtain

$$N \Lambda \mathbf{K} \boldsymbol{\alpha} = \mathbf{K}^2 \boldsymbol{\alpha} \quad (4.18)$$

where $\boldsymbol{\alpha}$ is a column vector with entries $\alpha_1, \dots, \alpha_N$. As $\mathbf{K}$ is nonsingular, (4.18) can be simply represented as

$$N \Lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha} \quad (4.19)$$

4.3.2 Weighted Kernel Construction

Since the patterns from the same class are quite different from the patterns in other categories, it becomes promising to enhance the discriminatory power for the extracted features if the class information can be incorporated into the kernel matrix. Recall our motivation, although LDA introduces the between-class and within-class scatters to show variations among different classes, it may still encounter difficulties such as linearity and the small sample size problem. In previous section, the EWPCA method is introduced, which has the capability of utilizing proper weights to integrate the class information into PCA with an aim on yielding good generalization performance.

The rationale behind WPCA is to compensate each pattern by incorporating the weights into the covariance matrix. Is it possible to use a similar method to extend KPCA to a weighted formulation? To answer this question, we have following theorem.

Theorem 4.1. *If function $f(a, b)$ in (4.4) is used to formulate the weighted KPCA, the resultant kernel matrix is not a Hermitian matrix.*

Proof. By incorporating the same weights used by WPCA to the covariance matrix of KPCA, we have

$$\Sigma_W^\Phi = \frac{1}{N} \sum_{j=1}^N \gamma_j \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)^T \quad (4.20)$$

Then (4.16) has following formulation

$$\begin{aligned}
 & N\Lambda \sum_{i=1}^N \alpha_i \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_i) \rangle \\
 &= \sum_{i=1}^N \alpha_i \left\langle \Phi(\mathbf{x}_n), \gamma_j \Phi(\mathbf{x}_j) \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i) \rangle \right\rangle
 \end{aligned} \tag{4.21}$$

The right side can be read as (4.22). After expansion, (4.22) becomes (4.23).

According to the definition of kernel matrix, (4.21) is simplified to $\mathbf{K}\mathbf{K}'\boldsymbol{\alpha}$ where $\mathbf{K}'$ shown as (4.24) is a variation of conventional kernel matrix.

$$\left(\begin{array}{ccc} \vdots & \dots & \vdots \\ \left\langle \Phi(\mathbf{x}_n), \sum_{j=1}^N \gamma_j \Phi(\mathbf{x}_j) \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_1) \rangle \right\rangle & \dots & \left\langle \Phi(\mathbf{x}_n), \sum_{j=1}^N \gamma_j \Phi(\mathbf{x}_j) \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_N) \rangle \right\rangle \\ \vdots & \dots & \vdots \end{array} \right) \boldsymbol{\alpha} \tag{4.22}$$

$$\left(\begin{array}{ccc} \vdots & \dots & \vdots \\ \left\{ \begin{array}{l} \gamma_1 \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_1) \rangle \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_1) \rangle + \\ \gamma_2 \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_2) \rangle \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_1) \rangle + \\ \dots + \\ \gamma_N \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_N) \rangle \langle \Phi(\mathbf{x}_N), \Phi(\mathbf{x}_1) \rangle \end{array} \right\} & \dots & \left\{ \begin{array}{l} \gamma_1 \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_1) \rangle \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_N) \rangle + \\ \gamma_2 \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_2) \rangle \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_N) \rangle + \\ \dots + \\ \gamma_N \langle \Phi(\mathbf{x}_n), \Phi(\mathbf{x}_N) \rangle \langle \Phi(\mathbf{x}_N), \Phi(\mathbf{x}_N) \rangle \end{array} \right\} \\ \vdots & \dots & \vdots \end{array} \right) \boldsymbol{\alpha} \tag{4.23}$$

$$\mathbf{K}' = \left(\begin{array}{cccc} \gamma_1 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_1) \rangle & \gamma_1 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle & \dots & \gamma_1 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_N) \rangle \\ \gamma_2 \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_1) \rangle & \gamma_2 \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_2) \rangle & \dots & \gamma_2 \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_N) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_N \langle \Phi(\mathbf{x}_N), \Phi(\mathbf{x}_1) \rangle & \gamma_N \langle \Phi(\mathbf{x}_N), \Phi(\mathbf{x}_2) \rangle & \dots & \gamma_N \langle \Phi(\mathbf{x}_N), \Phi(\mathbf{x}_N) \rangle \end{array} \right) \tag{4.24}$$

Then we can obtain (4.25).

$$N\Lambda\boldsymbol{\alpha} = \mathbf{K}'\boldsymbol{\alpha} \tag{4.25}$$

In evaluating the kernel matrix, each entry K'_{ij} reflects the relationship between pattern $\mathbf{x}_i$ and $\mathbf{x}_j$, and the weight of K'_{ij} in $\mathbf{K}'$ is γ_i . However, because $\mathbf{K}'$ is not a Hermitian matrix (i.e., not symmetric), the decomposition of $\mathbf{K}'$ will not give the exact solutions to (4.25). Therefore, it is important to find an approach to incorporate the weights so as to keep the kernel matrix positive semidefinite [10]. $\square$

To unify the formulations of WPCA and WKPCA, we intend to present the weighted kernel matrix as $K'_{ij} = f^\Phi(\beta_{ij}, K_{ij})$ using a customized function f^Φ where β_{ij} is one of $N \times N$ weights. In addition, like δ_i in (4.4), β_{ij} should be derived exclusively from the weights γ_i and γ_j .

A trial is using the compensated data $\mathbf{X}_W$ to infer the weighted kernel matrix which will appear as

$$K'_{ij} = \langle \Phi(\gamma_i \mathbf{x}_i), \Phi(\gamma_j \mathbf{x}_j) \rangle \quad (4.26)$$

As the Gaussian kernel is used, K'_{ij} becomes

$$K'_{ij} = \exp \left(- \frac{\|\gamma_i \mathbf{x}_i - \gamma_j \mathbf{x}_j\|^2}{2\sigma^2} \right) \quad (4.27)$$

Obviously, it is not possible to find an expression of β_{ij} with γ_i and γ_j merely.

Therefore, we have to construct a weight matrix and implement it to build the weighted kernel matrix $\mathbf{K}'$ using a simple function $f^\Phi(a, b)$ that associates β_{ij} with γ_i and γ_j . As a result, the weight matrix $\mathbf{W}$ is designed to have the

same dimension as the kernel matrix,

$$\mathbf{W} = \begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1N} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N1} & \beta_{N2} & \dots & \beta_{NN} \end{pmatrix} \quad (4.28)$$

Then $K'_{ij} = K_{ij}^{\beta_{ij}}$ is obtained by easily implementing a function $f^\Phi(a_{ij}, b_{ij}) = b_{ij}^{a_{ij}}$, and subsequently a new weighted kernel matrix $\mathbf{K}'$ is derived from the original $\mathbf{K}$,

$$\mathbf{K}' = \begin{pmatrix} K_{11}^{\beta_{11}} & K_{12}^{\beta_{12}} & \dots & K_{1N}^{\beta_{1N}} \\ K_{21}^{\beta_{21}} & K_{22}^{\beta_{22}} & \dots & K_{2N}^{\beta_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ K_{N1}^{\beta_{N1}} & K_{N2}^{\beta_{N2}} & \dots & K_{NN}^{\beta_{NN}} \end{pmatrix} \quad (4.29)$$

where $K'_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle^{\beta_{ij}}$. In the task of feature extraction for pattern recognition, the major objective is to construct an optimal subspace for extracting discriminative features from which enhanced generalization performance can be yielded. By selecting appropriate weights β_{ij} , discriminative features are likely to be extracted to attain high classification accuracy.

Having $\mathbf{K}'$, eigenvalues $\lambda_1, \dots, \lambda_N$ of the weighted kernel matrix in decreasing order are calculated and $\boldsymbol{\alpha}^1, \dots, \boldsymbol{\alpha}^N$ are the corresponding eigenvectors. The subspace is defined as $\mathbf{U}^n$, where $n = 1, \dots, m$, and λ_m is the last nonzero eigenvalue. We normalize the corresponding eigenvectors $\boldsymbol{\alpha}$ and have $\lambda_n \langle \boldsymbol{\alpha}^n, \boldsymbol{\alpha}^n \rangle = 1$. By constructing the nonlinear subspace, we can project both training and testing data onto it to extract the features. Given a pattern $\mathbf{x}_k$, we calculate its feature

vector of reduced dimension by projecting $\mathbf{x}_k$ onto the eigenvectors $\mathbf{U}^n$:

$$\begin{aligned} \mathbf{y}_k(n) &= \langle \mathbf{U}^n, \Phi(\mathbf{x}_k) \rangle \\ &= \sum_{i=1}^N \alpha_i^n \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_k) \rangle \end{aligned} \quad (4.30)$$

In other words, we can extract the first m ($1 \leq m \leq N$) nonlinear principal components which carry more variance than any other m orthogonal directions by using the kernel function.

We have so far made the assumption that the data are centered. However, it is difficult to center the input data in the feature space $\mathbf{F}$. Schölkopf and Smola [77] proposed an approach to solve this problem by calculating $\tilde{\mathbf{K}}$ from $\mathbf{K}'$.

$$\tilde{K}_{ij} = (\mathbf{K}' - \mathbf{1}_N \mathbf{K}' - \mathbf{K}' \mathbf{1}_N + \mathbf{1}_N \mathbf{K}' \mathbf{1}_N)_{ij} \quad (4.31)$$

where $(\mathbf{1}_N)_{ij} := \frac{1}{N}$ for all i, j . Then $\tilde{\mathbf{K}}$ is used instead of $\mathbf{K}'$ for building the weighted nonlinear subspace.

4.3.3 Estimation of Weights

Our idea in calculating β_{ij} is introduced as follows. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same class, they should be kept as close as possible. Thus we define the weight β_{ij} as 1 such that the relationship between these two patterns can be retained in WKPCA. Otherwise, β_{ij} should be variable so as to present the dissimilarity between two different classes.

Then, the calculation of $\mathbf{W}$ is discussed. As described in [47], the weights should represent the information relative to their corresponding patterns. For

instance, in WPCA, the weight δ_i shows the dominance of pattern $\mathbf{x}_i$ among the entire training set; in WKPCA, the weight β_{ij} should reflect the relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. Referring to the example shown in Figure 4.2, we notice that there are c classes. It is easy for human to recognize different persons, but is hard for the computer to distinguish. Different from the supervised methods like LDA, both PCA and KPCA are not able to incorporate the class information into the subspaces. Therefore, the weights consisting of the class information are proposed to be embedded into KPCA. Suppose that there are N weights $\gamma_1, \dots, \gamma_N$ corresponding to N patterns and each of them has been assigned to one of c values according to its class property, we can construct $\mathbf{W}$ as another $N \times N$ matrix

$$\mathbf{W} = \begin{pmatrix} \langle \Phi(\gamma_1), \Phi(\gamma_1) \rangle & \dots & \langle \Phi(\gamma_1), \Phi(\gamma_N) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(\gamma_N), \Phi(\gamma_1) \rangle & \dots & \langle \Phi(\gamma_N), \Phi(\gamma_N) \rangle \end{pmatrix} \quad (4.32)$$

so that $\beta_{ij} = \langle \Phi(\gamma_i), \Phi(\gamma_j) \rangle$. Because Gaussian kernel is used, the weighted kernel matrix $\mathbf{K}'$ has the following formulation

$$\begin{aligned} K'_{ij} &= \exp \left(- \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)^{\beta_{ij}} \\ &= \exp \left(- \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2 \exp(-\frac{\|\gamma_i - \gamma_j\|^2}{2\sigma^2})}{2\sigma^2} \right) \end{aligned} \quad (4.33)$$

From (4.33), we can analyze the effects of the weights under two conditions:

- If $\gamma_i = \gamma_j$, the patterns i and j belong to the same class so that β_{ij} will be equal to 1. It is observed in (4.33) that the weighted kernel matrix will remain the same as conventional kernel matrix, i.e., $K'_{ij} = K_{ij}$.

- If $\gamma_i \neq \gamma_j$, β_{ij} will be a positive value, in which the correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$ from different categories is embedded.

The relevance among different patterns will result in diversified generalization performances. Therefore, the estimation of weights γ is crucial to the recognition system. GAs are again employed to select the weights iteratively under the driven of the same fitness function (4.7) used by EWPCA.

4.4 Experiments

In this section, we evaluate EWPCA and EWKPCA on a large variety of applications for image analysis, and compare them with several feature extractors such as PCA, KPCA, and LDA etc to show their effectiveness in extracting discriminative features. We firstly show the results on face recognition [111], a challenging problem in which feature dimensions are extremely high. Moreover, three UCI image data sets [112] are used to exploit the appropriateness of the proposed weighted feature extractors on a broad spectrum of problems. Finally, our methods are extensively tested in accomplishing the task of object categorization on the Columbia Object Image Library (COIL-20) [113].

Within the problems related to pattern recognition, in conjunction with feature extraction methods that find low-dimensional representation for data, classifiers should be implemented for categorizing patterns. Numerous classifiers, such as Bayesian classifier [7], k -nearest neighbor algorithm (k -NN) [73], support vector machines (SVMs) [77], neural networks (NN), and recently introduced extreme learning machine (ELM) [18], have successfully demonstrated their merits in predicting properties of unseen data. Putting the classifica-

tion task in the framework of artificial evolution, k -NN is well investigated by Raymer et al. [74] and has shown its appropriateness to be used in evolution-based pattern recognition system. Because of k -NN's easy implementation and nonparametric nature (assumptions on the distributions of features are not required [74]), we combine the proposed weighted principal components extractors with k -NN to constitute a complete recognition system.

It is noted that some parameters should be specified for the experimental setup. In the experiments, both population size N_p and generation number N_G are set at 50 for GAs. It seems that N_p and N_G are probably insufficiently large for discovering the optimal solutions from traditional viewpoint in the evolutionary computation community. However, in pattern recognition, mature convergence may result in overfitting for training instances and subsequently degrade the generalization ability in testing. Other parameters in GAs such as p_c and p_m are heuristically set to 0.65 and 0.004, respectively. Furthermore, in determining the fitness values for chromosomes, five-fold cross-validation scheme is applied so as to give a trade-off between computational time and learning effects. Practically, different combinations of parameters may produce diversified outputs. We keep aforementioned variables with constant values for all testing cases, just for providing fair comparisons in evaluating algorithms, even though better results can be yielded by adjusting the parameters.

4.4.1 Face Recognition

Machine recognition of faces is emerging as an active research area spanning several disciplines such as image processing, pattern recognition, and computer vision. Face recognition from still images of a scene is a problem that, identify

one or more persons in the scene using a stored database of faces. The solution to the problem involves segmentation of faces (face detection) from cluttered scenes, feature extraction from the face region, and recognition. EWPCA and EWKPCA are implemented to extract facial features and the recognition results are compared with those of PCA, LDA, and KPCA. In face recognition, the number of training samples N usually need not be smaller than the number of pixels in each image (image dimension M). Hence the within-class scatter matrix $\mathbf{S}_W$ is possible to be singular when LDA is applied. In such a case, we are not able to calculate the optimal subspace directly. As an alternation, the fisherface [38] can be used, in which PCA is firstly deployed to reduce the dimension of the feature space to not more than $N - c$, then LDA is applied to further reduce the feature dimension to $c - 1$. Thus, the feature vector $\mathbf{y}_k$ for any query variable $\mathbf{x}_k$ can be calculated as $\mathbf{y}_k = \mathbf{\Psi}_{\text{opt}}^T \tilde{\mathbf{U}}^T \mathbf{x}_k$. Another problem should be addressed is the high computing complexity in implementing the proposed methods as decomposition of the covariance matrix involved in the evolutionary process. In face recognition, the dimensionality of feature vector is usually very high. For example, an image of 112 pixels by 92 pixels has 10304 (112×92) attributes. Therefore, when PCA-based methods are applied, a 10304×10304 covariance matrix is generated for finding optimal basis representations. Our solution is to apply the discrete cosine transform (DCT) to convert 2D face image to a low-dimensional vector of DCT coefficients, to which the feature extractors can be implemented. This approach is practical as DCT is a data independent technique, and has been widely used in face recognition [70]. In addition, the framework of combining DCT with feature extractors is commonly accepted [12], [114].

In assessing the performances of our proposals, three sets of face images are

Figure 4.6: Four face databases used in the experiments: (a) GTFD, (b) ORL, (c) UMIST, and (d) Yale.

employed. They are ORL database [100], a combo face data set (See Chapter 3), and Georgia Tech face database (GTFD) [115]. Since the combo data set encompasses ORL, UMIST [101], and Yale database [38], there are four stand-alone face image sets. The detailed descriptions of ORL, Yale, and UMIST databases can be found in Chapter 3. In summary, the combo set consists of 555 training samples and 575 testing images in total, and all the images belong to 75 different classes with large variations of illumination, poses, and facial expressions. In the Georgia Tech face database, each of 50 subjects has 15 images. All the color images with cluttered background are taken at resolution 640×480 pixels where frontal and/or tilted faces with different facial expressions, lighting conditions and scale are presented. In the experiments, a pre-processed set of images with the background removed is adopted, and for each subject, eight samples are randomly selected for training and the rest of seven images are used for validation. Before the experimental evaluations, images in Yale and

GTFD databases are manually cropped and resized to 112×92 to make their dimensions identical to those of samples in ORL and UMIST. Additional pre-processing on GTFD includes converting the cropped images to grey scale level. Figure 4.6 presents examples after pre-processing from the above mentioned face databases.

Table 4.1: Recognition rate on face databases.

Database	m	PCA	KPCA	LDA	EWPCA	EWKPCA
Combo	70	91.48	91.65	90.78	92.17	92.70
	65	91.30	91.65	90.78	91.83	92.00
	60	91.13	91.30	90.78	92.52	91.65
	55	90.09	90.09	90.96	91.13	92.18
	50	90.78	90.61	90.26	91.83	91.13
	45	89.74	89.91	91.13	90.96	91.13
	40	89.74	90.09	90.78	91.30	90.96
	35	89.74	89.91	91.48	92.18	90.78
	30	90.43	90.09	91.83	92.00	91.83
	25	90.09	90.26	91.30	90.96	90.78
	20	87.48	87.48	89.22	90.78	90.09
	15	86.43	86.43	89.57	87.83	89.21
	10	81.57	81.57	81.22	85.91	85.91
	5	62.96	63.13	66.96	72.70	73.22
ORL	35	88.50	90.00	88.00	92.00	93.00
	30	87.00	88.00	87.00	92.00	91.50
	25	88.00	88.00	87.50	91.00	91.00
	20	86.50	87.00	86.50	88.00	89.00
	15	84.00	83.50	84.50	88.00	86.50
	10	82.50	82.50	84.00	85.00	85.00
	5	70.50	70.50	75.00	71.50	72.00
GTFD	45	60.29	64.29	61.43	62.86	70.29
	40	61.71	65.71	60.86	63.43	70.86
	35	63.43	67.43	61.43	66.57	70.29
	30	61.71	64.57	62.86	64.29	68.00
	25	65.14	64.29	65.71	65.71	68.29
	20	62.86	64.57	66.29	67.14	64.86
	15	61.43	62.29	64.00	65.43	63.71
	10	54.57	55.71	65.14	58.00	62.86
	5	48.00	44.29	53.43	55.43	52.86

The testing results are summarized in Table 4.1 in which predictive accuracies on different feature dimensions are given. It can be observed that both EWPCA and EWKPCA perform generally better than their corresponding origins in terms of extracting discriminative facial features that are capable of leading to high classification rates on testing data. The improvements are particularly significant when low-dimensional feature vectors are extracted for recognition. In analyzing the results of different data sets, we can see that the increasing on CA of GTFD is slightly obvious. It is reasonable as the number of training samples in GTFD database is seven which is larger than those of ORL and the combo set. With more training images per class, EWPCA and EWKPCA are possibly able to learn sufficient details of the intrinsic structures of data, and consequently provide more accurate hypotheses on unseen data. However, maturely trained evolutionary systems on a large number of examples may also suffer from overfitting [108].

From the results of ORL database in Table 4.1, we notice that the maximum dimension of feature vector is constrained to 35, as LDA can extract at most $c - 1$ attributes, where c is the total number of classes (c is 40 for ORL). Nevertheless, our proposed weighted principal components extractors do not have such constraints. Ideally, it is possible to achieve higher CA when high-dimensional yet discriminative features are used. Therefore, we apply EWPCA and EWKPCA to extract more features for classification to illustrate that they can present better generalization performances than LDA but are not limited by the constraints on feature dimension. Figure 4.7 depicts the comparison results on ORL database, in which LDA is not applicable after feature dimension is larger than 35 (m increases with an interval of 5), and meanwhile both EWPCA and EWKPCA present high testing accuracies when $m > 35$. It is

also observed that classification accuracy will not always be improved with the increase of feature dimension, because high-dimensional feature vector may possess redundant information.

(a)

(b)

Figure 4.7: Results on ORL database. (a) LDA, PCA, and EWPCA; (b) LDA, KPCA, and EWKPCA.

Since the population size N_p (the number of weight vectors) is a key factor in controlling the evolutionary process, it is worth investigating the influence of

Figure 4.8: Recognition rate on GTFD data set using EWPCA with different population sizes where feature dimension is five.

variations of N_p on the classification results. To this end, we demonstrate with EPWCA on a wide range of population sizes. The results are visualized in Figure 4.8. Standing the viewpoint of evolution strategies, large N_p can bring great varieties to candidate solutions to avoid getting stuck into local optimum. Nevertheless, as claimed in [108], overtraining may occur with big population while evolving a pattern recognition system. Consequently, parameters acquired from an maturely evolved phase will put the testing on unseen data into an unstable situation that generalization performance could be very poor. Referring to the results in Figure 4.8, it is surprising that large N_p neither degrades predictive accuracy nor improve the performance. It is also noted that small changes on the testing outcomes are presented when either 20 individuals or 300 chromosomes comprise the population. Moreover, both population sizes can help to improve the validation performance significantly. Above observations support the rationale to set N_p as 50 to obtain a trade-off between the system performance and the consumption of computational resources.

Table 4.2: Comparisons with the state-of-the-art methods in handling the SSS problem on ORL and GTFD face databases.

Database	m	LDA	DLDA	RLDA	NLDA	EWPCA	EWKPCA
ORL	35	88.00	93.00	91.00	93.00	92.00	93.00
	30	87.00	92.00	90.00	91.00	92.00	91.50
	25	87.50	92.00	91.00	90.50	91.00	91.00
	20	86.50	87.50	90.00	90.50	88.00	89.00
	15	84.50	85.50	90.00	86.00	88.00	86.50
	10	84.00	75.00	84.50	86.00	85.00	85.00
GTFD	45	61.43	68.57	63.14	70.50	62.86	70.29
	40	60.86	67.71	63.71	70.00	63.43	70.86
	35	61.43	63.43	62.57	70.29	66.57	70.29
	30	62.86	60.29	63.43	70.86	64.29	68.00
	25	65.71	56.00	65.43	70.00	65.71	68.29
	20	66.29	47.71	63.43	69.71	67.14	64.86

The small sample size (SSS) problem [8], [41] is faced by many applications such as face recognition and microarray data analysis where feature vectors are in high dimension but limited samples are available for training. Generally speaking, LDA particularly suffers from the SSS problem so that it is not guaranteed to outperform PCA [116]. In consequence, several methods have been developed to address the SSS problem, such as the regularized LDA (RLDA) [117], the direct LDA (DLDA) [118], the Nullspace LDA (NLDA) [43], and the two-dimensional LDA (2DLDA) [119], [120]. To compare these state-of-the-art algorithms with ours, we conduct the experiments on both ORL and GTFD face databases. The experimental results are presented in Table 4.2. It is observed that DLDA, NLDA, and EWKPCA achieve comparable classification accuracies in higher dimensionality, while DLDA performs poorly when feature dimension is low. Furthermore, although EWPCA is not able to improve the classification performance as much as DLDA, NLDA, and EWKPCA, it still outperforms LDA significantly. Due to the differences in the reduced dimensionality after feature extraction between the 2DLDA algorithm and the other methods, the

experimental results using 2DLDA are given in Figure 4.9 separately. Comparing with the results of other LDA-based methods and our proposed algorithms in Table 4.2, it is obvious that 2DLDA achieves moderate performance but requires much more features than any other techniques.

(a)

(b)

Figure 4.9: Experimental results with the 2DLDA algorithm on two face databases: (a) ORL and (b) GTFD.

Figure 4.10: Experimental results by combining the state-of-the-art LDA-based methods with EWPCA on ORL and GTFD databases: (a) DLDA-based methods on ORL, (b) DLDA-based methods on GTFD.

Since the proposed methods are derived from PCA and KPCA, it is feasible to combine the PCA-based methods with the LDA-based techniques, as the PCA+LDA method [38] is an effective strategy in extracting discriminative features for classification. In this study, we incorporate PCA and EWPCA with

Figure 4.11: Experimental results by combining the state-of-the-art LDA-based methods with EWPCA on ORL and GTFD databases: (a) RLDA-based methods on ORL, (b) RLDA-based methods on GTFD.

DLDA and RLDA, and validate the generated algorithms on ORL and GTFD databases. As seen in Figure 4.10 and Figure 4.11, EWPCA+DLDA and EWPCA+RLDA outperform PCA+DLDA and PCA+RLDA in general. Moreover, the combined methods are capable of achieving higher classification accuracies

than DLDA and RLDA. These observations reveal that better prediction performances are possible to be obtained by integrating EWPCA and EWKPCA with other feature extractors in a straightforward manner.

4.4.2 UCI Image Data Sets

Table 4.3: Summary of UCI data sets. All data sets have been partitioned into two parts for training and testing.

Data set	Feature dimension	Class number	Sample size	Training size	Testing size
Landsat	36	6	2900	900	2000
Optdigits	64	10	5620	3823	1797
Segment	19	7	2310	210	2100

Three image data sets in UCI machine learning repository [112] are used for the evaluation. Their specifications are listed in Table 4.3. Landset satellite data set (Landsat) consists of the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image. The instances in Landset are originally separated into training and testing sets with 4435 and 2000 samples, respectively. To alleviate the computational burden in the learning phase, 150 instances per class are randomly selected for evaluating our proposals. The second database is optical recognition of handwritten digits set (Optdigits) where training and testing sets are partitioned by default. The third test is accomplished on the image segmentation (Segment) database in which the training set is small compared with the testing set.

Table 4.4 presents the recognition rates of five feature extraction algorithms. Similar to the results on the face databases, EWPCA and EWKPCA generally outperform PCA, KPCA, and LDA. Since the class number is small for each UCI data set, more examples per class than face images can be used for training.

Table 4.4: Recognition rate on UCI databases.

Database	m	PCA	KPCA	LDA	EWPCA	EWKPCA
Landsat	5	76.85	79.50	78.10	79.65	80.55
	4	74.75	78.50	76.60	78.15	79.50
	3	73.85	76.90	76.95	77.40	81.90
	2	67.55	74.20	67.70	76.00	74.35
	1	57.35	58.05	56.10	60.60	59.25
Optdigits	9	95.44	95.44	95.44	96.11	96.27
	8	94.16	94.49	94.88	94.71	94.88
	7	93.38	93.54	93.60	94.10	94.60
	6	92.54	92.43	92.26	92.82	92.43
	5	90.26	90.09	89.71	90.76	91.21
	4	81.75	81.97	82.25	84.75	85.14
	3	74.51	75.68	71.40	75.24	75.40
	2	56.04	57.48	53.53	61.99	64.00
	1	31.50	30.94	28.44	35.45	35.56
Segment	6	81.43	80.52	80.86	83.38	83.90
	5	81.00	80.57	80.43	81.29	83.81
	4	80.71	80.52	73.43	81.52	82.19
	3	60.71	77.38	72.29	78.67	78.91
	2	61.81	57.38	57.90	63.05	75.19
	1	39.38	50.00	55.90	55.76	59.48

This observation demonstrates that our methods can extract discriminative features for classification in a wide spectrum of applications. It is also noted that the improvements using EWPCA and EWKPCA on Segmentation and Landsat data sets are significant, especially when low-dimensional representations are used. Given the fact that there are more training samples in Optdigits set, it is concluded that our proposed weighted feature extractors are able to select discriminative features for classification, particularly when few examples are used for learning. This discovery provides the evidence that EWPCA and EWKPCA are suitable for building pattern recognition systems as training samples are very limited in many real-world applications. In addition, low-dimensional representations of the original data can reduce the computing requirements significantly.

(a)

(b)

(c)

Figure 4.12: Recorded variations on different parameters of EWPCA during evolution where κ is 0.02. The results are based on a subset of UCI Optdigits data set.

Table 4.5: Recognition rate with the EWPCA algorithm on a subset of UCI optdigits database where the feature dimension is two. The class separability R_{BW} of the original data set is 1.0549, and the testing accuracy using features extracted by PCA is 82.14%.

	$\kappa = 0.002$			$\kappa = 0.02$		
N_G	$\zeta(\mathcal{F})$	R_{BW}	CA_{test}	$\zeta(\mathcal{F})$	R_{BW}	CA_{test}
5	0.9359	1.3697	0.8984	0.9521	1.6199	0.8862
10	0.9546	1.2708	0.9051	0.9808	1.7497	0.8795
15	0.9638	1.2879	0.9085	0.9862	1.8632	0.8795
20	0.9653	1.3035	0.9051	0.9910	1.7747	0.8728
25	0.9668	1.3003	0.9063	0.9904	1.7754	0.8728
30	0.9662	1.3019	0.9040	0.9899	1.7764	0.8727
35	0.9680	1.3019	0.9040	0.9914	1.7792	0.8717
40	0.9676	1.3019	0.9040	0.9944	1.7861	0.8717
45	0.9691	1.3019	0.9040	0.9950	1.7872	0.8717
50	0.9686	1.3038	0.9051	0.9951	1.7946	0.8717
	$\kappa = 0.2$			$\kappa = 2$		
N_G	$\zeta(\mathcal{F})$	R_{BW}	CA_{test}	$\zeta(\mathcal{F})$	R_{BW}	CA_{test}
5	1.2797	2.5986	0.8147	4.7241	2.7013	0.7913
10	1.3958	2.9748	0.8315	6.0721	3.3663	0.6339
15	1.4716	3.4149	0.7623	7.4458	3.9954	0.6787
20	1.4886	3.5337	0.7567	8.4518	4.0363	0.6842
25	1.5248	3.5443	0.7545	8.5733	4.0557	0.6830
30	1.5394	3.5990	0.7478	8.6777	4.0729	0.6775
35	1.5423	3.5988	0.7500	8.7514	4.0793	0.6775
40	1.5492	3.6088	0.7489	8.7546	4.0798	0.6763
45	1.5519	3.6088	0.7489	8.8773	4.1006	0.6730
50	1.5478	3.6542	0.7411	8.8531	4.1006	0.6730

To obtain discriminative representations through evolution, a suitable fitness function is needed to guide the whole procedure. Therefore, it is important to know the effects of parameters in fitness function on the final results. To this end, the subset of Optdigits described in Section 4.2.2 is chosen for illustration. In detail, EWPCA is applied using the same parameters as those used in previous demonstration. First of all, we would like to investigate the relationship between the performance accuracy term $\zeta_{ca}(\mathcal{F})$ and the discrimination term $\zeta_d(\mathcal{F})$. Figure 4.12(a), (b), (c) depict the mean value of fitness, the class separa-

bility criterion R_{BW} , and the testing accuracy, respectively, on different number of generations. In this example, it is clear that the evolving process converges after 25 iterations. Moreover, R_{BW} and the testing accuracy are 1.0549 and 0.8214 when PCA is utilized where $m = 2$. In comparison, it is seen from Figure 4.12 that EWPCA achieves better generalization performance ($CA > 0.85$) and obtains larger class separability ($R_{BW} > 1.7$) than PCA by means of the optimal weights. Larger R_{BW} can be obtained by adjusting κ in the fitness, but it should be emphasized that our goal is not to find the “best” low-dimensional representation for patterns such that the training error is minimized and the underlying distribution of each class can be approximated. Rather, the features are expected to result in small training error and improved generalization performance. As a consequence, κ is determined empirically so that the predictive accuracy term in the fitness can contribute more than the class separability criterion does. Table 4.5 presents the results with different values of κ , where CA_{test} indicates the testing accuracy. It is obvious that R_{BW} becomes larger with the increase of κ while testing accuracy decreases accordingly. This observation reflects that κ can control the contribution of class separability criterion to the fitness function. An appropriately selected κ is capable of enlarging R_{BW} as well as achieving good generalization performance. We also notice that CA_{test} usually reaches to its peak value within several generations, and decreases until it stabilizes after 50 iterations of genetic operations. Refer to the previous discussion, mature convergence can have small training error but weaken the ability of generalization on unseen data. On the contrary, un-optimized weights may perform poorly in learning whereas achieve high classification accuracy in testing. That is the reason that the values of CA_{test} at the beginning of the evolution process are normally larger than the final testing results.

4.4.3 Object Categorization

Figure 4.13: Sample images in COIL-20 database. In this example, 6 out of 72 images of the same object are selected to show within-class and between-class differences. As seen from these pictures, large pose variation is observed in each category.

Columbia Object Image Library [113] is adopted as the object data set for algorithm evaluation. COIL-20 contains 1440 gray scale images of 20 objects. As shown in Figure 4.13, wide variety of geometric and reflectance characteristics can be observed from different images of the same subject. After operating normalization and histogram on each picture, 1440 gray scale images (128×128) are obtained, from which 36 images of each subject are selected to construct the training set, and the remaining 720 samples are used for testing. As 72 images per object are taken from a sequence at every 5 degrees of rotation, there are great variations in poses among images of single object. Hence, random selection of training and testing samples may introduce bias to data distribution. In practice, if the image index is even number, its corresponding image will be chosen into the training set so that images in testing set have similar pose orientation as the samples in training set. Furthermore, DCT is also applied to reduce dimensionality prior to implementing feature extraction algorithms.

Figure 4.14: Recognition rate on COIL-20 database where dimensionality of the feature vector is three.

In this study, we are interested in investigating the abilities of EWPCA and EWKPCA in handling the SSS problem with different number of training samples. Toward this end, six subsets are constructed using the original training data with various number of samples from each object. As there are 36 images per class, according to their indexes, the first six images forms the first subset. Then, the second subset is comprised of the first 12 images, and so forth. The results in Figure 4.14 reveal that LDA performs poorly in achieving high classification accuracy, and the effects of the SSS problem is obvious when only six examples are used for training. Meanwhile, PCA and KPCA, as well as our methods, are less sensitive to the number of training samples. These observations support our ideas that EWPCA and EWKPCA can inherit the advantages of PCA, KPCA and LDA, and simultaneously avoid involving their drawbacks. Furthermore, Table 4.6 summarizes the classification results using various feature extractors when the training is carried out with six samples per object. It can also be observed that the discriminatory power of features extracted by

Table 4.6: Recognition rate on COIL-20 database where six samples per class are used for training.

m	PCA	KPCA	LDA	EWPCA	EWKPCA
19	90.69	90.56	72.08	91.53	91.81
18	91.39	90.28	74.86	92.08	91.81
17	91.25	89.86	73.19	91.53	90.83
16	90.28	89.58	73.89	91.39	90.83
15	90.42	89.58	73.89	91.11	90.69
14	89.17	89.03	75.28	90.83	90.56
13	89.86	88.19	76.11	90.28	90.14
12	89.44	88.06	79.17	90.56	90.28
11	89.31	88.33	78.19	90.00	89.86
10	88.89	87.64	78.75	90.28	89.31

LDA is degraded significantly due to limited training samples.

4.5 Discussion

The proposed weighted principal components extraction methods have proven effective on various benchmark data sets. Apparently, EWPCA and EWKPCA outperform PCA, KPCA, and LDA in terms of achieving higher testing accuracies. The results in Table 4.1 reveal that our methods can tackle high-dimensional data effectively because face recognition is a challenging task where redundancy exists in a huge number of attributes. The evaluations on UCI data sets extend EWPCA and EWKPCA to be suitable for a broad spectrum of image analysis applications. Furthermore, both face databases and COIL-20 image library are employed to demonstrate the capability of the proposed algorithms in solving the SSS problem.

Other than the generalization performance, the issue of algorithmic complexity should be addressed as well. To have an intuitive understanding on

the time complexity, the combo face database (555 training samples and 575 testing samples) and the UCI Optdigits data set (3823 training samples and 1797 testing samples) are used for demonstration. The training and testing time with PCA and EWPCA methods are compared on the above two data sets. In detail, the population size N_p and the number of generations N_G are set as 50, and 5-fold cross-validation is used in the training phase. It is observed from the experimental results that the training and testing with PCA on both data sets can be finished within one second. Moreover, the testing with EWPCA can also be efficiently done in a second, whereas the training procedure in the EWPCA algorithm requires around 20 minutes for the combo face database and 200 minutes for the Optdigits database. Therefore, we conclude that the proposed methods have a limitation to be extended for online learning, which is also faced by most evolutionary systems. The major obstacle in accelerating the extraction of evolved features is brought about by the genetic algorithms because the feature extraction and classification procedures must be repeated at least $N_p \cdot N_G$ times in the training phase. Although several trials, such as decreasing values for N_p and N_G , choosing simple classifiers, can alleviate the computational burden, the time-consuming procedure originated from the nature of population-based random selection, is still a weak point in the evolutionary pattern recognition system.

In the remaining of this section, some interesting observations obtained from the results are summarized as follows.

- The combination of the merits of PCA, KPCA, and LDA, is considered as a key advantage of the weighted principal components extractors. Moreover, the proposals also benefit from the ability of GAs to seeking for global

optimum. In the training phase, the evolution strategy integrates feature extraction and classification together so as to determine the optimal weights with respect to providing high training rate and simultaneously increasing the class separability. Evidences that support the superiorities of EWPCA and EWKPCA over other feature extraction methods have been shown in the experiments.

- It is worth mentioning that the modified Fisher criterion plays a crucial role, because it not only utilizes class information to enhance EWPCA and EWKPCA to become supervised, but also helps to avoid the SSS problem usually faced by several LDA-based techniques. The experimental results in Section 4.4.1 show that EWPCA and EWKPCA achieve comparable generalization performances with RLDA, DLDA, and NLDA. In addition, the classification results are significantly improved when the proposed EWPCA algorithm is incorporated with LDA-based techniques.
- We also notice that the features extracted by EWPCA and EWKPCA are compact and discriminative. This observation is strongly supported by the results in Table 4.1 and Table 4.6. From the classification results on ORL in Table 4.1, it is noted that only 15 principal components extracted by EWPCA can achieve an accuracy of 88%, whereas when PCA, KPCA, and LDA are applied, 25, 25, and 35 dimensional features are required to obtain the same testing performance. As far as our knowledge goes, compact representations of the original data will not merely speed up the learning and testing processes, but also relieve the burden of storage.
- Since the feature extraction scheme is enhanced by GAs, several parameters, such as N_p , p_c , and p_m , are introduced and should be decided be-

forehand. Therefore, it is useful to evaluate our methods under different settings of parameters. In the experiments, the population size N_p and κ in the fitness function, are particularly discussed to show their effects on the final results. Figure 4.8 depicts the testing accuracies over a large variations of population size. The results convince us that even moderate value of N_p can lead to good generalization performance. This finding is fairly important as a small amount of individuals in population can save the learning time substantially. Moreover, the results in Table 4.5 reveal that κ is of primary importance in controlling the trade-off between accuracy and class separability in order to generalize well on unknown data.

In general, both EWPCA and EWKPCA can be employed in the applications where improving the classification accuracy is the major task while the time consumption is not the main concern. When conducting the experiments to evaluate our methods, we attempt to present the advantages of the weighted principal components extractors for pattern analysis, but not to target at achieving the highest classification accuracy. Hence, better results can be expected by selecting proper parameters. In practical settings, it is suggested to choose different combinations of parameters that are specifically designed for the evolutionary systems so that satisfactory outcomes can be obtained.

4.6 Summary

This chapter introduces two novel feature extraction algorithms, EWPCA and EWKPCA, and shows their superiorities in extracting compact yet discriminative representations from the original data to facilitate the classification. The

fundamental principles behind our proposals are the introduction of weights to PCA and KPCA, and the use of evolution strategy for selecting optimal parameters. The weighted feature extractor and the classifier are integrated by GAs to build a framework of training, from which optimal weights are obtained in terms of helping extract discriminative features to achieve high training accuracy. Furthermore, since PCA and KPCA are unsupervised methods, we take advantage of the weights in creating a modified Fisher criterion R_{BW} and incorporate it to the fitness function as the class separation term. As a consequence, we employ the fitness to evolve a balanced performance in achieving good learning results as well as enlarging the class separability.

One challenge should be addressed is that minimizing the overall learning error may make the GAs-based prediction model fit training data very well, and consequently degrades the generalization ability. In dealing with the above mentioned difficulty, we choose using five-fold cross-validation scheme rather than making use of the entire training set to estimate the learning error. In addition, we adopt a moderate population size to avoid pre-mature convergence. Another difficult problem is the great consumption of time for training which is an inevitable issue faced by almost all other evolutionary systems for pattern analysis. Therefore, fast evolution strategies, for example, particle swarm optimization, is worthwhile for further investigation to improve systematic efficiency. Overall, regardless of the aforementioned drawback, EWPCA and EWKPCA have been demonstrated with a variety of applications to be able to extract discriminative features. The effects of parameters in our methods are also well studied and illustrated in the experiments. Moreover, we have pointed out that better testing performances can be obtained than our reported results. Toward that end one is suggested to select the parameters specifically designed for applications.

Chapter 5

Classification with Modified Extreme Learning Machine

5.1 Introduction

Neural networks (NN) have been used actively in image analysis because of their capabilities to learn complex nonlinear input-output relationships and do predictions with trained networks. NNs also benefit from their low dependence on domain-specific knowledge [2], availability of efficient learning methods such as backpropagation, and robustness to errors in the training phase [75]. Among various types of network topology, feed-forward network is commonly employed [85]. Unlike the architecture of recurrent neural networks, there are no directed cycles in the network, and the information moves forwardly from the input nodes, through the hidden nodes and to the output nodes. Although feed-forward neural networks are the arguably simplest type of artificial neural networks devised, they may suffer problems such as local minima, improper

learning rate, learning epochs, slow training, and overfitting faced by conventional gradient-based learning algorithms [86], [88].

Extreme learning machine (ELM) was proposed recently as an efficient learning algorithm for single-hidden layer feedforward neural network (SLFN) [18]. It increases the learning speed by randomly generating weights and biases for hidden nodes rather than iteratively adjusting network parameters that is commonly adopted by gradient-based methods. Although ELM is fast and presents good generalization performance, it still has several drawbacks and thus leaves a lot of room for improvements. Zhu et al. [86] claimed that the random assignment of parameters will introduce un-optimal input weights and hidden biases. Consequently, evolutionary extreme learning machine (E-ELM) [86] was proposed by taking advantage of both ELM and differential evolution (DE) [121] to remove redundancy among hidden nodes and achieve satisfactory performance with more compact networks. Furthermore, pruned-ELM (P-ELM) was presented by Rong et al. [87]. The rationale behind P-ELM is to initialize a large network and prune it during learning. Apart from numerous improvements [122], [123], ELM was also implemented in microarray data classification [88] and showed its superiority to support vector machines.

In this chapter, we propose using cross-validation scheme for training ELM and E-ELM to improve the classification performance for image analysis. In supervised learning, classifiers usually suffer from overtraining which may degrade the generalization performance. During the training phase, complete training samples are categorized into several classes by classifier and the learning error is used to evaluate the efficiency of training. Normally, minimum training error is expected, but it cannot guarantee good recognition results on unseen data. The

main mechanism behind our proposals is partitioning the original training set using R -fold cross-validation scheme into R non-overlapped subsets and then R pairs of training and validation sets are obtained so that each training set consists of $(R - 1)$ subsets. Then in the new training procedure, each of the R learners is constructed using $(R - 1)$ subsets and validated with the remaining single subset. The cross-validation process is then repeated R times, with each of the R subset used exactly once for validation. Subsequently, in the improved ELM algorithm, a decision ensemble on testing data is constructed using multiple sets of R predictors. In the extension of E-ELM algorithm, the average classification accuracy (CA) across all R trials is employed as fitness in training phase for selecting the most fitting network parameters for testing. The above mentioned learning procedure is reasonable to avoid overfitting because the validation set ($\frac{1}{R}$ training samples) is used to replace the whole training set to evaluate learning error in each one of the R classifiers.

5.2 Preliminaries

5.2.1 Extreme Learning Machine

As one of the learning algorithms for SLFN, ELM randomly selects weights and biases for hidden nodes, and analytically determines the output weights by finding least squares solution. Given training set consisting of N samples $L = \{(\mathbf{x}_j, \mathbf{t}_j) | \mathbf{x}_j \in \mathbf{R}^n, \mathbf{t}_j \in \mathbf{R}^m, j = 1, 2, \dots, N\}$, where $\mathbf{x}_j$ is an $n \times 1$ input vector and $\mathbf{t}_j$ is an $m \times 1$ target vector, an SLFN with $\tilde{N}$ hidden nodes is

formulated as

$$f_{\tilde{N}}(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, 2, \dots, N \quad (5.1)$$

where additive hidden node is employed. $\mathbf{w}_i$ is n -dimensional weight vector connecting i th hidden node and input neurons. In approximating N samples using $\tilde{N}$ hidden nodes, β_i , $\mathbf{w}_i$, and b_i are supposed to exist if zero error is obtained. Consequently, (5.1) can be written in a more compact format as $\mathbf{H}\hat{\boldsymbol{\beta}} = \mathbf{T}$ where $\mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, \mathbf{x}_1, \dots, \mathbf{x}_N)$ is hidden layer output matrix of the network as shown in (5.2), $h_{ji} = g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)$ is the output of i th hidden neuron with respect to $\mathbf{x}_j$, $i = 1, 2, \dots, \tilde{N}$ and $j = 1, 2, \dots, N$; $\hat{\boldsymbol{\beta}} = [\beta_1, \dots, \beta_{\tilde{N}}]^T$ and $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_N]^T$ are output weight matrix and target matrix, respectively.

$$\begin{aligned} & \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, \mathbf{x}_1, \dots, \mathbf{x}_N) \\ &= \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \end{aligned} \quad (5.2)$$

Huang et al. [18] pointed out that in real applications training error cannot be made exactly zero as the number of hidden nodes $\tilde{N}$ will always be less than the number of training samples N . To obtain small non-zero training error, Huang et al. [18] proposed randomly assigning values for parameters $\mathbf{w}_i$ and b_i , and thus the system becomes linear so that the output weights can be estimated as $\boldsymbol{\beta} = \mathbf{H}^\dagger \mathbf{T}$, where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse [124] of the hidden layer output matrix $\mathbf{H}$.

In general, given a training set L_{trn} , activation function $g(x)$, and hidden node number $\tilde{N}$, the ELM algorithm can be summarized as three steps:

1. Generate parameters $\mathbf{w}_i$ and b_i randomly for $i = 1, \dots, \tilde{N}$.
2. Calculate the hidden layer output matrix $\mathbf{H}$.
3. Calculate the output weight using $\boldsymbol{\beta} = \mathbf{H}^\dagger \mathbf{T}$.

It has been shown in [122] that any continuous target functions in $\mathbf{R}^n$ can be universally approximated using single SLFN with randomly chosen additive (or RBF) hidden nodes.

5.2.2 Evolutionary Extreme Learning Machine

To eliminate possible non-optimum within hidden nodes and create more compact networks, an evolutionary extreme learning machine algorithm was introduced [86]. E-ELM deploys DE to select optimal weights and biases. At the beginning, E-ELM initializes a population of N_p parameter vectors $\{\mathbf{z}_{p,G} | p = 1, 2, \dots, N_p\}$, and then chooses best individual in terms of fitness to form a new generation in which the selection pool contains candidates from G th generation and their variants after operations of mutation and crossover. In E-ELM, the fitness of each individual is defined as root mean squared error (RMSE) on validation set instead of whole training set [86]. The RMSE is formulated as (5.3).

$$E = \sqrt{\frac{\sum_{j=1}^N \left\| \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right\|^2}{m \times N}} \quad (5.3)$$

In addition, the norm of output weights $\|\boldsymbol{\beta}\|$ is considered as another criterion to improve the generalization performance.

5.3 The Proposed Ensemble Based ELM

Depending on the random selection of weights and biases for hidden nodes, ELM decreases the learning time dramatically. However, the parameters do not incorporate prior knowledge about inputs and may contain non-optimum, and thus degrade the generalization performance. Consequently, we propose constructing an ensemble of several predictors on the training set using various sets of random parameters, in which parameters of each component are updated according to particular criteria, and then making decisions for testing samples through majority voting with the ensemble. The proposed ensemble based ELM algorithm (EN-ELM) is illustrated in Figure 5.1. Cross-validation is applied throughout the classification task. On the one hand, cross-validation scheme prohibits overfitting; on the other hand, it extends the number of predictors in the ensemble so as to guarantee a stable and accurate decision making.

The proposed EN-ELM algorithm starts its learning with partitioning the entire training set into R subsets and initializing $\mathbf{w}_i$ and b_i . Then each of R learners is trained using $(R - 1)$ subsets and validated with the remaining one subset. Afterward, results on the single validation set across R trials, including the mean of accuracies CA^{K+1} as well as mean value of norms of output weights $\|\boldsymbol{\beta}^{K+1}\|$ are calculated. In the next K iterations, if the parameters of current iteration result in higher classification accuracy, or smaller norm of $\boldsymbol{\beta}$, the values of parameters $\mathbf{w}_i^k$ and b_i^k are assigned to $\mathbf{w}_i^{K+1}$ and b_i^{K+1} , respectively; otherwise update current weights and biases with $\mathbf{w}_i^{K+1}$ and b_i^{K+1} . It is commonly agreed that smaller $\|\boldsymbol{\beta}\|$ could lead to better generalization performance [18], [86]. Hence, $\|\boldsymbol{\beta}\|$ is considered as one factor in creating the decision ensemble. After finishing learning all training samples, a hypothesis ensemble of $R \times K$ predictors

Algorithm: EN-ELM**Inputs**

- Sequence of N examples $L = \{(\mathbf{x}_j, \mathbf{t}_j) | \mathbf{x}_j \in \mathbf{R}^n, \mathbf{t}_j \in \mathbf{R}^m, j = 1, 2, \dots, N\}$.
- Number of iterations K , number of hidden nodes $\tilde{N}$, cross-validation fold number R .

Initialization

1. Randomly generate $\mathbf{w}_i$ and b_i where $i = 1, 2, \dots, \tilde{N}$.
2. Partition training set into R subsets so that R groups of data are obtained. In each group, $\frac{(R-1)N}{R}$ samples are used for training and the rest of un-overlapped $\frac{N}{R}$ examples in the entire training set form the validation set.
3. Assign $\mathbf{w}_i^{K+1}$ and b_i^{K+1} with the values of $\mathbf{w}_i$ and b_i , respectively. Then calculate mean values of classification accuracies as well as $\|\boldsymbol{\beta}\|$ for R validation sets and store the values into CA^{K+1} and $\|\boldsymbol{\beta}^{K+1}\|$.

For $k = 1, 2, \dots, K$

1. Randomly generate $\mathbf{w}_i^k$ and b_i^k where $i = 1, 2, \dots, \tilde{N}$.
2. Calculate mean of classification accuracy CA_r^k on R subsets where $r = 1, 2, \dots, R$ and store the values into CA^k and the norm of $\boldsymbol{\beta}^k$, where $\text{CA}^k = \frac{1}{R} \sum_{r=1}^R \text{CA}_r^k$, $\|\boldsymbol{\beta}^k\| = \frac{1}{R} \sum_{r=1}^R \|\boldsymbol{\beta}_r^k\|$.
3. **If** $\text{CA}^k < \text{CA}^{K+1}$ or $\|\boldsymbol{\beta}^k\| > \|\boldsymbol{\beta}^{K+1}\|$

Update parameters as $\mathbf{w}_i^k = \mathbf{w}_i^{K+1}$, $b_i^k = b_i^{K+1}$.

Else Set $\mathbf{w}_i^{K+1} = \mathbf{w}_i^k$, $b_i^{K+1} = b_i^k$.

End if

End for**Hypothesis ensemble construction****For $r = 1, 2, \dots, R$**

Given a testing instance $(\mathbf{x}, \mathbf{t})$, evaluate the hypothesis ensemble $\{\hat{h}_1^r, \hat{h}_2^r, \dots, \hat{h}_K^r\}$ on $\mathbf{x}$,

For $k = 1, 2, \dots, K$

- (a) Obtain updated hidden node weights and biases $\mathbf{w}_i^k$ and b_i^k and call ELM algorithm to predict target for $\mathbf{x}$ using partial L^r , i.e., $(R-1)$ subsets.
- (b) Set $v_{k,c}^r$ to 1 if $\hat{h}_k^r$ is predicted as class c , otherwise, its value is zero, where $c = 1, 2, \dots, C$.

End for**End for****Decision making with majority voting**

1. In ensemble, if $\|\boldsymbol{\beta}_r^k\| > \text{median}\{\|\boldsymbol{\beta}\|\}$, set $v_{k,c}^r$ to zero.
2. From above ensemble, obtain total vote received by each class as $V_c = \sum_{r=1}^R \sum_{k=1}^K v_{k,c}^r$.
3. The class that receives the highest votes is considered as the predicted label.

Figure 5.1: The architecture of the EN-ELM algorithm.

is obtained. As $\|\beta\|$ is available after training for every classifier, all predictors in ensemble are sorted according to the norm of β from small to large, and the first half of the ensemble is used to make decision through majority voting.

5.4 The Proposed Cross-Validation Based E-ELM

E-ELM was proposed by employing both RMSE and β on validation set for best candidate selection to achieve better classification accuracy with more compact networks. Since the original testing set is evenly separated into real testing and validation sets, E-ELM uses extra data for training which is not suitable for applications in which test samples are limited. Therefore, the validation set is crucial to E-ELM for learning. Alternatively, samples from training set can be used to form the validation set, but intuitively it may affect the generalization performance as the number of training samples decreases. Hence, we propose the cross-validation based E-ELM (E-ELMcv) algorithm to avoid using an extra validation set for training.

In order to inherit the merit of ELM, the proposed algorithm also deploys the differential evolution (DE) algorithm as a tool to select optimal weights and biases for hidden nodes. At first, a set of parameter vectors $\{\mathbf{z}_{p,G} | p = 1, 2, \dots, N_p\}$ is initialized, in which components of $\mathbf{z}_{p,G}$ have bound as $[-1, 1]$.

$$\mathbf{z}_{p,G} = [w_{11}, \dots, w_{1\tilde{N}}, \dots, w_{n1}, \dots, w_{n\tilde{N}}, b_1, \dots, b_{\tilde{N}}] \quad (5.4)$$

where G denotes the G th generation, and input weights $\mathbf{w}_i$ and hidden node

biases b_i form the candidate vector. The dimension of the vector depends on the number of hidden nodes $\tilde{N}$ and the feature dimension n . DE updates the population under the driven of fitness function. Before creating a new generation, mutation, crossover, and selection operations are applied. In detail, for each vector $\mathbf{z}_{p,G}$, a mutant vector is generated according to $\hat{\mathbf{z}}_{p,G+1} = \mathbf{z}_{r_1,G} + F \cdot (\mathbf{z}_{r_2,G} - \mathbf{z}_{r_3,G})$ with random indices $r_1, r_2, r_3 \in \{1, 2, \dots, N_p\}$ and positive real number F which is not larger than 2. F is a factor to control amplification of differential variation $(\mathbf{z}_{r_2,G} - \mathbf{z}_{r_3,G})$. Crossover operator is introduced to increase diversity among the population subsequently. As a result, the D -dimensional vector is constructed as

$$\tilde{\mathbf{z}}_{p,G+1} = (\tilde{z}_{1p,G+1}, \tilde{z}_{2p,G+1}, \dots, \tilde{z}_{Dp,G+1}) \quad (5.5)$$

where $D = \tilde{N}(n + 1)$, and we have

$$\tilde{z}_{qp,G+1} = \begin{cases} \hat{z}_{qp,G+1} & \text{randb}(q) \leq \text{CR} \\ & \text{or } q = \text{rnbr}(p) \\ z_{qp,G} & \text{randb}(q) > \text{CR} \\ & \text{and } q \neq \text{rnbr}(p) \end{cases} \quad (5.6)$$

In (5.6), $q \in \{1, 2, \dots, D\}$, and the q th evaluation of a uniform random number generator with outcome in $[0, 1]$ is determined by $\text{randb}(q)$. CR is a user-defined constant in $[0, 1]$. A random index $\text{rnbr}(p)$ is used to ensure that at least one parameter from $\hat{\mathbf{z}}_{p,G+1}$ is obtained by $\tilde{\mathbf{z}}_{p,G+1}$.

Prior to selection, fitness values are calculated for all $\mathbf{z}_{p,G}$ and $\tilde{\mathbf{z}}_{p,G+1}$ where $p = 1, 2, \dots, N_p$. The fitness function plays a key role in candidate selection. We apply classification accuracy only as the fitness function rather than the

combination of RMSE and β in E-ELM for selection. The training set L is partitioned into R pairs of data sets using cross-validation scheme. Then for each $\mathbf{z}_{p,G}$ and $\tilde{\mathbf{z}}_{p,G+1}$, $\text{CA}^{p,G}$ and $\text{CA}^{p,G+1}$ are evaluated as $\frac{1}{R} \sum_{r=1}^R \text{CA}_r^{p,G}$ and $\frac{1}{R} \sum_{r=1}^R \text{CA}_r^{p,G+1}$, respectively. If $\tilde{\mathbf{z}}_{p,G+1}$, the evolved candidate, appears fitter than the original parameter vector, i.e., $\text{CA}_r^{p,G+1} > \text{CA}_r^{p,G}$, $\tilde{\mathbf{z}}_{p,G+1}$ will be selected into the next generation instead of $\mathbf{z}_{p,G}$; otherwise, $\mathbf{z}_{p,G}$ is considered as elite candidate and continues to survive in the $(G+1)$ th generation as $\mathbf{z}_{p,G+1}$. After a number of iterations, the best candidate vector $\mathbf{z}^{\text{best}}$ in terms of achieving highest prediction accuracy is obtained for testing. Given new patterns, predictions are carried out using $\mathbf{w}_i$ and b_i in $\mathbf{z}^{\text{best}}$. In general, no extra validation set is needed to facilitate training in E-ELMcv algorithm.

5.5 Experiments

The experimental validations are carried out on three face databases (Combo, ORL, and GTFD) and two image data sets in UCI machine learning repository with ELM, E-ELM, and the proposed algorithms. The characteristics of these data sets are summarized in Table 5.1. In the face databases, all the images are resized to 112×92 and pre-processed by DCT. Details of processing on face images are described in Section 4.4.1 (Chapter 4). All of these computerized simulations are run in MATLAB on a workstation equipped with Intel Pentium 4 3.2GHz CPU and 1G RAM. The learning and testing processes are repeated for 50 times with sigmoidal function $g(x) = 1/(1 + e^{-\lambda x})$ as the activation function. In this chapter, 10-fold cross-validation is adopted for training. EN-ELM chooses 50 iterations to create the decision ensemble. In E-ELM and E-ELMcv, N_p , F , and CR are 200, 1, and 0.8, respectively. The number of

Table 5.1: Data sets used in the experiments.

Data set	Training	Testing	Dimension	Class
Combo	555	575	81	75
ORL	200	200	81	40
GTFD	400	350	81	50
Landsat	3217	3218	36	6
Segment	1500	810	19	7

generations is heuristically determined as 20. In the experiments, all approaches use the original training and testing sets, except for the E-ELM algorithm which divides testing data into two groups equally and chooses one part as validation set to avoid overtraining.

Table 5.2 presents the comparison results. It is shown that ELM is the fastest learner but performs the worst in classification. The proposed E-ELMcv algorithm outperforms ELM and E-ELM in terms of achieving higher testing accuracies on all data sets. In summary, EN-ELM is a stable and efficient extension as it can provide satisfactory generalization performance within comparable short training time. Before recording the results for the E-ELMcv method, several trials have been done and the testing outcomes indicate that E-ELMcv need more time than E-ELM to be trained. Hence, we reduce the population size N_p and the number of generations to half of their values, and discover that the learning time decreases dramatically. The results using the above new parameters for E-ELMcv in Table 5.2 show that although the population size is shrunk and the evolving procedure is shortened, our proposed E-ELMcv algorithm can still achieve higher testing accuracies than E-ELM in comparable learning time. Moreover, referring to the results reported in Zhu et al. [86], conventional gradient-based learners cost much longer time for training and unfortunately, the validation results are far from satisfactory when compared with

Table 5.2: Comparisons of the experimental results on three face databases and two UCI image data sets.

Data set	Classification algorithm	Training time (s)	Testing accuracy (%)	Hidden nodes
Combo	ELM	0.0672	81.82 ± 1.35	100
	EN-ELM	32.488	84.67 ± 1.55	100
	E-ELM	88.641	81.90 ± 1.23	100
	E-ELMcv	103.47	84.02 ± 1.19	100
ORL	ELM	0.0391	78.45 ± 2.72	100
	EN-ELM	18.959	82.33 ± 2.50	100
	E-ELM	62.813	80.70 ± 3.48	100
	E-ELMcv	69.188	82.30 ± 2.74	100
GTFD	ELM	0.0555	54.31 ± 1.88	100
	EN-ELM	28.941	57.20 ± 2.56	100
	E-ELM	81.141	57.17 ± 2.82	100
	E-ELMcv	92.969	60.29 ± 2.32	100
Landsat	ELM	0.8297	88.10 ± 0.32	200
	EN-ELM	445.79	88.27 ± 0.52	200
	E-ELM	4324.9	88.44 ± 0.62	200
	E-ELMcv	4416.8	88.72 ± 0.73	200
Segment	ELM	0.4180	94.83 ± 0.80	200
	EN-ELM	212.93	95.22 ± 0.63	200
	E-ELM	1440.8	95.46 ± 1.24	200
	E-ELMcv	2000.5	95.69 ± 1.02	200

the results achieved by E-ELMcv.

The effects of parameters in EN-ELM are depicted in Figure 5.2. It is observed from Figure 5.2(a) that the classification accuracies increase monotonically till the number of folds is larger than eight. A small fold number results in poor generalization performance because less training samples involve in learn-

(a)

(b)

Figure 5.2: Results on the ORL database using EN-ELM method: (a) Classification results with different number of folds for cross-validation where $\tilde{N}$ is 100; (b) testing results with different number of hidden nodes where R is 10.

ing process. Figure 5.2(b) illustrates that a large number of hidden nodes may give higher accuracies in testing. However, a complex network could also overfit the training data. For example, the generalization performance decreases when the hidden node number is larger than 80.

5.6 Summary

In this chapter, a new cross-validation based strategy is introduced into the training processes of ELM and E-ELM to alleviate the problem of overfitting and improve generalization ability. As a result, EN-ELM and E-ELMcv algorithms are proposed. EN-ELM is an enhanced version of ELM by employing the cross-validation mechanism and the ensemble learning, and E-ELMcv is an extension of E-ELM. The validation results illustrate that our proposals outperform their original algorithms in terms of the classification accuracy. Although the proposed methods spend more time for training than ELM, they are still effective when compared with E-ELM and traditional gradient-based learning algorithms. In addition, it is also possible to alleviate the computational burden significantly by selecting appropriate network parameters.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, several novel feature extraction and classification algorithms are proposed, which have been successfully validated with face databases, COIL object database, and UCI image data sets.

To enhance the performance of an image analysis system, the MTF is proposed as a novel representation for the original image. The MTF benefits from the assumption that it is possible to make explicit differences among various 2D Trace transforms of the same image which are implicitly presented in the original representation. During the construction of MTF, GAs are utilized to tune the parameters for fusing multiple Trace transforms. Furthermore, the Bagging algorithm is coupled with the cross-validation scheme to produce a novel fitness function such that the overfitting problem in training phase may be avoided. The testing conducted on a combo face database reveals that MTF outperforms the features extracted by PCA and LDA. In addition, the Bagging-based fit-

ness helps in achieving highest classification accuracy. The MTF method can be implemented to any image-based pattern recognition systems although face patterns are used for validation in this thesis.

Two novel feature extraction algorithms, namely EWPCA and EWKPCA, are presented in Chapter 4. They have been extensively tested on face databases, COIL object database, and UCI image data sets. The experimental results show their superiorities in extracting compact yet discriminative features to facilitate the classification. EWPCA and EWKPCA can be considered as evolution-based weighted extensions of PCA and KPCA, respectively. To extract discriminative features, GAs initialize a set of class-dependent weight vectors from which linear or nonlinear weighted subspace can be constructed. Then the evolutionary process guided by the fitness function is carried out to search the optimal weights in terms of achieving the highest training accuracy. The fitness function consists of two components, one is the accuracy term, another is the class separation term which is the proposed modified Fisher criterion. Therefore, maximizing the fitness will lead to balanced performances in achieving good learning results as well as enlarging the class separability.

In the third proposal, the training processes of ELM and E-ELM are extended by incorporating a new cross-validation based structure such that the overfitting problem may be avoided and the generalization ability is possible to be improved. These improvements result in two novel classifiers, namely EN-ELM and E-ELMcv. The superiorities of these proposed algorithms are illustrated by comparing against their original formulations.

6.2 Recommendations for Future Research

Although the presented works have been shown to perform successfully, they have the potentials to be further enhanced to build reliable pattern recognition systems for a variety of applications. Several possible directions for future investigation are summarized as follows.

- The proposed MTF utilizes eight Trace transforms obtained from a single image. One feasible way to improve the discriminability of MTF is fusing more available Trace transforms. In addition, it is very promising to create an ensemble of classifiers with different sets of transforms to enhance the reliability of the prediction, where the classifier combiner can be either majority voting or AdaBoost.
- Since high computational complexity is usually an inevitable difficulty in most evolution-based pattern recognition systems, we would like to investigate the possibilities to alleviate the computational burden. As discussed in Chapter 4, selecting appropriate parameters could be useful. Furthermore, it may be a wise choice to replace GAs with PSO to accelerate the learning process, because PSO does not adopt the crossover and mutation operators while it can still receive satisfactory performance.
- ELM and its variants are mainly deployed as classification tools for pattern analysis. Besides, it is observed that the inherent characteristics of ELM can enable it to be a good feature selection method. The rationale behind this hypothesis is that smaller $\|\beta\|$ can lead to better generalization performance [18], [86]. Consequently, the feature subsets can be evaluated by simply calculating the values of their corresponding $\|\beta\|$.

Author's Publications

Journal Papers

1. N. Liu and H. Wang, "Weighted Principal Components Extraction with Genetic Algorithms," *Pattern Recognition*. Submitted.
2. N. Liu and H. Wang, "Classification with Ensemble Based Extreme Learning Machine," *Neurocomputing*. Submitted.
3. N. Liu and H. Wang, "Pattern Analysis of Face Images Using Trace Transform," *Digital Signal Processing*. Submitted.
4. N. Liu and H. Wang, "Modeling Images with Multiple Trace Transforms for Pattern Analysis," *IEEE Signal Processing Letters*, vol.16, no.5, pp.394-397, 2009.
5. N. Liu and H. Wang, "Improving Predictive Accuracy by Evolving Feature Selection for Face Recognition," *IEICE Electronics Express*, vol.5, no.24, pp.1061-1066, 2008.

Conference Papers

6. N. Liu and H. Wang, "Feature Selection in Frequency Domain and Its Application to Face Recognition," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, Hong Kong, Jun. 2008.
7. N. Liu and H. Wang, "Recognition of Human Faces using Discrete Cosine Transform Filtered Trace Features," in *Proceedings of IEEE International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore, Dec. 2007.
8. N. Liu and H. Wang, "Classification of Transformed Face Images with Majority Voting," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Montréal, Canada, Oct. 2007.

9. N. Liu and H. Wang, "Extraction of Hybrid Trace Features with Evolutionary Computation for Face Recognition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, Singapore, Sep. 2007.
10. N. Liu, H. Wang, and W. Y. Yau, "Face Recognition with Weighted Kernel Principal Component Analysis," in *Proceedings of International Conference on Control, Automation, Robotics, and Vision (ICARCV)*, Singapore, Dec. 2006.
11. N. Liu and H. Wang, "Feature Extraction with Genetic Algorithms Based Nonlinear Principal Component Analysis for Face Recognition," in *Proceedings of IAPR International Conference on Pattern Recognition (ICPR)*, Hong Kong, Aug. 2006.
12. N. Liu and H. Wang, "Invariant Facial Feature under Pose Variation for Face Recognition," in *Proceedings of IEEE International Conference on Information, Communications and Signal Processing (ICICS)*, Bangkok, Thailand, Dec. 2005.
13. H. Wang and N. Liu, "Design and Recognition of a Ring Code for AGV Localization," in *Proceedings of Joint Conference of IEEE International Conference on Information, Communications and Signal Processing and IEEE Pacific-Rim Conference on Multimedia (ICICS-PCM)*, Singapore, Dec. 2003.

Bibliography

- [1] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 491–502, 2005.
- [2] A. K. Jain, R. P. W. Duin, and J. C. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, Jan. 2000.
- [3] M. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. Oxford: Academic Press, 2008.
- [4] A. R. Webb, *Statistical Pattern Recognition*, 2nd ed. New York: John Wiley & Sons, 2002.
- [5] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds. Amsterdam: North-Holland, 1982, vol. 2, pp. 835–855.
- [6] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks Theory and Applications*. New York: Wiley, 1996.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, 2001.
- [8] P. Howland, J. Wang, and H. Park, "Solving the small sample size problem in face recognition using generalized discriminant analysis," *Pattern Recognition*, vol. 39, pp. 277–287, 2006.
- [9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge University Press, 2004.

- [10] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [11] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, A. Smola, and K. R. Müller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 623–628, May 2003.
- [12] M. J. Er, W. L. Chen, and S. Q. Wu, "High-speed face recognition based on discrete cosine transform and rbf neural networks," *IEEE Transactions on Neural Networks*, vol. 16, pp. 679–691, Mar. 2005.
- [13] J. H. Lai, P. C. Yuen, and G. C. Feng, "Face recognition using holistic fourier invariant features," *Pattern Recognition*, vol. 34, pp. 95–109, 2001.
- [14] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 811–828, Aug. 2001.
- [15] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, pp. 21–45, 2006.
- [16] S. A. Fahmy, C. S. Bouganis, P. Y. K. Cheung, and W. Luk, "Real-time hardware acceleration of the trace transform," *Journal of Real-Time Image Processing*, vol. 2, pp. 235–248, 2007.
- [17] S. Srisuk, M. Petrou, W. Kurutach, and A. Kadyrov, "A face authentication system using the trace transform," *Pattern Analysis and Applications*, vol. 8, pp. 50–61, 2005.
- [18] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.
- [19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, Nov. 1986.
- [20] S. M. Smith and J. M. Brady, "Susan-a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, pp. 45–78, 1997.
- [21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

- [22] I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 1089–1094, Oct. 1999.
- [23] L. Xu, "A unified perspective and new results on rht computing, mixture based learning, and multi-learner based problem solving," *Pattern Recognition*, vol. 40, pp. 2129–2153, 2007.
- [24] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.
- [25] F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig, "Principal component analysis in ecg signal processing," *EURASIP Journal on Advances in Signal Processing*, 2007.
- [26] X. Chen, L. Wang, J. D. Smith, and B. Zhang, "Supervised principal component analysis for gene set enrichment of microarray data with continuous or survival outcomes," *Bioinformatics*, vol. 24, pp. 2474–2481, 2008.
- [27] D. Skočaj, A. Leonardis, and H. Bischof, "Weighted and robust learning of subspace representations," *Pattern Recognition*, vol. 40, pp. 1556–1569, 2007.
- [28] F. D. L. Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.
- [29] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional pca: A new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 131–137, 2004.
- [30] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 696–710, 1997.
- [31] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [32] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

- [33] K. V. Kumar and A. Negi, "Subxpca and a generalized feature partitioning approach to principal component analysis," *Pattern Recognition*, vol. 41, pp. 1398–1409, 2008.
- [34] S. Chen and Y. Zhu, "Sub-pattern based principal component analysis," *Pattern Recognition*, vol. 37, pp. 1081–1083, 2004.
- [35] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1945–1959, Dec. 2005.
- [36] R. Saegusa, H. Sakano, and S. Hashimoto, "Nonlinear principal component analysis to preserve the order of principal components," *Neurocomputing*, vol. 61, pp. 57–70, 2004.
- [37] H. G. Hiden, M. J. Willis, M. T. Tham, and G. A. Montague, "Non-linear principal components analysis using genetic programming," *Computers & Chemical Engineering*, vol. 23, pp. 413–425, 1999.
- [38] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, July 1997.
- [39] M. L. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1274–1286, Aug. 2006.
- [40] E. K. Tang, P. N. Suganthan, X. Yao, and A. K. Qin, "Linear dimensionality reduction using relevance weighted lda," *Pattern Recognition*, vol. 38, pp. 485–493, 2005.
- [41] J. Liu, S. C. Chen, and X. Y. Tan, "A study on three linear discriminant analysis based methods in small sample size problem," *Pattern Recognition*, vol. 41, pp. 102–116, 2008.
- [42] W. Zhao, R. Chellappa, and P. J. Phillips, "Subspace linear discriminant analysis for face recognition," University of Maryland, MD, Technical Report CS-TR-4009, 1999.
- [43] W. Liu, Y. H. Wang, S. Z. Li, and T. N. Tan, "Null space approach of fisher discriminant analysis for face recognition," in *Proceedings of Biometric Authentication Workshop*, 2004, pp. 32–44.

- [44] X. G. Wang and X. O. Tang, "Dual-space linear discriminant analysis for face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 564–569.
- [45] W. S. Zheng, J. H. Lai, and P. C. Yuen, "Ga-fisher: A new lda-based face recognition algorithm with selection of principal components," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 35, pp. 1065–1078, Oct. 2005.
- [46] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385–2404, 2000.
- [47] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 762–766, 2001.
- [48] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
- [49] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2003.
- [50] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
- [51] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.
- [52] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," Stanford University, CA, Technical Report TR-2003-08, 2003.
- [53] B. Li, C. H. Zheng, and D. S. Huang, "Locally linear discriminant embedding: An efficient method for face recognition," *Pattern Recognition*, vol. 41, pp. 3813–3821, 2008.
- [54] X. L. Li, S. Lin, S. C. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 38, pp. 342–352, 2008.

- [55] N. Mekuz, C. Bauckhage, and J. K. Tsotsos, "Subspace manifold learning with sample weights," *Image and Vision Computing*, vol. 27, pp. 80–86, 2009.
- [56] O. Kouropteva, O. Okun, and M. Pietikäinen, "Incremental locally linear embedding," *Pattern Recognition*, vol. 38, pp. 1764–1767, 2005.
- [57] H. Chang and D. Y. Yeung, "Robust locally linear embedding," *Pattern Recognition*, vol. 39, pp. 1053–1065, 2006.
- [58] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1450–1464, 2002.
- [59] K. C. Kwak and W. Pedrycz, "Face recognition using an enhanced independent component analysis approach," *IEEE Transactions on Neural Networks*, vol. 18, pp. 530–541, 2007.
- [60] C. J. Liu and H. Wechsler, "Independent component analysis of gabor features for face recognition," *IEEE Transactions on Neural Networks*, vol. 14, pp. 919–928, 2003.
- [61] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.
- [62] X. M. Huo, "A statistical analysis of fukunaga-koontz transform," *IEEE Signal Processing Letters*, vol. 11, pp. 123–126, Feb. 2004.
- [63] S. Zhang and T. Sim, "When fisher meets fukunaga-koontz: A new look at linear discriminants," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 323–329.
- [64] Y. H. Li and M. Savvides, "Kernel fukunaga-koontz transform subspaces for enhanced face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [65] S. Zhang and T. Sim, "Discriminant subspace analysis: A fukunaga-koontz approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1732–1745, Oct. 2007.
- [66] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 570–582, 2000.

- [67] B. Li, D. S. Huang, C. Wang, and K. H. Liu, "Feature extraction using constrained maximum variance mapping," *Pattern Recognition*, vol. 41, pp. 3287–3294, 2008.
- [68] S. C. Yan, D. Xu, Q. Yang, L. Zhang, X. O. Tang, and H. J. Zhang, "Discriminant analysis with tensor representation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 526–532.
- [69] X. Y. Jing and D. Zhang, "A face and palmprint recognition approach based on discriminant dct feature extraction," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 34, pp. 2405–2415, Dec. 2004.
- [70] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *International Journal of Computer Vision*, vol. 43, pp. 167–188, 2001.
- [71] L. Shen, L. Bai, and M. Fairhurst, "Gabor wavelets and general discriminant analysis for face identification and verification," *Image and Vision Computing*, vol. 25, pp. 553–563, 2007.
- [72] D. V. Jadhav and R. S. Holambe, "Radon and discrete cosine transforms based feature extraction and dimensionality reduction approach for face recognition," *Signal Processing*, vol. 88, pp. 2604–2609, 2008.
- [73] G. Shakhnarovich, T. Darrell, and P. Indyk, Eds., *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. Cambridge, MA: The MIT Press, 2006.
- [74] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 164–171, 2000.
- [75] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
- [76] J. R. Quinlan, *C4.5: Programs for Machine Learning*. CA: Morgan Kaufmann, 1993.
- [77] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: The MIT Press, 2002.

- [78] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
- [79] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [80] P. J. Phillips, "Support vector machines applied to face recognition," NISTIR, MA, Technical Report 6241, 1999.
- [81] G. D. Guo, S. Z. Li, and K. L. Chan, "Support vector machines for face recognition," *Image and Vision Computing*, vol. 19, pp. 631–638, 2001.
- [82] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas, "Support vector machines for face authentication," *Image and Vision Computing*, vol. 20, pp. 369–375, 2002.
- [83] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, pp. 906–914, 2000.
- [84] C. C. Chang and C. J. Lin. (2001) Libsvm: A library for support vector machines. Department of Computer Science, National Taiwan University. Taipei, Taiwan. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [85] A. K. Jain, J. C. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, pp. 31–44, 1996.
- [86] Q. Y. Zhu, A. K. Qin, P. N. Suganthan, and G. B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, pp. 1759–1763, 2005.
- [87] H. J. Rong, Y. S. Ong, A. H. Tan, and Z. X. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, pp. 359–366, 2008.
- [88] R. X. Zhang, G. B. Huang, N. Sundararajan, and P. Saratchandran, "Multicategory classification using extreme learning machine for microarray gene expression cancer diagnosis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, pp. 485–495, 2007.
- [89] L. I. Kuncheva, *Combining Pattern Classifiers, Methods and Algorithms*. New York: Wiley Interscience, 2005.

- [90] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.
- [91] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [92] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990.
- [93] R. Polikar, "Bootstrap-inspired techniques in computation intelligence," *IEEE Signal Processing Magazine*, vol. 24, pp. 59–72, 2007.
- [94] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.
- [95] A. Kadyrov and M. Petrou, "Affine parameter estimation from the trace transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1631–1645, 2006.
- [96] M. Petrou and A. Kadyrov, "Affine invariant features from the trace transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 30–44, Jan. 2004.
- [97] A. Kadyrov and M. Petrou, "Object signatures invariant to affine distortions derived from the trace transform," *Image and Vision Computing*, vol. 21, pp. 1135–1143, 2003.
- [98] S. A. Fahmy, "Investigating trace transform architectures for face authentication," in *Proceedings of International Conference on Field Programmable Logic and Applications*, Aug. 2006.
- [99] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [100] F. S. Samaria and A. C. Harter, "Parameterization of a stochastic model for human face identification," in *Proceedings of The Second IEEE Workshop on Applications of Computer Vision*, Florida, Dec. 1994, pp. 138–142.
- [101] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, Eds., 1998, vol. 163, pp. 446–456.

- [102] R. Salomon, "The influence of different coding schemes on the computational complexity of genetic algorithms in function optimization," in *Proceedings of Parallel Problem Solving from Nature*, Berlin, Germany, Sept. 1996, pp. 227–235.
- [103] L. Wang, "Feature selection with kernel class separability," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1534–1546, Sept. 2008.
- [104] S. Petridis and S. J. Perantonis, "On the relation between discriminant analysis and mutual information for supervised linear feature extraction," *Pattern Recognition*, vol. 37, pp. 857–874, 2004.
- [105] A. Sierra and A. Echeverría, "Evolutionary discriminant analysis," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 81–92, 2006.
- [106] L. Nanni and A. Lumini, "Evolved feature weighting for random subspace classifier," *IEEE Transactions on Neural Networks*, vol. 19, pp. 363–366, Feb. 2008.
- [107] Z. H. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection," *Pattern Recognition*, vol. 37, pp. 2165–2176, 2004.
- [108] X. Wang and H. Wang, "Classification by evolutionary ensembles," *Pattern Recognition*, vol. 39, pp. 595–607, 2006.
- [109] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intelligent Systems*, vol. 13, pp. 44–49, 1998.
- [110] D. Masip and J. Vitrià, "Shared feature extraction for nearest neighbor face recognition," *IEEE Transactions on Neural Networks*, vol. 19, pp. 586–595, Apr. 2008.
- [111] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*. New York: Springer, 2005.
- [112] A. Asuncion and D. J. Newman. (2007) Uci machine learning repository. School of Information and Computer Science, University of California, Irvine. California, USA. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [113] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Columbia University, New York, NY, Technical Report CUCS-005-96, 1996.

-
- [114] D. Ramasubramanian and Y. V. Venkatesh, "Encoding and recognition of faces based on the human visual model and dct," *Pattern Recognition*, vol. 34, pp. 2447–2458, 2001.
 - [115] L. Chen, H. Man, and A. V. Nefian, "Face recognition based on multi-class mapping of fisher scores," *Pattern Recognition*, vol. 38, pp. 799–811, 2005.
 - [116] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 228–233, Feb. 2001.
 - [117] D. H. Lin and X. O. Tang, "Recognize high resolution faces: From macrocosm to microcosm," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
 - [118] H. Yu and J. Yang, "A direct lda algorithm for high-dimensional data - with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.
 - [119] M. Li and B. Yuan, "2d-lda: A statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters*, vol. 26, pp. 527–532, 2005.
 - [120] Z. Liang, Y. Li, and P. Shi, "A note on two-dimensional linear discriminant analysis," *Pattern Recognition Letters*, vol. 29, pp. 2122–2128, 2008.
 - [121] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
 - [122] G. B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, pp. 879–892, 2006.
 - [123] G. B. Huang, M. B. Li, L. Chen, and C. K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, pp. 576–583, 2008.
 - [124] D. Serre, *Matrices: Theory and Applications*. New York: Springer, 2002.