Feature Selection in Image Databases

Mahdi Yektaii

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

July 2013

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Mahdi Yektaii**

Entitled: **Feature Selection in Image Databases**

and submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. S. Rakheja	Chair
Dr. O.A. Basir	External Examiner
Dr. W. F. Xie	External to Program
Dr. A. Ben-Hamza	Examiner
Dr. M.N.S. Swamy	Examiner
Dr. M.O. Ahmad	Thesis Co-Supervisor
Dr. P. Bhattacharya	Thesis Co-Supervisor

Approved by _____
Dr. A.R. Sebak, Graduate Program Director

August 22, 2013

Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

ABSTRACT

Feature Selection in Image Databases

Mahdi Yektaii, Ph.D.
Concordia University, 2013

Even though the problem of determining the number of features required to provide an acceptable classification performance has been a topic of interest to the researchers in the pattern recognition community for a few decades, a formal method for solving this problem still does not exist. For instance, the well-known dimensionality reduction method of principal component analysis (PCA) sorts the features it generates in the order of their importance, but it does not provide a mechanism for determining the number of sorted features that need to be retained for a meaningful classification. Discrete wavelet transform (DWT) is another linear transformation used for data compaction, in which the coefficients in the transform domain can be sorted in different orders depending on their importance. However, the question of determining the number of features to be retained for a good classification of the data remains unanswered.

The objective of this study is to develop schemes for determining the number of features in the PCA and DWT domains that are sufficient for a classifier to provide a maximum possible classifiability of the samples in these transform domains. The energy content of the DWT and PCA coefficients of practical signals follow a specific pattern. The proposed schemes, by exploiting this property of the signals, develop criteria that are based on maintaining the energy of the ensemble of the feature vectors as their dimensionality is reduced. Within this unifying theme, in this thesis, the problem of dimension reduction is investigated when the features are generated by the linear transformation techniques of the discrete wavelet transform and the principal

component analysis, and by the nonlinear technique of kernel principal component analysis.

The first part of this study is concerned with developing a criterion for determining the number of coefficients when the features are represented as wavelet coefficients. The reduction in the dimensionality of the feature vectors is performed by letting the matrices of the wavelet coefficients of the data samples to undergo the process of Morton scanning and choosing a set of a fixed number of coefficients from these matrices whose energy content approaches to that of the original set of all the samples.

In the second part of the thesis, the problem of determining a reduced dimensionality of feature vectors is investigated when the features are PCA generated. The proposed method of finding a reduced dimensionality of feature vectors is based on evaluating a cumulative distance between all the pairs of distinct clusters with a reduced set of features and examining its proximity to the distance when all the features are included.

The PCA methods for data classification work well when the distinct clusters are linearly separable. For clusters that are nonlinearly separable, the kernel versions of PCA (KPCA) prove to be more efficient for generating features. The method developed in the second part of this thesis for obtaining the reduced dimensionality of the PCA based feature vectors cannot be readily extended to the kernel space because of the lack of availability of the feature vectors in an explicit form in this space. Therefore, the third part of this study develops a suitable criterion for obtaining reduced dimensionality of the feature vectors when they are generated by a kernel PCA.

Extensive experiments are performed on a series of image databases to demonstrate the effectiveness of the criteria developed in this study for predicting the number of features to be retained. It is shown that there is a direct correlation between the expressions developed for the criteria and the classification accuracy as functions of

the number of features retained. The results of the experiments show that with the use of the three feature selection techniques, a classifier can provide its maximum classifiability, that is, a classifiability attained by the uncompressed feature vectors, with only a small fraction of the original features. The robustness of the proposed methods is also investigated by applying them to noise-corrupted images.

To my father who always loved to learn and did his job with passion

ACKNOWLEDGEMENTS

Although this thesis bears the name of one author, this work would not have been possible without the support of numerous people. My family and my supervisors have been very supportive during the development of this study. I would like to especially thank Professor M. Omair Ahmad whose close supervision and guidance have played a very important role in the formation of this thesis. He has also kindly spent a significant amount of time for carefully reading the manuscript of the thesis and giving constructive suggestions and feedback. His support was crucial in the completion of this study. The early discussions with Professor Prabir Bhattacharya were the main source of ideas developed in this study. His persistence and encouragement since the beginning of this study have been encouraging. I feel obliged to appreciate Professor A. Ben Hamza for his useful comments during the proposal and seminar presentations. The emotional support of my mother, my sisters and my brother has been a blessing to me. I am indebted forever to them for their love, support and care. Several of my friends also helped me during the course of this study; their kindness is truly appreciated. I also would like to express my sincere thanks to the various sources from where I downloaded the data sets and used them in the experiments of this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C_p	Ratio of cumulative sum and sum of eigenvalues
$E_{\boldsymbol{w}}$	Weighted partial energy
J	Number of additional features included to verify saturation of criteria
$\boldsymbol{K}$	Kernel matrix
L	Lowest number of features for a maximum classification accuracy
l	Number of retained features
M	Number of data vectors in training set
N	Original dimensionality of data or feature vectors
$\boldsymbol{S}, \bar{\boldsymbol{S}}$	Covariance matrices
$\Re$	Set of real numbers
S	Length, in number of pixels, of a square image
s	Maximum possible number of DWT stages for an image
α, β, γ	Small numbers used for investigating the saturation of criteria
η	Cumulative global sample scattering distance
$\boldsymbol{\Phi}$	Transformation matrix employing a reduced set of eigenvectors
ϕ	Nonlinear function for mapping of samples to a kernel space
λ	Eigenvalue of a covariance matrix
$\boldsymbol{\mu}$	Mean vector of samples in a cluster, distribution or database
ζ	Cumulative global mean distance

LIST OF ACRONYMS

BGD	Bhattacharyya global distance
CGMD	Cumulative global mean distance
CGSSD	Cumulative global sample scattering distance
DCT	Discrete cosine transform
DWT	Discrete wavelet transform
KPCA	Kernel principal component analysis
MGD	Mahalanobis global distance
PCA	Principal component analysis
SNR	Signal-to-noise ratio

Chapter 1

Introduction

Advances in the collection and storage of data in recent years have led to a phenomenon known as *data explosion*. To cope with the difficulties associated with high-dimensional data, it is necessary to reduce their dimensionality. The main purposes of dimensionality reduction (DR) are the following:

- To compress the data in order to reduce the transmission and storage requirements;
- To facilitate the process of learning or recognition in pattern classification applications.

In pattern classification problems, there are a huge number of features[1] from which those that are most relevant to a specific classification problem need to be retained. Even though the rest of the features may be useful in other applications, they are discarded as irrelevant for the purpose of classification. The process of discarding irrelevant features is known as *feature subset selection*, *feature reduction*, *dimensionality reduction* and sometimes *data compression* [1–4]. In this regard, one can benefit from the existing compression techniques such as those used in the domain

[1]By definition, a feature vector is a set of numbers that represents uniquely an object within a database of objects.

of telecommunication, even though feature selection is not always a motivation in compression. The goal of compression, for example, could be to maintain a certain level of closeness between the original and the reconstructed version of the data [5].

Reducing the number of features also helps the designer of a pattern recognition algorithm to avoid a phenomenon known as *curse of dimensionality* [6–8]. This phenomenon occurs, since the data in high-dimensional spaces are extremely sparse, and designing a system that can function properly given all the possible points in that high-dimensional space becomes extremely difficult, if not impossible. Feature reduction, especially when the number of training vectors is considerably less than their original dimensionality, reduces the risk of over-adaptation in classifiers [9]. Finally, working with shorter data vectors is computationally less expensive. Retaining a minimum number of features becomes especially important since the speed of the matching algorithm depends on the dimension of the data vectors stored.

Dimensionality reduction is sometimes referred to as manifold learning [10, 11] or finding the intrinsic dimensionality [12], since the original high-dimensional data have a structure that could be viewed as a manifold in the original space [13]. In the case of pattern classification, the different classes of the data could be on a single manifold or on separate manifolds in the form of different clusters. Reducing the dimensionality of the data vectors, in this case, means finding the manifold(s) on which the data vectors reside, since each manifold requires a few parameters for its full description.

Some of the feature generation techniques can be categorized as dimensionality reduction methods. Among these techniques are the linear principal component analysis (PCA) [7] and its two-dimensional version [14], discrete cosine transform (DCT) [15] and discrete wavelet transform (DWT) [16] and its successor, discrete shearlet transform (DST) [17, 18]. The basic characteristic of these transformations is that they compact most of the energy of the practical signals in the first few coefficients they generate. Since in the original spatial or temporal domain, each of the huge number

of features has the same importance, this characteristic of the transformation plays an important role in feature subset selection. These feature generation methods are especially useful for template matching classifiers [19] in which all the features are treated equally.

The above feature generation techniques have yet another property: they automatically sort the generated features in a certain order based on their importance. In PCA, the most important features correspond to directions along which the data have the highest variation. In DWT, the coefficients in the low subbands carry most of the energy of the spatial-domain data. If the generated features are sorted based on their importance, the problem of feature subset selection is reduced to only determining the number of features to be retained.

Data reduction techniques are usually classified as either lossy or lossless. A lossless reduction is a reversible process, in that the original data can be retrieved exactly as they were before the reduction process. On the other hand, a lossy reduction is an irreversible process. Part of the data is permanently lost during the reduction. The majority of reduction techniques, including PCA, DCT and DWT are lossy transformations. The amount of compression depends on the application. If the application is to transfer some signals on a communication channel, they can be compressed up to a certain degree that guarantees a close to perfect reconstruction of the received data. However, if the compressed feature vectors are going to be classified, they may be compressed even more. The subject of this study is to determine the compression rate when the lossy reduction techniques of DWT and PCA are employed for generating the feature vectors for the purpose of classification.

1.1 Feature Subset Selection Schemes

This section briefly discusses the process of feature subset selection in a pattern classification system in relation to its other tasks. Regardless of the method applied to the task of feature selection, the goal of the feature selector is to provide the classifier with a set of features that result in the maximum possible classification efficiency of the classifier. The algorithms for feature selection produce a set of features that are either independent of the classifier or make use of the classification results in some fashion. These two approaches are illustrated in the pattern classification schemes of Figures 1.1 and 1.2 respectively. In the scheme of Figure 1.1, a set of reduced feature vectors is obtained from the entire set of features generated and fed directly to the classifier. In the classification scheme of Figure 1.2, the reduced number of the features and or their quality is refined based on the results of the classifier. A simple method of feature selection for this type of classification scheme is to increase the number of features l so that a classification accuracy close to the maximum possible accuracy is achieved. Another example of the second scheme is the one in which the set of the reduced features are translated or rotated based on the classification results in order to refine the quality of the features.

Figure 1.1: A simple classifier highlighting the feature selection block.

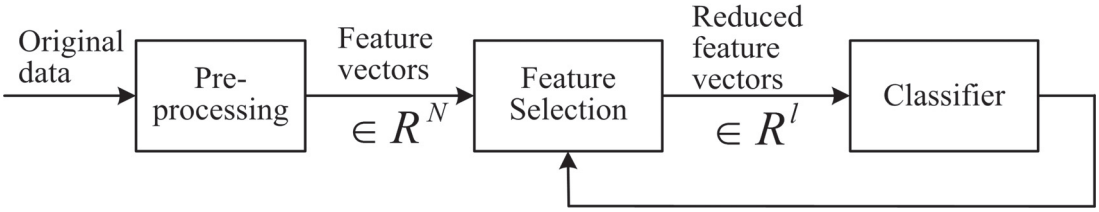


Figure 1.2: Feature selection sub-system using the results of the classifier.

In the scheme of Figure 1.1, it is possible to divide the subset selection task into serializing the raw features in terms of their importance and then determining the right number of features L as shown in Figure. 1.3. This approach is especially attractive in view of the fact that some of the linear feature generation techniques naturally arrange the components of the feature vectors in a certain order. For example, in the case of PCA, the generated features are arranged automatically in a decreasing order of their importance or in the case of DWT, the generated features are naturally arranged in some order that facilitates their serialization.

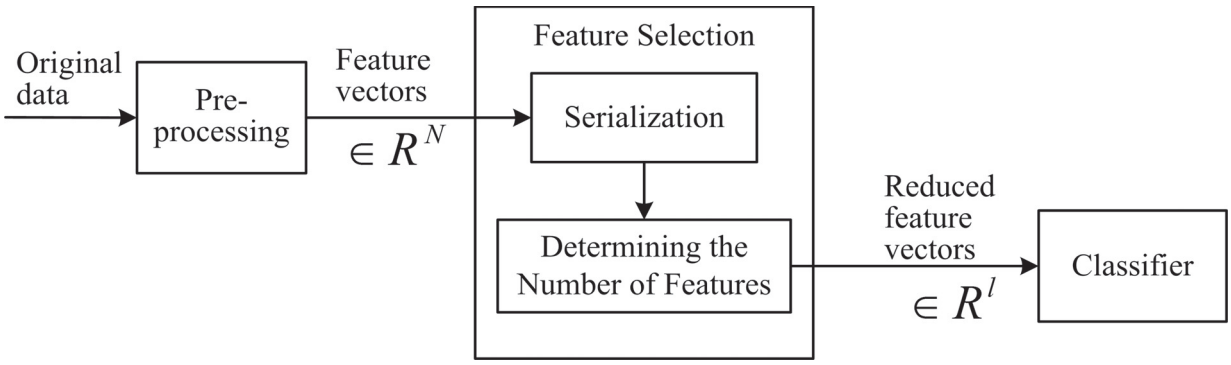


Figure 1.3: Feature selection divided into two sub-problems.

1.2 Motivation

A number of feature subset selection schemes employ the classification results for disregarding some of the features (Figure 1.2). The well-known method of recursive feature elimination [9] is one of these schemes. In this scheme, individual features from the original set of features are successively removed one by one until such a time when the remaining set of features still provides a satisfactory classification accuracy. At any time of this successive removal of features, a feature that affects the classification accuracy least is chosen to be removed. This method of subset selection results in a set of features that is most discriminatory. However, the use of this scheme results in a large computational complexity, since one does not know *a priori* as to how

many features can be removed, and, at any point during the successive elimination of features, it is not known as to which feature from the remaining set of features needs to be removed. Obviously, the method of recursive feature elimination to obtain the reduced set of features would be computationally inefficient in cases where the features are generated using DWT or PCA, where the generated features are already arranged in a certain order.

Classical distance measures such as Mahalanobis distance [7] between clusters, at first thought, may seem to be suitable for determining the reduced dimensionality, L, in the case of sorted features: one can compute the distance between all the pairs of distinct clusters with increasing dimensionality. The dimensionality at which the distance measure enters into its steady state can be regarded as the number of features to be retained. However, as it will be shown in Chapter 4, Mahalanobis distance measure does not work well for determining L in cases where the number of samples in classes are small, since the covariance matrix obtained from the set of feature vectors of a cluster becomes singular.

Another method of determining L is based on maximizing the ratio of the among-class-scatter and within-class-scatter [20]. In this approach, the number of features is increased until this ratio achieves its maximum. However, this approach fails if the number of samples in the database is less than the dimensionality of the original data vectors [7]. For example, the dimension of the original feature vectors in a face recognition database is in the order of 10,000, which is usually much larger than the number of samples available.

In the following, we now review the existing methods for determining the number of features that are generated specifically by a discrete wavelet transform as well as by the principal component analysis and its kernel version in view of the importance of these feature generation schemes in data compression applications.

1.2.1 Reduction of Features Generated by DWT

The *discrete wavelet transform* (DWT) has been used in standards such JPEG-2000 [4] for image compression. It has also been used for generating features for classification problems (see [21], for example). When DWT is applied to an image, a matrix of coefficients is generated. The serialization of the elements of this matrix results in a feature vector whose dimension is the number of elements in the coefficient matrix. Discrete wavelet transform compacts the energy of the image only in a few coefficients whose number is much smaller than the dimension of this vector. This is an attractive characteristic of this transform that can be employed in feature subset selection.

For determining the compression ratio, sometimes all the entries in the original feature vectors that have magnitudes less than a specific threshold are discarded. Depending on the threshold and the type of the wavelet used, in this method, a subset of features get selected whose energy corresponds to a percentage of the original energy of the image. This approach of obtaining a reduced set of features has been employed in a variety of applications including signal de-noising and compression [22]. Different methods for determining the threshold have been proposed. Donoho and Johnston [23] have given a general guideline for selecting the value of the threshold based on the statistical behavior of the noise present in the data. Since the noise variance is not known in advance, it is difficult to compute the threshold. Note that pattern classifiers, in general, compare the corresponding coefficients from all the reduced feature vectors. However, a threshold-based technique for obtaining the dimensionality-reduced feature vectors cannot guarantee that all these corresponding coefficients of the feature vectors are fetched from the same location of the transform matrices, which is essential for a good classification. Moreover, a threshold-based technique may generate feature vectors of different sizes for different samples, which is not commensurate to the functioning of a classifier. Rajoub [24] has given a method in which the DWT coefficients are sorted in terms of their magnitude and a number

of coefficients that contain 99.9% of the total energy of all the coefficients is retained. This method and in that matter, any other method of determining the number of features based on preserving a certain percentage of the energy of the original features would also suffer from the disadvantages mentioned just above.

Some researchers have developed tables giving classification accuracy corresponding to different compression ratios for features of the samples of a database using a specific wavelet [5]. Some others have developed such tables based on the peak signal-to-noise ratio (PSNR) metric of the reconstructed images [25]. These methods are useful, if a pre-computed table for the samples of the database that one is interested in and the type of the wavelet that one uses to generate features does exist.

1.2.2 Reduction of Features Generated by PCA

The *Principal Component Analysis* (PCA), also called the *Karhunen-Loève Transform*, the *Hotelling Transform* or factor analysis [26], is one of the most popular methods for dimension reduction [6, 7]. It is often used in the feature extraction phase of classification problems [27–29]. It is also the basis of generating the eigen-faces in face recognition [30]. PCA obtains the directions in the original space of the data along which the samples are dispersed most widely by performing an eigen analysis on the covariance matrix of the samples. The feature extraction is then the selection of the number of eigenvectors corresponding to the largest eigenvalues of the covariance matrix. In PCA, the dimension of the feature space after the reduction process is usually chosen in an ad-hoc manner [2, 31] as explained in the following.

An often-used procedure for determining the reduced dimensionality, L, is first to sort the eigenvalues of the covariance matrix in the decreasing order and then to select L such that the ratio of the sum of the first L eigenvalues and the sum of all the eigenvalues is greater than a certain threshold. This ratio is known as *cumulative percentage* [2]. For instance, [32] uses a threshold value of 0.95. Another

widely used method for determining L is to plot the sorted eigenvalues as a function of the eigenvalue number. This plot is known as *scree graph* [2]. Then, L is chosen to be the eigenvalue number at which the scree graph has steep slope to the left and non-steep slope to the right. This approach is subject to one's interpretation of steep and non-steep slopes, since there is no mathematical definition of a steep slope. There are a few other methods for determining L that assume a certain distribution of the data that may not necessarily be realistic. Cross-validation [33] and bootstrapping [34] are two other methods for determining the reduced dimensionality, but they are computationally intensive [2].

The above-mentioned methods for determining the reduced number of eigenvectors have been devised originally for data compression and not necessarily for pattern classification applications where the data samples reside in separate classes. For pattern classification problems, an objective method for determining the number of principal components does not exist. For example, the authors of the well-known method for face classification based on eigen-faces [30] retain the first 100 features without providing an objective justification for their choice. Silda et al. retain 20 features in their work for vehicle recognition [27] based on the experience. Benediktsson et al. [35] use four principal components of the hyper-spectral images for the purpose of classification. Melo et al. [28] reduce the number of principal components from 260 to 80 based on acceptable classification accuracy. Some researchers have provided a table or a graph with different reduction ratios and the corresponding classification performances [36]. In all these cases, the experience and the classification results play an important role in selecting the number of features to be retained.

1.2.3 Reduction of Features Generated by KPCA

The PCA method of generating features assumes that the data vectors reside in linearly separable clusters [6, 37]. Kernel-based methods have been introduced for

the classification problems in which the data samples are not linearly separable. The main idea in kernel and nonlinear methods for classification is to map the data vectors to a higher dimensional space by employing a non-linear kernel function so that, in the new space, the samples are linearly separable [38]. Therefore, in this space, referred to as *feature or kernel space*, it is possible to use PCA to reduce the dimensionality of the mapped samples. In general, a better classifiability in the kernel space is achieved at the expense of a larger number of features that have to be retained. The use of kernel and the increased number of features lead to a larger computational complexity.

Just as for the linear case, the existing techniques of kernel PCA for pattern classification have not focused on developing a formal method for reducing the number of features. For instance, the schemes in [38–40] have generated tables providing classification accuracies as the number of features retained is varied. Cao et al. in [29] have performed a classification experiment using all the possible number of features and then reported the number of features that resulted in the highest classification accuracy.

1.3 Problem Statement

The majority of the methods for determining the number of features that need to be retained in a classification problem are based on one's personal experience or on the results of classification experiments performed using the test samples within a database. Removing the features whose magnitude is less than a threshold, providing tables of different number of features and the corresponding classification accuracy, and choosing the number of features based on scree graph or cumulative percentage are examples of informal approaches that are generally used in classification problems.

The objective of this investigation is to develop formal criteria for selecting a set of reduced number of features for a homogenous database in which the sample

classes are linearly or nonlinearly separable. Specifically, this study is focused on cases in which features are generated through a discrete wavelet transform (DWT) and principal component analysis (PCA) in the linear case, and through a kernel principal component analysis (KPCA) in the nonlinear case. Criteria for obtaining a reduced set of features generated by DWT, PCA or KPCA are devised, that are based on preserving the energy of the original set of feature vectors. The validity of the criteria developed is examined by studying the correlation between the criteria devised and the classification performance.

1.4 Organization of the Thesis

Chapter 2 provides an overview of the discrete wavelet transform, principal component analysis and kernel principal component analysis that are used in this study for generating features. This chapter also introduces the notion of global distance metric as a measure of separability among the data clusters. Global distance measures are useful for devising the criteria for determining L, the number of features that need to be retained. Chapters 3 and 4 are devoted to developing the algorithms for determining L for the class of data sets in which the samples belonging to distinct clusters are linearly separable. For this class, the features are generated by the linear transformation techniques of discrete wavelet transform and principal component analysis. The algorithm in Chapter 3 is based on preserving the energy of the original feature vectors. The feature vectors to be retained are chosen through a process of Morton scanning of the wavelet coefficients. The algorithm in Chapter 4 is based on preserving the distance between the feature vectors of distinct clusters during the reduction process. Chapter 5 introduces a criterion for determining L in the case of non-linearly separable sample clusters with kernel principal component analysis used for generating features. In this chapter, the approach of Chapter 4 is essentially extended to a

kernel space using the so called kernel trick.

Chapters 3, 4 and 5 also present the results of extensive experiments performed for verifying the usefulness of the criteria proposed in these chapters for obtaining a reduced set of features. The evaluations of the criteria involve a classification of the test samples from each of the benchmark databases. In order to investigate the robustness of the proposed criteria, experiments are also performed with noise-contaminated versions of the samples. Chapter 6 concludes the thesis by highlighting the main findings of this study and suggesting a few problems related to this study for future investigation.

Chapter 2

Background Material

The discrete wavelet transform, principal component analysis and kernel principal component analysis are often employed for feature generation and data compression. In this chapter, we review these three transforms, since they are used in subsequent chapters for devising the criteria for determining the number of features to be retained for data classification. We also discuss several distance measures commonly used to measure the distance between two data points or two distributions. These measures are used in this thesis to determine the number of features to be retained for classifiability of data clusters.

2.1 Discrete Wavelet Transform

This section is a brief overview of the wavelet transform and its application to digital image compression. The material in this section is mainly based on a tutorial by Usevitch [16]. Figure 2.1 shows the generic form of a one-dimensional wavelet transform. The input signal is passed through a low-pass filter h and a high-pass filter g. The outputs from both filters undergo an operation of down sampling by a factor of two. The filtering and down sampling operations together form a single level of wavelet decomposition. In practice, multiple levels of wavelet transform are performed

on a digital sequence as in Figure 2.1. Note that the recursion is usually performed on the output of the low-pass filter. The resulting sequences on the high-pass side $d_{ih}(n), i = 1, 2, \cdots, s$ together with the output sequence of the last low-pass stage $d_{sl}(n)$ are called *wavelet coefficients* $w(n)$. Assuming the number of points in the original signal is 2^q, it could be easily verified that the number of wavelet coefficients is also 2^q. This conclusion is valid as long as the filtering process does not generate longer sequences than its input.

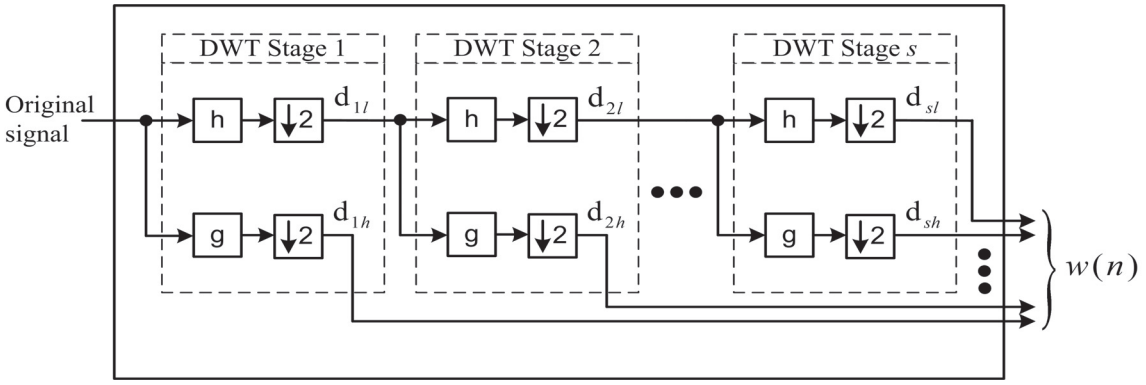


Figure 2.1: Digital signal decomposition using discrete wavelet transform.

The extension of the one dimensional transform to two dimensional transform is possible by using separable wavelet filters. With these filters, the two dimensional transform can be performed by first applying a one dimensional transform on all the rows and then repeating the transform on all the columns. The four subbands that result from one level of wavelet transform on a 2-D signal, are shown in Figure 2.2. The LL subband is the result of low-pass filtering on both rows and columns of the signal. The HL subband is the output of high-pass filtering on the rows followed by low-pass filtering on the columns. The LH subband is obtained by first applying the low-pass filter on rows and then the high-pass filter on the columns. Finally the HH subband is the result of high-pass filtering on rows and columns of the input signal. Similar to the one-dimensional case, the number of wavelet coefficients generated is the same as the number of pixels in the original image.

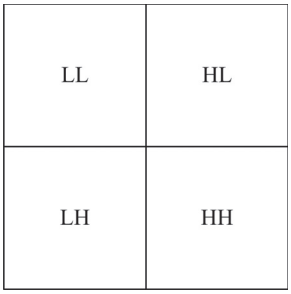


Figure 2.2: 2-D signal decomposition using discrete wavelet transform.

Figure 2.3 shows an example of applying one level of 2-D wavelet transform on a digital image chosen from the MIT CBCL [41] database. Figures 2.4 and 2.5 show, respectively, a two-level decomposition and a corresponding example on the same image as in Figure 2.3.

Figure 2.3: An example of applying one level of wavelet decomposition on an image from MIT CBCL database [41].

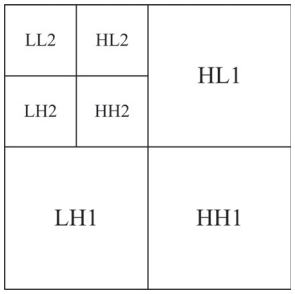


Figure 2.4: 2-D signal decomposition using two levels of DWT.

If the low-pass and high-pass filters of the wavelet transform satisfy the orthogonality conditions, then the transform will preserve the energy of the signal. The orthogonality conditions are defined as follows:

Figure 2.5: An example of a two-level DWT decomposition on a sample from CBCL database [41].

$$\begin{cases} \sum_n h(n-2i)h(n-2j) = \delta(i-j) \\ \sum_n g(n-2i)g(n-2j) = \delta(i-j) \\ \sum_n g(n-2i)h(n-2j) = 0 \end{cases} \tag{2.1}$$

where $\delta(\cdot)$ is the unit impulse function.

The preservation of the energy, similar to the Parseval's theorem in Fourier analysis, can be expressed as

$$\sum_{n=1}^{N} x^2(n) = \sum_{n=1}^{N} w^2(n) \tag{2.2}$$

where N is the length of the signal and its wavelet transform. The preservation of the energy makes it possible to perform the compression algorithm completely in the transform domain or wavelet domain in this case.

The simplest form of wavelet filters is the Haar filter. The low-pass and high-pass Haar filters are two-point filters defined as

$$h(0) = h(1) = 1; \quad g(0) = -g(1) = 1; \tag{2.3}$$

2.2 Principal Component Analysis

This section presents an overview on the principal component analysis as a technique for feature generation. More specifically, Section 2.2.1 presents the basics of this technique and Section 2.2.2 shows how it is possible to perform the eigen analysis when the number of data vectors is less than their original dimensionality. The material in Section 2.2.2 is especially important in the study of this thesis, since the dimensionality of the data samples in most of the image databases is very high.

2.2.1 Basics

The method of principal component analysis was introduced as one of the first compression techniques in statistical analysis [2]. The central theme of this technique is to reduce the dimensionality of a data set with a large number of components in each sample. For implementing this idea, an eigen analysis is performed on the covariance matrix of the data vectors comprising the data set.

Let $\boldsymbol{S}$ be the $N \times N$ covariance matrix of the training data defined by

$$\boldsymbol{S} = \frac{1}{M} \sum_{i=1}^{M} \left(\boldsymbol{x}^i - \boldsymbol{\mu}\right) \left(\boldsymbol{x}^i - \boldsymbol{\mu}\right)^T \tag{2.4}$$

where $\boldsymbol{x}^i \in \Re^N$ is the i^{th} training data point and $\boldsymbol{\mu}$ denotes the mean vector of the training points $\boldsymbol{x}^i, i = 1, 2, \cdots, M$, and M is the number of data points[1]. The dimension, N, of the data points should be reduced to $L(< N)$ (assuming that L is known). The eigenvectors $\boldsymbol{V}^i (1 \leq i \leq L)$ corresponding to the L largest eigenvalues of $\boldsymbol{S}$ are then employed for defining a linear transformation that reduces the data by transferring them to the space $\Re^L$. This linear transformation is given by the matrix

$$\boldsymbol{\Phi} = \left[\boldsymbol{V}^1 \vdots \boldsymbol{V}^2 \vdots \cdots \vdots \boldsymbol{V}^l\right]^T \tag{2.5}$$

[1]The unbiased sample covariance matrix is computed by dividing the summation in Eq. (2.4) by $M - 1$ rather than by M.

The sample point $\boldsymbol{x}^i$ of the input data is then transformed using $\boldsymbol{\Phi}$ as

$$\boldsymbol{y}^i = \boldsymbol{\Phi}\left(\boldsymbol{x}^i - \boldsymbol{\mu}\right) \tag{2.6}$$

where $\boldsymbol{y}^i \in \Re^L$ is the reduced feature vector. A block diagram form of this transformation is shown in Figure 2.6.

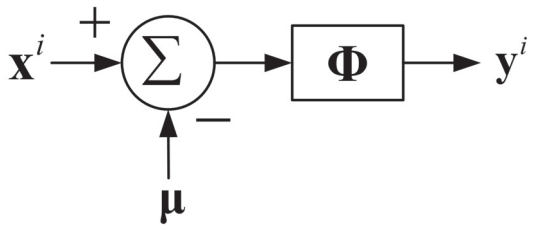


Figure 2.6: PCA transformation from $\Re^N$ to $\Re^L$ based on Eq. (2.6).

As discussed in Chapter 1, finding L in Eq. (2.5) is important. A large feature vector not only requires more computational resources, but also brings more noise to the system. In the following, we briefly review two of the well-known methods for determining L in the context of PCA.

Assume that the eigenvalues λ_i of $\boldsymbol{S}$ are sorted in descending order. There are several ad-hoc methods available for choosing L [2], [31]. A straightforward method is to plot λ_i versus i, called the *scree plot*, and search for a saturation or an "elbow" after which the slope of the curve reduces significantly. Another approach is to consider the ratio of the cumulative sum of the eigenvalues and the sum of all eigenvalues – this ratio is called the *cumulative percentage* given by

$$Cp(l) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \tag{2.7}$$

where M is the number of eigenvalues of the covariance matrix. The value of l at which $Cp(l)$ gets very close to unity, is chosen to be the number of eigenvectors to be retained. For instance, in [32] a value of L is chosen for which $Cp(l) = 0.95$.

2.2.2 Eigen Analysis in a High Dimensional Space

In some databases the dimensionality N of the data points is much higher than the number of vectors M. In this case, the rank of the $N \times N$ covariance matrix in Eq. (2.4) is less than N. Moreover, its huge size makes it difficult to perform the eigen analysis especially because the covariance matrices are not sparse. Fortunately, as explained below, there is a trick [42] to find the eigenvalues and eigenvectors more easily.

We begin by defining the matrix $\boldsymbol{X}$ as

$$\boldsymbol{X} = \left[\boldsymbol{x}^1 - \boldsymbol{\mu} \vdots \boldsymbol{x}^2 - \boldsymbol{\mu} \vdots \cdots \boldsymbol{x}^M - \boldsymbol{\mu}\right]_{N \times M} \tag{2.8}$$

We then introduce a new matrix as

$$\bar{\boldsymbol{S}} = \frac{1}{M}\boldsymbol{X}\boldsymbol{X}^T \in \Re^{N \times N} \tag{2.9}$$

$\bar{\boldsymbol{S}}$ is equal to the covariance matrix of Eq. (2.4), since

$$\bar{S}_{ij} = S_{ij} = \frac{1}{M}\sum_{k=1}^{M}\left(x_i^k - \mu_i)\right)\left(x_j^k - \mu_j)\right), \; i, j = 1, ..., N \tag{2.10}$$

The eigen analysis relationship states

$$\frac{1}{M}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{V}^k = \lambda_k \boldsymbol{V}^k \tag{2.11}$$

Multiplying the two sides of this equation by $\boldsymbol{X}^T$ and after some manipulation, we obtain

$$\left(\frac{1}{M}\boldsymbol{X}^T\boldsymbol{X}\right)\left(\boldsymbol{X}^T\boldsymbol{V}^k\right) = \lambda_k\left(\boldsymbol{X}^T\boldsymbol{V}^k\right) \tag{2.12}$$

This last equation states that $\boldsymbol{X}^T\boldsymbol{V}^k$ and λ_k are the corresponding eigenvector/ eigenvalue pair of a new matrix defined as

$$\boldsymbol{S^*} = \left(\frac{1}{M}\boldsymbol{X}^T\boldsymbol{X}\right), \boldsymbol{S^*} \in \Re^{M\times M} \tag{2.13}$$

The size of $\boldsymbol{S^*}$ is much less than that of the original covariance matrix $\bar{\boldsymbol{S}}$. In order to obtain the original eigenvectors $\boldsymbol{V}^k$ we just need to multiply the new eigenvectors by $\boldsymbol{X}$, since

$$\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^k\right) = \left(\boldsymbol{X}\boldsymbol{X}^T\right)\boldsymbol{V}^k = \left(M\bar{\boldsymbol{S}}\right)\boldsymbol{V}^k = M\lambda_k\boldsymbol{V}^k \tag{2.14}$$

which differs from the original eigenvectors $\boldsymbol{V}^k$ only by a multiplicative scalar factor $M\lambda_k$.

2.3 Kernel Principal Component Analysis

The underlying assumption in the PCA technique is that the data reside in linearly separable sub-spaces. Nonlinear kernel-based transformations have been introduced to relax this assumption of the linear methods [43, 44]. The idea is to first increase the dimensionality of the data vectors so that the vectors from different clusters of the database have no overlap or less overlap compared with that in their initial space. The new space is referred to as *kernel space* or *feature space*.

The high dimensionality of the kernel space is the main difficulty in handling directly the data vectors in that space. Therefore, the so called *kernel trick* is used in deriving the basic formulation of the kernel-based PCA. We now describe the basic formulation of KPCA making use of this trick.

Assume a set of centered (zero-mean) data vectors $\boldsymbol{x}^k \in \Re^N, k = 1, ..., M$. The principal component analysis is a linear transformation based on eigen analysis of the covariance matrix of the data points, given by

$$\boldsymbol{C} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{x}^i\left(\boldsymbol{x}^i\right)^T \tag{2.15}$$

where T denotes the transpose of the associated vector. The eigenvalues and eigenvectors of this matrix are obtained by solving the equation

$$\boldsymbol{C}\boldsymbol{v} = \lambda \boldsymbol{v}, \;\; \lambda \geq 0, \;\; \boldsymbol{v} \in \Re^N \setminus \boldsymbol{0} \tag{2.16}$$

Combining Eqs. (2.15) and (2.16) gives

$$\lambda \boldsymbol{v} = \frac{1}{M} \sum_{i=1}^{M} \langle \boldsymbol{x}^i, \boldsymbol{v} \rangle \boldsymbol{x}^i \tag{2.17}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot-product of the two associated vectors. This means that all solutions $\boldsymbol{v}$ with corresponding $\lambda \neq 0$ must lie in the span of the original data vectors, $\boldsymbol{x}^1, ..., \boldsymbol{x}^M$.

Now we describe the eigen analysis process in the kernel space assuming a possible nonlinear map ϕ given by

$$\phi : \Re^N \rightarrow F, \boldsymbol{x}^k \rightarrow \boldsymbol{X}^k \tag{2.18}$$

where $\boldsymbol{x}^k$ denotes the data elements in $\Re^N$ and $\boldsymbol{X}^k$ denotes those in F. Note that the feature space F may have an arbitrarily large, possibly infinite, dimensionality.

Before continuing the eigen analysis in kernel space, it needs to be mentioned that neither $\phi(\cdot)$ nor the explicit values of $\boldsymbol{X}^k$ are known. Hence, the kernel trick is employed to facilitate the formulation of feature generation in kernel space. The kernel trick states that even though the data samples $\boldsymbol{X}^k$ or the nonlinear mapping ϕ are unknown, the inner product of any pair of data vectors in kernel space can be evaluated using a *kernel function*

$$\left\langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j) \right\rangle = \left\langle \boldsymbol{X}^i, \boldsymbol{X}^j \right\rangle = f\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right) \tag{2.19}$$

where f is the kernel function and is easy to evaluate. The selection of kernel function is based on the database to which the KPCA technique is applied. For example, [44]

uses a polynomial kernel function for handwritten digit character recognition. To continue the eigen analysis in kernel space, it is again assumed that the $\boldsymbol{X}^k$ are centered, that is, $\sum_{i=1}^{M} \phi(\boldsymbol{x}^i) = \boldsymbol{0}$. Therefore, the covariance matrix in the feature space F is obtained as

$$\bar{\boldsymbol{C}} = \frac{1}{M} \sum_{i=1}^{M} \phi(\boldsymbol{x}^i) \left(\phi(\boldsymbol{x}^i)\right)^T \tag{2.20}$$

Next, we search for positive eigenvalues and eigenvectors $\boldsymbol{V} \in F \setminus \boldsymbol{0}$ satisfying the relation

$$\bar{\boldsymbol{C}}\boldsymbol{V} = \lambda \boldsymbol{V} \tag{2.21}$$

Similar to the linear PCA (Eq. (2.17)), we can express each eigenvector with corresponding nonzero eigenvalue as a linear combination of feature space vectors, as given by

$$\boldsymbol{V} = \sum_{i=1}^{M} \alpha_i \phi(\boldsymbol{x}^i) \tag{2.22}$$

Knowing from Eq. (2.21) that $\lambda\langle\phi(\boldsymbol{x}^k), \boldsymbol{V}\rangle = \langle\phi(\boldsymbol{x}^k), \bar{\boldsymbol{C}}\boldsymbol{V}\rangle$ for k = 1, ..., M and using Eqs. (2.20) and (2.22), we have

$$\lambda \sum_{i=1}^{M} \alpha_i \langle\phi(\boldsymbol{x}^k), \phi(\boldsymbol{x}^i)\rangle = \frac{1}{M} \left\langle \sum_{i=1}^{M} \alpha_i \phi(\boldsymbol{x}^k), \sum_{j=1}^{M} \phi(\boldsymbol{x}^j) \left\langle \phi(\boldsymbol{x}^j), \phi(\boldsymbol{x}^i) \right\rangle \right\rangle, \; k = 1, ..., M \tag{2.23}$$

Having an $M \times M$ matrix $\boldsymbol{K}$ whose elements are defined as

$$K_{ij} = \langle\phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j)\rangle = f(\boldsymbol{x}^i, \boldsymbol{x}^j) \tag{2.24}$$

we can re-write Eq. (2.23) as

$$M\lambda \boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{K}^2\boldsymbol{\alpha} \tag{2.25}$$

where $\boldsymbol{\alpha}$ is the column vector of the entries $\alpha_1, ..., \alpha_M$. It is shown in [44] that solving Eq. (2.25) is equivalent to solving

$$M\lambda\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha} \tag{2.26}$$

Eq. (2.26) is, in fact, an eigen analysis equation of the matrix $\boldsymbol{K}$. This matrix is called the *kernel matrix* or the *feature matrix*. Let $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_M$ denote the eigenvalues of $\boldsymbol{K}$ and $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, ..., \boldsymbol{\alpha}^M$ the corresponding eigenvectors. The eigenvectors in the feature space are not explicitly available, but their normalization constraint leads to the following

$$\begin{aligned} \langle \boldsymbol{V}^k, \boldsymbol{V}^k \rangle = 1 = \sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_i^k \alpha_j^k \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j) \rangle \\ = \sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_i^k \alpha_j^k K_{ij} = \langle \boldsymbol{\alpha}^k, \boldsymbol{K}\boldsymbol{\alpha}^k \rangle = M\lambda_k \langle \boldsymbol{\alpha}^k, \boldsymbol{\alpha}^k \rangle \end{aligned} \tag{2.27}$$

where α_i^k denotes the i^{th} element in the k^{th} eigenvector of $\boldsymbol{K}$. To obtain the principal components or the so called features of a given vector $\boldsymbol{x}$, we need to compute the projections of $\phi(\boldsymbol{x})$ along the eigenvectors $\boldsymbol{V}^k$ in F, as given by

$$\langle \boldsymbol{V}^k, \phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{M} \alpha_i^k \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}) \rangle \tag{2.28}$$

For the sake of simplicity, it has been assumed that the data points are centered, meaning that $\sum_{i=1}^{M} \phi(\boldsymbol{x}^i) = \boldsymbol{0}$. However, the formulation of this section can be generalized for the case of non-centered data with the resulting equations being slightly different from those in the above development [43, 44].

2.4 Distance Measures

The distance measures may be useful tools for determining the number of features to be retained in a classification system. Usually, the distance measures are defined

between two vectors or between two distributions. In this section, we will review some of the distance measures and introduce the notion of global distance for the cases involving more than two vectors or distributions.

The *Euclidean* distance is the most commonly used distance measure between two vectors. For two vectors $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$, it is defined as

$$D = \|\boldsymbol{V}^1 - \boldsymbol{V}^2\| = \sqrt{\left\langle (\boldsymbol{V}^1 - \boldsymbol{V}^2), (\boldsymbol{V}^1 - \boldsymbol{V}^2) \right\rangle} \tag{2.29}$$

The *Manhattan* distance between two vectors is defined as the distance that should be traveled from one vector to the other provided that the path is traversed along the coordinate axes. In other words it is defined as

$$D = \sum_{i=1}^{p} \left| v_i^1 - v_i^2 \right| \tag{2.30}$$

in that p is the dimensionality of vectors $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$ with their elements represented by v_i^1 and v_i^2, respectively.

The Euclidean and Manhattan distances have been defined for two vectors. Sometimes it is necessary to measure the distance between two populations. Mahalanobis and Bhattacharyya distances have been introduced to measure the distance between two random distributions that are completely described with the mean and covariance information. For two random distributions D_1 and D_2 with means $\boldsymbol{\mu}^1$ and $\boldsymbol{\mu}^2$ and covariance matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, Mahalanobis distance is defined as [45]

$$MD(D_1, D_2) = \left(\boldsymbol{\mu}^1 - \boldsymbol{\mu}^2\right)^T \left(\frac{\boldsymbol{S}_1 + \boldsymbol{S}_2}{2}\right)^{-1} \left(\boldsymbol{\mu}^1 - \boldsymbol{\mu}^2\right) \tag{2.31}$$

The Mahalanobis distance is, in fact, a weighted version of the Euclidean distance between the means of the two distributions. The weighting factor is inversely proportional to the variance of both of the distributions. It means that the distance is more if the variance of the two distributions are not large. In comparison with the Mahalanobis distance, the Bhattacharyya distance has an additional term. This distance is given by [45]

$$BD(D_1, D_2) = \frac{1}{8} MD(D_1, D_2) + \frac{1}{2} \ln \left(\frac{\left| \frac{\boldsymbol{S}_1 + \boldsymbol{S}_2}{2} \right|}{\sqrt{|\boldsymbol{S}_1||\boldsymbol{S}_2|}} \right) \tag{2.32}$$

The Mahalanobis and Bhattacharyya distance measures are useful in applications requiring a measure of distance between two data clusters [7, 8, 46, 47]. However, pattern classification applications usually involve more than two clusters in a database. Therefore, global versions of these two distance measures need to be defined. *Mahalanobis global distance* (MGD) for a collection of c classes is defined as the sum of Mahalanobis distance between all pairs of classes:

$$MGD = \sum_{i=2}^{c} \sum_{j=1}^{i-1} MD(D_i, D_j) \tag{2.33}$$

where D_i and D_j are the i^{th} and j^{th} clusters in the database. *Bhattacharyya global distance* (BGD) is similarly defined as

$$BGD = \sum_{i=2}^{c} \sum_{j=1}^{i-1} BD(i, j) \tag{2.34}$$

2.5 Summary

In this chapter, we have provided a brief review of three popular feature generation methods in the field of pattern classification: the discrete wavelet transform (DWT), principal component analysis (PCA) and kernel principal component analysis (KPCA). We have also discussed several distance measures between a pair of vectors or between a pair of distributions. We have described the notion of global distances that can be used to measure the overall distance among a set of distributions. The material presented in this chapter constitutes the basis for the work in this thesis for developing the criteria for determining the number of features that need to be retained for classifiability of data clusters when DWT, PCA and KPCA are used for generating features.

Chapter 3

Feature Selection in DWT Domain

Discrete wavelet transform (DWT) is an important transformation that is used in various applications such as image de-noising and compression. It can also be employed in generating feature vectors for pattern classification problems. For pattern classification applications, it is important to know the number of features that need to be retained in order to provide a classifiability that is close to that obtained by using all the original features. The objective of this chapter is to determine the number of the DWT-generated coefficients that need to be retained for a satisfactory classification for a given database. To achieve this goal, a criterion called weighted partial energy is introduced that is computed using the original feature vectors and a weighting vector [48]. The elements of the wavelet coefficient matrix of each image are serialized by the Morton scanning [49], appropriately weighted and then used to compute the weighted partial energy in order to determine the number of features[1] to be retained for a successful classification of the images of the database. Using a number of different databases, it is shown that weighted partial energy can be efficiently used to determine a minimum number of features that could provide a classification accuracy very close to that of using all the features.

This chapter is organized as follows. Section 3.1 explores the relationship between

[1]In this chapter, "features" and "DWT coefficients" are used interchangeably.

the classifiability of the reduced DWT feature vectors and the ratio of their energy to the total energy of all the images in the training set. Since reducing the dimensionality of the feature vectors is equivalent to removing the least important features, Section 3.2 discusses several different sorting schemes of the DWT coefficient matrices into 1-D feature vectors. Section 3.3 introduces a criterion for determining the number of features to be retained from the entire set of DWT coefficients of an image database. Section 3.4 presents the final algorithm for obtaining the reduced dimensionality along with a discussion on the choice of the parameters used in the algorithm. Section 3.5 describes several experiments that are performed on different databases to verify the applicability of the proposed criterion. In Section 3.6, the results of using different wavelets, sequencing approaches and weighting schemes are discussed and compared. The experiments for studying the effect of noise on the proposed criterion are presented in Section 3.7. Section 3.8 concludes the chapter.

3.1 Classifiability of Wavelet Compressed Images

Classification methods based on template matching [19], are dependent on the distance between the data vectors in distinct classes. When the feature vectors in the database are reduced by removing some of the features, the distance between the vectors, in general, gets reduced. Therefore, it is possible for the feature vectors belonging to different clusters to become less and less recognizable one from another, especially when these feature vectors are produced directly from the raw data. The classifiability of the feature vectors is less affected by removing elements from these vectors when they are produced using energy compacting transforms such as PCA and DWT. A large fraction of the transform coefficients can be removed without affecting the overall classifiability of the data vectors from distinct clusters [45]. To achieve high reduction rates, availability of a criterion using the characteristics of the specific transform could

be useful.

The goal of feature subset selection is to remove unnecessary features, and at the same time, maintain the classifiability. Since the energy of the ensemble of the feature vectors is closely related to the geometrical distance among them, feature reduction criteria that are based on energy preservation should be able to produce a set of features that maintain their classifiability.

If the filters used in obtaining the DWT coefficients are orthonormal, according to the Parseval's theorem [50] the image energy is preserved [16]. That is,

$$\sum_{i=1}^{S}\sum_{j=1}^{S}(x_{ij})^2 = \sum_{m=1}^{S}\sum_{n=1}^{S}(X_{mn})^2 \tag{3.1}$$

where x_{ij} and X_{mn} are, respectively, the elements of the image matrix $\boldsymbol{x}$ of size $S \times S$ and those of its wavelet transform matrix $\boldsymbol{X}$. Since most of the energy is contained in the first few coefficients of the DWT, by keeping the first l coefficients and ignoring the rest, we have

$$\sum_{i=1}^{S}\sum_{j=1}^{S}(x_{ij})^2 \approx \sum_{k=1}^{l}(v_k)^2, 1 \leq l \leq S^2 \tag{3.2}$$

provided that the elements of the one-dimensional vector $\boldsymbol{v}$ are those of $\boldsymbol{X}$ when serialized taking into consideration their energy contents.

Now, the problem is to find l in Eq. (3.2), so that the classifiability of the data points in the new space $\Re^l$ is close to maximum possible. The new dimension or the number of features to be retained is denoted by L. However, before trying to determine the right number of features it has to be decided how to serialize the elements of the coefficient matrix $\boldsymbol{X}$. In other words, it is first necessary to sort, that is, to serialize the coefficients of the DWT matrix in terms of their importance before selecting the first l coefficients. This is the subject of Section 3.2.

3.2 Serialization of the DWT Coefficients

The method of principal component analysis automatically sorts the features generated by this method. However for two-dimensional DWT, it is not quite obvious how the designer has to serialize the wavelet coefficients into a 1-D feature vector. For example, consider the method of *embedded zero-tree wavelet*. EZW is an important method for serializing and encoding the DWT coefficients [51]. Even though EZW puts into order the important coefficients, the corresponding coordinates of these coefficients are not necessarily the same for every image in the database. Figure 3.1 shows a hypothetical example to illustrate this. The second element of the feature vector on the left side is the coefficient located at $(1, 2)$ in the DWT matrix, whereas the second element of the vector on the right side is from location $(1, 3)$. Therefore, EZW produces vectors $\boldsymbol{v}$ in which the elements v_k for different images may not necessarily correspond to the coefficients located at the same position of all the transform matrices. This correspondence is essential in a pattern classification task. Therefore, we need to consider other sorting approaches of the DWT coefficients that do not have this problem.

The traditional approach to eliminating all the coefficients having a magnitude smaller than a certain threshold suffers from the same problem as mentioned above. Two other well-known methods for scanning of 2-D transform coefficients are zigzag and Morton scanning. The zigzag scanning as shown in Figure 3.2, has been used in traversing the discrete cosine transform (DCT) coefficient matrices [52]. As seen from Figure 2.3, the LL subband of the DWT is more important than the other ones. However the zigzag sorting method does not fully take into consideration this importance.

The Morton scanning of the wavelet coefficients [49], as demonstrated in Figure 3.3, sorts the coefficients according to the importance of the wavelet subbands of the DWT. This approach ensures that the wavelet subbands are traversed according to

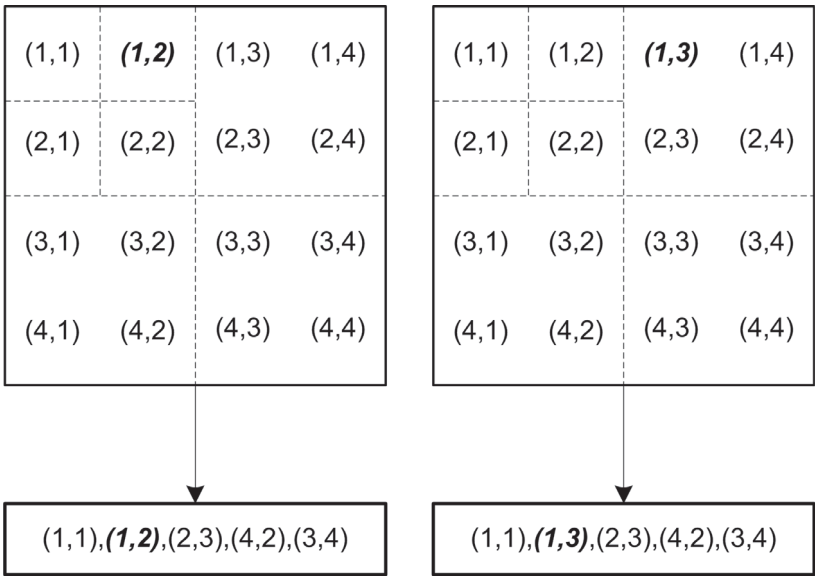


Figure 3.1: A hypothetical serialization method that does not necessarily place a specific DWT coefficient corresponding to two different images in the same location of the feature vector.

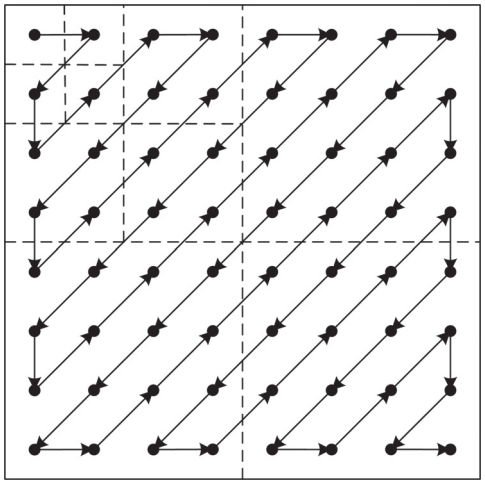


Figure 3.2: Zigzag serialization of DWT coefficient matrix.

their importance. In this scanning, the order of traversal is LL, HL, LH and HH subbands (see Figure 2.2). Thus, in a wavelet-based separability of clusters, it would be more appropriate to use the Morton scanning.

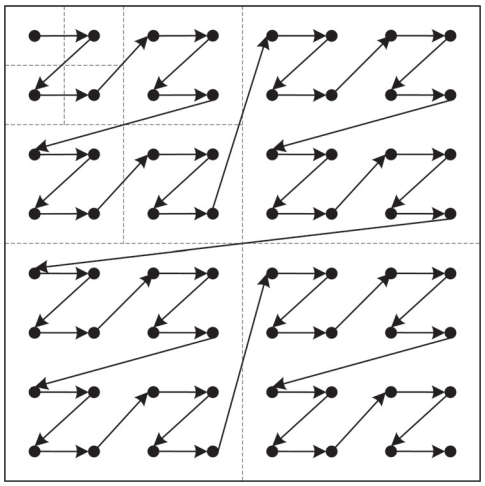


Figure 3.3: Morton scanning of DWT coefficient matrix.

3.3 Proposed Criterion

Assume that $\boldsymbol{v}^i \in \Re^l$ is the vector of first l entries obtained through the Morton scanning of the DWT coefficient matrix of i^{th} image in a database, where l is an arbitrary positive integer less than the number of entries in the coefficient matrix. An approximate value of energy of the ensemble of the images in the database, using Eq. (3.2), is given by

$$E(l) = \sum_{i=1}^{M} \sum_{j=1}^{l} \left(v_j^i\right)^2, \quad 1 \leq l \leq S^2 \tag{3.3}$$

where M is the number of training images in the database and v_j^i is the j^{th} element of the coefficient vector $\boldsymbol{v}^i \in \Re^l$.

Note that when $l = S^2$, Eq. (3.3) provides the complete energy of the ensemble of the images. Our objective is to obtain L, the minimum number of coefficients that when retained in the feature subset selection task, will result in a close to the

maximum classifiability. In order to give different importance to different coefficients, Eq. (3.3) can be modified as

$$E_{\boldsymbol{w}}(l) = \sum_{i=1}^{M} \sum_{j=1}^{l} w_j \left(v_j^i\right)^2, \quad 1 \leq l \leq S^2 \tag{3.4}$$

where w_j are the elements of a weighting vector $\boldsymbol{w}$. We call $E_{\boldsymbol{w}}$ *weighted partial energy*. The weighting vector is included in order to address the importance of the locations of the DWT coefficients. The weighting vector $\boldsymbol{w}$ is obtained by the Morton scanning of the waiting matrix of Figure 3.4. The idea is to give the same importance to each of the three subbands, namely, LH, HL and HH subbands, and this importance has to be less than that given to the LL subband. We propose a weight matrix whose elements assume values according to the following scheme

$$wm(i,j) = \begin{cases} 1 & i = 1;\ j = 1 \\ \frac{1}{3} \times 2^{-2k} & i = 1, \cdots, 2^k;\ j = 2^k + 1, \cdots, 2^{k+1} \\ \frac{1}{3} \times 2^{-2k} & i = 2^k + 1, \cdots, 2^{k+1};\ j = 1, \cdots, 2^k \\ \frac{1}{3} \times 2^{-2k} & i = 2^k + 1, \cdots, 2^{k+1};\ j = 2^k + 1, \cdots, 2^{k+1} \end{cases} \tag{3.5}$$

where $k = 0, 1, \cdots, log_2 S - 1$. Thus, the coefficients in the highest level of DWT decomposition are assigned weights of 1, $\frac{1}{3}$, $\frac{1}{3}$ and $\frac{1}{3}$ in the LL, HL, LH and HH subbands, respectively. For any other level of decomposition, the weight given to each of the coefficients in HL, LH and HH subbands is progressively reduced by a factor $\frac{1}{4}$ to that given to each of the coefficients in the three subbands of the next higher level of decomposition. Figure 3.4 shows an example of the weight matrix $\boldsymbol{wm}$ for $S = 8$.

The magnitude of the DWT coefficients, serialized by Morton scanning, generally decreases. The same holds for the elements of the coefficient matrix of Figure 3.4. These two factors contribute to the saturation of $E_{\boldsymbol{w}}(\cdot)$. If the difference between $E_{\boldsymbol{w}}(l)$ and $E_{\boldsymbol{w}}(l+1)$ is negligible, it means that employing an additional coefficient

1	1/3	1/12	1/12	1/48	1/48	1/48	1/48
1/3	1/3	1/12	1/12	1/48	1/48	1/48	1/48
1/12	1/12	1/12	1/12	1/48	1/48	1/48	1/48
1/12	1/12	1/12	1/12	1/48	1/48	1/48	1/48
1/48	1/48	1/48	1/48	1/48	1/48	1/48	1/48
1/48	1/48	1/48	1/48	1/48	1/48	1/48	1/48
1/48	1/48	1/48	1/48	1/48	1/48	1/48	1/48
1/48	1/48	1/48	1/48	1/48	1/48	1/48	1/48

Figure 3.4: Weight matrix taking into consideration the importance of the DWT coefficients.

only increases the length of the compressed vectors without contributing to the energy of the coefficients so far included. This means that the curve for $E_{\boldsymbol{w}}(l)$ gets saturated beyond this value of l. The minimum number of coefficients, L, that must be retained can then be determined by satisfying the condition

$$|E_{\boldsymbol{w}}(l+1) - E_{\boldsymbol{w}}(l)| \leq \alpha E_{\boldsymbol{w}}(S^2) \tag{3.6}$$

where α is a pre-specified small positive number.

In order to avoid a false determination of the number of coefficients to be retained due to some local saturation of $E_{\boldsymbol{w}}$, Eq. (3.6) is modified as

$$|E_{\boldsymbol{w}}(l+j+1) - E_{\boldsymbol{w}}(l+j)| \leq \alpha E_{\boldsymbol{w}}(S^2), \quad j = 0, 1, \cdots, J-1 \tag{3.7}$$

where J is a positive integer.

3.4 Algorithm

In Section 3.3 we have provided a set of formulae for determining L, the right number of features to be retained. In this section, we present this method in the form of an

algorithm and provide details for selecting certain parameters and functions used in this algorithm.

The method of determining the reduced dimensionality of the coefficient vectors as descried in the preceding section can be summarized as an algorithm given below.

Algorithm 1 : Determining the Number of Features in DWT Domain

1. Determine s the number of DWT stages.
2. Compute the 2-D transform matrix of all training images in the database using s recursions of DWT.
3. Scan each 2-D transform matrix using Morton scanning according to the scheme given in Figure 3.3.
4. Set $l \leftarrow 1$.
5. Compute $E_{\boldsymbol{w}}(l)$ using Eq. (3.4).
6. Set $l \leftarrow l + 1$.
7. If $l < S^2$, go to step 5.
8. Set $l \leftarrow 1$.
9. If Eq. (3.7) is satisfied, set $L \leftarrow l$ and terminate.
10. Set $l \leftarrow l + 1$;
11. If $l > S^2$, L cannot be determined, terminate.
12. Go to step 9.

An important parameter in this algorithm is the number of DWT stages in Figure 2.1. Assuming the size of the image to be an integral power of two, the value of this parameter can be obtained as

$$s = log_2 S \tag{3.8}$$

An experiment of running the proposed algorithm repeatedly for different values of α in Eq. (3.7) on certain number of databases is conducted. Values of $\alpha \leq 0.0005$,

result in an unnecessarily large L. However, for $\alpha \geq 0.005$ an insufficient number of features is obtained causing poor classifiability among the compressed vectors. It is found that a suitable choice for the value of α is 0.001. The value of J in Eq. (3.7) is not a sensitive parameter. It has been chosen to be 15, a value that adequately ensures that a steady state of $E_{\boldsymbol{w}}$ with respect to l has been achieved.

The introduction of our criterion uses the fact that the energy of the transform coefficients is the same as that of the image. Therefore, it is necessary to use an orthonormal wavelet to perform the transform described in Section 2.1. We employ the Haar wavelet [53] with scaling and wavelet functions given by

$$\Phi(x) = \begin{cases} 1 & 0.0 \leq x \leq 1.0 \\ 0 & otherwise \end{cases} \tag{3.9}$$

$$\Psi(x) = \begin{cases} 1 & 0.0 \leq x < 0.5 \\ -1 & 0.5 \leq x \leq 1.0 \\ 0 & otherwise \end{cases} \tag{3.10}$$

3.4.1 Complexity Analysis of Algorithm 1

In this section, the computational complexity of Algorithm 1 is investigated. First, the cost of generating the coefficients DWT and then that of determining L, the number of features that need to be retained, is studied. The number of operations required to run the algorithm on a database depends on the number and size of the training images in that database. It is assumed that the database contains M training images each of size $S \times S$, where S is an integer power of 2.

Let us assume that at each level of decomposition, the number of operations

needed for obtaining a filtered pixel output to be d. Thus, at the first level of decomposition, dS^2 operations are needed. For the second level of decomposition, the number of operations is reduced to $dS^2/4$, and so on and so forth. Therefore, the total number of operations required to perform DWT filtering involving $log_2 S$ decomposition levels for an $S \times S$ image is given by

$$dS^2(1 + \frac{1}{4} + \cdots + \frac{4}{S^2}) = \sum_{i=0}^{log_2 S} (\frac{1}{4})^{-i} < \frac{4}{3} dS^2 \tag{3.11}$$

This means that the total number of operations required for performing all the stages of the DWT operations on all training images is upper-bounded by $\frac{4}{3} MdS^2$. In other words, the computational complexity of the feature generation part of the algorithm is $O(dMS^2)$.

The weighting process of a scanned coefficient vector requires S^2 multiplication operations. This means that computing $E_{\boldsymbol{w}}(l), l = 1, 2, \cdots, S^2$, involves MS^2 operations. The number of operations needed for obtaining the saturation point of $E_{\boldsymbol{w}}(l)$ is not known in advance, but as it will be shown in the next section, the saturation happens at the L^{th} iteration, where $L \ll S^2$. Thus, the dominant part of the computational complexity of feature subset selection arises from the computation of partial weighted energy. In other words, the computational complexity of determining L is $O(MS^2)$, provided that the DWT coefficients are available.

From the above discussion, the overall computational complexity of Algorithm 1 is found to be $O((d + 1)MS^2)$. However, it is noted that dMS^2 operations are required for generating the DWT coefficients regardless of the method of determining the number used for features that need to be retained.

3.5 Implementation

In this section, the algorithm (Algorithm 1) presented in the preceding section is applied for feature selection in a series of face and object classification problems. Each image in the training set undergoes a recursive process of two-dimensional DWT. If the size of the image is not an integral power of two, the image is zero-padded before the transform is applied. The resulting coefficient matrix is then Morton scanned into a one-dimensional vector. These vectors are used in Algorithm 1 for determining L, the number of coefficients that need to be retained.

To demonstrate the ability of the proposed approach in predicting the right number of features, we need to perform a classification task in each database. This necessitates splitting the database into two sets: *training set* and *test set*. The training set has more samples than the test set has. In this work, we run the proposed and the other algorithms on the data points of the training set. The classifier used is a k-nearest neighbor (k-NN) classifier [7] with $k = 5$ and Euclidean distance used as a metric for the distance measure. In fact, we have carried out the experiments with $k = 1, 3$ and have noticed that the results are very similar.

The classification efficiency is measured using l features; $1 \leq l \leq L$, where L is given by Eq. (3.7). We also measure the accuracy of classification using all the original features in order to determine the classification accuracy in case of no compression. With L features, we expect an efficiency close to the one when all the features are used. Therefore, if with all the original features, the classifier used provides a certain classification rate, Algorithm 1 gives the minimum number of features needed to obtain almost the same classification rate. We begin our experiment first with a simple database and then continue with more complex ones. The results are presented in the following sub-sections.

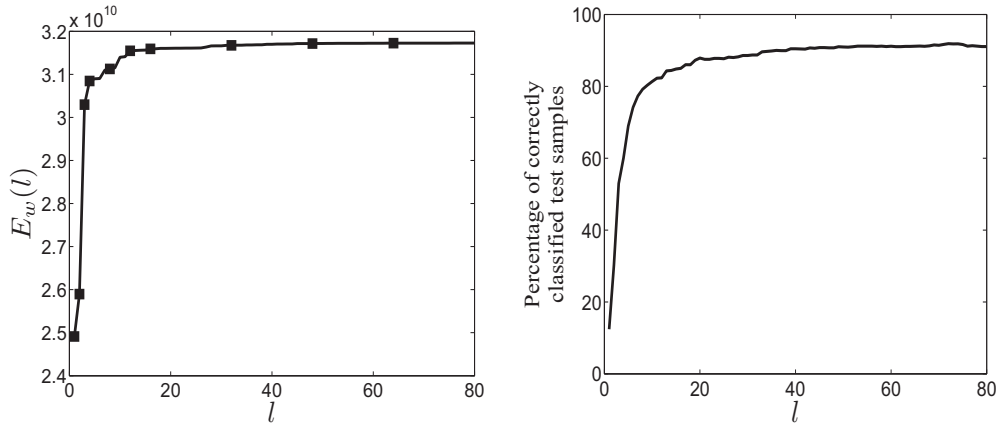
3.5.1 AT&T-Olivetti Face Database

The AT&T-Olivetti database [54] contains 40 subjects with 10 images per subject. The database is divided into training and test sections with seven and three images per person, respectively. All the images are of size 112×92. They are zero-padded to become 128×128 for the DWT process. Thus, there are $16,384$ DWT coefficients or features before the compression. Figure 3.5 shows two samples of each of the two subjects chosen from this database. As seen from this figure, the different face images of the same subject are the rotated versions of the corresponding front image. Figure 3.6(a) shows the graph of the proposed criterion versus the number of coefficients retained. A solid square on the plot indicates the transition point from one subbands to the next. A k-NN classifier is used after the wavelet compression on the test subset of the face images of 40 subjects. The results of this categorization are shown in Figure 3.6(b). From Eq. (3.7), the dimensionality is obtained to be 49. With 49 features, 91.02% of the test vectors are classified correctly. The maximum classification performance is 91.87% attained with 69 features. The computation of the proposed criterion takes 1.1 seconds.

Figure 3.5: A few samples from AT&T-Olivetti database [54].

3.5.2 Columbia Object Image Library (COIL-20) Database

The COIL-20 database [55] contains 20 objects each having 72 images with different orientations. Figure 3.7 shows a sample image of each of the objects. Each image has a resolution of 128×128. From the 72 images of each of the objects, 40 are selected for computing the weighted partial energy, $E_{\boldsymbol{w}}$, and the remaining 32 are

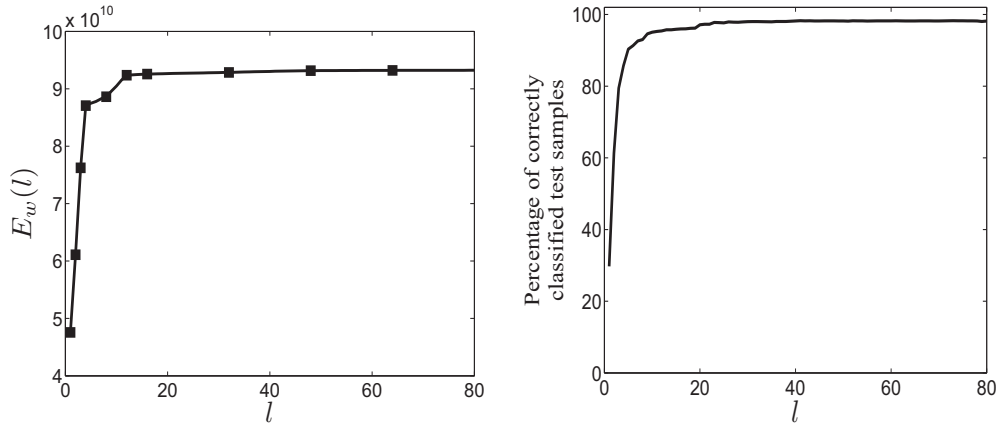


(a) (b)

Figure 3.6: AT&T-Olivetti database. (a) Weighted partial energy as a function of the number of coefficients retained. (b) Percentage of correctly classified samples as a function of the number of coefficients retained.

used as test images. Figure 3.8 shows $E_{\boldsymbol{w}}$ and the percentage of correctly classified test points as functions of the number of features used. It is seen that there is a good correspondence between the criterion and the classifier efficiency. By applying the algorithm described in Section 3.4, a minimum number of features is found to be 49. For $L = 49$ features, 98.20% of the test images are classified correctly. A maximum classification efficiency of 98.27% is obtained by using 73 features. For this database, the run time of the algorithm is 4.3 seconds.

Figure 3.7: A few samples from COIL-20 database [55].



(a) (b)

Figure 3.8: COIL-20 database. (a) Weighted partial energy as a function of the number of coefficients retained. (b) Percentage of correctly classified samples as a function of the number of coefficients retained.

3.5.3 MIT-CBCL Face Database

The MIT Center for Biological and Computational Learning (CBCL) face recognition database [41] contains face images of ten subjects. Each subject with nine different poses has 36 levels of illuminations per pose. The total number of images is $3,240$ each of size 200×200. They are zero-padded to become 256×256. Out of 324 images in each class, 216 and 108 are randomly selected for training and testing, respectively. The number of test images are $1,080$ for all ten classes. Figure 3.9 gives three examples of the faces of each of the two subjects chosen from the database with different poses and lighting conditions. For each l, the number of features, $E_{\boldsymbol{w}}(l)$ is computed using the DWT coefficients of the training images of the database. Figure 3.10(a) shows the plot of $E_{\boldsymbol{w}}$ as a function of the number of features l. Figure 3.10(b) shows the percentage of correctly classified test points versus l. The classification performance obviously improves as the dimensionality of the feature vectors increases, but the main point to note is that there is a strong similarity in the saturation behaviors of the two plots in Figure 3.10. Applying Eq. (3.7) gives 53 as the best number of features. With 53 features, the classification accuracy is 99.66%. When 91 or greater number

of features are used, all the 1,080 test images are categorized correctly. Computation of the criterion takes 33.0 seconds.

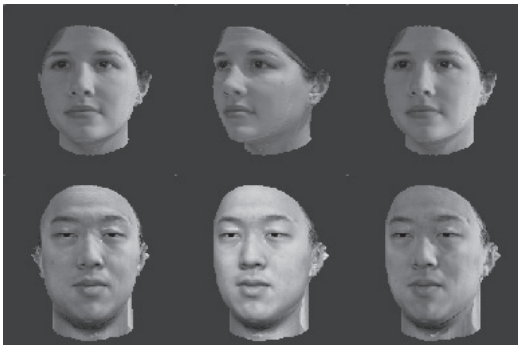
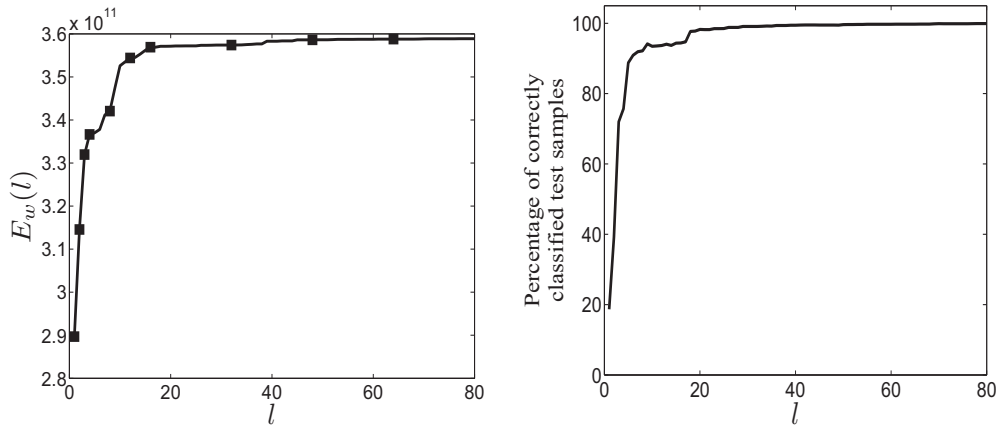


Figure 3.9: A few samples from MIT-CBCL Face database [41].



(a) (b)

Figure 3.10: MIT-CBCL database. (a) Weighted partial energy as a function of the number of coefficients retained. (b) Percentage of correctly classified samples as a function of the number of coefficients retained.

3.5.4 MNIST Handwritten Digit Database

MNIST handwritten digit database [56] of the US National Institute of Standards and Technology (NIST) is the next example considered in our experiments. There are 60,000 training samples and 10,000 test samples in this database stored in two separate files. Any classification algorithm that involves all these images, requires very long time. Therefore, in order to validate Algorithm 1 using this database, we use a

fraction of those samples: the first 18,000 samples from the training file are used for computing $E_{\boldsymbol{w}}$ and the first 3,000 samples from the test file are used for classification. Figure 3.11 shows several 28×28 images of the database. They are zero-padded to become 32×32 before undergoing the DWT operation of Figure 2.1. Figure 3.12 shows the plots of $E_{\boldsymbol{w}}$ and the result of the classification, both as functions of the number of features used. Based on Eq. (3.7), at $l = 83$ features, $E_{\boldsymbol{w}}$ becomes completely saturated. This could also be observed from Figure 3.12. For this same number of features, 94.63% of 3,000 test images are classified correctly. Increasing the number of features to 211, results an accuracy of 96.40%, the maximum accuracy attainable. The algorithm needs about three minutes of run time.

Figure 3.11: A few samples from MNIST handwritten digit database [56].

3.5.5 The Caltech-101 Database

The Caltech-101 database [57] comprises 101 categories each containing a fixed number of samples from 31 to 800. The huge number of images in this database makes it hard to conduct a classification using all of them. Therefore, we select 2,000 images for training and 1,000 other images for testing the algorithm. These images are drawn from a number of randomly selected classes. Figure 3.13 shows a few samples of this database. The sizes of the images in this database are different. They are all zero-padded to become 512×512. The plots of $E_{\boldsymbol{w}}$ and the classification accuracy of

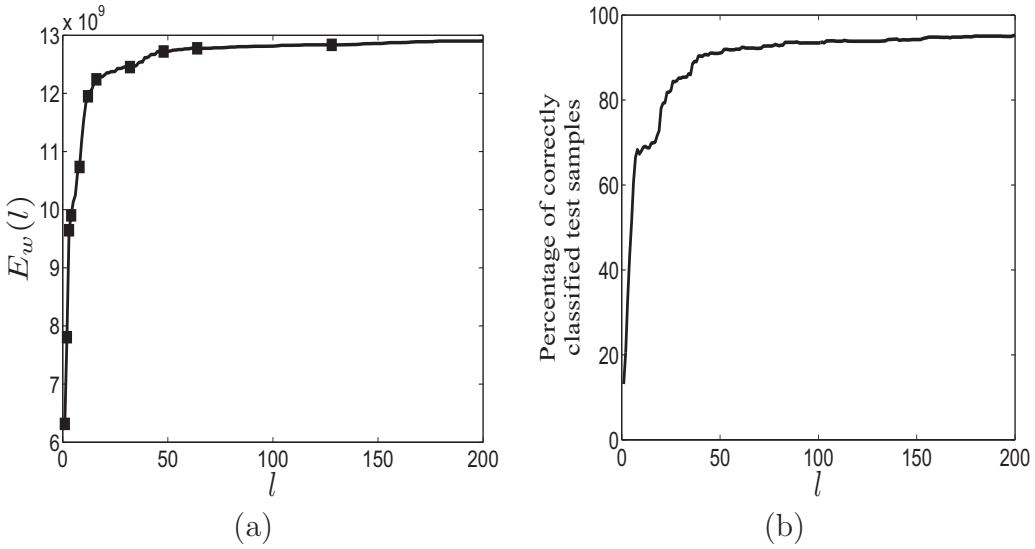


Figure 3.12: MNIST Handwritten Digit database. (a) Weighted partial energy as a function of the number of coefficients retained. (b) Percentage of correctly classified samples as a function of the number of coefficients retained.

the test vectors are shown in Figure 3.14. Just as the other databases considered in this section, a k-NN ($k = 5$) classifier is used in the classification experiment. For this database, even though all the original features are used, only about 30% of the test images are classified correctly. Applying Algorithm 1 to this database, gives $L = 55$ features. With this number of features, 325 out of 1,000 test images are classified correctly, that is, with $L = 55$, one achieves almost the same accuracy as that by using all the $256 \times 256 = 65,536$ coefficients.

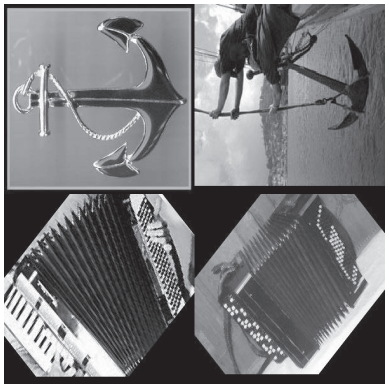


Figure 3.13: Four samples from two classes of Caltech-101 database [57].

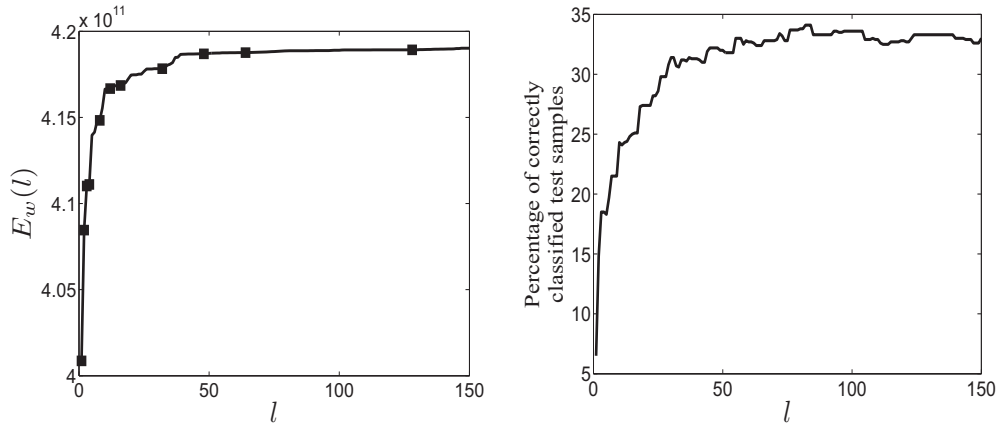


Figure 3.14: Caltech-101 database. (a) Weighted partial energy as a function of the number of coefficients retained. (b) Percentage of correctly classified samples as a function of the number of coefficients retained.

3.6 Comparisons

The experiments of Section 3.5 are based on Haar wavelet transform, Morton scanning of the DWT coefficients and the weighting scheme of Figure 3.4. In this section, the experiments performed in Sections 3.5.1-3.5.5 using different databases are repeated for the following four cases: (i) zigzag scanning of Figure 3.2 instead of Morton scanning of Figure 3.3, (ii) Morton scanning with random selection of the coefficients in undivided subbands of coefficient matrix of Figure 3.3, (iii) Morton scanning with weighting scheme of Figure 3.15 instead of that of Figure 3.4, and (iv) using a bi-orthogonal wavelet instead of the orthonormal wavelet.

The results of these four sets of experiments as mentioned above along with those obtained in Sections 3.5.1-3.5.5 are presented in Tables 3.1-3.3. The entries in these tables are the number of features that need to be retained and the corresponding classification accuracies. The results in Table 3.1 show that using the weighting scheme of Figure 3.15 provides a smaller number of features to be retained generally at the expense of a slight decrease in the classification accuracy. Therefore, as long as the LL subband is given more importance, the values of the actual weights do not have

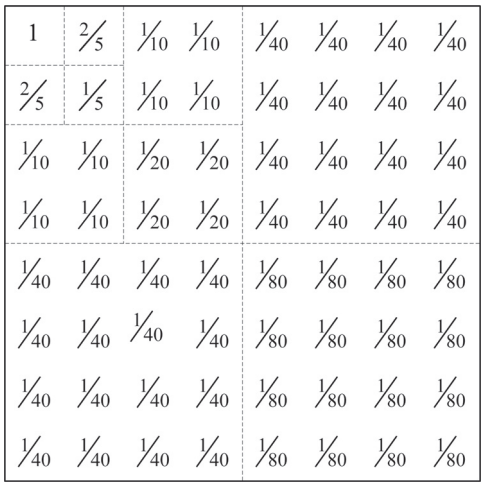


Figure 3.15: Another weight matrix emphasizing the importance of the location of the DWT coefficients.

Database	Weighting scheme of Fig.3.4		Weighting scheme of Fig.3.15	
	L	Accuracy	L	Accuracy
AT&T	49	91.02	45	90.78
COIL	49	98.20	47	98.22
MIT	53	99.66	45	99.49
MNIST	83	94.63	83	94.63
Caltech	55	32.50	39	32.30

Table 3.1: Effect of weighting scheme on the value of L and the classification accuracy.

a significant impact on the results. The choice of wavelet is important as seen from Table 3.2: the bi-orthogonal wavelets obviously produce inferior results, since the Parseval's theorem (Eq. (3.1)) is not valid in this case. It is seen from Table 3.3 that depending on the database, the choice of a weighting scheme may have a significant impact on the value of L. However, this variation in the value of L is not reflecting in the corresponding variation in the classification accuracy except for the MNIST database in which a very significant increase in the value of L is observed along with a significant increase in the classification accuracy when the scanning scheme is changed from the Morton scanning to the zigzag scanning. These comparisons show that the asymptotic behavior of the proposed criterion, $E_{\boldsymbol{w}}$, is achieved regardless of the above-mentioned variations.

Database	Orthonormal wavelet		Bi-orthogonal wavelet	
	L	Accuracy	L	Accuracy
AT&T	49	91.02	45	86.52
COIL	49	98.20	45	97.83
MIT	53	99.66	48	99.39
MNIST	83	94.63	61	90.07
Caltech	55	32.50	40	31.60

Table 3.2: Effect of different wavelets on the value of L and the classification accuracy.

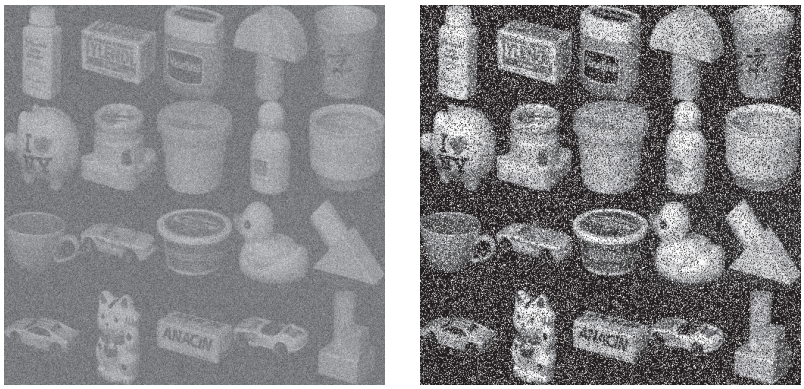
Database	Morton scanning		Zigzag scanning		Random scanning	
	L	Accuracy	L	Accuracy	L	Accuracy
AT&T	49	91.02	51	91.19	45	89.98
COIL	49	98.20	57	98.08	50	98.20
MIT	53	99.66	39	99.38	61	99.67
MNIST	83	94.63	149	96.50	79	94.40
Caltech	55	32.50	64	33.20	63	32.40

Table 3.3: Effect of scanning approaches on the value of L and the classification accuracy.

3.7 Robustness

To demonstrate the robustness of the proposed algorithm for determining the number of the features to be retained for classification, we conduct two sets of experiments in the presence of noise. The experiments are carried out on the Gaussian noise corrupted images with the signal-to-noise ratios (SNR) ranging from 12 dB to 30 dB in steps of 1 dB and salt and pepper (impulsive) noise corrupted images with SNRs ranging from 3 dB to 25 dB, again in steps of 1 dB. For each level of SNR, the average results over ten repetitions of the experiments are obtained. Images in Figure 3.16 are the noisy versions of that in Figure 3.7. The SNR values of the images in Figure 3.16(a) and Figure 3.16(b) are 14 dB and 4 dB, respectively.

Figures 3.17(a) and (b) show, respectively, the average values of L and the average number of correctly classified test samples as functions of the SNR values when the noise type is Gaussian. Figures 3.17(c) and (d) show the corresponding results when the noise type is impulsive. It is seen from the plots of these figures that with the



(a) (b)

Figure 3.16: A few samples of noisy COIL-20 images. (a) Gaussian noise with $SNR = 14dB$. (b) Impulsive noise with $SNR = 4dB$.

Gaussian noise, the performance of the algorithm starts to deteriorate when the SNR is reduced below 16 dB. On the other hand, for the salt and pepper noise, the algorithm is still able to determine the right dimensionality with the SNR value as low as 6 dB.

3.8 Summary

In this chapter, we have introduced a simple criterion and developed an efficient algorithm for the prediction of a reduced dimension of feature vectors generated by using a wavelet. It has been shown that with this reduced dimensionality of feature vectors, the efficiency of a k-NN classifier is very close to that using all the features. Extensive experiments on AT&T/Olivetti face database, Columbia Object Image Library (COIL-20) database, MIT-CBCL face database, MNIST handwritten digit database, and Caltech-101 database have demonstrated the effectiveness and efficiency of the proposed algorithm. It has been shown that the use of appropriate weights given to the features in computing the proposed criterion enhances the saturation behavior of the criterion and facilitates the process of determining the number of features to be retained. The robustness of the algorithm has also been demonstrated by repeating the experiments on the images of COIL-20 databases corrupted by Gaussian and

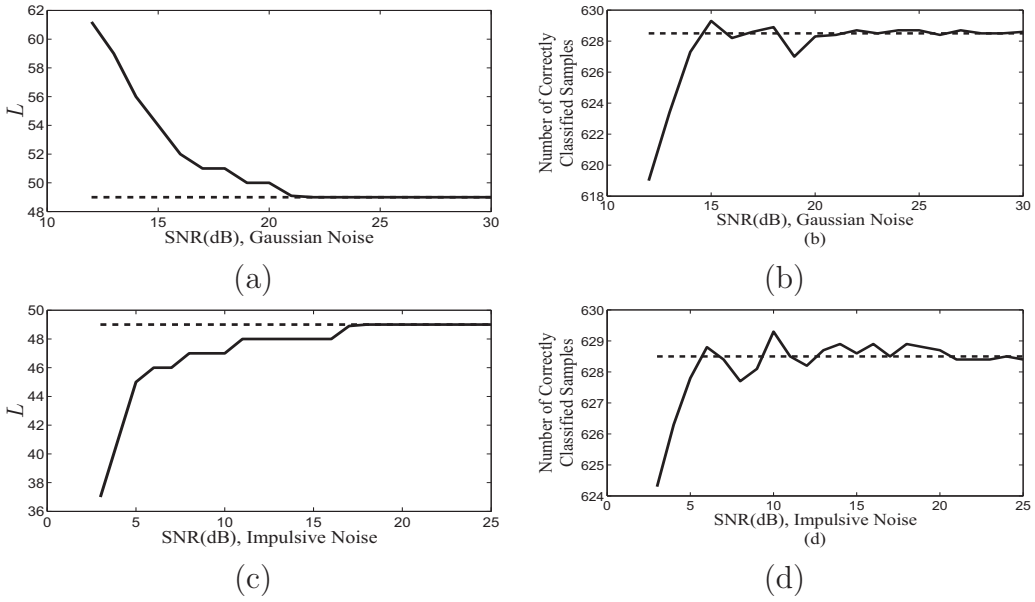


Figure 3.17: The effect of noise on Algorithm 1 using COIL-20 database reflected on L and the classification accuracy. (a) L with Gaussian noise. (b) Classification accuracy with Gaussian noise. (c) L with impulsive noise. (d) Classification accuracy with impulsive noise.

impulsive noise.

Chapter 4

Feature Selection in PCA Domain

Principal component analysis is used extensively for generating features in the field of pattern classification. Since the process of PCA automatically sorts the generated features (Section 2.2), determining the number of features to be retained is the main concern in feature subset selection in a PCA-based technique. In this chapter, a criterion and an algorithm for determining the number of features generated by the principal component analysis is developed so that the set with reduced number of features provides almost the same classifiability as that of using all the features [58,59]. To this end, we first introduce a criterion for determining the number of features to be retained based on maintaining the distance among the mean points of the database clusters while the dimensionality of the feature vectors in these clusters is reduced. However, it is possible for the distinct clusters to have some overlap in a reduced dimensionality even if their mean vectors are well separated. Therefore, we next introduce a second criterion that is based on expressing the reduced feature vectors as a linear combination of the eigenvectors of the covariance matrix. A combination of these two criteria is obtained as the final criterion that it used for determining the reduced dimensionality of the feature vectors.

This chapter is organized as follows. Section 4.1 introduces a criterion called

cumulative global mean distance ($CGMD$). The algorithm devised based on this criterion relies on the separability of the clusters' means in a database. Section 4.2 proposes another criterion that brings into account the separability of the individual data vectors of the clusters. This second criterion is called the cumulative global sample scattering distance ($CGSSD$). Section 4.3 presents an algorithm for determining the number of features to be retained based on these two criteria. Section 4.4 presents a number of experiments performed on different databases to verify the applicability of the proposed criteria. In Section 4.5, the robustness of the proposed algorithm is studied. In order to compare the proposed criterion with a method of determining a reduced number of features based on some standard distances, the results of experiments based on using Mahalanobis and Bhattacharyya distances are presented in Section 4.6. Section 4.7 concludes the chapter.

4.1 Cumulative Global Mean Distance

In this section, a criterion for determining a minimum number of features that need to be retained for providing a classifiability that is close to that of using all the features is introduced. Assume that there are c distinct classes in the training data and let $\boldsymbol{\mu}^i(1 \leq i \leq c)$ denote the mean vectors of those classes, and $\boldsymbol{V}^k(1 \leq k \leq l)$ denote the eigenvectors corresponding to the l largest eigenvalues λ_k of the covariance matrix of all the training feature vectors $\boldsymbol{x}^i, i = 1, 2, \cdots M, \boldsymbol{x}^i \in \Re^N$. Let us now consider an expression given by

$$D(i, j, k) = \left[\lambda_k(\boldsymbol{\mu}^i - \boldsymbol{\mu}^j)^T \boldsymbol{V}^k\right]^2 \tag{4.1}$$

where $\boldsymbol{\mu}^i$ and $\boldsymbol{\mu}^j$ are, respectively, the mean vectors of the i^{th} and j^{th} training classes of the data set and λ_k represents the eigenvalue corresponding to the eigenvector $\boldsymbol{V}^k$. The expression given by Eq. (4.1) can be considered to represent a measure of the

distance between the means of the i^{th} and j^{th} classes of the data set. It indicates how "distant" the two means are in the orthogonal coordinate system formed by the PCA. The projection in Eq. (4.1) is performed along the eigenvectors, since the main variation of the data points is indeed along the eigenvectors of the covariance matrix. The multiplication by the corresponding eigenvalues λ_k gives a greater significance to more important eigenvectors in this projection process. If there are more than two classes in the data set, it is possible to formulate a global distance (GD) measure as

$$GD(k) = \sum_{i=2}^{c} \sum_{j=1}^{i-1} D(i, j, k) \tag{4.2}$$

where c denotes the number of classes ($c \geq 2$). Note that GD is a *non-increasing* function of k, since the eigenvalues λ_k in Eq. (4.1) are sorted in decreasing order, that is $\lambda_{k+1} \leq \lambda_k$. In fact, the eigenvalues of the covariance matrix of a set of natural data are highly decreasing (see the discussion in Section 4.2.1). Therefore, GD is a rapidly deceasing function of k.

Recall that the main objective of this chapter is to determine the number of features, i.e., the number of principal components in the case of PCA that need to be retained. Therefore, a cumulative distance is introduced that sums up all the $GD(k)$, up to the number of new dimension l. This new measure is called the *cumulative global mean distance* given by

$$\zeta(l) = \sum_{k=1}^{l} GD(k), l = 1, 2, \cdots, l_{Max} \tag{4.3}$$

It is observed that ζ is an increasing function of l, that is, its value will increase monotonically as the number of eigenvectors that are included is increased. However, since $GD(\cdot)$ is a rapidly decreasing function, in practice, $\zeta(\cdot)$ can be expected to have a steady state behavior, represented by a plateau, as more eigenvectors are introduced. If such a behavior is obtained, the number of features that need to be retained (L) is the number beyond which the growth in $\zeta(\cdot)$ is insignificant. In Section 4.4, this

argument will be further substantiated through several experiments conducted on a number of data sets.

Using the saturation characteristic of $\zeta(\cdot)$, L, the number of features to be retained can be determined to be the smallest l at which $\zeta(l+1)$ - $\zeta(l)$ is a very small fraction of $\zeta(l_{Max})$, that is,

$$\zeta(l+1) - \zeta(l) < \alpha(\zeta(l_{Max})) \tag{4.4}$$

where α is a pre-specified small positive number and l_{Max} is the number of eigenvalues of the covariance matrix.

4.2 Cumulative Global Sample Scattering

The cumulative global mean distance ($CGMD$) defined by Eq. (4.3), used to determine the number of features to be retained, yields for some databases a value of L that is less than the number of features necessary for classification. The inadequacy of $CGMD$ results from the fact that it ignores the influence of the higher order statistics of the training vectors while focusing mainly on the clusters means. It is true that the farther apart the class means, the more accurate is the classification. However, the scatter within classes also plays an important role even if the means of the classes are well-separated in a given dimensionality-reduced sub-space created by a certain number of eigenvectors. There could be instances of databases in which even though the mean vectors of two clusters are far from each other, there may be vectors in the two clusters that are close to one another.

The training vectors of the database have the majority of their variations along the first L eigenvectors. This means that only an insignificant fraction of the total variation of the training vectors falls outside the coordinate system created by these L eigenvectors and that the training vectors can be almost completely expressed as a

linear combination of these eigenvectors. However, since the value of L is not known in advance, the following approach is proposed to determine it. The values of the projection of all the training vectors along the first eigenvector are summed up. The summation is then calculated along the second eigenvector. This process is repeated using the next most important eigenvectors until the summation of the values of the projection along an eigenvector becomes insignificant. The number of eigenvectors up to this eigenvector is then selected to be the number of features that need to be retained.

The above discussion could be formalized as follows. The magnitude square of the projection of a training vector along the k^{th} eigenvector $\boldsymbol{V}^k$, scaled by the square of the corresponding eigenvalue λ_k is given by

$$P(i,j,k) = \lambda_k^2 \left| \langle (\boldsymbol{x}^{ij} - \boldsymbol{\mu}), \boldsymbol{V}^k \rangle \right|^2 \tag{4.5}$$

where $\boldsymbol{x}^{ij}$ denotes the j^{th} training vector of the i^{th} class and $\boldsymbol{\mu}$ is the mean vector of all the training vectors. If $\boldsymbol{x}^{ij}$ does not have significant components outside the space generated by the eigenvectors $\boldsymbol{V}^k$, $(k = 1, 2, ..., L)$, then it is almost perpendicular to any of the rest of eigenvectors $\boldsymbol{V}^k$, $(k > L)$ and the value of the inner product of $\boldsymbol{x}^{ij}$ with any of these eigenvectors is negligible. In Eq. (4.5), the multiplication of the inner product by λ_k^2, is simply to provide appropriate importance to the eigenvector $\boldsymbol{V}^k$ on the projection. The eigenvectors corresponding to larger eigenvalues are indeed more important than the remaining ones. As in the case of $CGMD$, a cumulative global version of this new distance is now defined as

$$GP(k) = \sum_{i=1}^{c} \sum_{j=1}^{M_i} P(i,j,k) \tag{4.6a}$$

$$\eta(l) = \sum_{k=1}^{l} GP(k), \quad l = 1, 2, \cdots, l_{Max} \tag{4.6b}$$

where M_i is the number of training points in the i^{th} class. This last summation given by $\eta(\cdot)$ is called the *cumulative global sample scattering distance* ($CGSSD$). Here, the objective is to obtain L, the number of eigenvectors needed to represent all the training vectors. Each of the training vectors can be expressed as a linear combination of the first L eigenvectors. The concern is to determine L given the values of $\eta(l), 1 \leq l \leq l_{Max}$ which is equivalent to determining the point $l = L$ at which the saturation begins in the graph of η versus l. Mathematically, that corresponds to the point where the slope of the curve in this graph becomes zero (note that η is a non-decreasing function of l.) In practice, the saturation occurs where $\eta(l+1)$ and $\eta(l)$ are very close, that is,

$$\eta(l+1) - \eta(l) < \beta(\eta(l_{Max})) \tag{4.7}$$

where β is a small positive number and l_{Max} is the number of positive eigenvalues of the covariance matrix. The smallest l satisfying Eq. (4.7) is selected to be the value of L.

4.2.1 Saturation of $\zeta(\cdot)$ and $\eta(\cdot)$

Feature subset selection in the context of principal component analysis means removing less important components from the feature vectors so that their dimensionality is reduced from N to $L < N$. Assuming L features are enough for a classification, then increasing the number of features to $L+1$ does not significantly improve the classification accuracy. Equivalently, the distance among the feature vectors in $\Re^{L+1}$ is not noticeably greater than that in $\Re^{L}$. From another perspective, the feature vectors representing the original data points form a structure in $\Re^{N}$ that has the majority of its variation along only L directions given by the L eigenvectors of the covariance matrix corresponding to the L largest eigenvalues of this matrix. Increasing the number of features from L to $L+1$ is equivalent to introducing a new direction to the feature

space along which the variation of the feature vectors is insignificant. The amount of variation along an eigenvector is reflected in the magnitude of the corresponding eigenvalue. This means that in the vicinity of $l = L$, we have $\lambda_{l+1} << \sum_{k=1}^{l} \lambda_k$. In other words, it can be claimed that the value of $(L+1)^{\text{th}}$ eigenvalue is not significant, since otherwise, the variation along the newly introduced eigenvector cannot be negligible and L cannot represent a sufficient number of features, which contradicts the assumption that L features are enough.

As an example of a practical case, Figure 4.1 shows the eigenvalue spectrum of the MNIST handwritten digit database [56]. The plot in this figure illustrates experimentally the above discussion.

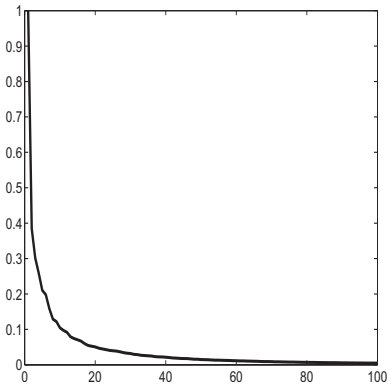


Figure 4.1: Eigenvalue spectrum of the MNIST handwritten digit database.

The distance measures of Eqs. (4.1) and (4.5) are directly proportional to the square of the eigenvalues of the covariance matrix. This is why these two measures reduce as the number of eigenvalues increases. In other words, $\zeta(\cdot)$ and $\eta(\cdot)$, the cumulative global versions of the two measures introduced in this chapter attain saturation.

4.3 Algorithm

Either of the two criteria developed in the preceding sections could be used for determining the number of features to be retained. However, in our approach for selecting

L, we use both the cumulative global mean distance and the cumulative global sample scattering distance. While the former gives a significant cluster separability, the latter resolves the problem of inseparability of mixed vectors in the case of overlapping clusters in spaces of insufficient dimensionality. There are at least two approaches as to how make a conclusion based on these two measures. One approach is to evaluate L based on Eqs. (4.4) and (4.7) separately and then to determine the final reduced dimensionality based on a fusion technique. A second approach is to evaluate a new measure based on both these measures and to determine L based on this new measure. In this work, this second way of determining the reduced number of features is adopted.

A simple integration of the two measures is a linear combination with equal weights for the two criteria. Since it is possible for the two measures to have very different values, we scale the values of the two measures to have the same maximum value by dividing $\zeta(\cdot)$ and $\eta(\cdot)$ by $\zeta(l_{Max})$ and $\eta(l_{Max})$, respectively. The combined measure is a simple addition of these two scaled quantities given by

$$cc(l) = \zeta(l)/\zeta(l_{Max}) + \eta(l)/\eta(l_{Max}) \tag{4.8}$$

and is referred to as *combined criterion*. Since both ζ and η tend to saturate, the value of $cc(l_{Max})$ gets very close to 2. Based on an argument similar to that presented in Sections 4.1 and 4.2 for determining the saturation point of the criteria, L is chosen to be the smallest positive integer satisfying

$$cc(l+1) - cc(l) < \gamma(cc(l_{Max})) \tag{4.9}$$

where γ is a pre-specified positive small constant.

Recalling that M and N are, respectively, the number and the original dimensionality of the training vectors, the various steps of the technique for finding the reduced number of features are summarized in Algorithm 2.

Algorithm 2 : Determining the Number of Features in PCA Domain

1. Compute the covariance matrix of the training data and perform eigen analysis to obtain its eigenvalues and eigenvectors.
2. Set $l \leftarrow 1$.
3. Set $l_{Max} = \min$ (M,N).
4. Compute $cc(l)$ using Eq. (4.8).
5. $l \leftarrow l + 1$.
6. If $l < l_{Max}$ go to step 4.
7. Set $l \leftarrow 1$.
8. If Eq. (4.9) is satisfied, set $L \leftarrow l$ and terminate.
9. Set $l \leftarrow l + 1$;
10. If $l > l_{Max}$, L cannot be determined, terminate.
11. Go to step 8.

In a practical application, all the available data are used in computing the covariance matrix and obtaining the value of L. However, in the experiments of this study, in order to demonstrate the effectiveness of the introduced criterion, the data samples in each database are partitioned into two sets. The first set, which is usually more populated, is used as the *training set* and the less populated one is used as the *test set*. The former set is used in computation of the covariance matrix and to conduct eigen analysis. The members of the test set undergo a dimension reduction process using the transform matrix of Eq. (2.6). The test vectors of reduced dimensionality are then used in a classification experiment in order to determine the classifiability of the feature vectors in a given dimension, i.e. for a specific value of l.

For above algorithm, the value of the parameter γ needs to be specified. To determine the value of this parameter, an experiment of running the algorithm repeatedly on certain number of databases is conducted. It is found that, in general, for values

of $\gamma \leq 0.0005$, more features than necessary are retained, and for $\gamma \geq 0.005$, the algorithm gives an insufficient number of features. The value of this parameter is chosen to be 0.001.

4.3.1 Complexity Analysis of Algorithm 2

In this section, the computational complexity of Algorithm 2 is studied. First, the complexity of eigen analysis part of the algorithm that is needed for computing the eigenvalues and eigenvectors of the covariance matrix is studied and then the complexity of the part of algorithm dealing with the computation of the criteria and determining L, the number of features that need to be retained is considered. Note that according to Section 2.2.2, since for the majority of image databases M, the number of training samples is less than N, the original dimensionality of these samples, l_{Max}, the number of eigenvalues of the covariance matrix is equal to M.

The computational complexity of eigen analysis of a matrix of size $M \times M$ is $O(M^3)$ [60]. Computing the $M \times M$ covariance matrix based on Section 2.2.2 involves the computation of $\boldsymbol{X}$ from Eq. 2.8. This requires $2NM$ subtraction and addition operations. With $\boldsymbol{X}$ now known, computing the covariance matrix using Eq. (2.13) requires $2MN^2$ multiplication and addition operations. Hence, the computational complexity of the first two steps of Algorithm 2 is $O(M^3 + 2MN^2)$.

Now, let us investigate the computational complexity of determining L. For $CGMD$, it is noted that for each computation of Eq. (4.1), N subtraction and N multiplication operations have to be performed. Evaluating $GD(k)$ using Eq. (4.2) requires $\frac{c}{2}(c-1)$ computations of Eq. (4.1), where c is the number of distinct classes in the database. Therefore, evaluating $GD(k)$ corresponding to an eigenvalue involves a total of $c(c-1)N$ subtraction and multiplication operations, and $\frac{c}{2}(c-1)$ addition operations. This means that the computation of $\zeta(l),\ l = 1, 2, \cdots, l_{Max} = M$

would require $c(c-1)NM$ operations. As for $CGSSD$, $P(i,j,k)$ requires N subtraction and N multiplication operations to be evaluated (Eq. (4.5)). Hence, the computation of $GP(k)$ using Eq. (4.6a) needs $2MN$ operations. Computing $\eta(l),\ l = 1, 2, \cdots, l_{Max} = M$ requires $2M^2N$ operations. Computing the combined criterion $cc(l),\ l = 1, 2, \cdots, l_{Max} = M$ using Eq. (4.8) requires $2M$ divisions and M addition operations. Determining L by following the last five steps of Algorithm 2 needs a maximum of M iterations. It is noted that the cost of computing $cc(\cdot)$ and searching for its saturation is much less than that of computing $\zeta(\cdot)$ and $\eta(\cdot)$. The computation of both of these latter criteria requires $NM\,(c(c-1)+2M)$ operations according to the above discussion. Therefore, the computational complexity of the part of Algorithm 2 dedicated to determining L is $O(2NM^2 + NMc^2)$.

It is concluded that the overall complexity of Algorithm 2 is $O(M^3 + 2MN^2 + 2NM^2 + NMc^2)$. However, it is important to notice that the part of this algorithm for performing the eigen analysis of the covariance matrix needs to be executed regardless of the method of determining the number of features that need to be retained. The complexity of this part is $O(M^3 + 2MN^2)$.

4.4 Implementation

The proposed method is next applied for feature reduction to a series of face and object recognition problems. Each image is raster scanned into a vector. For most of the image databases, it is necessary to conduct the eigen analysis of Section 2.2.2, since the original dimensionality of the training vectors in these databases is higher than the number of data points in the training set.

4.4.1 AT&T-Olivetti Face Database

The AT&T and Olivetti database [54] contains 40 subjects with 10 images per subject. The different images of the same subject are simply rotated versions of its frontal image and also changes in facial expression. Each class of the database is randomly divided into training and test sections with seven and three images per subject, respectively. The size of the images in this database is 112×92. Figure 4.2 shows a sample of images from this database. The combined criterion is computed based on the eigen analysis of the covariance matrix of the training data. To see the relationship between this graph and the classification efficiency, a nearest neighbor classifier is used to classify the test subset of the face images of 40 subjects. Figure 4.3 shows the graph of cc and the percentage of correctly classified test images as functions of the reduced dimension l. It is seen from this figure that there is a good correspondence between the saturation behavior of the two curves. This demonstrates the effectiveness of the proposed criterion in determining the number of features. From Eq. (4.9), the number of features that need to be retained is obtained to be 17. With this number of features retained, the classification efficiency is 95.4%. The best accuracy of 97.0% accuracy is achieved with 47 features. The computation of the criterion needs 12 seconds.

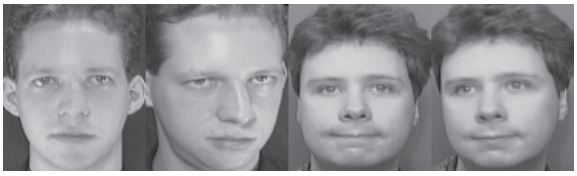
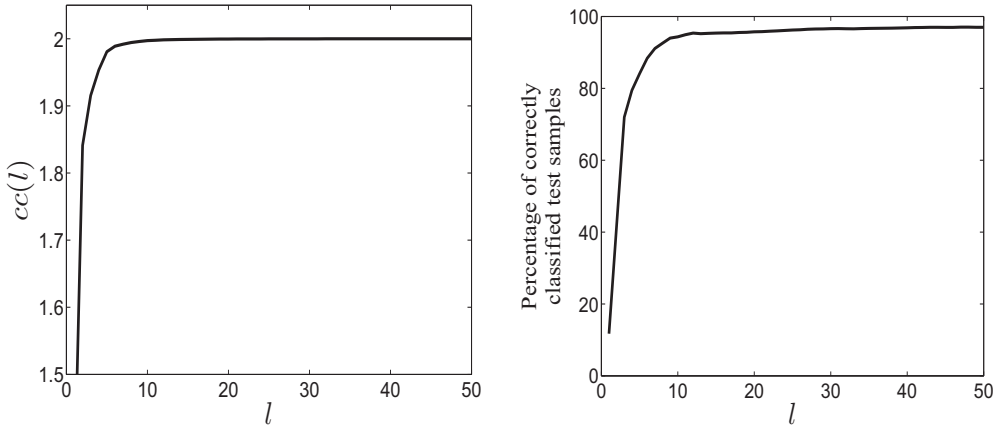


Figure 4.2: A pair of two samples of two of the subjects chosen from AT&T-Olivetti database [54].

4.4.2 Columbia Object Image Library Database

The Columbia Object Image Library (COIL-20) database [55] contains 20 objects each with 72 images corresponding to 72 different orientations. Figure 4.4 shows a

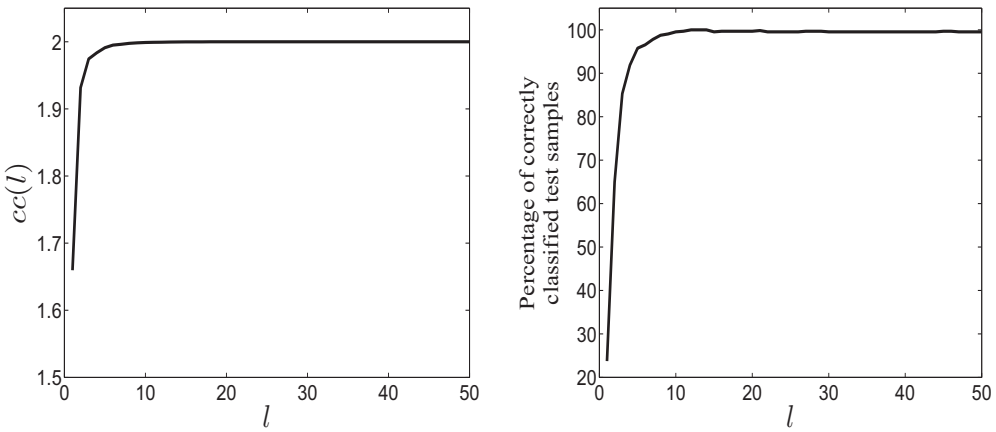


(a) (b)

Figure 4.3: AT&T-Olivetti database. (a) Combined criterion as a function of the reduced number of features. (b) Correctly classified samples as a function of the reduced number of features.

sample image of each of the objects in the database. The images in this database have a resolution of 128×128. Of the 72 images for each object, 40 are chosen for training the system and the remaining 32 are used as test images. The selection of the test set is done randomly. Figure 4.5 shows the plots of the combined criterion cc and the percentage of correctly classified test points, both as functions of l, the reduced number of features. It is seen from this figure that a high level of correspondence exists between the criterion and the classifier efficiency.

Figure 4.4: A sample from each of the 20 objects COIL-20 database [55].



(a) (b)

Figure 4.5: COIL-20 database. (a) Combined criterion as a function of the reduced number of features. (b) Correctly classified samples as a function of the reduced number of features.

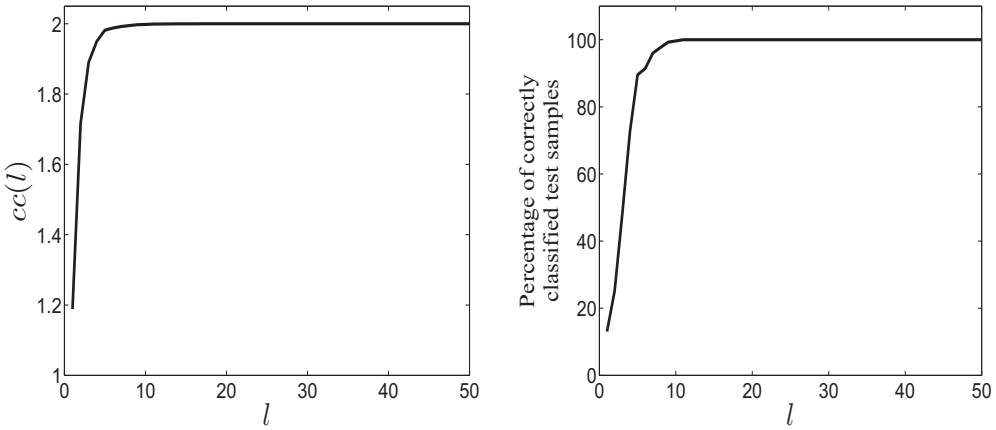
By applying Algorithm 2, the number of features is found to be 14. With this number of features, 99.5% of the images are classified correctly. The best classification efficiency attainable is 99.8% which is achievable with 31 features. Increasing the number of features to 800, the number of training samples, results in a classification accuracy of 98.9%. The slight reduction in the accuracy is attributed to the inclusion of eigenvectors that mainly represent the noise in the data. Computation of the proposed criterion needs 40 seconds.

4.4.3 MIT-CBCL Face Database

The MIT Center for Biological and Computational Learning (CBCL) face recognition database [41] contains face images of ten subjects. The images of each person consist of nine different poses and 36 levels of illuminations per pose. The total number of images is $3,240$ each of size 200×200. The number of training images is 216 for each cluster or $2,160$ in total. The number of test images per class is 108. Figure 4.6 gives several examples of the faces of two subjects chosen from this database showing different poses and lighting conditions. For each number of features, $cc(\cdot)$ is computed

among the ten classes (subjects). Figure 4.7(a) shows a curve where we have plotted the cc versus l, the number of features retained, whereas Figure 4.7(b) presents the percentage of correctly classified test points versus l. As in the case of the previous two databases, the classification accuracy improves as the number of retained features increases and there is a strong similarity in the saturation behavior, between the two plots in Figure 4.7. Applying Eq. (4.9) gives 13 as the number of features that need to be retained. Using these 13 features, the classification accuracy is 100%. The computation of the criterion takes about 50 seconds.

Figure 4.6: A few samples from MIT-CBCL face database [41]. For the top subject, the pose varies; whereas for the bottom one, illumination is altered.



(a) (b)

Figure 4.7: The MIT-CBCL face database. (a) Combined criterion as a function of the reduced number of features. (b) Correctly classified samples as a function of the reduced number of features.

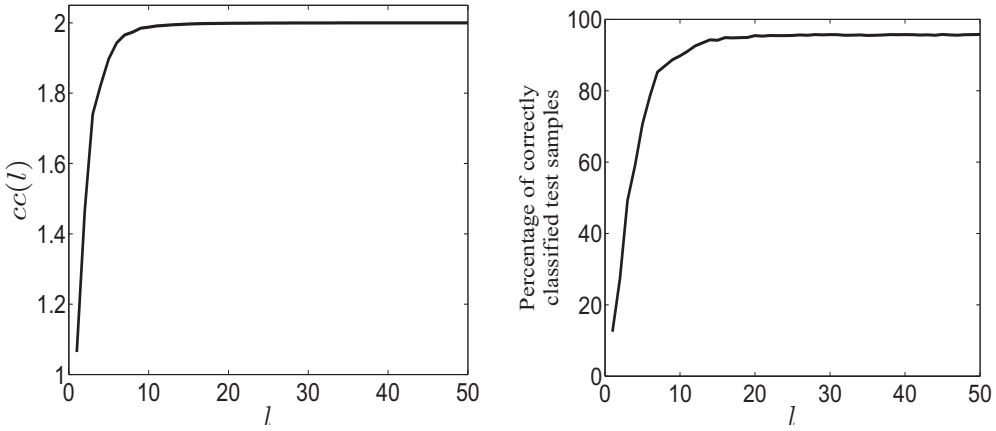
4.4.4 MNIST Handwritten Digit Database

MNIST handwritten digits database of the US National Institute of Standards and Technology (NIST) [56] is the last database considered in our experiments. There are $60,000$ training samples and $10,000$ test samples in this database. In order to reduce the time of the experiment, we use only $18,000$ samples for training, that is, for eigen analysis of the covariance matrix and only $3,000$ of the test samples for determining the effectiveness of the proposed criterion. Figure 4.8 shows several samples chosen from this database. The size of the samples is 28×28. Figure 4.9 shows the graph of $cc(\cdot)$ and the result of classification. With 22 features, $cc(\cdot)$ is completely saturated. This can also be observed from Figure 4.9. For this number of features, the classification accuracy is 95.5%. The maximum achievable accuracy, which is 96.0%, is attained with 78 features. This is only a minimal improvement in accuracy over that when only 22 features are used. Computing the criterion takes 10 seconds of CPU time.

Figure 4.8: A few samples from the MNIST handwritten database [56].

4.5 Robustness

In this section, we investigate the robustness of the proposed algorithm for determining the number of features when the image samples are corrupted by noise. For this



(a) (b)

Figure 4.9: MNIST database. (a) Combined criterion as a function of the reduced number of features. (b) Correctly classified samples as a function of the reduced number of features.

purpose, we conduct two sets of experiments on COIL-20 database. In the first set, the image samples are corrupted with Gaussian noise to provide a signal-to-noise ratio (SNR) in the range of 13 dB to 30 dB in steps of 1 dB, whereas in the second set, impulsive (salt and pepper) noise is added to provide SNR in the range of 3 dB to 26 dB, again in steps of 1 dB.

In the first set of experiments, it is observed that for Gaussian noise of SNR level of 23 dB or higher, the proposed algorithm is able to correctly determine L, the number of features needed to be retained, and at the same time, the classification accuracy remains very close to that at SNR=∞. The value of L is 14 for SNR$\geq$ 23 dB. For SNR level between 22 dB and 19 dB, the predicted number of features grows from 14 to 18, and the classification efficiency stays intact. However, the classifier loses its accuracy if the number of features employed is more than L. For SNR values below 18 dB, the number of features to be retained increases rapidly and the classification efficiency decreases sharply. For example, at SNR value of 17 dB, the algorithm predicts 174 number of features. At SNR value of 12 dB, the algorithm is not able to determine L any more. Figures 4.10(a) and (b), respectively, show cc and the classification efficiency as functions of l at SNR levels of 23 dB, 18 dB, 16 dB and 12

dB, summarizing the above discussion. The results of the second set of experiments, that is, when the corrupting noise is impulsive, are depicted in Figure 4.11. It is seen from this figure that the four levels of performance discussed for Gaussian noise (Figure 4.10) are achieved at SNR levels of 15 dB, 5 dB, 4 dB and 2 dB, respectively, in the case of impulsive noise.

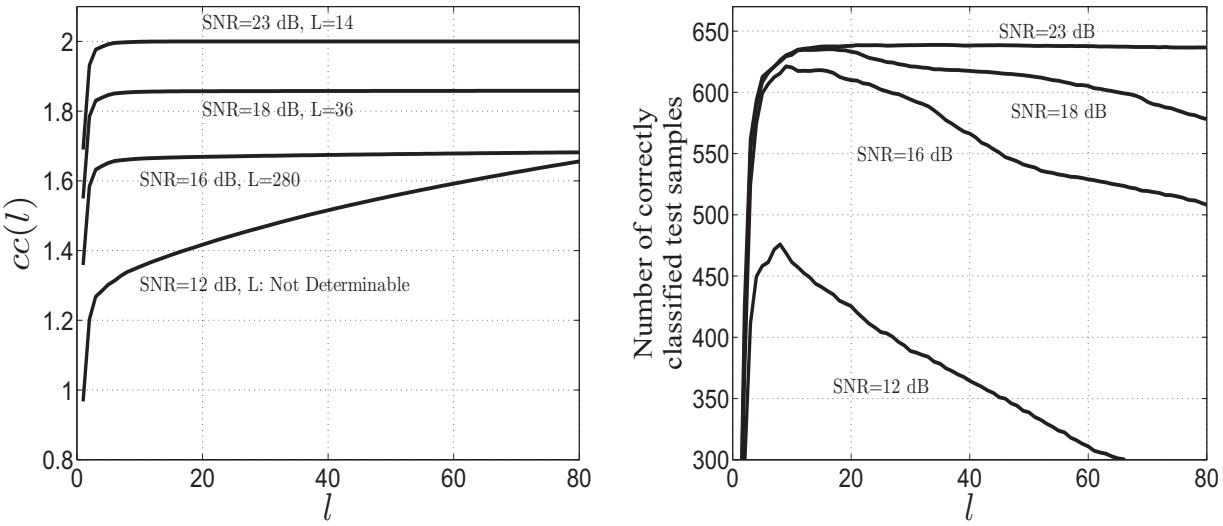


Figure 4.10: The influence of Gaussian noise on the proposed algorithm at four different SNR levels. (a) Cumulative global mean distance as a function of the reduced number of features. (b) The classification accuracy as a function of the reduced number of features.

The above experiments have thus shown that the proposed algorithm is quite robust in determining the required number of features to be retained when the SNR level of the corrupted images is as low as 19 dB in the case of Gaussian noise and 5 dB in the case of impulsive noise.

4.6 Other Measures for Determining L

In Section 2.4 the global versions of Mahalanobis and Bhattacharyya distances (Eqs. (2.33) and (2.34)) were introduced. In view of the discussions of Sections 4.1 and 4.2, it may be interesting to study the applicability of the cumulative versions of these

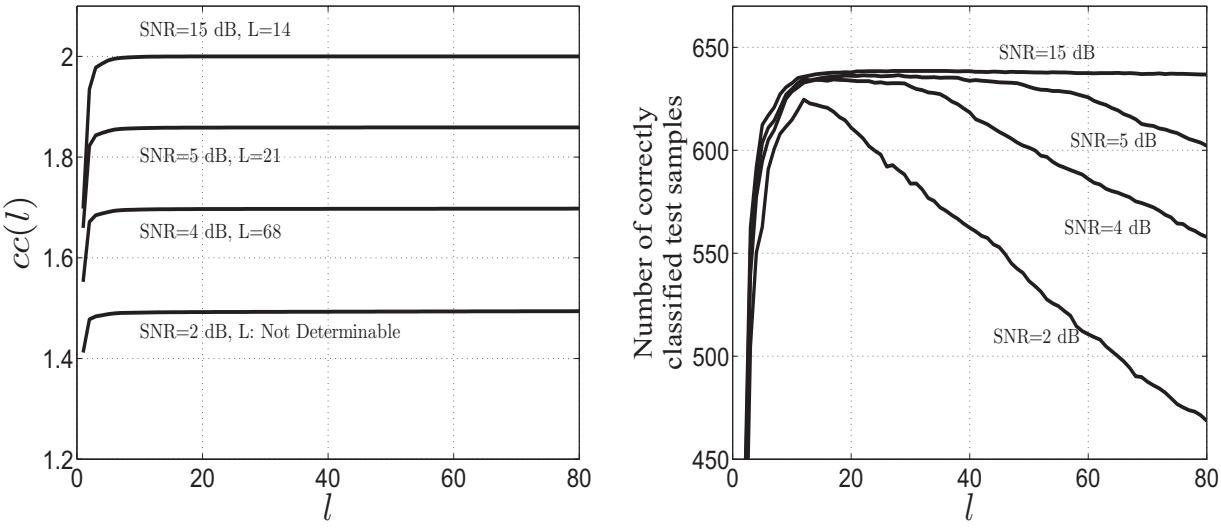


Figure 4.11: The influence of impulsive noise on the proposed algorithm at four different SNR levels. (a) Cumulative global mean distance as a function of the reduced number of features. (b) The classification accuracy as a function of the reduced number of features.

distance measures for determining the number of features that need to be retained. The cumulative versions are given by

$$CMGD(l) = \sum_{k=1}^{l} MGD(k) \tag{4.10}$$

$$CBGD(l) = \sum_{k=1}^{l} BGD(k) \tag{4.11}$$

An experiment is carried out for computing these cumulative distance measures for the Columbia Object Image Library (COIL-20) database [55]. Figures 4.12(a) and 4.12(b) show the plots of the two cumulative distance measures as functions of the number of features retained. It is seen from these plots that both these measures grow sharply and that there is no correspondence between their behavior and that of the classification efficiency given in Figure 4.5(b). Therefore, the cumulative versions of the Mahalanobis and Bhattacharyya distances cannot be used to determine the number of features to be retained for the classifiability of the data clusters.

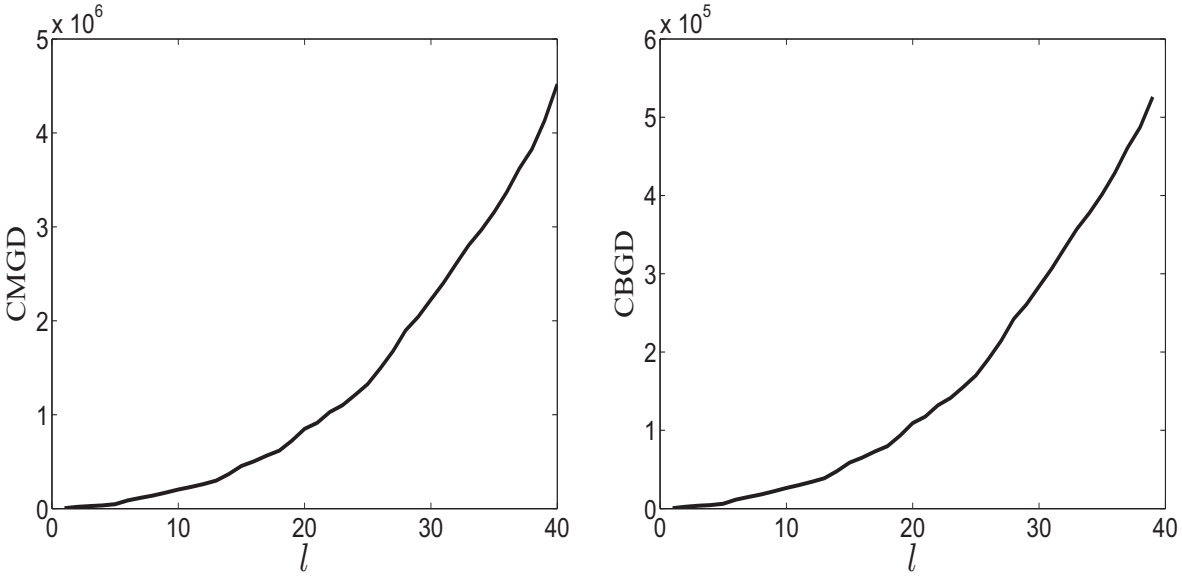


Figure 4.12: Cumulative global distance measures for COIL-20 database. (a) Mahalanobis cumulative global distance as a function of the reduced number of features. (b) Bhattacharyya cumulative global distance as a function of the reduced number of features.

4.7 Summary

In this chapter, two criteria for accurately determining a reduced dimensionality of the feature vectors generated by the process of principal component analysis have been developed. The first criterion, the cumulative global mean distance, is based on preserving a reasonable distance between the mean vectors of each pair of clusters within a database as the dimensionality of the training samples is reduced. However, for some databases, where such a distance preservation is achievable, the individual training vectors from different clusters may still get overlapped as their dimensionality is reduced. For this reason, a second criterion called cumulative global sample scattering distance has been introduced. This criterion is based on expressing the entire set of training vectors of all the clusters as a linear combination of a certain number of eigenvectors of the covariance matrix of the training vectors. Using a combination of these two criteria, an algorithm has been developed for determining the number

of features that need to be retained for a reasonable classifiability of the clusters. To demonstrate the effectiveness of the proposed algorithm, several experiments have been carried out using AT&T/Olivetti face database, Columbia Object Image Library (COIL-20) database, MIT-CBCL face database and MNIST handwritten digit database. The proposed algorithm is computationally inexpensive and exhibits a much better performance compared to that obtained by using some of the traditional methods such as scree diagram. In order to study the robustness of the proposed algorithm, experiments have been performed by adding different levels of Gaussian and impulsive noise to the original images. It has been seen from these experiments that the algorithm remains resilient against the additive noise. By performing a set of experiments, it has also been shown that the cumulative global version of Bhattacharyya and Mahalanobis distance measures cannot be used for determining the number of features that need to be retained.

Chapter 5

Feature Selection in KPCA Domain

Kernel based classification methods are employed when the traditional linear techniques such as principal component analysis do not provide acceptable classification accuracy for a database in which the original clusters of samples are non-linearly distributed. Kernel based methods can be used to make the different classes of samples of such a database to get separated in different clusters by mapping the data vectors to a space of a higher dimensionality than that of the original space of the vectors. In the new space, called *kernel space*, the classification efficiency could be improved at the expense of a larger number of features. On the other hand, the need for determining the number of features in this new space remains a valid concern.

In Chapter 4, two criteria for determining the number of features that need to be retained for a reasonable classifiability were introduced when the principal component analysis was used for generating the features. These criteria, namely, the cumulative global mean distance and the cumulative global sample scattering distance, can be suitable candidates for determining the number of features in a kernel space. However, a direct computation of these criteria for determining the number of features in the

kernel space may not be possible, since the data samples in this space are not explicitly available. In this chapter, a method for computing the cumulative global mean distance in the kernel space is presented [61]. The applicability and the robustness of the proposed method is demonstrated by carrying out several experiments.

This chapter is organized as follows. Section 5.1 shows as to how the cumulative global mean distance can be evaluated in kernel space. Section 5.2 explains how it is possible to find the saturation point of this criterion. The saturation point of this criterion is shown to be the number of features that need to be retained for a reasonable classifiability. Section 5.3 presents a formal algorithm for determining this number when the features are generated by kernel principal component analysis. To validate the ideas developed in Sections 5.1 - 5.3, two experiments are conducted and their results presented in Section 5.4. Section 5.5 studies the robustness of the proposed algorithm in the presence of Gaussian and impulsive noise. Section 5.6 concludes the chapter.

5.1 Cumulative Global Mean Distance in Kernel Space

Since the cumulative global mean distance ($CGMD$) focuses mainly on preserving the distance between the mean vectors of the clusters in a database as the dimensionality of the samples in the space of the database is reduced, it makes sense to apply it in a space in which the various classes of the database are separated. In this section, it is shown how $CGMD$ can be computed in a kernel space. The objective here is to re-express the $CGMD$ in a form that can readily be used in the kernel space. Since the individual sample values in the kernel space are not explicitly available, the expression for $CGMD$, as developed in Section 4.1, cannot be used directly. However, since in the kernel space, we have available to us the inner products of the pairs of the data

vectors, this criterion needs to be re-expressed in a form involving the inner products. That is to say that the so called *kernel trick*, explained in Section 2.3, needs to be used in transforming the original expression for $CGMD$ into a form that can be used in the kernel space.

Similar to Eq. (4.3), the cumulative global mean distance is defined in kernel space as

$$\zeta(l) = \sum_{k=1}^{l} \left[\sum_{i=2}^{c} \sum_{j=1}^{i-1} D(i,j,k) \right], \quad l = 1, 2, \cdots, M \tag{5.1}$$

where c is the number of clusters in the database and $D(i,j,k)$, the main component of $\zeta(\cdot)$ is given by

$$D(i,j,k) = \left[\lambda_k \langle \left(\boldsymbol{\Omega}^i - \boldsymbol{\Omega}^j \right), \boldsymbol{V}^k \rangle \right]^2 \tag{5.2}$$

and is referred to as the *distance element* of $CGMD$. In the above expression for the distance element, $\boldsymbol{V}^k$ is the eigenvector corresponding to λ_k, the k^{th} largest eigenvalue of the covariance matrix of training vectors in the kernel space, and $\boldsymbol{\Omega}^i$ is the mean vector of the i^{th} cluster in that space, that is,

$$\boldsymbol{\Omega}^i = \frac{1}{M_i} \sum_{m=1}^{M_i} \boldsymbol{X}^m = \frac{1}{M_i} \sum_{m=1}^{M_i} \phi(\boldsymbol{x}^m) \tag{5.3}$$

where M_i is the number of training vectors in the i^{th} cluster, i.e., $\sum_{i=1}^{c} M_i = M$, and $\phi(\cdot)$ is the nonlinear function for mapping $\boldsymbol{x}^m \in \Re^N$ to the kernel space. Direct computation of $D(i,j,k)$ is not possible, since the values of the mapped data vectors $\boldsymbol{X}^m = \phi(\boldsymbol{x}^m),\ m = 1, ..., M$ are not known explicitly. However, as shown next, it is possible to compute this quantity using the kernel trick.

The key term in the computation of $\zeta(\cdot)$ is the following (see Eq. (5.2)):

$$\langle \boldsymbol{\Omega}^i, \boldsymbol{V}^k \rangle \tag{5.4}$$

Note that the above expression represents the projection of the i^{th} mean onto the k^{th}

eigenvector in the kernel space. Considering Eq. (2.22), the above mentioned inner product can be expressed as

$$\left\langle \boldsymbol{\Omega}^i, \boldsymbol{V}^k \right\rangle = \frac{1}{M_i} \left\langle \sum_{m=M_{is}}^{M_{ie}} \phi(\boldsymbol{x}^m), \sum_{n=1}^{M} \alpha_n^k \phi(\boldsymbol{x}^n) \right\rangle \tag{5.5}$$

where $\boldsymbol{x}^{M_{is}}, \boldsymbol{x}^{M_{is}+1}, ..., \boldsymbol{x}^{M_{ie}}$ are the data vectors comprising the i^{th} cluster having M_i data points and α_n^k is the n^{th} element of the k^{th} eigenvector of the feature matrix $\boldsymbol{K}$ (see Eq. (2.24)). After some manipulation, Eq. (5.5) can be re-expressed as

$$\begin{aligned} \left\langle \boldsymbol{\Omega}^i, \boldsymbol{V}^k \right\rangle &= \frac{1}{M_i} \sum_{n=1}^{M} \alpha_n^k \sum_{m=M_{is}}^{M_{ie}} \left\langle \phi(\boldsymbol{x}^n), \phi(\boldsymbol{x}^m) \right\rangle \\ &= \frac{1}{M_i} \sum_{n=1}^{M} \alpha_n^k \sum_{m=M_{is}}^{M_{ie}} K_{mn} = \frac{1}{M_i} \left\langle \boldsymbol{\xi}^i, \boldsymbol{\alpha}^k \right\rangle \end{aligned} \tag{5.6}$$

where $\boldsymbol{\xi}$ is defined as

$$\boldsymbol{\xi}^i = \left(\sum_{m=M_{is}}^{M_{ie}} K_{m1}, ..., \sum_{m=M_{is}}^{M_{ie}} K_{mM} \right)^T \tag{5.7}$$

Using Eq. (5.6), the measure in the kernel space corresponding to Eq. (5.2) can be obtained as

$$D_F(i,j,k) = \lambda_k^2 \left| \frac{1}{M_i} \langle \boldsymbol{\xi}^i, \boldsymbol{\alpha}^k \rangle - \frac{1}{M_j} \langle \boldsymbol{\xi}^j, \boldsymbol{\alpha}^k \rangle \right|^2 \tag{5.8}$$

If $M_i = M_j = \frac{M}{c}$, the above equation can be further simplified as

$$D_F(i,j,k) = \lambda_k^2 \left(\frac{c}{M} \right)^2 \left| \left\langle \left(\boldsymbol{\xi}^i - \boldsymbol{\xi}^j \right), \boldsymbol{\alpha}^k \right\rangle \right|^2 \tag{5.9}$$

From this development, the computation of the distance element of $CGMD$ becomes straightforward. Now, with the availability of Eq. (5.8), the inner product in (5.4) can be computed only by using the elements of the matrix $\boldsymbol{K}$ and its eigenvectors $\boldsymbol{\alpha}^k$. Since the $\boldsymbol{\xi}$ vectors are computed only once using Eq. (5.7), the evaluation of $\zeta(\cdot)$ does not require a huge computational cost.

5.2 Search for Saturation of $\zeta(\cdot)$

If the difference between $\zeta(l)$ and $\zeta(l+1)$ is negligible, it means that employing an additional eigenvector $\boldsymbol{\alpha}^{l+1}$ only increases the length of the compressed vectors without contributing to the separability of the clusters. This means that the curve for $\zeta(l)$ gets saturated beyond this value of l. A minimum number of coefficients, L, of the compressed vectors that must be calculated can then be determined by ensuring the condition

$$(\zeta(l+1) - \zeta(l)) \leq \beta\zeta(M) \tag{5.10}$$

where β is a small pre-specified positive number. In order to avoid a false determination of the number of coefficients to be calculated due to some local saturation of ζ versus l plot, the saturation condition is modified as

$$(\zeta(l+j) - \zeta(l)) \leq \beta\zeta(M), \quad j = 1, \cdots, J \tag{5.11}$$

where $J > 1$ is a positive integer.

5.3 Algorithm

In Sections 5.1 and 5.2, we have provided a method of determining a reduced number of features. The various steps of this method are now formally given as Algorithm 3.

In this algorithm, the values of the parameters β and J need to be specified, since they are used in Eq. (5.11). To determine the value of β, an experiment of running the algorithm repeatedly is conducted. It is found that, in general, for values of $\beta \leq 0.0005$, more features than what is necessary are retained, and for $\beta \geq 0.005$, the algorithm gives an insufficient number of features. The value of this parameter is chosen to be 0.001. We have chosen $J = 15$, a value that adequately ensures that a steady state of ζ with respect to l is achieved.

Algorithm 3 : Determining the Number of Features in Kernel Space

1. Normalize all the vectors of the database to have length of unity.
2. Compute the matrix $\boldsymbol{K}$ by applying a kernel function on the training vectors $\boldsymbol{x}^i,\ i = 1, 2, \cdots, M$
3. Perform the eigen analysis on $\boldsymbol{K}$ to obtain λ_k and $\boldsymbol{\alpha}^k$.
4. Set $l \leftarrow 1$.
5. Compute $\zeta(l)$ using Eq. (5.1).
6. $l \leftarrow l + 1$; if $l < M$, go to step 5.
7. Set $l \leftarrow 1$.
8. If Eq. (5.11) is satisfied, set $L \leftarrow l$ and terminate.
9. Set $l \leftarrow l + 1$;
10. If $l > S^2$, L cannot be determined, terminate.
11. Go to step 8.

Although the proposed algorithm is independent of the type of kernel chosen, we conduct experiments with radial basis function (RBF) kernels. With this type of kernel, the elements of the matrix $\boldsymbol{K}$ are computed as [44]

$$K_{ij} = \left\langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j) \right\rangle = exp\left(-\frac{\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2}{2\sigma^2} \right) \tag{5.12}$$

where σ is a parameter of the kernel function used.

5.3.1 Complexity Analysis of Algorithm 3

In this section, the computational cost of Algorithm 3 is studied. First the complexity of computing kernel matrix and that of its eigen analysis is considered, and then the complexity of the part of algorithm concerning the computation of the proposed criterion and finding L, the number of features that need to be retained, is studied.

Assuming the number of addition and multiplication operations needed for computation of the kernel function once to be t, computing the kernel matrix requires tM^2 operations. The complexity of the eigen analysis of this matrix is $O(M^3)$ [60]. Therefore, the overall complexity of performing the kernel eigen analysis is $O(M^3)$ for sufficiently large values of M.

Next, it is noted that evaluating $\zeta(l),\ l = 1, 2, \cdots, M$, involves computation of the vectors $\boldsymbol{\xi}^i,\ i = 1, 2, \cdots, c$ using Eq. (5.7), where c is the number of distinct classes in the database. According to this equation, computing these c vectors requires $M \sum_{i=1}^{c} M_i = M^2$ additions. With the vectors $\boldsymbol{\xi}^i$ now available, computing Eq. (5.8) requires $2M + 2$ multiplications and a single subtraction. Computing $\zeta(l),\ l = 1, 2, \cdots, M$, involves $M\left[\frac{c}{2}(c-1)\right]$ computations of Eq. (5.8). Therefore, assuming $M \gg 1$, the computation of $\zeta(\cdot)$ in the kernel space requires $2M^2\left[\frac{c}{2}(c-1)\right]$ operations. This means that the computational complexity of the part of Algorithm 3 for determining L is $O(M^2c^2)$.

The overall complexity of Algorithm 3 is $O(M^3 + M^2c^2)$. However, it has to be noted that the kernel eigen analysis with a complexity of $O(M^3)$ needs to be carried out regardless of the method used for determining the number of features that need to be retained.

5.4 Implementation

The proposed method is applied for feature reduction in two databases: United States Postal Service (USPS) handwritten digit database and Yale Face database. Each database is divided into two mutually exclusive sets, *training set* and *test set*. The training set is used to compute the matrix $\boldsymbol{K}$. The computation of the elements of $\boldsymbol{K}$ requires the images to be in a vector form. To satisfy this requirement, the images of the training set are unfolded into N-dimensional vectors.

Algorithm 3 is then applied to determine L, the number of features to be retained. Note that L cannot be more than M, the number of images in the training set. The set of the compressed test vectors, obtained using Eq. (2.28), is used in a classifier to demonstrate the ability of the proposed approach in determining the right number of features. The classifier used here is a k-NN (k nearest neighbor) classifier with $k = 5$. Euclidean distance is used to measure the distance measure between the test vector in question and a training vector. The value of the kernel parameter σ used in the kernel function given by Eq. (5.12) is empirically determined to be $\frac{\sqrt{2}}{2}$, since this value provides the best classification accuracy with the k-NN classifier employed in our experiments.

The classification efficiency is computed using l features, $l = 1, \cdots, M$. As will be seen from the experiments, the classification accuracy does not necessarily increase by the inclusion of each additional feature, but in general, it is a non-decreasing function of the number of features. Also, the results of the KPCA experiments are compared with those of the linear PCA by performing another experiment.

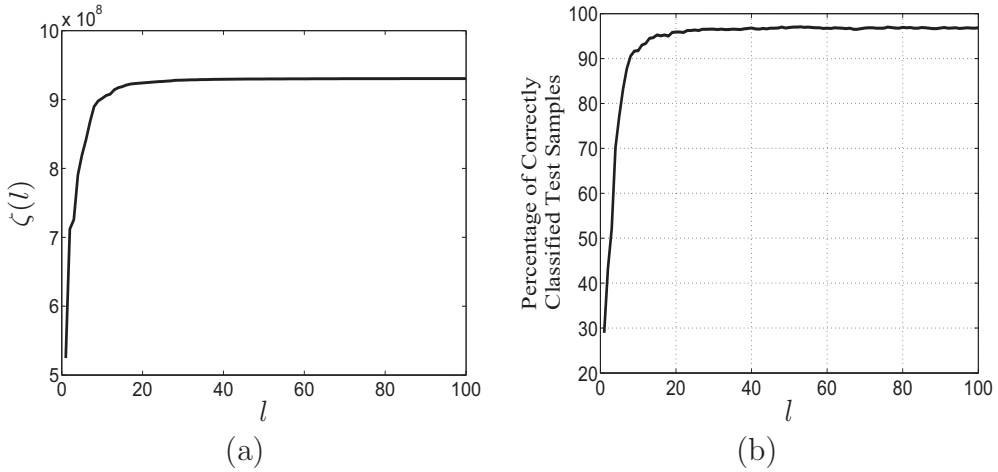
5.4.1 US Postal Service Handwritten Digit Database

The United States Postal Service digit database [62] is a collection of 9,298 handwritten digits collected from the mail envelopes. Among them, 7,291 are training samples and the rest are labeled to be test samples by the authors of the database. Each sample is a 16×16 ($N = 256$) binary (black and white) image. Figure 5.1 shows some of the samples of this database. From these samples, $M = 7,000$ samples are taken from the training set for the purpose of computing $\boldsymbol{K}$, the kernel matrix. In order to validate the usefulness of Algorithm 3, 2,000 images are classified from the set of test samples.

The pixel values of the raw images of the database are re-arranged into 256-dimensional vectors and then normalized to have a unit norm. Figure 5.2 shows the

Figure 5.1: A few samples from the USPS handwritten digit database.

plot of ζ and that of the percentage of the correctly classified samples as functions of l, the number of features retained. By employing Algorithm 3, we find $L = 35$. With 35 features, 96.6% of test vectors are classified correctly. The maximum classification accuracy is attained by using 48 features. With 48 features, the percentage of the correctly classified test vectors will be 97.1%. With 256 features, an accuracy of 96.8% is achieved.



(a) (b)

Figure 5.2: USPS handwritten digit database. (a) Cumulative global mean distance as a function of the number of features retained. (b) Classification accuracy as a function of the number of features retained.

We now repeat the experiment using the linear PCA. The classification results are depicted in Figure 5.3. With the linear PCA, ζ gets saturated at $L = 15$. With 15 features, 95.0% of the test vectors are classified correctly. The additional number of features required by KPCA could be justified in view of the ability of the kernel

method to provide improved efficiency.

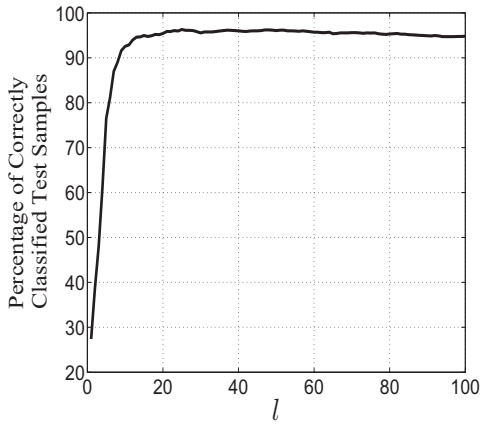


Figure 5.3: Classification results for USPS handwritten digit database using PCA features.

5.4.2 Yale Face Database

The Yale Face database [63] contains the gray-scale images of 15 individuals with 11 samples per subject. These images have been taken with variations in facial expressions, eyeglasses and lighting conditions. The images are in *tagged image file* format (TIFF) with a resolution of 243×320 pixels. The raw images of the database are unfolded into (N=243×320)-dimensional vectors and then normalized to have a unit norm. The image samples of one of the subjects in this database are shown in Figure 5.4. From the 11 images in each cluster, seven are chosen for training and the rest for testing. This results in an overall set of $M = 105$ training images and 60 test images.

The results of this experiment are shown in Figure 5.5. Figure 5.5(a) shows ζ and Figure 5.5(b) the percentage of correctly classified test samples as functions of l. Applying Eq. (5.11) gives $L = 20$ as the number of features to be retained. With 20 features, 76.24% of the test samples are classified correctly.

For the purpose of comparison, a classification experiment is also performed using the linear PCA as feature generating technique. The result of this experiment is shown in Figure 5.6. For PCA, ζ gets saturated at $L = 13$. With the feature vectors

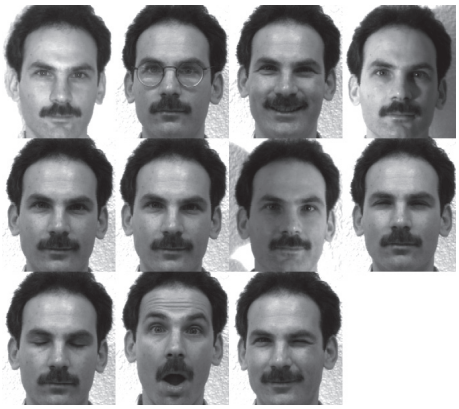


Figure 5.4: All the image samples of one of the subjects from Yale Face database [63].

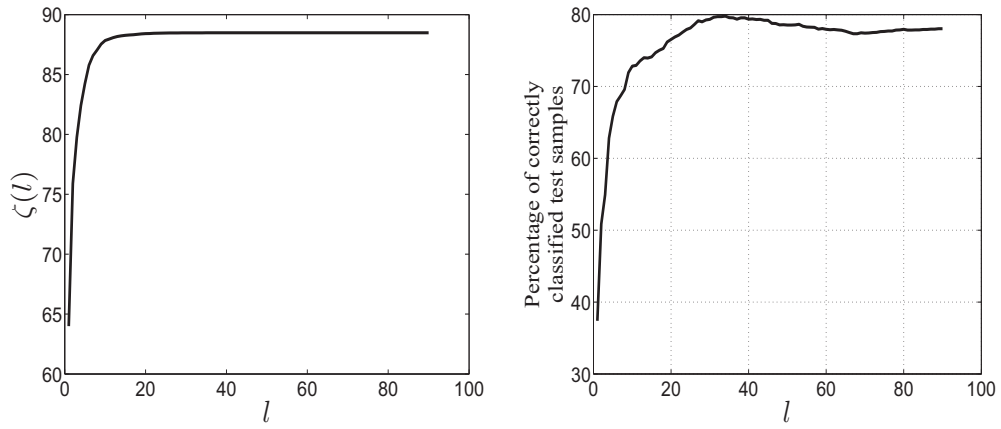


Figure 5.5: Yale Face database. (a) Cumulative global mean distance as a function of the number of features retained. (b) Classification accuracy as a function of the number of features retained.

reduced to $\Re^{13}$, 73.87% of the test samples are classified correctly. In KPCA, the use of additional features over that in PCA, improves the classification accuracy of the former by 2.37%.

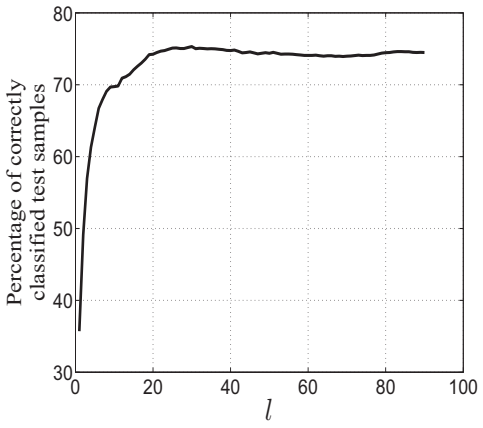


Figure 5.6: Classification results for Yale Face database using PCA features.

5.4.3 Caltech 101 Data Set

The Caltech 101 database [57] comprises 101 categories each containing a number of samples ranging from 31 to 800. A few of these samples are shown in Fig. 5.7. The huge number of images in this database makes it difficult to conduct a classification using all the data. Therefore, we select 2,000 images for training and 1,000 other images for testing the proposed algorithm. These images are drawn from a number of randomly selected classes. The sizes of different images in this database are not the same. Since the algorithm devised in this chapter requires that all the samples to have the same size, they are all zero-padded to become 512×512. As in the experiments involving other databases, here too, we use an RBF kernel with $\sigma = \frac{\sqrt{2}}{2}$.

Fig. 5.8 shows the plot of ζ and that of the classification accuracy of the test vectors as functions of l, the reduced number of features. For this database, when all $2,000$ features are used, only 29.59% of the test images are classified correctly. Applying Algorithm 3 to this database, gives $L = 24$. With 24 features, a classification

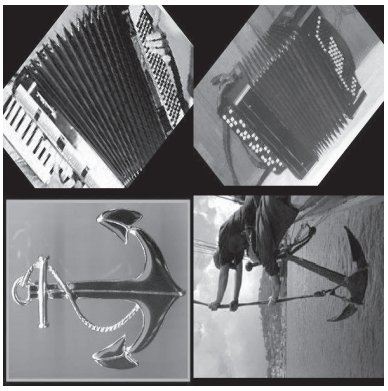


Figure 5.7: Four samples chosen from two of the classes of Caltech 101 database [57].

accuracy of 34.2% is attained.

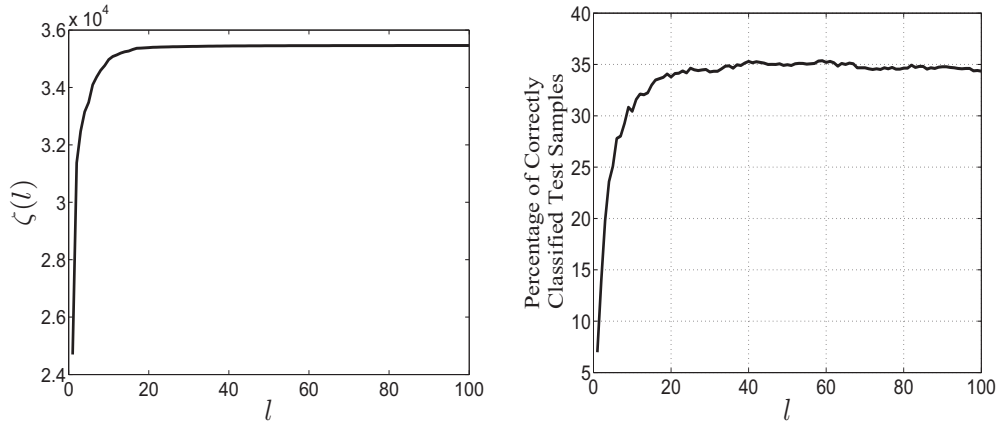


Figure 5.8: Caltech 101 database. (a) Cumulative global mean distance as function of l, the reduced number of features. (b) Correctly classified samples as function of l, the reduced number of features.

To compare the above classification results with those of the linear PCA, another experiment is performed using the PCA generated features. Fig. 5.9 shows the results of this experiment. For PCA, ζ is saturated at $L = 12$. With 12 features, 31.8% of the test samples are classified correctly. With KPCA, an improvement of 2.4% in classification accuracy over that with PCA is attained at the expense of 12 additional features.

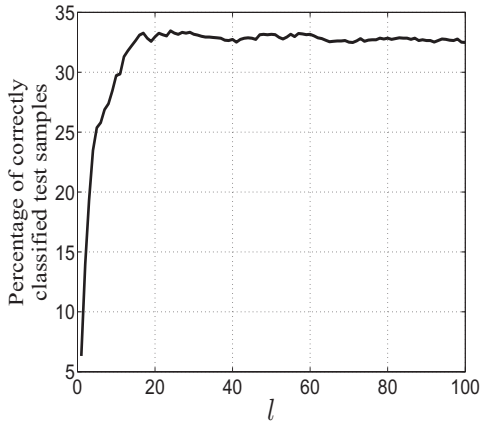


Figure 5.9: Classification results for Caltech 101 database using PCA features.

5.5 Robustness

In this section, the robustness of the proposed algorithm for determining the number of features is investigated when the image samples are corrupted by noise. For this purpose, two sets of experiments are conducted on USPS handwritten digit database. In the first set, the image samples are corrupted with Gaussian noise to provide a signal-to-noise ratio (SNR) in the range 10 dB to 30 dB in steps of 1 dB, whereas in the second set, impulsive (salt and pepper) noise is added to provide SNR in the range 0 dB to 30 dB, again in steps of 1 dB.

In the first set of experiments, it is observed that for the Gaussian noise of SNR levels of 24 dB or higher, the proposed algorithm is able to correctly determine L, the number of features needed to be retained, and at the same time, the classification accuracy remains very close to that at SNR=∞. As the SNR level is lowered below 24 dB, it is seen that the algorithm is able to predict the value of L correctly, but the classification efficiency is somewhat decreased as the SNR level approaches 20 dB. For SNR levels between 17 dB and 20 dB, the algorithm provides a value for L which does not correspond to maximum achievable classification accuracy. For SNR levels below 17 dB, $\zeta(\cdot)$ does not converge and hence the algorithm is not able to provide a value for L. Figures 5.10(a) and (b) show ζ and the classification efficiency as functions of

l at SNR levels of 24 dB, 20 dB, 17 dB and 15 dB.

The results of the second set of experiments, that is, when the corrupting noise is impulsive, are depicted in Figure 5.11. It is seen from this figure that the four levels of performance corresponding to those in Figure 5.10 are achieved at SNR levels of 12 dB, 5 dB, 2 dB and 1 dB, respectively, in the case of impulsive noise.

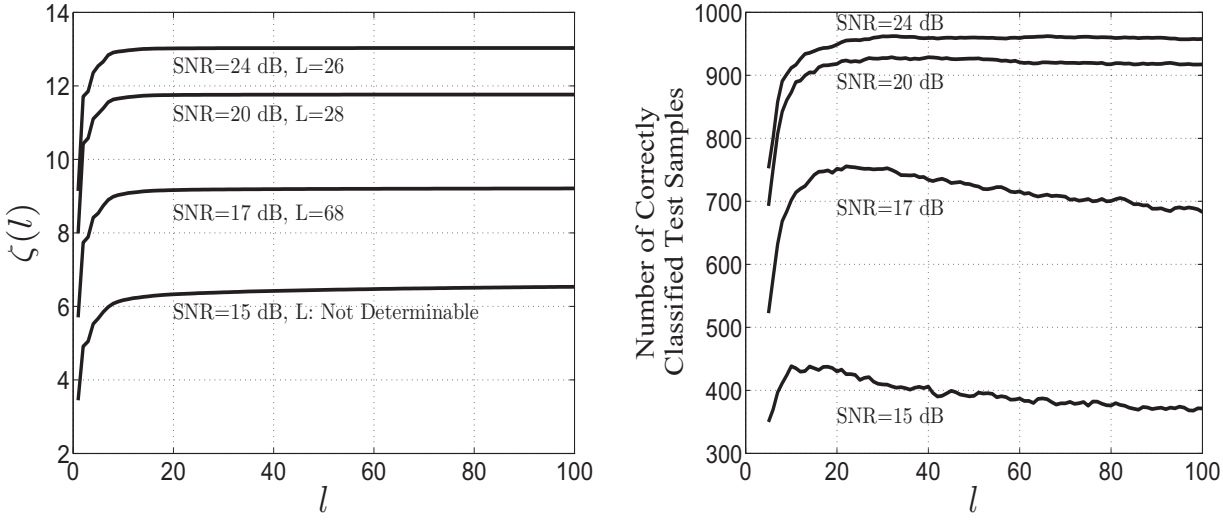


Figure 5.10: The influence of Gaussian noise on the proposed algorithm at four different SNR levels. (a) Cumulative global mean distance as a function of the number of features retained. (b) Classification accuracy as a function of the number of features retained.

The above experiments have thus shown that the proposed algorithm is quite robust in determining the required number of features to be retained when the SNR level of the corrupted images is as low as 20 dB in the case of Gaussian noise and 5 dB in the case of impulsive noise.

5.6 Summary

Even though the principal component analysis (PCA) is one of the most popular feature generation techniques for pattern classification, it can only be employed for

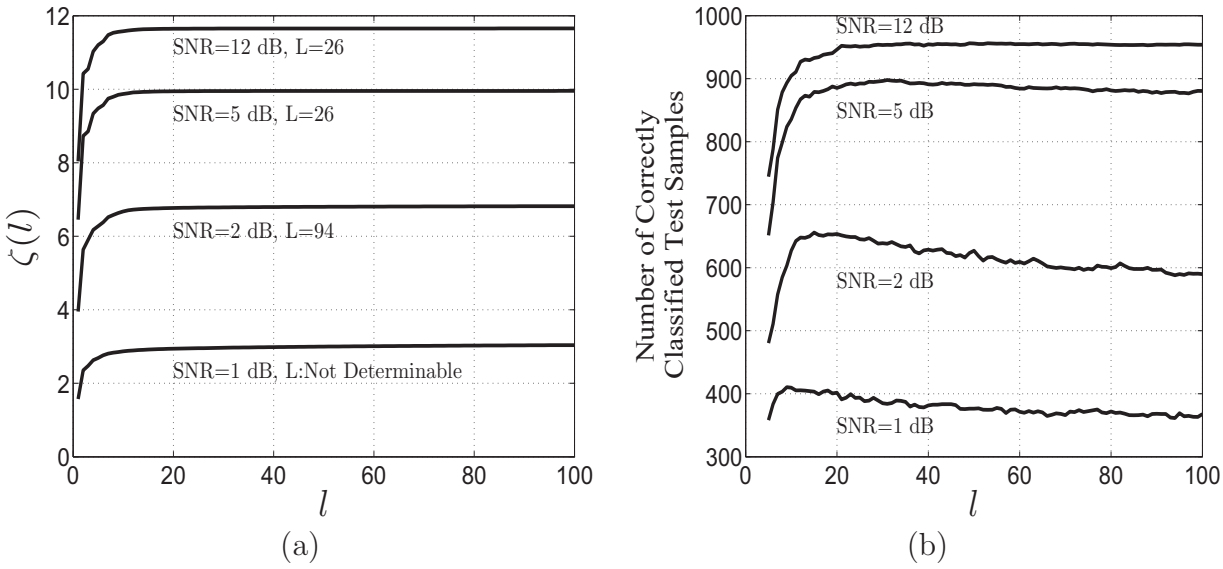


Figure 5.11: The influence of impulsive noise on the proposed algorithm at four different SNR levels. (a) Cumulative global mean distance as a function of the number of features retained. (b) Classification accuracy as a function of the number of features retained.

databases in which the different classes are linearly separable. For data sets of non-linearly separable classes, the kernel version of PCA generally proves to be more useful for feature generation. Determining the number of features that need to be retained in order to maintain a classification accuracy that is close to the maximum has been a major concern in pattern recognition. In Chapter 4, a method for determining the required number of features generated by a linear PCA was proposed. However, this method cannot be extended in a straightforward manner to the kernel space. In this chapter, a criterion and a corresponding algorithm for determining the number of features required to be retained, when a kernel PCA is used for feature generation, has been developed. The proposed algorithm has been applied to USPS handwritten digit, Yale Face and Caltech-101 databases. For all these databases, the proposed algorithm has been shown to predict the correct number of features, that is, the number of features giving the maximum achievable classification accuracy. The results of the experiments have been compared to those obtained by applying linear PCA to the same databases. It has been found that KPCA provides higher classification accuracy

at the expense of retaining a larger number of features. It has also been shown that the new algorithm is robust and is able to determine the number of features correctly in the presence of Gaussian and impulsive noise within certain levels.

Chapter 6

Conclusion

6.1 Concluding Remarks

Feature subset selection and classification algorithms are the main building blocks of pattern recognition tasks. The objective of feature subset selection is to obtain a subset of the original features that provides the maximum possible classification accuracy. An optimum solution to this problem involves an exhaustive search for the most discriminative subset, which can be computationally very expensive. If the features are assigned an order based on their discriminatory importance, this problem is reduced to determining only the number of features that need to be retained. However, the current literature in this area has not provided a satisfactory mechanism to deal with this problem. This thesis has been concerned with developing techniques of determining a minimum number of features necessary for classifiability when the feature vectors are generated by the lossy compression techniques of discrete wavelet transform (DWT), principal component analysis (PCA) and kernel principal component analysis (KPCA). It has been known that when these compression techniques are applied to practical signals, the energy content of the coefficients generated follows a specific pattern. In this thesis, it has been shown that by exploiting this property, one

can determine a minimum number of features that need to be retained. The idea is to maintain the energy of the ensemble of the feature vectors as their dimensionality is reduced.

For each of the three compression techniques mentioned above, a criterion and a corresponding algorithm for determining a minimum number of features required for classifiability have been developed. The proposed criteria are functions of the number of features retained. The main characteristic of these criteria is that the functions representing these criteria get saturated as the number of features retained is increased. The proposed algorithms have employed this characteristic for determining the number of features required for maintaining the separability of the classes of a database. It has been shown that the number of features at which the function representing a criterion attains saturation is the number of features that provides a classification accuracy close to the maximum achievable. Although the algorithms in this study have been developed without assuming a specific data type for databases, they have been applied to image databases.

The objective of the first part of this study has been to determine a minimum number of DWT-generated coefficients to provide a classifiability close to that when all the coefficients are employed. To achieve this goal, a criterion, called partial weighted energy, has been introduced that is computed using the original DWT coefficients and a weighting vector. The elements of the wavelet coefficient matrix of each image are serialized to a vector form by using the Morton scanning, a scanning method that ensures the subbands with high energy concentration to have priority over other sub bands. Next, the resulting vectors are appropriately weighted and used to compute the partial weighted energy. An algorithm has been proposed for determining the number of coefficients that need to be retained by using this criterion.

The second part of this study has been devoted to determining a reduced number of PCA-generated features for classifiability. Two criteria have been introduced that

are based on preserving the distance among the data vectors while their dimensionality is reduced. The first one, called cumulative global mean distance, essentially focuses on the mean vectors of the clusters within the database. Since in a dimensionality reduction task, it is possible that even when the mean vectors are separated the individual data vectors might still get overlapped, a second criterion, called cumulative global sample scattering distance, has also been introduced by expressing the training vectors as a linear combination of a minimum number of eigenvectors of the covariance matrix of these vectors. Finally, an algorithm that uses both these criteria has been presented for a maximum possible classifiability.

Since a PCA-based compression technique cannot be effectively applied in situations in which data clusters are not linearly distributed, the third part of this investigation has aimed at determining a reduced dimensionality of the KPCA-generated feature vectors. The cumulative global mean distance and the cumulative global sample scattering distance can be suitable candidates for solving this problem in a kernel space. However, a direct computation of these criteria is not possible, since the data vectors are not explicitly available in this space. In this part of the study, an expression for the cumulative global mean distance has been obtained in terms of the inner products of the training vectors mapped onto the kernel space by using the kernel trick.

Extensive experiments have been conducted throughout the thesis on several homogenous benchmark databases containing large numbers of images of faces, objects or handwritten digits. The proposed algorithms have been employed for determining the number of features that need to be retained for each database. In order to demonstrate the effectiveness of the proposed algorithms in predicting this number, a classification experiment has also been conducted on these databases using different number of features retained. The classifier employed in the classification experiments is the k-nearest neighbor (k-NN) classifier with $k = 5$, and the Euclidean distance is

used for measuring the distance between a test vector and the training vectors.

For each database, it has been demonstrated that there is a strong correspondence between the proposed criterion and the classification accuracy as the number of features retained is varied. It has been shown that the number of features at which the function representing a criterion attains saturation is a minimum number of features required to be retained for achieving a classification accuracy close to the maximum possible. It has been shown that, this number of required features is only a small fraction, generally less than one percent, of the number of original features. The robustness of the proposed algorithms has been investigated by carrying out a set of experiments on the Gaussian and impulsive noise corrupted data vectors with different levels of signal-to-noise ratio (SNR). The algorithms have been found to be resilient against these two types of additive noise. The proposed algorithms have been shown to have a low computational complexity.

6.2 Scope for Future Investigation

In this section, some of the directions along which the ideas and schemes developed in this study can be further investigated are explored. The focus in this work has been on developing criteria and the corresponding algorithms for determining a reduced number of features that are generated by using the discrete wavelet transform, principal component analysis and kernel version of principal component analysis. It could be worth exploring new or different criteria for determining a reduced number of features for classifiability of data samples when other types of compression schemes, such as the discrete cosine transform, is used. Even though the algorithms developed in this study have been applied exclusively to image databases, the derivation of the criteria has not pre-assumed a particular type for the databases. One could experiment of applying the proposed techniques of dimension reduction to other types of

databases such as speech databases. The classifier used in this thesis has been k-NN with Euclidean distance. One could study the possibility of reducing the number features even further by using other types of classifiers.

The development of the algorithm for determining a reduced number of features generated by a discrete wavelet transform has assumed that the DWT filters are orthonormal. This assumption is to ensure that the Parseval's theorem holds, which is necessary for the development of the criterion. Since other types of filters, such as bi-orthogonal filters, might offer a greater compression, one could look into developing criteria employing other types of filters that do not necessarily observe the Parseval's theorem as the original data get transformed by DWT. Among the existing scanning methods of the DWT matrix, the Morton scanning has been chosen for serializing the wavelet coefficients. One could explore the possibility of using other innovative scanning approaches that, similar to Morton scanning, prioritize the subbands with higher energy.

Recent two-dimensional PCA techniques are known to provide certain advantages over the conventional one-dimensional PCA technique. One could investigate two-dimensional PCA approaches for feature generation and develop a criterion for determining a reduced feature subset.

Feature subset selection in a kernel space has been the final topic dealt with in this study. The experiments of this part have been based on choosing a radial basis function as kernel function. One could also experiment by employing other types of kernel functions with a view of reducing the number of features even further for classifiability.

References

[1] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, New York, NY, June 17–22, 2006, pp. 1735–1742.

[2] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Berlin: Springer, 2002.

[3] M. L. Rayner, 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, no. 2, pp. 164–171, July 2000.

[4] P. Schelkens, A. Skodras, and T. Ebrahimi, *The JPEG-2000 Suite*. Hoboken, NJ: John Wiley & Sons, 2009.

[5] S. Grgic, M. Grgic, and B. Zovko-Cihlar, "Performance analysis of image compression using wavelets," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 3, pp. 682–695, June 2001.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[7] R. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY: John Wiley, 2001.

[8] A. Jain, P. Moulin, M. Miller, and K. Ramachandran, "Information theoretic bounds on target recognition performance based on degraded image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1153–1164, September 2002.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, pp. 389–422, March 2002.

[10] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 796–809, May 2008.

[11] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 253–265, February 2012.

[12] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 17, Cambridge, MA, 2005, pp. 777–784.

[13] U. Kruger, J. Zhang, and L. Xie, "Developments and applications of nonlinear principal component analysis - a review," in *Principal Manifolds for Data Visualization and Dimension Reduction*, ser. Lecture Notes in Computational Science and Engineering, A. N. Gorban, B. Kgl, D. C. Wunsch, and A. Y. Zinovyev, Eds. Springer Berlin Heidelberg, 2008, vol. 58, pp. 1–43.

[14] Y. Choi, T. Tokumoto, M. Lee, and S. Ozawa, "Incremental two-dimensional two-directional principal component analysis (I(2D)2PCA) for face recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Rebublic, May 22–27, 2011, pp. 1493–1496.

[15] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 23, no. 1, pp. 90–93, January 1974.

[16] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: Foundations of JPEG-2000," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 22–35, September 2001.

[17] K. Guo, D. Labate, W. Lim, G. Weiss, and E. Wilson, "Wavelets with composite dilations and their MRA properties," *Applied and Computational Harmonic Analysis*, vol. 20, no. 2, pp. 202–236, 2006.

[18] W. Lim, "The discrete shearlet transform: A new directional transform and compactly supported shearlet frames," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1166–1180, May 2010.

[19] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, October 1993.

[20] J. A. Richards and K. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed. Berlin: Springer, 2006.

[21] P. P. C. Tsui and O. A. Basir, "Wavelet basis selection and feature extraction for shift invariant ultrasound foreign body classification," *Ultrasonics*, vol. 45, no. 1-4, pp. 1–14, December 2006.

[22] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, September 2000.

[23] D. L. Donoho and I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data," in *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Engineering Advances: New Opportunities for Biomedical Engineers*, vol. 1, Baltimore, MD, November 3–6, 1994, pp. A24–A25.

[24] B. A. Rajoub, "An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 4, pp. 355–362, April 2002.

[25] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 429–441, October 1993.

[26] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Transactions on Computers*, vol. 20, no. 2, pp. 176–183, February 1971.

[27] O. Sidla, L. Paletta, Y. Lypetskyy, and C. Janner, "Vehicle recognition for highway lane survey," in *Proceedings of the 7th IEEE International Conference on Intelligent Transportation Systems*, Washington, DC, October 3–6, 2004, pp. 531–536.

[28] J. C. B. Melo, G. D. C. Cavalcanti, and K. S. Guimaraes, "PCA feature extraction for protein structure prediction," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 4, Portland, OR, July 20–24, 2003, pp. 2952–2957.

[29] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu, "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1-2, pp. 321–336, September 2003.

[30] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, July 1997.

[31] J. E. Jackson, *A User's Guide to Principal Components*. New York, NY: John Wiley & Sons Inc., 1991.

[32] A. Vailara, H. Zhang, C. Yang, F. I. Liu, and A. K. Jain, "Automatic image orientation detection," *IEEE Transactions on Image Processing*, vol. 11, no. 7, pp. 746–755, July 2002.

[33] B. Mertens, T. Fearn, and M. Thompson, "The efficient cross-validation of principal components applied to principal component regression," *Statistics and Computing*, vol. 5, no. 3, pp. 227–235, 1995.

[34] D. A. Jackson, "Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches," *Ecology*, vol. 74, no. 8, pp. pp. 2204–2214, 1993.

[35] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–491, March 2005.

[36] W. Liao, A. Pizurica, W. Philips, and Y. Pi, "A fast iterative kernel PCA feature extraction for hyperspectral images," in *Proceedings of the 17th IEEE International Conference on Image Processing*, Hong Kong, September 26–29, 2010, pp. 1317–1320.

[37] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York, NY: Academic Press, 1990.

[38] M. H. Yang, N. Ahuja, and D. Kriegman, "Face recognition using kernel eigenfaces," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, Vancouver, Canada, September 10–13, 2000, pp. 37–40.

[39] Q. Liu, J. Cheng, H. Lu, and S. Ma, "Distance based kernel PCA image reconstruction," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, Cambridge, UK, August 23–26, 2004, pp. 670–673.

[40] K. I. Kim, K. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 40–42, February 2002.

[41] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz, "Component-based face recognition with 3D morphable models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, June 27–July 2, 2004, pp. 85–85.

[42] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, N.J.: Prentice Hall Inc., 1998.

[43] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.

[44] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, July 1998.

[45] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. London, UK: Academic Press, 2006.

[46] N. Ayache, *Artificial Vision for Mobile Robots*. Cambridge, MA: MIT Press, 1991.

[47] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," in *Proceedings of the Forth International Conference on Spoken Language*, vol. 4, Philadelphia, PA, October 3–6, 1996, pp. 2005–2008.

[48] M. Yektaii, M. O. Ahmad, and P. Bhattacharya, "A method for preserving the classifiability of digital images after performing a wavelet-based compression," *Signal, Image and Video Processing*, DOI:10.1007/s11760-013-0509-3.

[49] G. Lafruit, F. V. M. Catthoor, J. P. H. Cornelis, and H. J. J. De Man, "An efficient VLSI architecture for 2-D wavelet image coding with novel image scan," *IEEE Transactions on Very Large Scale Integration*, vol. 7, no. 1, pp. 56–68, March 1999.

[50] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall Inc., 1989.

[51] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[52] F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, "Variable bit-rate coding of video signals for atm networks," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, pp. 801–806, June 1989.

[53] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, August 1989.

[54] AT&T Bell Laboratories, Cambridge, UK. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[55] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library," Columbia University. Dept. Comp. Sci., New York, NY, Tech. Rep. CUCS-005-96, 1996.

[56] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[57] F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, April 2006.

[58] M. Yektaii and P. Bhattacharya, "Cumulative global distance for dimension reduction in handwritten digits database," in *Proceedings of the 9th International Conference on Advances in Visual Information Systems*, ser. VISUAL'07. Berlin, Heidelberg: Springer-Verlag, June 28–29, 2007, pp. 216–222.

[59] ——, "A criterion for measuring the separability of clusters and its applications to principal component analysis," *Signal, Image and Video Processing*, vol. 5, no. 1, pp. 93–104, March 2011.

[60] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins University Press, 1996.

[61] M. Yektaii, M. O. Ahmad, and P. Bhattacharya, "A method for determining the number of features in the kernel space required for preserving classifiability," *Signal, Image and Video Processing*, under review.

[62] AT&T Bell Laboratories, Holmdel, NJ. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html.

[63] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versus Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.