FEATURE EXTRACTION AND FUSION TECHNIQUES FOR PATCH-BASED FACE RECOGNITION

by

BERKAY TOPÇU

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
August 2009

FEATURE EXTRACTION AND FUSION TECHNIQUES FOR PATCH-BASED FACE RECOGNITION

APPROVED BY

Assist. Prof. Dr. Hakan ERDOĞAN .....
(Thesis Supervisor)

Assoc. Prof. Dr. Özgür GÜRBÜZ .....

Assist. Prof. Dr. Gözde ÜNAL .....

Assist. Prof. Dr. İlker HAMZAOĞLU .....

Assist. Prof. Dr. Yücel SAYGIN .....

DATE OF APPROVAL: .....

to my family

Acknowledgements

I would like to express my gratitude to my thesis supervisor Hakan Erdoğan for his invaluable guidance, support and encouragement throughout my thesis. I appreciate his intimate attitude, patience and confidence in my ability to succeed.

I would like to thank TÜBİTAK-BİDEB for providing the necessary financial support for my graduate education.

Many thanks to my thesis jury members, Özgür Gürbüz, Gözde Ünal, İlker Hamzaoğlu and Yücel Saygın, for having kindly accepted to read and review this thesis.

My valuable thanks go to all my colleagues at Vision and Pattern Analysis Laboratory of Sabanci University, for sharing my troubles during the preparation of this thesis, and for turning the lab into a warm environment with their presence.

I owe special thanks to Serkan Belkaya, Berkay Kaya, my roommate Özer Koca and Muharrem Bayraktar for being there for me in both good and bad times. I would also like to thank all my friends, especially Ahmet Can Erdoğan, Ahmet Yasin Yazıcıoğlu, who joined us in our enjoyable football hours.

Finally, I would like thank my family for their endless love, patience and understanding throughout my life. Their limitless tolerance made everything about me possible.

FEATURE EXTRACTION AND FUSION TECHNIQUES FOR PATCH-BASED FACE RECOGNITION

BERKAY TOPÇU

EE, M.Sc. Thesis, 2009

Thesis Supervisor: Hakan Erdoğan

Keywords: face recognition, dimensionality reduction, decision fusion

Abstract

Face recognition is one of the most addressed pattern recognition problems in recent studies due to its importance in security applications and human computer interfaces. After decades of research in the face recognition problem, feasible technologies are becoming available. However, there is still room for improvement for challenging cases. As such, face recognition problem still attracts researchers from image processing, pattern recognition and computer vision disciplines. Although there exists other types of personal identification such as fingerprint recognition and retinal/iris scans, all these methods require the collaboration of the subject. However, face recognition differs from these systems as facial information can be acquired without collaboration or knowledge of the subject of interest.

Feature extraction is a crucial issue in face recognition problem and the performance of the face recognition systems depend on the reliability of the features extracted. Previously, several dimensionality reduction methods were proposed for feature extraction in the face recognition problem. In this thesis, in addition to dimensionality reduction methods used previously for face recognition problem, we have implemented recently proposed dimensionality reduction methods on a patch-based face recognition system. Patch-based face recognition is a recent method which uses the idea of analyzing face images locally instead of using global representation, in order to reduce the effects of illumination changes and partial occlusions.

Feature fusion and decision fusion are two distinct ways to make use of the extracted local features. Apart from the well-known decision fusion methods, a novel approach for calculating weights for the weighted sum rule is proposed in this thesis. On two separate databases, we have conducted both feature fusion and decision fusion experiments and presented recognition accuracies for different dimensionality reduction and normalization methods. Improvements in recognition accuracies are shown and superiority of decision fusion over feature fusion is advocated. Especially in the more challenging AR database, we obtain significantly better results using decision fusion as compared to conventional methods and feature fusion methods.

YAMA-TABANLI YÜZ TANIMA İÇİN ÖZNİTELİK ÇIKARIMI VE BİRLEŞTİRME TEKNİKLERİ

BERKAY TOPÇU

EE, Yüksek Lisans Tezi, 2009

Tez Danışmanı: Hakan Erdoğan

Anahtar Kelimeler: yüz tanıma, boyut düşürme, karar birleştirme

Özet

Yüz tanıma, güvenlik uygulamaları ve insan bilgisayar arayüzündeki öneminden dolayı, son dönemde en fazla incelenen örüntü tanıma problemlerinden biridir. Yüz tanıma problemi üzerinde on yıllardır süre gelen araştırmalar sonucu, uygulanabilir teknolojiler mevcut hale gelmiştir. Fakat, karşılaşılan zor durumlar için gelişime açık bir konudur. Öyle ki, yüz tanıma problemi halen imge işleme, örüntü tanıma ve bilgisayarla görü gibi farklı disiplinlerden araştırmacıların ilgisini çekmektedir. Yüz tanıma dışında, parmak izi tanıma ve retina/iris taraması gibi farklı kişisel kimlik tanıma sistemleri bulunsa da, tüm bu sistemlerde kişinin işbirliğine ihtiyaç duyulmaktadır. Yüz bilgisi ise kişinin işbirliği ya da bilgisi olmadan da elde edilebildiği için, yüz tanıma sistemleri diğer sistemlerden ayrılmaktadır.

Öznitelik çıkarımı, yüz tanıma probleminde önemli bir yer teşkil eder ve yüz tanıma sistemlerinin performansı, çıkarılan özniteliklerin güvenilirliğine dayanır. Daha önce, yüz tanıma problemi için çeşitli boyut düşürme yöntemleri sunulmuştur. Biz de bu tezde, daha önce yüz tanıma problemi için sunulmuş boyut düşürme yöntemlerine ek olarak, yakın zamanda sunulan boyut düşürme yöntemlerini bir yama-tabanlı yüz tanıma sistemi üzerinde uyguladık. Yama-tabanlı yüz tanıma, yakın zamanda sunulmuş bir yüz tanıma yöntemidir. Yüz imgelerinin bütünsel temsili yerine bölgesel analizi fikrine dayanarak, ışıklandırma değişimlerinin ve kısmi kapatmaların (oklüzyon) etkisini azaltmayı amaçlar.

Öznitelik birleştirme ve karar birleştirme, çıkarılan özniteliklerin değerlerdirilmesi için iki farklı yoldur. Bu tezde, herkesçe bilinen karar birleştirme yöntemlerinin dışında, ağırlıklı toplam kuralı için ağırlık hesaplanması için yeni bir yaklaşım sunmaktayız. İki farklı yüz veritabanı üzerinde, hem öznitelik birleştirme hem de karar birleştirme deneyleri gerçekleştirdik ve farklı boyut düşürme ve normalizasyon yöntemleri için tanıma oranlarını sunduk. Tanıma oranlarındaki artışları ve karar birleştirmenin öznitelik birleştirmeye göre üstünlüğünü gösterdik. Özellikle, daha zorlu AR veritabanı üzerinde, karar birletirme uygulayarak geleneksel yöntemlerden ve öznitelik birleştirmeden, önemli bir şekilde daha yüksek tanıma oranları elde ettik.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In today's high capability of data capturing and collection, researchers from various disciplines such as engineering, economics and biology, have to deal with large observations and simulations. These large observations are generally high dimensional data which depends on several numbers of features measured in each observation. As the number of features increase, it becomes harder to process this multi-dimensional data. Dimensionality reduction is the process of decreasing the number of features into a reasonable number so that the data can be analyzed much more easily. Also, not all the features are independent from each other and sometimes some features follow similar patterns. So, they bring computational complexity although they do not carry any additional information.

One of the application areas of dimensionality reduction is face recognition problem. In face recognition problem observations are usually 2-D face images in which features are equal to the number of pixels in the image. For a 64x64 image, 4096 pixels (features) make it hard for a recognition system to operate and as most of the pixels are correlated with each other, some of the features do not carry any additional information. Therefore, dimensionality reduction is essential for a face recognition system.

Decision fusion is a relatively new research area that has attracted interest in the last decade. It is a common method to increase reliability and accuracy of pattern recognition systems by combining outputs of several classifiers. Instead of relying on a single decision making scheme, multiple schemes can be combined using their individual decisions [3].

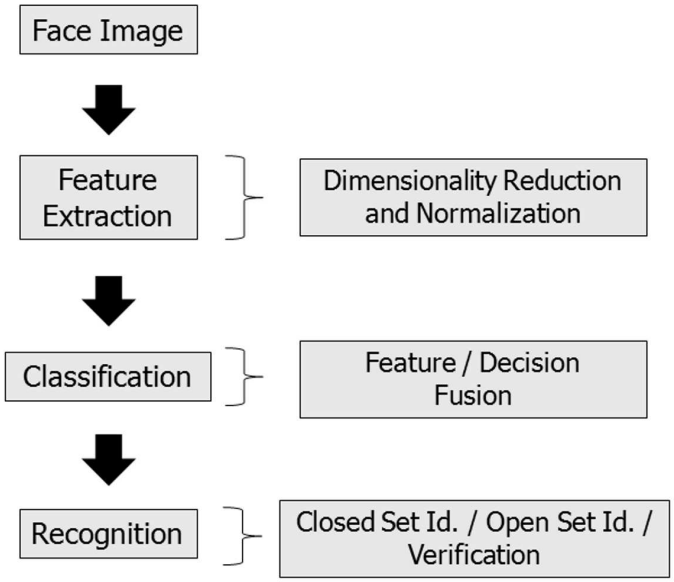

Figure 1.1: General face recognition scheme.

In this study, our main motivation is to overcome some of the difficulties that face recognition systems face, especially illumination differences and partial occlusion in face images, by applying different dimensionality reduction techniques that are enhanced by image and feature normalization methods and by applying decision fusion techniques. To tackle these problems, instead of using a face image as a whole, patch-based methods are proposed in [4]. In patch-based face recognition, face images are divided into overlapping or non-overlapping blocks and feature extraction and normalization methods are applied on these blocks. By dividing image into different regions and handling each region separately brings some advantages such as decreasing the effect of illumination changes and partial occlusions in face images. One way to approach face recognition problem is to extract features from separate blocks and then concatenate those features in order to use in the recognition system. In addition, features extracted from each block can be classified within the same blocks of different images and by decision fusion, recognition results of different blocks of a test sample can be combined in order to provide more accurate decision. In our study, we examine each approach, feature fusion and decision fusion, and present recognition rates for each dimensionality reduction technique and normalization method.

1.2 Literature Review

Face recognition is one of the most addressed pattern recognition problems in recent studies due to its importance in security applications and human computer interfaces. This is evidenced by the presence of face recognition conferences such as the International Conference on Automatic Face and Gesture Recognition (AFGR), evaluation standards of face recognition techniques such as FERET [5], FRVT [6] and databases such as XM2VTS [7] and several commercial systems. One of the main reason behind this attention on face recognition is the commercial and law enforcement applications. An example to security applications of face recognition is that the German Federal police use a face recognition system to allow voluntary subscribers to pass fully automated border controls at Frankfurt Rhein-Main international airport. Also in the 2000 president election, the Mexican government employed facial recognition software to prevent voter fraud [8]. After decades of research in face recognition problem, feasible technologies became available that attract researchers from image processing, pattern recognition and computer vision disciplines. The reasons behind this interest comes from the increase in commercial opportunities, availability of real-time harware and the increasing importance of surveillance-related applications [9].

A major application of a face recognition system is in the area of biometric personal identification which can replace any password needed to protect privacy such as ATM PIN, PC login and internet passwords. For instance, some laptop companies implement face verification systems in their products. Although there exists other type of personal identifications such as fingerprint recognition and retinal/iris scans, all these methods require collaboration of the subjects. However, face recognition differs from these systems as facial information can be acquired without collaboration or knowledge of the subject of interest [10].

Face recognition can be briefly described as identifying or verifying a person from an image or a sequence of images. Verification differs from identification in a way that, both an identity and an image of the assumed identity is provided to the system and a two class classification is done resulting in correct identity or incorrect identity. Inputs to a face recognition system can be either 2-D images and/or sequences of images (videos) or 3-D images. 2-D color or intensity images are

widely used in recent face recognition systems due to their availability. It is difficult to capture 3-D images in order to use in face recognition and it is possible only under controlled conditions. Although 2-D face images or video can be captured easily even without the collaboration of the subject, to create a 3-D image, stereo cameras are needed. When recorded from distance, stereo cameras are also incapable of capturing enough information to reconstruct a face in 3-D as the view angle is almost same for stereo cameras. So, most of the current face recognition systems operate on 2-D color or intensity images, although 3-D images are beneficial as they carry more information. Also, in 2-D images pose changes are problematic and affect recognition rates negatively, whereas 3-D facial recognition is affected less by the changes in pose and lighting. It can identify a face from a range of different viewing angles.

Despite the intense research efforts on face recognition, it is still a difficult problem in real-world applications. Recognition of face images acquired in an outdoor environment with changes in illumination and pose remains a largely unsolved problem [9]. These unavoidable problems occur when face images are acquired in an uncontrolled and uncooperative environment. In addition, face images may be partially occluded or taken some time ago which makes it difficult for the system to recognize successfully. Prior to face recognition, face localization which is crucial for face detection, is another challenge in outdoor images.

The history of face recognition dates back to 1950s in psychology [11]. This research concerned with whether face recognition is a dedicated process and whether it is done holistically or by local feature analysis. But research on automatic facial recognition systems started in the 1970s in the engineering literature. In the early studies, face recognition is treated as a typical pattern recognition problem in which measured attributes of features in face images are used [12]. Parallel to developments in other pattern recognition and image processing disciplines such as design of classifiers for accurate face recognition, i.e. neural networks, support vector machines, research has focused on making face recognition an automated process by localizing a face, eyes, eyebrows, nose and mouth in an image and extraction of meaningful features.

Feature extraction is an important issue in face classification problems. For

example, a 64x64 face image has 4096 pixels which is a huge number of features for the classifiers to operate. Apart from that, most of these pixels are highly correlated with each other so all of the 4096 features do not provide beneficial information for classification purposes. In order to reduce dimensionality of face images and obtain meaningful feature vectors, several feature extraction methods are applied up to now.

In 1990s, appearance-based holistic approaches are presented and their accuracies in large databases are shown. In 1990, Kirby and Sirovich [13] and in 1991, Turk and Pentland [14] introduced face recognition using eigenfaces. Following these studies in 1997, Belhumeur [15], Etemad and Chelappa [16] presented usage of fisherfaces in face recognition. Apart from holistic appearance-based methods, feature-based approaches are also presented in 1990s and proved to be also successful. Feature-based methods are advocated as being less sensitive to illumination changes and pose differeneces [9]. But also there exists problems with the feature-based methods. Feature extraction such as localization of eyes and mouth is problematic and does not work always accurately for example when eyes are closed or mouth is wide opened. Face recognition problem can be divided into three subtasks that need to be completed in this order: detection of faces, extraction and normalization of features and identification or verification. Earlier face detection techniques were able to detect single faces or a few numbers of well-seperated frontal face images with simple backgrounds [9]. Current face detectors can detect several faces and their poses in complex backgrounds [17]. By extensive training, detection of faces by computer has become very successful as the face images are very similar to each other and different from non-face objects. In the study of Viola and Jones [17], huge number of face and non-face objects are used to extract Haar-like features [18] and fed into cascaded classifiers that allow background regions of the image to be quickly discarded while spending more computation on promising face-like regions.

Next step following face detection is the feature extraction that is crucial for face recognition system in addition to the holistic face. One class of feature extraction is the accurate localization of eyes, nose and mouth. A statistical shape model, Active Shape Model(ASM) [19] is proposed that matches a predefined template to a face image. ASM is then expanded more robust and flexible statistical appearance

models such as Flexible Appearance Model(FAM) [20] and Active Appearance Model (AAM) [21]. Following the extraction of eyes, nose and mouth, a face recognition system is built by using location and local statistics of these local features.

Apart from local feature extraction, features extracted from holistic view of faces are also used as inputs to face recognition systems. PCA [13] and LDA [16] are two well-known feature extraction methods which are proven to be successful for face recognition problem. In a recent study, it is shown that use of eigenfaces and fisherfaces in deteriorated face images is also valid at some levels of different kind of noise, such as salt and pepper, gaussian noise and blurring [22]. In addition to PCA and LDA, Liu and Wechsler [23] presented use of Independent Component Analysis (ICA) [24] together with extracted Gabor features from a set of downsampled Gabor wavelet representations of face images. Integrating Gabor features and ICA brings strong characteristics of spatial locality, scale and orientation selectivity providing salient local features suitable for face recognition [23]. Another image based linear projection used in face recognition is laplacianfaces. In [25], laplacianfaces which has been shown to be successful despite the nonlinearity of image space for dimensionality reduction, is used as a dimensionality reduction method that can preserve the locality in face images. In contrast to eigenfaces and fisherfaces, which seek for optimal projection by analyzing global patterns of data intensity, the laplacianfaces find an optimal solution by examining the local geometry of the training data.

Effectiveness of support vector machines(SVM) for face recognition has been reported in recent studies. In study of Hotta [26], together with local features, use of SVM with local Gaussian summation kernel is shown to provide successful recognition performances under partial occlusion. A single kernel is applied to global features that are influenced easily by noise or occlusion but application of local kernels to local features provides robustness to partial occlusion as only some of the local features are affected by occlusion. However, the kernel based methods improve the linear separability of the data at the cost of increasing dimensions and therefore high computational cost. Furthermore, how to select different kernels and how to assign the optimal parameters remain unclear [27]. Unlike kernel-based methods, local linear embedding(LLE) is straightforward in finding the structure in the obervation space. Several improved versions of LLEs are also used in face recog-

nition problem. One modification on LLE, locally linear discriminant embedding (LLDE) is proposed in [27], which is shown to increase class seperability and data from the same class to be clustered closer. A marginal study on face recognition is modelling face recognition algorithms. In [28], a linear transformation is sought to model recognition algorithms based on match scores. In this study, transformation of face images by PCA is followd by a nonrigid transformation that aims to preserve pair-wise distances between face images.

Although there exists studies on variations in face images taken in uncontrolled conditions such as illumination, pose and expression variations, most of the work assume the existence of one challenge at a time. In study of Geng, Zhou and Miles [29], all difficult problems of uncontrolled conditions are tackled together by using individual stable space (ISS) and neural network. In addition, by synthesizing an illumination normalized image, an illumination invariant representation of a face is extracted from a raw facial image [30].

To tackle the general problems associated with holistic approaches, in 2003 modular PCA is presented [31]. In this method, face images are divided into subimages and PCA is applied on these smaller images. Extracted features from each block are then combined and fed into a classification system. As the face image is divided into sub-images, the variations in pose or illumination will affect only some of the sub-images and more accurate features will be extracted. In an earlier study of Pentland et al. [32], a similar method is used by performing PCA on eyes and nose of the face image. Following these studies, Ekenel and Stiefelhagen proposed using Discrete Cosine Transform(DCT) on blocks of face images instead of PCA [33]. In this study, feature fusion and block selection were proposed in addition to two feature normalization methods. Use of DCT on holistic approach was presented by Hafed and Levine in 2001 [34]. In our study, we have studied on the block-based approach and improved this approach with the contributions listed as follows.

1.3 Contributions

In this thesis, we have developed a patch-based face recognition system and contributions of this thesis can be listed as follows:

- We have applied recently proposed dimensionality reduction methods to patch-

based face recognition.

- New image level and feature level normalization methods to be applied in patch-based face recognition are introduced.
- We introduced the use of decision fusion techniques for patch-based face recognition.
- We have estimated weights in "weighted sum rule" decision fusion using a novel method.

1.4 Outline

This thesis is organized in five chapters including the Introduction chapter. In Chapter 2, feature extraction methods, dimensionality reduction and normalization techniques for face recognition are given. Proposed feature fusion and decision fusion types for patch-based face recognition are presented in Chapter 3. The experimental results are provided and discussed in Chapter 4. In the last chapter, the conclusions and future work are expressed.

Chapter 2

Feature Extraction

In this chapter, feature extraction methods for face recognition are described. In the first section, different dimensionality reduction methods are presented. In the remaining sections of this chapter, image and feature normalization techniques are introduced.

2.1 Dimensionality Reduction

Decreasing the number of features of a multidimensional data under some constraints is desired in many applications. One way of decreasing feature number is to select some of the features and discard remaining features which are less relevant or carry less information. This is called feature selection. Another way is linear or nonlinear transform of the whole data into another feature set. This process is called dimension reduction. For dimension reduction, multidimensional data is projected or mapped into a space with less number of dimensions. Therefore, by applying a dimension reduction method, a d-dimensional data is mapped or transformed into a p-dimensional data, where $p < d$.

Parallel to improvements in data collection and storage capabilities, researchers from various disciplines have to deal with large observations. By large observations, we mean multidimensional data with high number of samples. As both dimension and quantity of the data increase, it becomes harder for systems to analyze and process these data. Dimensionality reduction is one of the essential methods which aims to extract relevant structures and relationships from multidimensional data.

An important problem with the high dimensional data is that, some features may be unimportant at describing the structure of data. Also, in some cases, features

are highly correlated with each other and some of them do not carry additional information. All dimension reduction methods aim to present high dimensional data in a lower dimensional space, in a way that captures the desired structure of the data [35]. Dimensionality reduction is a helpful tool for multidimensional observations, that is applied prior to any analysis or processing application such as clustering and classification.

In mathematical terms, the problem we investigate is: given the d-dimensional sample $\boldsymbol{x} = [x_1, x_2, ..., x_d]^T$, we want to find a lower dimensional (p-dimensional) representation of $\boldsymbol{x}$, $\boldsymbol{f} = [f_1, ..., f_p]^T$ where $p < d$, that captures the content in the original data, according to some criterion. This criterion can be lower dimensional representation of a single class data, or separability of multi-class data in the reduced dimensional space. For linear dimensionality reduction, we need to create a pxd transformation matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_p]^T$, such that $\boldsymbol{f} = \boldsymbol{W}^T\boldsymbol{x}$. We need to find d-dimensional column vectors $\boldsymbol{w}_i$'s (or so called basis) that will constitute the rows of the transformation matrix $\boldsymbol{W}$. Then we project our data $\boldsymbol{x}$ onto these basis by multiplying with $\boldsymbol{W}$.

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^T \\ \boldsymbol{w}_2^T \\ . \\ . \\ . \\ \boldsymbol{w}_p^T \end{bmatrix}.$$

Assuming orthonormality of the rows of $\boldsymbol{W}$, we find the coefficients f_i's that represent $\boldsymbol{x}$ as a linear combination of basis elements $\boldsymbol{w}_i$'s. We can calculate the approximation of $\boldsymbol{x}$, which is represented with $\hat{\boldsymbol{x}}$, by using basis coefficients, as following:

$$\hat{\boldsymbol{x}} \cong \sum_{i=1}^{p} f_i \boldsymbol{w}_i. \tag{2.1}$$

2.1.1 Discrete Cosine Transform (DCT)

Discrete Cosine Transform expresses a sequence of data points in terms of sum of cosine functions oscillating at different frequencies and amplitudes. The 2D DCT transform equation of an NxM image is given in Equation 2.2 where $\Omega(u) = 1$ for $u \neq 0$ and $\Omega(u) = \frac{1}{\sqrt{2}}$ for $u = 0$.

$$\boldsymbol{f}(u,v) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \boldsymbol{x}(i,j)\Omega(u)\Omega(v) \cos\left[\frac{\pi}{N}(i+\frac{1}{2})u\right] \cos\left[\frac{\pi}{M}(j+\frac{1}{2})v\right]. \tag{2.2}$$

Discrete Cosine Transform (DCT) uses an orthonormal basis and is widely used in visual feature extraction as well as image compression. One of its advantages is that, DCT has a strong energy compaction property so that most of the signal information is concentrated in a few low frequency components. So, by using the first low frequency components, most of the information in the data is captured. In Figure 2.1, 8x8 DCT basis is illustrated. The first three DCT basis elements contain general information about the global statistics of an image. The first basis element represents the average intensity of the image and the second and third basis elements represent the average horizontal and vertical intensity change in the image, respectively. In addition, DCT has a fast implementation which is an advantage in real time processing. Also, it requires no training data. In this study, we perform two dimensional DCT on face images, remove the first three coefficients that correspond to the first three basis elements and pick p low frequency components (coefficients of p number of basis following the first three basis) to use them as visual features. Note that, we order the 2D DCT basis vectors in zig-zag scan order starting from top-left.

2.1.2 Principal Component Analysis (PCA)

DCT is preferred in image processing due to its approximation of the Karhunen-Loeve Transform (KLT) for natural images. However, if there is enough training data, one can obtain the data-dependent version of KLT, which is the principal component analysis (PCA) transform. Principal component analysis (PCA) is an orthogonal linear transformation that maps the data into a lower dimension by preserving most of the variance in the data. PCA provides an orthonormal basis

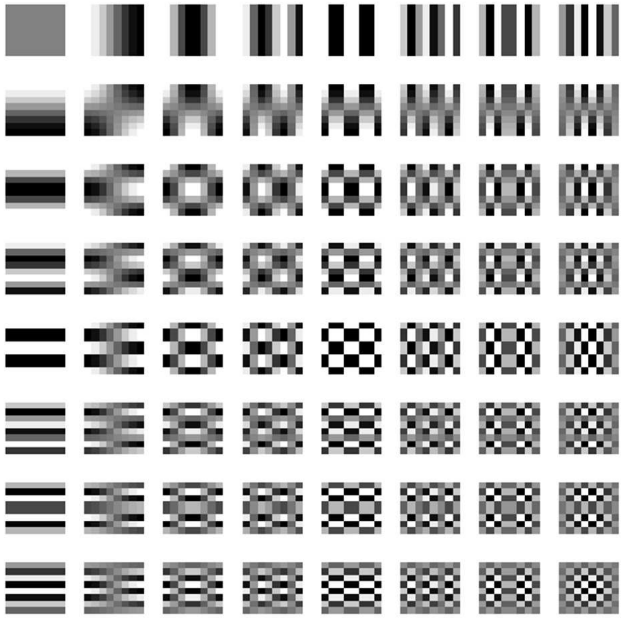

Figure 2.1: 8x8 DCT basis.

for the best subspace that gives minimum least squared error on training samples. First principal component is in the direction of the maximum variance in the data and the second component is in the direction of the second maximum variance in the data and so on. In dimension reduction by using PCA, characteristics of the data that contribute most to its variance are kept by keeping lower-order principal components. So, by using less amount of information, most of the variance of the data is captured. We select the rows of the transformation matrix, $\boldsymbol{W}$, as the eigenvectors that corresponds to the p highest eigenvalues of the scatter matrix $\boldsymbol{S}$,

$$\boldsymbol{S} = \sum_{k=1}^{n} (\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^T, \tag{2.3}$$

where $\boldsymbol{x}_k$ represents the k^{th} sample and $\boldsymbol{m}$ is the sample mean.

The main weakness of PCA is that, it is lighting and background variant so that changes in lighting conditions and background decreases the success of reliable mapping and classification performance. However, advantages it brings are that it is fast, computationally easy and needs less amount of memory. On the other hand, PCA does not take class information into account, so there is no guarantee that the direction of the maximum variance will contain good features for discrimination.

Figure 2.2: First 16 principal components.

Figure 2.3: First 12 principal components for block corresponding to eye region.

2.1.3 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a method used to find the linear combination of features which best separate two or more classes of objects. LDA finds the vectors in the lower dimensional space that best discriminate among classes. In Figure 2.4, a transformation from 3-dimensions to 2-dimensions is illustrated [1]. The goal is to maximize between-class scatter while minimizing within-class scatter. Between-class scatter and within-class scatter matrices are defined as follows:

$$\boldsymbol{S}_B = \sum_{i=1}^{N} p_i(\boldsymbol{m}_i - \hat{\boldsymbol{m}})(\boldsymbol{m}_i - \hat{\boldsymbol{m}})^T, \tag{2.4}$$

$$\boldsymbol{S}_W = \sum_{i=1}^{N} p_i \boldsymbol{S}_i, \tag{2.5}$$

where $\hat{\boldsymbol{m}}$ equals $\sum_{i=1}^{N} \boldsymbol{m}_i$ and $\boldsymbol{S}_i$ is the within-class covariance matrix of class i and p_i is the prior probability for the i^{th} class. This goal can be achieved by maximizing the ratio of the determinant of the between-class scatter $\boldsymbol{S}_B$ and the determinant of the within-class scatter $\boldsymbol{S}_W$ in the projected space.

$$J(\boldsymbol{W}) = \frac{|\boldsymbol{W}\boldsymbol{S}_B\boldsymbol{W}^T|}{|\boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^T|}. \tag{2.6}$$

We want to find the transformation $\boldsymbol{W}$ that maximizes the ratio of the between-class scatter to the within-class scatter and rows of the transformation matrix, $\boldsymbol{W}$, are eigenvectors that corresponds to the p highest eigenvalues of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ [1].

One of the possible deficiencies of LDA is that there are computational difficulties in a situation with large numbers of highly correlated feature values. In face recognition case, as pixel values are highly related with the neighbor pixels, correlation is high and scatter matrices might become singular. When there is little data for each class, scatter matrices are not reliably estimated and there are also numerical problems related to the singularity of scatter matrices.

2.1.4 Approximate Pairwise Accuracy Criterion (APAC)

One of the main drawbacks of LDA is that as it tries to maximize the squared distances between pairs of classes, outliers dominate the eigenvalue decomposition. So, LDA tends to overweight the influence of classes that are already well separated. The

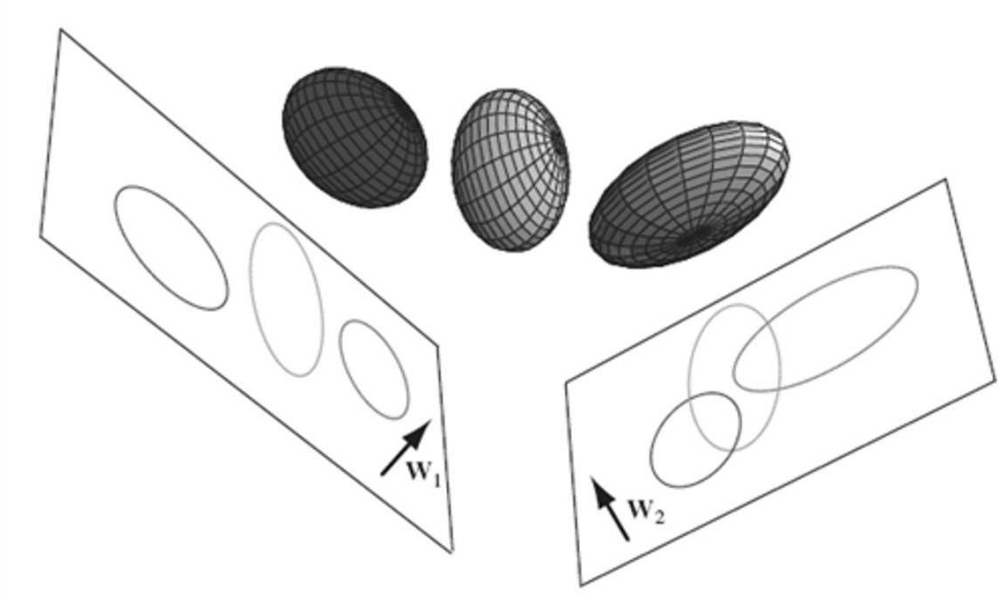

Figure 2.4: LDA projection vectors (taken from [1]).

resulting transformation preserves the distances of already well-seperated classes, causing a large overlap of neighboring classes, which decreases the classification performance. Approximate pairwise accuracy criterion (APAC) method has been proposed in order to prevent the domination of outliers [36]. Using the transformation matrix as $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_p]^T$ and p_i and p_j as prior probabilities of class i and j respectively, overall criterion, J_w, to be maximized can be expressed as the following:

$$J_w(\boldsymbol{W}) = \sum_{m=1}^{p} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} p_i p_j w(\Delta_{ij}) tr(\boldsymbol{w}_m \boldsymbol{S}_{ij} \boldsymbol{w}_m^T). \tag{2.7}$$

N-class LDA can be decomposed into a sum of $\frac{1}{2}N(N-1)$ two-class LDA problems and contribution of each two-class LDA to the overall criterion is weighted by w depending on the Mahanalobis distance ($\Delta_{ij} = \sqrt{(\boldsymbol{m}_i - \boldsymbol{m}_j)^T \boldsymbol{S}_w^{-1} (\boldsymbol{m}_i - \boldsymbol{m}_j)}$) between the classes i and j in the original space. $\boldsymbol{S}_{ij}$ is the pairwise between-class scatter matrix calculated as $\boldsymbol{S}_{ij} = (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^T$. Regular LDA is equivalent to using $\boldsymbol{S}_B = \sum_i \sum_{j \geq i} p_i p_j \boldsymbol{S}_{ij}$ and the idea of APAC is to weight each pairwise between-class scatters. In the study of Loog and Duin [36], weighting function is expressed as $w(\Delta_{ij}) = \frac{1}{2\Delta_{ij}^2} \text{erf}(\frac{\Delta_{ij}}{2\sqrt{2}})$. The solution that maximizes the above criterion is the eigenvectors of $\sum \sum p_i p_j w(\Delta_{ij}) \boldsymbol{S}_w^{-\frac{1}{2}} \boldsymbol{S}_{ij} \boldsymbol{S}_w^{-\frac{1}{2}}$ where $\boldsymbol{S}_w = \sum p_i \boldsymbol{S}_i$ is the pooled within-class scatter given that $\boldsymbol{S}_i$ is the within-class covariance matrix for class i. Although this approach can be viewed as a generalization of LDA, it does

not bring any additional computational complexity cost and it is designed to confine the influence of outlier classes which makes it more robust than LDA.

2.1.5 Normalized PCA (NPCA)

Normalized PCA is a generalization of regular PCA. In [2], it is shown that PCA maximizes the sum of all squared pairwise distances between the projected vectors. So solving the maximization of this sum in the projected space yields the same result with regular PCA. In regular PCA, an unweighted sum of the squared distances is maximized and by introducing a weighting scheme, elements from different classes can be placed further from each other in the projected space.

If we show the sum of squared distances in the projected space as $\sum_{i<j}(\text{dist}_{ij}^p)^2$ where dist_{ij}^p is the distance between elements i and j in the projected space, we seek the projection that maximizes the weighted sum:

$$\sum_{i<j} d_{ij}(\text{dist}_{ij}^p)^2. \tag{2.8}$$

d_{ij}'s are called pairwise dissimilarities, so by defining these pairwise dissimilarities, we can place elements from different classes further from each other. If we set $d_{ij} = 1$, we get the same result with regular PCA. In [2], pairwise dissimilarities are introduced as $d_{ij} = \frac{1}{\text{dist}_{ij}}$ where dist_{ij} is the distance between elements i and j from different classes, in the original space. The rows of the transformation matrix, W, are selected as the generalized eigenvectors that corresponds to the p highest eigenvalues of $(\boldsymbol{X}^T\boldsymbol{L}^d\boldsymbol{X}, \boldsymbol{X}^T\boldsymbol{X})$, where $\boldsymbol{L}^d$ is a Laplacian matrix derived by pairwise dissimilarities and $\boldsymbol{X}$ is data matrix (one sample in each row). What we are trying to accomplish here is to place elements of different classes apart from each other. By selecting pairwise dissimilarities as inversely proportional to their distances in the original space, on the overall criterion we emphasize the elements that are close to each other and give less importance to the elements that are already apart. If elements i and j belong to same class, d_{ij} can be set to 0, which means we are not interested in separating elements within the same class. So, normalized PCA becomes able to discriminate classes in the projected space where PCA may fail as it does not take class information into account.

In the Figure 2.5, a 2-D dataset is projected to 1-D by using both PCA and Nor-

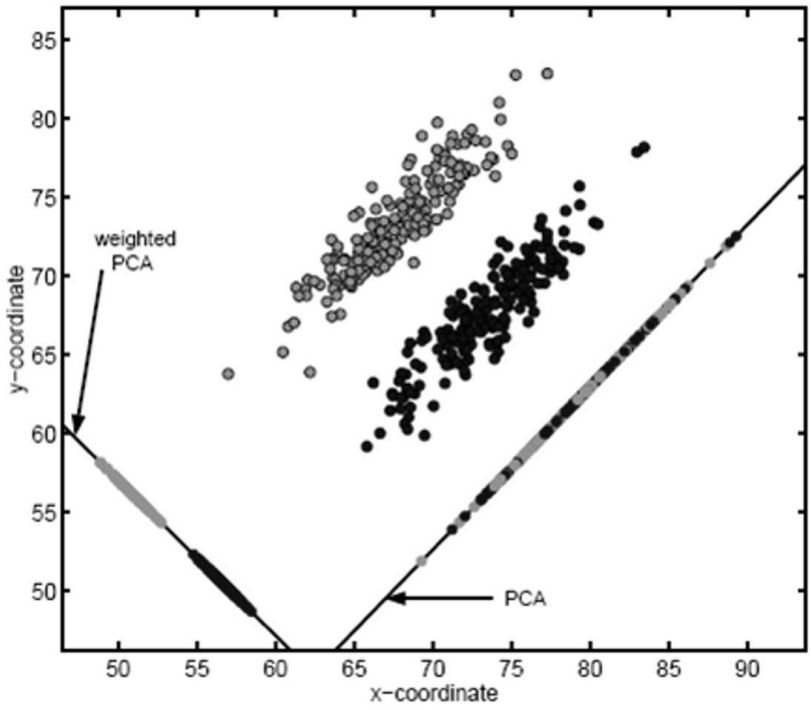

Figure 2.5: PCA vs Normalized PCA (taken from [2]).

malized PCA into two different directions. In PCA case, PCA fails to discriminate classes in the projected space. However, by the introduction of pairwise dissimilarities Normalized PCA is able to capture the class decomposition.

2.1.6 Normalized LDA (NLDA)

An improved version of Normalized PCA is Normalized LDA (NLDA), in which pairwise similarities (s_{ij}) are introduced in addition to pairwise dissimilarities (d_{ij}). The maximization criterion of Normalized PCA which depends on the sum of pairwise distances can also be written in a different way as $\sum_{i<j} s_{ij}(\text{dist}_{ij}^{p})^2$ to be minimized. In [2], pairwise similarities are introduced as $s_{ij} = \frac{1}{\text{dist}_{ij}}$, inversely proportional with the distance between elements i and j in the original space, for the elements of the same class and 0 for the elements belonging to different classes. On the overall criterion of the Normalized LDA, unlike the criterion of the Normalized PCA, we emphasize the distance between elements of the same class that are apart from each other and attach less importance to the elements of the same class that are already close. When we combine the second criteria to be minimized with the first one to be

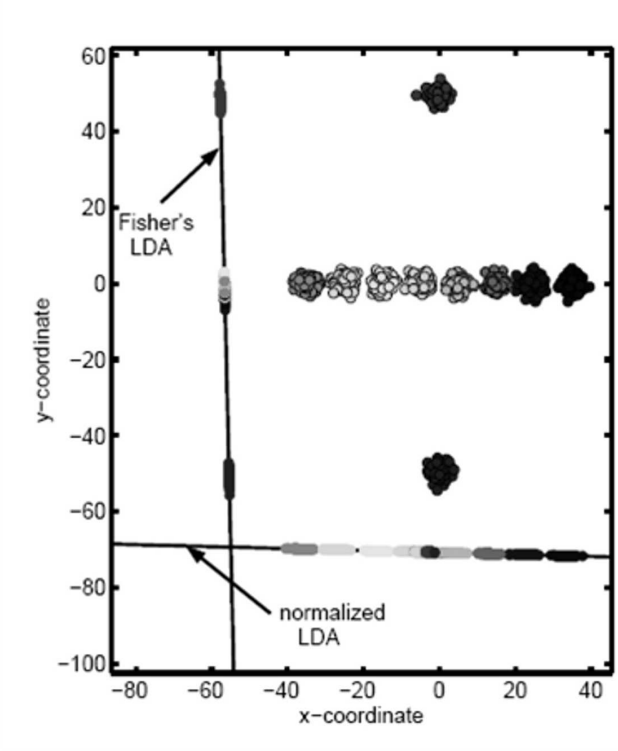

Figure 2.6: LDA vs Normalized LDA (taken from [2]).

maximized (criteria of NPCA), we obtain the following problem to be maximized:

$$\frac{\sum_{i<j} d_{ij}(\text{dist}_{ij}^{p})^2}{\sum_{i<j} s_{ij}(\text{dist}_{ij}^{p})^2}. \tag{2.9}$$

The rows of the transformation matrix, W, are selected as the generalized eigenvectors that corresponds to the p highest eigenvalues of $(\boldsymbol{X}^T\boldsymbol{L}^d\boldsymbol{X}, \boldsymbol{X}^T\boldsymbol{L}^s\boldsymbol{X})$, where $\boldsymbol{L}^d$ is a Laplacian matrix derived by pairwise dissimilarities, $\boldsymbol{L}^s$ is a Laplacian matrix derived by pairwise similarities and $\boldsymbol{X}$ is data matrix (one sample in each row).

Therefore, the labeled data can be discriminated in the projected space, as Normalized LDA can induce "attraction" between elements of the same cluster, and "repulsion" between elements of different clusters [2]. Figure 2.6 illustrates an example of a data with 10 different classes. As two classes are outlier classes with respect to the remaining data, LDA fails to discriminate classes that are placed close to each other in the original space. When Normalized LDA is applied on the data, the effect of the outlier classes are normalized and the classes are well-separated.

2.1.7 Nearest Neighbor Discriminant Analysis (NNDA)

Nearest neighbor discriminant analysis (NNDA) is a linear mapping that aims to optimize nearest neighbor classification performance in the projected space [37]. We seek to find the transformation $\boldsymbol{W}$ that maximizes the criterion below.

$$J(\boldsymbol{W}) = \boldsymbol{W}(\boldsymbol{S}_B^{'} - \boldsymbol{S}_W^{'})\boldsymbol{W}^T. \tag{2.10}$$

$\boldsymbol{S}_B^{'}$ and $\boldsymbol{S}_W^{'}$ are nonparametric between-class and within-class scatter matrices, defined as:

$$\boldsymbol{S}_B^{'} = \sum_{n=1}^{N} w_n(\Delta_n^E)(\Delta_n^E)^T \quad \boldsymbol{S}_W^{'} = \sum_{n=1}^{N} w_n(\Delta_n^I)(\Delta_n^I)^T, \tag{2.11}$$

where N is the number of samples and the other variables are described in the following. Let $\boldsymbol{x}^E$ and $\boldsymbol{x}^I$ be extra-class nearest neighbor and intra-class nearest neighbor for a sample $\boldsymbol{x}$. The nonparametric extra-class differences Δ^E, intra-class differences Δ^I and sample weight w_n are defined as

$$\Delta^E = \boldsymbol{x} - \boldsymbol{x}^E, \quad \Delta^I = \boldsymbol{x} - \boldsymbol{x}^I \text{ and} \tag{2.12}$$

$$w_n = \frac{||\Delta_n^I||^\alpha}{||\Delta_n^I||^\alpha + ||\Delta_n^E||^\alpha}, \tag{2.13}$$

where α is a control parameter to deemphasize the samples in the class center and give emphasis to the samples closer to the other classes. Notice that, the nonparametric extra-class and intra-class differences are calculated in the original high dimensional space, then projected to the low dimensional space, so that we have no guarantee that these distances are preserved in the low dimensional space. To solve this problem, the projection matrix $\boldsymbol{W}$ is calculated in a stepwise manner such that, at each step dimensionality is reduced to a higher dimension than the desired low dimension (at each step we decreased the dimensionality to half) and we calculate the nonparametric extra-class and intra-class differences in its current dimensionality at each step. The final projection matrix is the multiplication of projection matrices calculated at each step.

NNDA is an extension of nonparametric discriminant analysis, but it does not depend on the nonsingularity of the within-class scatter. Also unlike LDA, NNDA does not assume normal class densities.

2.2 Normalization Methods

In patch-based face recognition, every image is processed over non-overlapping square blocks. We define an image in a vector form as $\boldsymbol{x}^T = [\boldsymbol{x}_1^T ... \boldsymbol{x}_B^T]$ where B is the number of blocks and $\boldsymbol{x}_b$ denotes the vectorized b^{th} block of the image. For dimension reduction, we try to find a linear transform matrix for each block, $\boldsymbol{W}_b$, such that $\boldsymbol{f}_b = \boldsymbol{W}_b \boldsymbol{x}_b$. Then for each image, the feature vector is formed as $\boldsymbol{f}^T = [\boldsymbol{f}_1^T ... \boldsymbol{f}_B^T]$. On features extracted from separate blocks, we have applied some normalization methods that are described below.

2.2.1 Image Domain Mean and Variance Normalization

Image domain mean and variance normalization is a preprocessing step that is applied on the images before any dimension reduction method is used. So, it is a normalization of intensity values of pixels. In each block, mean intensity value of the current block μ_b is subtracted and the result is divided by the standard deviation σ_b in the block.

$$\tilde{\boldsymbol{x}}_b = \frac{1}{\sigma_b}(\boldsymbol{x}_b - \mu_b). \tag{2.14}$$

By image domain normalization, we aim to be able to extract similar visual feature vectors from each block across sessions of the same subject. Figure 2.7 shows the resulting image before and after this normalization as well as the effects of the normalization on one row of the image.

2.2.2 Feature Normalizations

As image domain normalization, feature normalizations may also be important in a patch-based face recognition scheme to reduce inter-session variability and intra-class variance. We have worked on different kinds of feature normalization methods as detailed below.

Norm Division (ND):

$\tilde{\boldsymbol{f}} = \boldsymbol{f}/||\boldsymbol{f}||$. In this method, we divide each feature vector to its Euclidean norm, which makes the norm of the normalized vector one. Blocks with different brightness levels lead to visual feature vectors with different value levels. To balance the effect

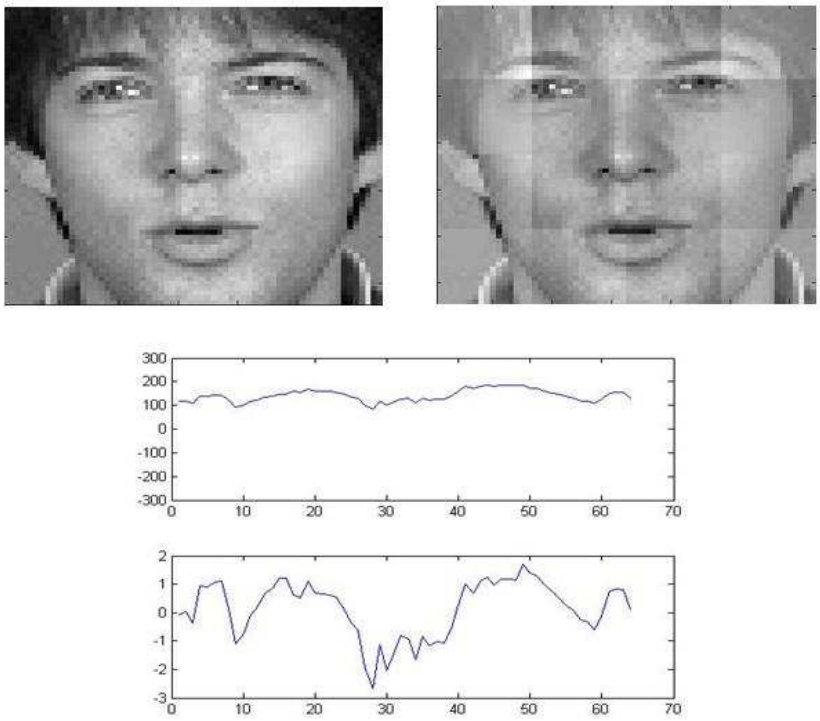

Figure 2.7: Effect of image domain normalization on a face image (above) and on a single row of the same image (below) using 16x16 blocks.

of features that come from blocks with higher or lower brightness levels, we divide each feature vector to its norm.

Sample variance normalization (SVN):

$\tilde{f}_i = f_i/\sigma(f_i)$. Here, each feature vector component is divided by its sample standard deviation computed over a training set. Due to the value range of visual feature vectors, higher numbers in each feature vector dominates the classification results. To balance the contribution of each value in a feature vector, each vector is divided by its standard deviation.

Block mean and variance normalization (BMVN):

$\tilde{\boldsymbol{f}}^b = \frac{1}{\sigma_f^b}(\boldsymbol{f}^b - \mu_f^b)$. The mean and variance normalization is done over the smaller feature vectors corresponding to each block separately as in the image domain normalization case. As each block corresponds to different parts in human face, brightness levels of each block differs even for the same subject. Also due to lighting

conditions, pixel values for each block differ greatly from pixel values of another block. Therefore, resulting visual feature vectors of different samples from same objects differ from each other which makes it impossible to classify correctly. To overcome these effects, one way is to normalize each block in itself. This is a new feature normalization technique proposed by us.

Feature vector mean and variance normalization (FMVN):

$\tilde{\boldsymbol{f}} = \frac{1}{\sigma_f}(\boldsymbol{f} - \mu_f)$. With the similar motivation as variance normalization, we introduced another normalization method on feature vectors which we call feature vector normalization. Here, the mean and standard deviation are computed over the components of the overall feature vector. This is also a new feature normalization method introduced by us.

Chapter 3

Patch-Based Face Recognition

In this chapter, patch-based face recognition is introduced and its advantages are discussed. Following the description of patch-based methods, feature fusion and decision fusion methods for extracted local features are presented.

3.1 Patch-Based Methods

Variation on the facial appearance caused by illumination changes, occlusion and expression changes, affect global transformation coefficients that represent the whole face information. Instead of describing a face image as a whole, analyzing faces locally might be beneficial and improve recognition accuracies. As the local changes will affect only the features extracted from the corresponding region of the face, overall representation coefficients will not be changed completely. The main motivation behind local appearance-based (or so called patch-based) face recognition is to eliminate or lower the effects of illumination changes, occlusion and expression changes by analyzing face images locally. The resulting outputs of this analysis is then combined at the feature level or decision level [33].

As in [38], modular and component based approaches require detection of local regions such as eyes and nose. However, patch-based face recognition is a generic local approach. Patch-based face recognition can be briefly explained as follows: A detected and normalized face image is divided into blocks of 16x16 or 8x8 pixels size and dimensionality reduction techniques are applied on each block separately. Selection of block size is important because blocks should be big enough to provide sufficient information about the region it represents and should be small enough to provide stationarity and to prevent complexity in dimensionality reduction. Two

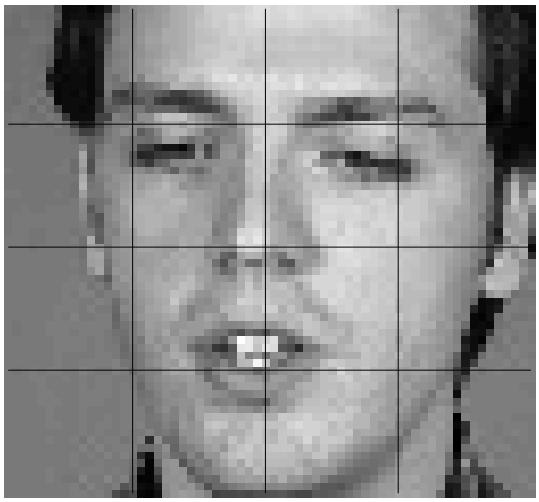

Figure 3.1: 16x16 blocks on a detected face.

examples of blocks with different block sizes (8 and 16) are illustrated in Figures 3.1 and 3.2.

Following the feature extraction process from blocks, one approach is to concatenate features from each block in order to create visual feature vector of an image which is called as feature fusion. Another approach is to classify each block separately and then combine individual recognition results of each block. This approach is named as decision fusion.

The originating point of our study is that by using patch-based methods, we can get recognition rates higher than the global eigenface approach. In global eigenface approach, PCA is applied on the whole image and each image is reduced to 192 dimensions from 4096 dimensions. Eigenfaces are found from the training set. We have chosen to reduce 4096 dimensions to 192 dimensions, so that we can capture 85% of the variance of the data. To preserve 85% of the variance, we have chosen the lowest number, 192, that can be divided by 16 and 64 (number of blocks for block sizes 16 and 8, respectively).

Comparison of global eigenface (PCA) and patch-based face recognition results are given in Table 3.1. It can be seen that, patch-based methods have close recognition rates to global PCA but do not perform significantly better when no normalization is applied to local visual feature vectors. However, experimental results show

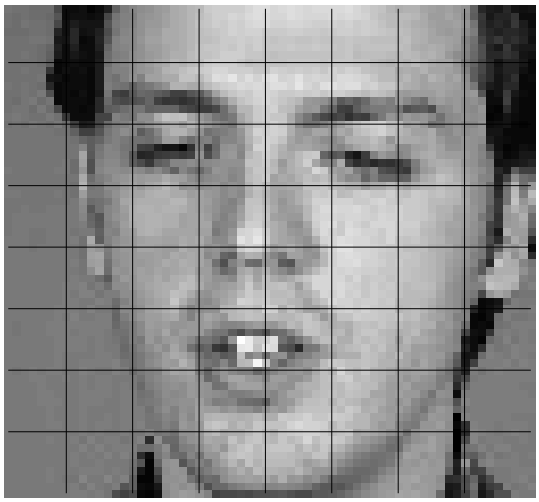

Figure 3.2: 8x8 blocks on a detected face.

Table 3.1: Global and block PCA results

Global PCA	83.45%
Block PCA (8x8)	83.78%
Block PCA (16x16)	83.78%

that, other dimensionality reduction and normalization methods improve recognition performance and patch-based methods have higher correct classification rates than global eigenface method after normalization.

In addition to improvements from block based feature extraction and fusion methods, decision fusion also provides higher correct classification rates. Experimental results show that decision fusion outperforms feature fusion and global approaches.

3.2 Classification Method: Nearest Neighbor Classifier

In our face recognition experiments, we use nearest neighbor classification with one nearest neighbor. The choice of nearest neighbor classifier instead of other type of classifiers is due to the nature of the face recognition problem. Data obtained from

face images are sparse therefore for other type of classifiers, extracting a statistical pattern that represents the nature of training data, is a difficult task.

For nearest neighbor classification, distances between samples are to be calculated and there exists several distance metrics. One of the most commonly used metrics is the L_p-norm between d-dimensional training sample, $\boldsymbol{f}_{\text{train}}$, and test sample, $\boldsymbol{f}_{\text{test}}$, which is defined as:

$$L_p = (\sum_{n=1}^{d} (\boldsymbol{f}_{\text{train},n} - \boldsymbol{f}_{\text{test},n})^p)^{\frac{1}{p}}. \tag{3.1}$$

In our experiments we have used nearest neighbor classifier with L_2-norm as the distance metric. Apart from that, for some of the successful methods, we have evaluated also effects of different distance metrics: L_1-norm and cosine angle, which is defined as:

$$\text{COS} = \frac{\boldsymbol{f}_{\text{train}}^T \boldsymbol{f}_{\text{test}}}{||\boldsymbol{f}_{\text{train}}||.||\boldsymbol{f}_{\text{test}}||}. \tag{3.2}$$

Decision fusion requires extraction of class posterior probabilities $p(C_i|\boldsymbol{x})$ for the classifiers used. For nearest neighbor classifier, it is not immediately clear how to assign posterior probabilities. Following [39], we calculated the class posterior probabilities depending on the distance of $\boldsymbol{x}$ to the nearest training sample from each class. If we denote this distance vector as $\boldsymbol{D} = [\boldsymbol{D}(1), \boldsymbol{D}(2), ..., \boldsymbol{D}(N)]$, posterior probabilities associated with class i is calculated as:

$$p(C_i|\boldsymbol{x}) = \text{norm}(\text{sigm}(\log(\frac{\sum_{j \neq i} \boldsymbol{D}(j)}{\boldsymbol{D}(i)}))), \text{ where} \tag{3.3}$$

$$\text{sigm}(x) = \frac{1}{1 + \text{e}^{-x}}. \tag{3.4}$$

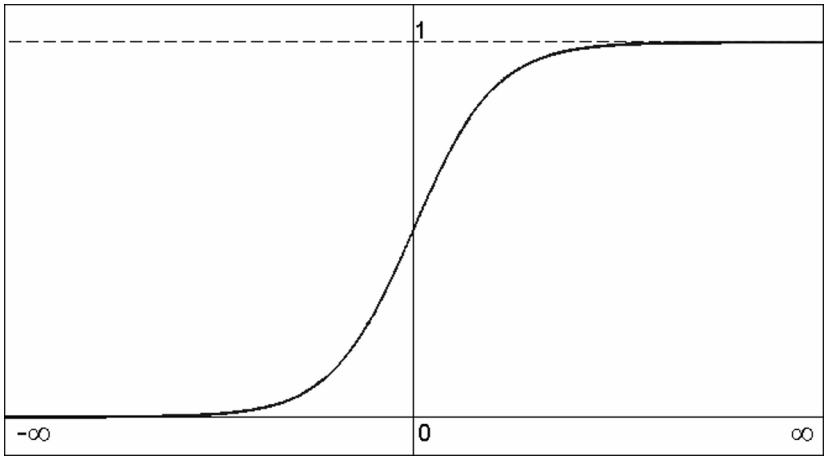

Figure 3.3: Sigmoid function.

In this calculation, sigmoid function which nonlinearly maps $-\infty$ to 0 and $+\infty$ to 1, is used. After calculating posterior probability for each class, they are normalized to sum up to 1.

3.3 Feature Fusion

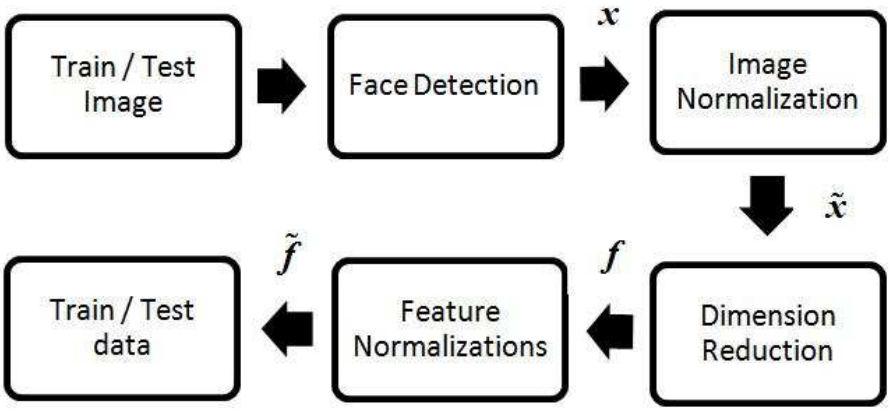

Figure 3.4: General schema for the proposed patch-based face recognition feature fusion system.

Following the feature extraction (dimension reduction and normalization), local visual feature vectors are obtained. In holistic approaches such as global PCA, a single feature vector is obtained for each sample, and this feature vector is used either in training or testing stage. Unlike holistic approaches, in patch-based face

recognition, several (equal to the number of blocks) local features are extracted and we need to combine these feature vectors.

One way of combining these feature vectors is to concatenate vectors extracted from an image. In patch-based face recognition, every image is processed over non-overlapping square blocks. We define an image in a vector form as $\boldsymbol{x}^T = [\boldsymbol{x}_1^T ... \boldsymbol{x}_B^T]$ where B is the number of blocks and $\boldsymbol{x}_b$ denotes the vectorized b^{th} block of the image. For dimension reduction, we try to find a linear transform matrix for each block, $\boldsymbol{W}_b$, such that $\boldsymbol{f}_b = \boldsymbol{W}_b \boldsymbol{x}_b$. Then for each image, the feature vector is formed as $\boldsymbol{f}^T = [\boldsymbol{f}_1^T ... \boldsymbol{f}_B^T]$.

In our system, we use images of size 64x64 which corresponds to 4096 pixels. Dividing an image into blocks of 16x16 provides 16 blocks each having 256 pixels. By applying dimensionality reduction methods, we decrease these number, 256 pixels/dimensions, into 12 dimensions for each block. Similarly, by using 8x8 blocks we obtain 64 blocks with 64 pixels and reduce 64 pixels/dimensions to 3 for each block. Therefore, by concatenating these local feature vectors, for each block size either 16 or 8, we end up with a 192 dimensional (number of blocks x feature vector dimension) feature vector for each sample in the database (16x12=192 or 64x3=192). So, by applying dimension reduction on blocks separately, we reduce the dimension of each image from 4096 to 192.

Once the feature vectors for train and test images are created, we perform classification using nearest neighbor classifier using Euclidean distance.

3.4 Decision Fusion

Decision fusion or classifier combination can be interpreted as making a decision by combining the outputs of different classifiers for a test image. One of the methods to combine outputs of multiple classifiers is by majority voting. In our case, instead of different type of classifiers, we combined outputs of nearest neighbor classifiers trained by different blocks that correspond to different regions on a face image.

For 16x16 blocks, we have 16 different block positions and we evaluate each block separately. For every block position, a separate nearest neighbor classifier is trained by using the features extracted over the training data for that block. From a given test image, 16 feature vectors each corresponding to a different block are extracted,

$\boldsymbol{f}_b$ representing the feature vector extracted from the b^{th} block. For each test image, local feature vector is given to the corresponding classifier and the outputs of the classifiers are then combined to make an ultimate decision for the test image.

In a classification system, output of a classifier for a test sample is the label of the decided class. For a given test dataset, we come up with a recognition rate if the true labels of test samples are provided. The decision of a Bayesian classifier depends on the posterior probabilities of classes given the sample, $\boldsymbol{x}$, denoted as $p(C|\boldsymbol{x})$, where C is the label of a class. For other classifiers, it is possible to estimate posterior probabilities as well. These posterior probabilities adds up to 1 and the class with the highest posterior probability is the decision of the classifier.

Two well-studied ways of combining outputs of several classifiers are fixed and trainable combiners. Fixed combiners operate directly on the outputs of the classifiers. Fixed combination rules can be listed as maximum, median, mean, minimum, sum, product and majority voting. Decision fusion with fixed combination for $b = 1 : B$ (number of blocks) and $i = 1 : N$ (number of classes) can be formulated as:

$$\hat{i} = \text{argmax}_i P(C_i|\boldsymbol{x}) = \text{rule}(\{P(C_i|\boldsymbol{x}_b) : b = 1...B\}), \tag{3.5}$$

where rule can be taking the mean, maximum, minimum, median, sum, product of the argument set. Majority voting does not work with posterior probabilities but decides on the classifier decision output by majority voting of the individual classifier decisions.

Unlike fixed combination methods, trainable combiners use the outputs of the classifier, class posterior probabilities, as a feature set. From the class posterior probabilities of several classifiers each corresponding to a block, a new classifier is trained to provide an ultimate decision by combining the posteriors of separate classifiers. To train a combiner, training dataset is divided into two parts as train and validation data. Validation data is tested by the classifiers trained by train data part of the training dataset. Another type of partitioning the database for calculating posterior probabilities is illustrated in Figure 3.5. This process is called stacked generalization [40]. The database is divided into several partitions, first level classifier is trained with some partitions and tested with validation part of the

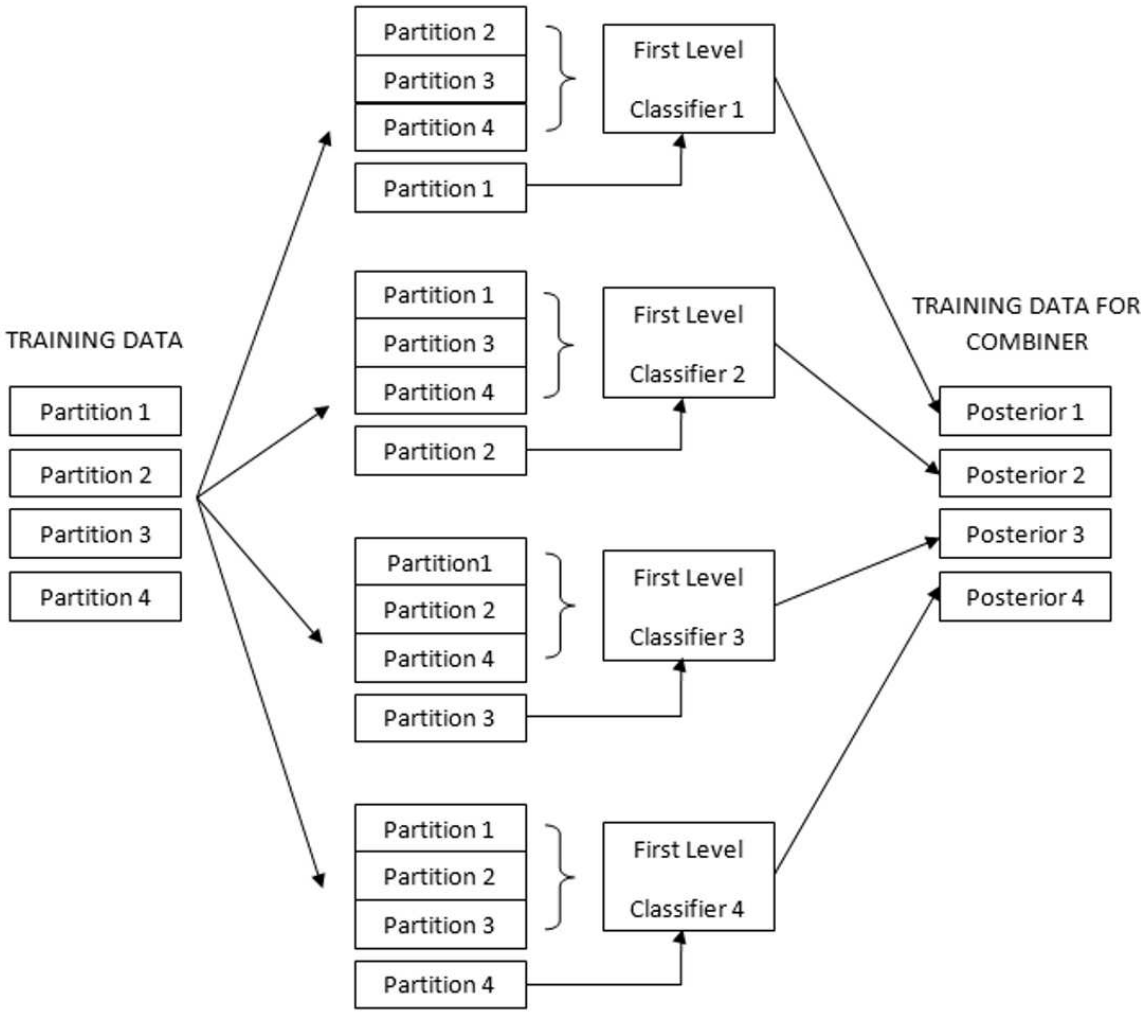

Figure 3.5: Partition of the database for stacked generalization.

data. This process is repeated by changing the validation part and training first level classifiers with remaining data. At the last stage, outputs of the first level classifiers, class posterior probabilities are stacked as in Figure 3.5. This data is used for training the combiner

The resulting class posterior probabilities of the classifiers are then trained by a separate classifier. The last level classifier that is trained from the posterior probabilities, does not need to be the same type of classifiers that are used for calculating posterior probabilities. Once the class posterior probabilities for each block are calculated from validation data, these posterior probabilities are concatenated into a long vector ($[p(C_1|\boldsymbol{x}_1), p(C_2|\boldsymbol{x}_1), ..., p(C_{N-1}|\boldsymbol{x}_B), p(C_N|\boldsymbol{x}_B)]^T$) which is then used to train the combiner. However, the length of input feature vectors of the combiner,

makes it difficult to train a classifier for multi-class classification problems. The length of the class posterior probabilities from each classifier are equal to the number of classes (N). As each classifier is trained by features extracted from separate blocks, classifier number is equal to the number of blocks (B). So, input feature set of the last level classifier is (NxB)-dimensional. Therefore, we did not prefer to build a conventional trainable combiner for decision fusion.

In sum rule, the posterior probabilities for one class from each classifier are summed. Similar to the sum rule, one can also perform weighted summation of posterior probabilities. Intuitively, we would like to weight successful classifiers more. It is not clear how to learn those weights. So, we developed methods to determine those weights in a weighted sum rule in this thesis.

If we denote the contribution or weight of each block with w_b and for a given sample $\boldsymbol{x}$ posterior probability of i^{th} class for the b^{th} block as $p(C_i|\boldsymbol{x}_b)$, weighted sum of posterior probabilities for class i is given by:

$$p(C_i|\boldsymbol{x}) = \sum_{b=1}^{B} w_b p(C_i|\boldsymbol{x}_b). \tag{3.6}$$

In the remaining part of this chapter, several weighting schemes are presented to combine outputs of classifiers for decision fusion. Note that this method can also be considered under the umbrella of trainable combiners since the weights can be learned from data as we show in the following. However, it is not a conventional trainable combiner.

3.4.1 Block Weighting

In block weighting, weights calculated from the whole training dataset are used for all samples of test dataset which means we assume contribution of blocks to the recognition performance is constant and independent from the variations in the test samples. For a block size of 16x16, 16 weights are found for all blocks and for each sample in the test dataset, posterior probabilities of blocks are multiplied by these weights. Final decision is given depending on the value of the summation of these weighted posterior probabilities. In our study, we use several different weighting methods.

Equal Weights (EW)

One of the weighting schemes is to assign equal weight to all blocks. This is equivalent to the sum rule or mean rule of fixed combiners. So, contribution of each block is assumed to be the same and equal to 1/number of blocks.

$$w_b = \frac{1}{B}. \tag{3.7}$$

For the other methods that are described in the following parts, we employ stacked generalization on the M2VTS database to train the weights. For the AR database, training dataset is partitioned into two as train and validation. Using train part, classifiers are trained and by using validation part as input, class posterior probabilities from first level classifiers are obtained in order to calculate block weights.

Score Weighting (SW)

The first weighting scheme, which we name as score weighting, depends on the posterior probability distribution of true and wrong labels on 16 blocks. In this method, for a single sample in the validation dataset, class posterior probabilities are calculated and posterior probability of the true class (let's say true class is i) at each block, $(p(C_i|\boldsymbol{x}_b))$, (16x1 vector) is labeled as positive score. For a sample $\boldsymbol{x}$ in the validation data, positive score vector is shown as:

$$\mathbf{PS} = \left[\begin{array}{cccc} p(C_i|\boldsymbol{x}_1) & p(C_i|\boldsymbol{x}_2) & ... & p(C_i|\boldsymbol{x}_B) \end{array} \right].$$

Remaining posterior probabilities of wrong classes, where $j = 1 : N$ and $j \neq i$, $[p(C_j|\boldsymbol{x}_1), p(C_j|\boldsymbol{x}_2), ..., p(C_j|\boldsymbol{x}_B)]$ are labeled as negative score vectors.

For each sample, this procedure is repeated and positive score and negative score matrices are combined in order to create two datasets which consist of class posterior probabilities of blocks.

Our aim is to find a weight for each block so that successful blocks are weighted more. Linear Discriminant Analysis (LDA) finds the linear combination of vectors, such that these vectors are most separated in the projected space. If we successfully project our positive score and negative score vectors to 1-dimension where they can

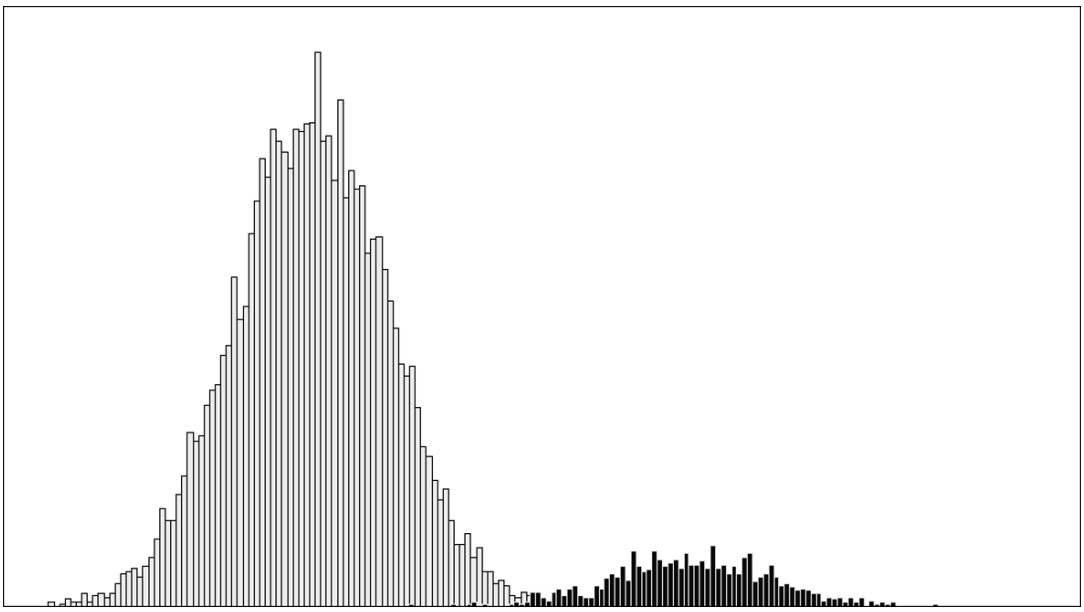

Figure 3.6: Distribution of positive scores (on the right handside) and negative scores (on the left handside) in 1-dimension. Note that, there are more negatives than positives.

be separated, we can use the coefficients used for this mapping as our weights for each block.

By combining these two datasets, we get a 16-dimensional and two-class dataset. Then the dimension of this dataset is reduced to one from 16 by using LDA and elements of the resulting dimension reduction vector of LDA are used as block weights. Distribution of positive scores and negative scores, after projecting to 1-dimension is presented in Figure 3.6. Note that, in this example, positive scores are projected to the right side and negative scores are projected to the left side. However, LDA may have projected these two classes in an opposite way, so that negative scores are higher than in the projected space and this is not the case we seek for. Therefore, in the projected space positive scores should be higher than negative scores and if the projection results in the opposite way, a change of signs on the weights is required. Note that, this procedure may yield negative weights for some blocks which may be counter-intuitive. In practice, we observed some small negative weights in the weight vector, but this did not cause any problems.

Validation Accuracy Weighting (VAW)

Another weighting scheme, which we name as validation accuracy, depends on individual recognition rates of each block on validation data. Using training data, a single classifier is trained for each block and each block of a sample in the validation data is classified using the classifier that corresponds to the block of interest. Individual block recognition rates for all samples in the validation data are acquired separately and weights are assigned proportional to the recognition accuracy of each block. If $\mathrm{acc}(k)$ denotes the recognition accuracy for the k^{th} block, weight of the b^{th} block is given as:

$$w_b = \frac{\mathrm{acc}(b)}{\sum_{k=1}^{B} \mathrm{acc}(k)}. \tag{3.8}$$

Therefore, blocks are weighted depending on their recognition capacity independently from each other. In addition to weights that are calculated proportionally to the validation accuracy, their second or higher powers might also be assigned as weights if we want to attach more importance to the blocks that are more accurate at recognition.

However, the most trusted blocks in the validation data might not contain that much information in a test image. Because, that blocks in a test sample can be partially or fully occluded and by assigning higher weights to these blocks may lead to misclassification. Therefore, a weighting scheme that depends on the training dataset may not be trustworthy and a more interactive scheme that is related with the test sample is believed to provide better weight assignments to blocks.

3.4.2 Confidence Weighting and Block Selection

Confidence weighting differs from block weighting in the sense that, in block weighting, weights are fixed and for all test samples, same weights for blocks are used. In confidence weighting, each test sample is treated separately and individual block weights for each test sample is calculated. Two weighting schemes can be termed as offline weighting and online weighting. In online weighting, we would like to estimate block weights from features extracted from current testing data. For each block of data, we would like to determine the reliability or confidence of that block

online. For that purpose, we would like to define some "confidence features" which could be used to determine the reliability of each block. Still, for learning, we use noisy validation data (with artificially added noise) to determine the online weights from confidence featurs.

As in the block weighting, training dataset is divided into train and validation parts but instead of class posterior probabilities, class distances are used in confidence weighting as decision criteria. As we use nearest neighbor classifier for training, the distance used is the distance of a feature vector to the nearest feature vector belonging to each class.

By using the validation data, class distances are calculated and several different confidence features are extracted from these distances. We have used the following confidence features:

1. First feature is the distance of a feature vector of a block to the mean feature vector of the corresponding block. Representing a normalized feature vector as $\tilde{\boldsymbol{f}}^b$ and mean block feature vector of that block as $\boldsymbol{\mu}^b$, first feature is $||\tilde{\boldsymbol{f}}^b - \boldsymbol{\mu}^b||$.

$$f_1 = ||\tilde{\boldsymbol{f}}^b - \boldsymbol{\mu}^b||. \tag{3.9}$$

   This first feature provides information about the closeness of the current block to the mean block. If the block of interest is close to the mean block, it can be concluded that this block is useful and carries information for recognition.

2. Second feature is extracted from class distances and it is the difference between the distance of the feature vector to the closest class and to the second closest class.

$$f_2 = (\boldsymbol{D}(2) - \boldsymbol{D}(1)). \tag{3.10}$$

   This feature gives idea about how close is the closest class and whether that distance is reliable in deciding the true class. If the difference between these two distances is not small, then the effect of block of interest is positive and weight of this block should be accordingly high.

3. The third feature is similar to the second one. It is the difference between the distance of the feature vector to the closest class and to the furthest class.

$$f_3 = (\boldsymbol{D}(N) - \boldsymbol{D}(1)). \tag{3.11}$$

Having the same motivation as previous feature, this difference also provides information about the reliability of the block. If this difference is small,it can be said that the block of interest is not useful in deciding the true class.

4. The final feature is named as similarity and is a measure of closeness of a block to the mean block as in the first feature.

$$f_4 = \frac{(\tilde{\boldsymbol{f}}^b_{test})^T \boldsymbol{\mu}^b}{||\tilde{\boldsymbol{f}}^b_{test}||.||\boldsymbol{\mu}^b||}. \tag{3.12}$$

From validation data, class distances are calculated and for each block of every sample in the validation data, these four confidence features are extracted. We concatenate these four features, create a 4-dimensional feature vector and label each vector as correctly classified or misclassified according to the individual classification result of that block.

$$\boldsymbol{f} = [f1, f2, f3, f4]. \tag{3.13}$$

With this 4-dimensional dataset a 2-class linear discriminant classifier (ldc) is trained. For each block of a test sample, same 4 features are extracted and tested by this classifier. The output of the classifier is two posterior probabilities for two classes: correct or incorrect classification. If the extracted features of the current block is similar to the features extracted from validation data, the block will be helpful in recognition process and vice versa. So, for a block of a test sample, the posterior probability of correct classification is assigned as the weight of that block.

In addition to the confidence weighting on blocks, applying block selection may be beneficial against the problem of facial occlusion. According to some criteria, we can sort the blocks depending on their importance and select the first few blocks which are thought to be important. This is equivalent to using a weight of zero for

the discarded blocks. This way, both the performance and speed of face recognition can be increased by using only important local regions.

For the criteria to sort blocks according to their importance, we use block similarity (f_4) which is introduced in confidence weighting. These calculated similarity scores are ordered and the blocks that have higher scores are used for face recognition. Although confidence weighting scheme is a reasonable method, it is a difficult problem to learn confidence or reliability of a block. Therefore, it does not provide promising recognition accuracies as discussed in Chapter 4.

Chapter 4

Experimental Results and Discussions

4.1 Databases and Experiment Set-Up

For experiments, we used two different face databases, the M2VTS and the AR face database. Details regarding each database will be presented in the remaining of this chapter. Face images are detected from databases using Viola-Jones face detector [17] and no human interaction is required such as marking eye centers. Therefore all experiments implement a fully automatic face recognizer. For classification, we used the nearest neighbor classifier with Euclidean distance. In our experiments, we analyzed the effects of different block sizes (8 and 16), several dimensionality reduction and normalization techniques and decision fusion methods.

Figure 4.1: Sample face images from M2VTS database. In each column, there are sample images from the same subject.

4.1.1 M2VTS - Multi Modal Verification for Teleservices and Security applications

The M2VTS database is made up from faces of 37 different people and provides 5 video shots for each person. These shots were taken at different times and drastic face changes occurred in this period. The database consists of two different videos of 37 people in 5 different tapes and we used few frames extracted from the videos. During each session, people have been asked to count from '0' to '9' in their native language in the first video and rotate their head from 0 to -90 degrees, again to 0, then to +90 and back to 0 degrees in the second video. We only used the counting videos. For each person in the database, the most difficult tape is the fifth one in which several variations exist. In the fifth tape variations such as tilted head, closed eyes, different hairstyle and accessories such as hat or scarf are present.

Apart from the fifth tape, the database can be considered as having been produced under ideal shooting conditions such as good picture quality, nearly constant lighting and uniform background. However, some impairments that are not expected can be noticed.

This kind of imperfections together with the occlusions and lighting variation are present in real life problems and will appear when implementing a practical face recognition system. In addition, people will expect the recognition algorithms to be able to deal with such problems and require this kind of databases to test the robustness of their recognition algorithms on these imperfections.

The M2VTS database consists of five videos of 37 subject recorded at different times. We selected random 8 frames from each video, so a total of 40 images are extracted for each subject. The first four sessions (tapes) are used as training data (8x4=32 images for each subject) and the last tape which includes variation such as different hairstyles, hats and scarfs, is used as test data (8 images for each subject). Thus, in our dataset we have 1184 (37x32) training images and 296 (37x8) test images. For validation purposes, we use 1 tape in the training data as validation tape and the remaining 3 tapes as train data and we repeat this step for each tape in the training data.

Figure 4.2: Sample images of a subject for tape number 1 (from the M2VTS database).

Figure 4.3: Sample images of a subject for tape number 5 (from the M2VTS database).

4.1.2 AR Face Database

This face database was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B [41]. It contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). Each person participated in two sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions. Figures 4.4 and 4.5 illustrates images of the same subject in both sessions. In each session, there are 13 images of the subject and each subject has 26 face images totally in

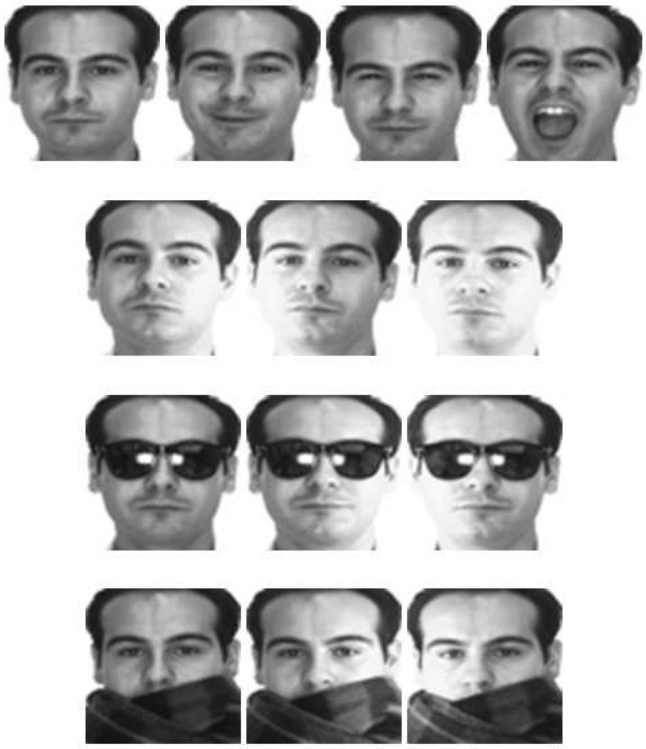

Figure 4.4: Sample images of a subject for first session (from the AR database).

two sessions. We have selected 65 male and 55 female subjects within 126 people due to some missing images. Totally there are 120 subjects in the subset of the AR database that we use and each subject have 26 images taken in two different sessions.

In each session, first 7 images are faces with different facial expressions and illumination conditions and remaining 6 images are partially occluded images (either wearing sun glasses or scarf). We separated our database into two as training and testing. In training dataset, we have the first 7 images of each subject for both sessions (7x2=14 images for each subject) and remaining 6 images are reserved as test dataset (6x2=12 images for each subject). Therefore, in this dataset we have 1680 (120x14) training images and 1440 (120x12) test images. For validation purposes, we use the first 7 images of the first session as validation data and the first 7 images of the second session as train data.

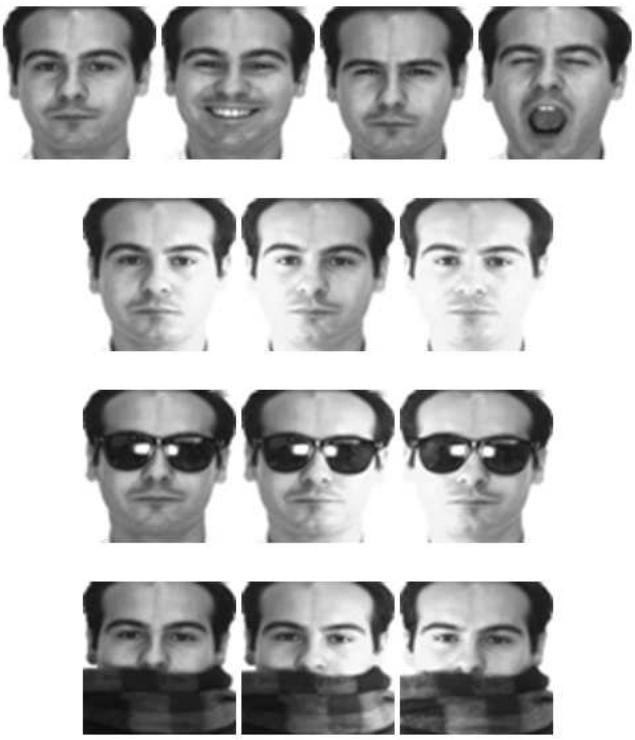

Figure 4.5: Sample images of a subject for second session (from the AR database).

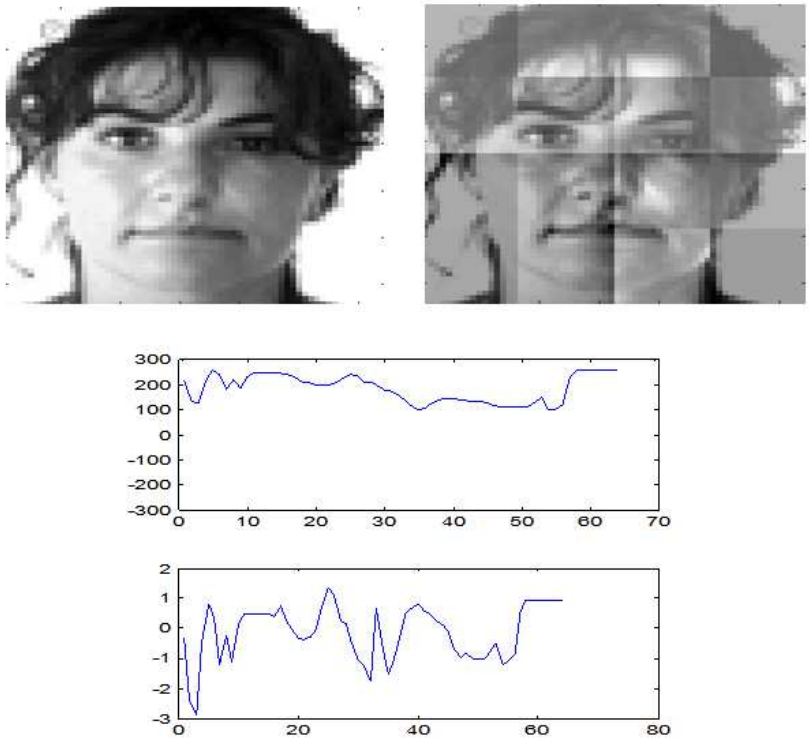

Figure 4.6: Effect of image domain normalization on a face image (above) and on a single row of the same image (below) using 16x16 blocks (image from the AR database).

4.2 Closed Set Identification

Face recognition process of identifying an unknown individual if the individual is known to be in the database is called closed set identification. The term "face recognition" is mostly used to mean closed set identification in the literature. Most of our results are closed set identification accuracies as well. For both of the databases, we performed closed set identification by either feature fusion or decision fusion.

4.2.1 Experiments with the M2VTS Database

On the M2VTS database, we have performed both feature fusion experiments and decision fusion experiments and we present the effects of different block sizes, dimensionality reduction and normalization techniques.

Feature Fusion Experiments

First set of experiments are done on the database to see the effect of image domain mean and variance normalization. As this normalization is supposed to eliminate illumination differences across sessions, recognition rates are expected to increase. For 16x16 blocks a comparison of recognition rates without and with image domain normalization can be found in Table 4.1.

Table 4.1: Effect of image domain normalization for 16x16 blocks (on the M2VTS database)

	w/o image domain normalization	**with image domain normalization**
DCT	85.47%	87.84%
PCA	83.78%	87.50%
LDA	84.80%	84.46%
APAC	86.15%	86.15%
NPCA	83.78%	87.50%
NLDA	87.16%	83.11%
NNDA	85.47%	87.84%

Table 4.2: Feature fusion results on the M2VTS database for all normalization methods with image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	87.16%	88.18%	83.11%	89.19%	87.16%
PCA	87.50%	86.15%	85.47%	86.82%	86.49%
LDA	84.46%	**92.23%**	83.78%	**91.89%**	88.51%
APAC	86.15%	**91.55%**	85.14%	**90.20%**	**90.88%**
NPCA	87.50%	86.82%	87.16%	84.80%	86.82%
NLDA	83.11%	82.43%	78.38%	82.77%	**93.45%**
NNDA	87.84%	87.16%	85.47%	85.47%	86.15%

Test results indicate that image domain normalization indeed increases recognition performance. It can be seen as a preprocessing step before feature extraction and normalization and it has a positive effect on the success of other procedures and methods. For all dimensionality reduction and feature normalization methods, results are presented in Table 4.2.

In Table 4.2, recognition rates with 16x16 blocks are shown in the presence of image domain normalization. Results of DCT and PCA follow a similar pattern and they both produce mediocre results. But, in some cases PCA's accuracies are slightly higher than DCT. NPCA and NNDA provide similar results to those two methods. Although LDA does not perform well in the absence of image domain normalization, together with image domain normalization, LDA provided recognition rates as high as APAC and in some cases outperform APAC with accuracies such as 92.23% and 91.98%. The highest recognition rate is again obtained by NLDA with 93.45%.

For 16x16 blocks with image domain normalization, ND and BMVN are helpful for recognition performance especially for LDA and APAC. In addition, FMVN provides high recognition rates when combined with LDA and APAC, and the highest recognition rate is again obtained by FMVN.

When we consider the 16x16 block size, image domain normalization has positive effect and improves recognition rates of all dimensionality reduction methods.

APAC and NLDA are the two dimensionality reduction methods which provide the highest recognition accuracies. In the presence of image domain normalization, LDA also performs well. Due to the fact that, there are enough training samples from each class for LDA and its derivatives, APAC and NLDA, these methods are very helpful in discriminating classes in the projected space. When we consider feature normalization methods, FMVN is a helpful tool that increases recognition rates and for both block sizes and whether image domain normalization is applied or not, highest recognition rates are provided in the presence of FMVN. In addition, although other feature normalization methods perform inconsistently, image domain normalization also increases their performances. Recognition results for all block sizes and normalization methods on the M2VTS database are presented in Appendix A.

Decision Fusion Experiments

We have conducted several experiments on the M2VTS database that shows the effects of decision fusion methods. After concluding that using 16x16 blocks performs better than using 8x8 blocks, we have tried several fusion methods on 16x16 blocks for different dimensionality reduction and normalization methods. We do not include the recognition results for all cases for brevity but accuracies for all dimensionality reduction and normalization methods are presented in Appendix B.

In Tables 4.3 and 4.4, decision fusion accuracies both in the absence and presence of image domain normalization are presented. In both tables, results when no feature normalization method is applied are given. Except DCT, image domain normalization plays a positive role in increasing recognition accuracies of different dimensionality reduction techniques.

The most successful dimensionality reduction methods for block weighting are DCT and NNDA. DCT, independent of any normalization method, always provide high recognition rates for both score weighting (SW) and validation accuracy weighting (VAW). The highest recognition rate of 97.30% is provided by DCT with ND (Table 4.5). In the absence of normalization methods, NNDA does not perform significantly but with or without image domain normalization, NNDA performs close results to DCT in most of the cases. The second highest recognition rate which is 96.96% is provided by NNDA when SVN (Table 4.6) is used. Other dimensionality

Table 4.3: Decision fusion results on the M2VTS database without any feature normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	96.28%	96.96%	96.28%	93.58%	96.62%
PCA	88.85%	88.51%	88.85%	88.18%	88.85%
LDA	85.81%	86.15%	85.81%	85.47%	85.47%
APAC	86.15%	88.18%	86.82%	87.16%	86.82%
NPCA	88.85%	88.85%	89.19%	88.51%	88.85%
NLDA	89.19%	89.53%	89.19%	89.53%	89.19%
NNDA	89.19%	89.19%	89.19%	89.53%	89.19%

Table 4.4: Decision fusion results on the M2VTS database without any feature normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	92.91%	94.26%	94.26%	94.26%	94.26%
PCA	90.54%	92.57%	91.55%	91.22%	91.55%
LDA	86.82%	90.20%	88.18%	90.20%	86.82%
APAC	87.50%	90.54%	88.51%	89.86%	88.18%
NPCA	91.22%	93.24%	91.55%	91.22%	91.55%
NLDA	87.84%	87.84%	88.85%	91.22%	88.85%
NNDA	93.92%	95.27%	94.93%	94.59%	94.93%

reduction methods perform inconsistently and in some cases, they provide accuracies as high as 94.93% for PCA with SVN (Table 4.6) and 93.92% (Table B.10) for LDA with FMVN. However, dimensionality reduction methods apart from DCT and NNDA, do not perform significantly higher for all normalization and weighting methods. In addition, it can be said that all normalization methods are useful on the M2VTS database and increase recognition performances.

Table 4.5: Decision fusion results on the M2VTS database with norm division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	95.95%	96.96%	**97.30%**	96.96%	96.96%
PCA	88.51%	88.85%	88.85%	89.19%	88.85%
LDA	85.47%	84.80%	84.80%	83.78%	85.47%
APAC	91.22%	90.20%	90.54%	90.54%	90.88%
NPCA	88.51%	89.19%	88.85%	89.19%	89.19%
NLDA	92.57%	90.88%	92.91%	92.57%	92.91%
NNDA	89.19%	89.19%	89.19%	89.19%	89.19%

Table 4.6: Decision fusion results on the M2VTS database with sample variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	93.92%	94.26%	96.28%	95.61%	94.59%
PCA	94.59%	92.57%	94.29%	93.24%	94.93%
LDA	86.15%	89.19%	88.85%	90.20%	87.50%
APAC	86.82%	90.88%	89.19%	88.15%	88.85%
NPCA	94.26%	92.91%	94.93%	92.91%	94.26%
NLDA	92.23%	90.54%	92.23%	92.93%	92.23%
NNDA	94.59%	**96.96%**	95.61%	95.61%	95.95%

By block weighting, we aim to find the contribution of each block to the recognition. Therefore, our goal is to find weights that result in a performance better than using equal weights. Although there are few exceptions, in almost all cases, using the weights we have calculated, provide higher recognition rates than using equal weights. As an example, weights for 16x16 blocks when DCT and ND is applied on the M2VTS database, are illustrated for SW and WAV.

$$w_{SW} = \begin{bmatrix} 0.0632 & 0.0699 & 0.0811 & 0.0480 \\ 0.0737 & 0.1126 & 0.0790 & 0.0553 \\ 0.0426 & 0.0654 & 0.1035 & 0.0502 \\ 0.0027 & 0.0738 & 0.0851 & -0.0062 \end{bmatrix}.$$

$$w_{VAW} = \begin{bmatrix} 0.0569 & 0.0853 & 0.0707 & 0.0642 \\ 0.0646 & 0.0890 & 0.0866 & 0.0589 \\ 0.0459 & 0.0715 & 0.0744 & 0.0459 \\ 0.0232 & 0.0618 & 0.0731 & 0.0280 \end{bmatrix}.$$

After presenting and discussing feature and decision fusion results on the M2VTS database, it is evident that although feature fusion provides promising results as high as 93.45%, decision fusion outpowers feature fusion and provide higher accuracies such as 97.30% and 96.96%.

4.2.2 Experiments with the AR Database

Same set of experiments are also conducted on the AR database and the results are presented.

When compared with the M2VTS database, the AR database has almost four times more subjects and training sample/subject ratio is much smaller for the AR database (this ratio is 32 in the M2VTS for 37 subjects and 14 in the AR for 120 subjects). Illumination changes are much more drastic in the AR database. In addition, wide variety of accessories are present in the AR database where the M2VTS database does not include that much variation. As a result, recognition rates for the AR database is much lower than accuracies obtained in the M2VTS database.

Table 4.7: Feature fusion results on the AR database for all normalization methods without image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	41.35%	**46.15%**	41.28%	45.26%	43.91%
PCA	45.32%	**45.71%**	42.18%	45.19%	44.55%
LDA	31.09%	27.88%	24.36%	27.63%	31.35%
APAC	31.86%	26.22%	29.74%	24.87%	31.92%
NPCA	45.32%	**45.71%**	42.24%	44.74%	44.04%
NLDA	32.76%	35.58%	27.88%	35.19%	33.33%
NNDA	42.31%	**48.08%**	39.81%	**47.24%**	43.40%

Feature Fusion Experiments

If we analyze the performance of dimension reduction methods for the AR database, we can say that less data dependent transforms such as DCT, PCA and NNDA generally provide higher recognition rates. Also, a generalized version PCA, NPCA performs better than LDA, APAC and NLDA. DCT is independent from the nature of the data and its performance is not affected by the lack of training data. However, LDA and its derivatives, APAC and NLDA, face problems when there is not enough training sample for each class.

In Table 4.7, recognition rates with 16x16 blocks when image normalization is not applied, are given. DCT, PCA, NPCA and NNDA performs better than LDA, APAC and NLDA for all situations. The highest recognition rates are provided by NNDA for two different normalization methods.

In Table 4.8, recognition rates with 16x16 blocks when image normalization is applied, are given and the recognition rates follow similar patterns. Again DCT, PCA, NPCA and NNDA provide higher accuracies than LDA, APAC and NLDA due to the fact that is mentioned before.

For both cases, whether image domain normalization is applied or not, recognition rates are very close to each other, which shows that image domain normalization is not working for the AR database and oppose to its functionality in the

Table 4.8: Feature fusion results on the AR database for all normalization methods with image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	44.62%	**46.15%**	43.97%	**45.26%**	**45.19%**
PCA	42.95%	41.92%	43.72%	42.05%	42.76%
LDA	30.19%	42.05%	29.29%	41.67%	36.86%
APAC	30.45%	36.92%	29.74%	35.51%	37.05%
NPCA	43.01%	41.86%	43.85%	42.69%	42.56%
NLDA	29.62%	37.76%	26.99%	37.82%	33.08%
NNDA	41.54%	43.72%	39.55%	42.56%	41.60%

M2VTS database, it decrease recognition rates for the AR database. We attribute this situation to the function and aim of image domain normalization. By image domain normalization, we aim to decrease variations between images of the same subject. Images of the subjects are taken in different sessions and inside a session, there are illumination differences across images. Image domain normalization tries to makes image of same subject as close to each other. This idea works for the M2VTS database because the images of the same subject are very apart from each other across sessions. Illumination changes are high across sessions and image domain normalization decreases these variations to some degree and its positive effect is proved in the recognition results of the M2VTS database. However, in the AR database train and test data have almost identical illuminations. If we analyze the images of same person shown in Figure 4.4, we see that test images (last two rows with sun glasses and scarf) has three types of illuminations, none, light from right and left. In the training data, we have similar images of the subject having none illumination and light from left and right. Therefore, nearest neighbor classifier is able to match the test images with train images. In the presence of image domain normalization (an example for the AR database is provided in Figure 4.6), train and test images become similar in terms of illumination, which is almost uniform, but this does not help in recognition success of nearest neighbor classifier as it helps in

the M2VTS database.

Almost for all dimensionality reduction methods, ND has a positive influence and the highest recognition rates are provided by DCT and NNDA when combined with ND. Also, BMVN affects DCT and NNDA in a good way, increasing their accuracies.

Recognition results for all blocks sizes and normalization methods on the AR database are presented in Appendix A.

Decision Fusion Experiments

Same set of experiments which are conducted on the M2VTS database are also conducted on the AR database. We have seen that 16x16 blocks provide higher recognition rates than 8x8 blocks on the AR database, similar to the M2VTS database. Therefore, we have tried decision fusion methods on 16x16 blocks for different dimensionality reduction and normalization methods. We do not include the recognition results for all cases for brevity but accuracies for all dimensionality reduction and normalization methods are presented in Appendix B.

In Table 4.9, decision fusion results on the AR database without any normalization is presented. Similar to feature fusion, image domain normalization does not affect decision fusion results in a positive way due to the reason discussed above. However, feature normalization methods increase recognition rates most of the time.

The most successful dimensionality reduction methods that provide higher recognition rates are DCT, PCA and NNDA. The highest recognition rate of 85.90% and 85.97% (Table 4.10) are obtained by NNDA. In any case, NNDA provides higher results than other dimensionality reduction. However, there are some exceptions where DCT and PCA performs slightly better than NNDA. The second highest accuracies after NNDA are provided by DCT as 84.65% and by PCA as 84.58% in the presence of SVN (Table 4.11).

When the decision fusion results on the AR database are analyzed, it is clear that, both weighting schemes (SW and VAW) are successful. For all dimensionality reduction and normalization methods, both weighting schemes provide higher accuracies than equal weights for each block.

In addition to these experiments, we have also conducted experiments with single

Table 4.9: Decision fusion results on the AR database without any feature normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	74.58%	74.86%	76.74%	76.11%	76.11%
PCA	65.49%	65.90%	67.57%	65.63%	66.81%
LDA	55.35%	57.85%	64.24%	67.43%	61.18%
APAC	65.83%	66.60%	69.10%	68.96%	68.26%
NPCA	65.35%	65.97%	67.64%	65.90%	66.94%
NLDA	69.79%	70.28%	74.72%	77.01%	72.29%
NNDA	75.76%	76.60%	77.85%	78.82%	77.29%

Table 4.10: Decision fusion results on the AR database with norm division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	75.90%	76.18%	77.57%	77.29%	76.94%
PCA	78.82%	79.58%	80.83%	81.25%	80.21%
LDA	66.32%	66.60%	69.79%	71.67%	68.61%
APAC	67.78%	70.21%	71.39%	70.90%	70.56%
NPCA	78.54%	79.86%	80.49%	81.04%	80.28%
NLDA	73.40%	76.74%	77.99%	79.86%	76.74%
NNDA	83.75%	83.75%	**85.90%**	**85.97%**	85.14%

Table 4.11: Decision fusion results on the AR database with sample variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	82.71%	83.96%	**84.65%**	83.89%	83.82%
PCA	81.67%	**84.58%**	83.96%	82.78%	83.82%
LDA	62.08%	66.25%	67.57%	68.89%	65.83%
APAC	63.47%	66.67%	68.75%	69.72%	67.57%
NPCA	82.01%	84.79%	84.38%	82.99%	84.10%
NLDA	69.72%	72.99%	74.10%	75.97%	72.57%
NNDA	79.24%	82.92%	82.43%	82.01%	81.39%

Table 4.12: Accuracy of single training data experiment on the AR database

NN	42.36%
ND	44.03%
BMVN	43.82%
FMVN	45.14%

training data from each class. The aim of this experiment is to see the effects of normalization methods which are not helpful for the AR database, in both feature fusion and decision fusion experiments. As mentioned before, training dataset of the AR database consists of images with similar illumination conditions as test dataset of the AR database. By using a single training sample for each subject, we expect different normalization methods to make difference. By using DCT, we have conducted decision fusion experiment and we have used EW for weighting as we cannot compute any weights due to absence of validation data in training dataset. Recognition accuracies are presented in Table 4.12. It is clear that feature normalization methods increase recognition rates. The accuracy of 42.36% increases to 45.14% when FMVN is applied and other normalization techniques perform better than no normalization.

After presenting and discussing feature fusion and decision fusion recognition results on the AR database, similar to the M2VTS database, it is evident that decision fusion is much more successful than feature fusion. The highest recognition rate obtained by feature fusion is 48.08% where as with decision fusion recognition rates as high as 85.97% and 85.56% are obtained.

4.2.3 Confidence Weighting and Block Selection

Although confidence weighting seems a reasonable and a promising method for face recognition, recognition accuracies that we obtained were not satisfactory. The weights calculated are very close to each other and confidence weighting provides lower results than using EW. Confidence weighting is a difficult problem in our case and is not helpful, however, it might be helpful for other cases.

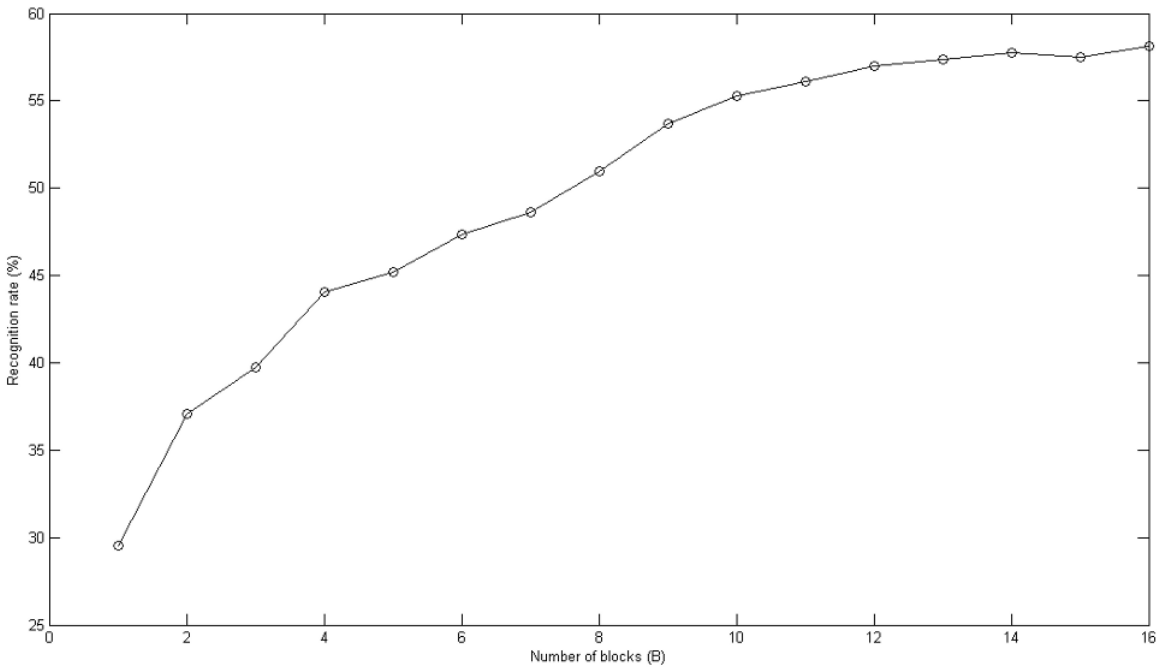

Figure 4.7: Confidence Weighting and Block Selection on 16x16 blocks

As an example to confidence weighting and block selection, we present a single case on the AR database. We have conducted confidence weighting and block selection with PCA in the absence of all normalization methods for block sizes of 16 and 8 in Figures 4.7 and 4.8.

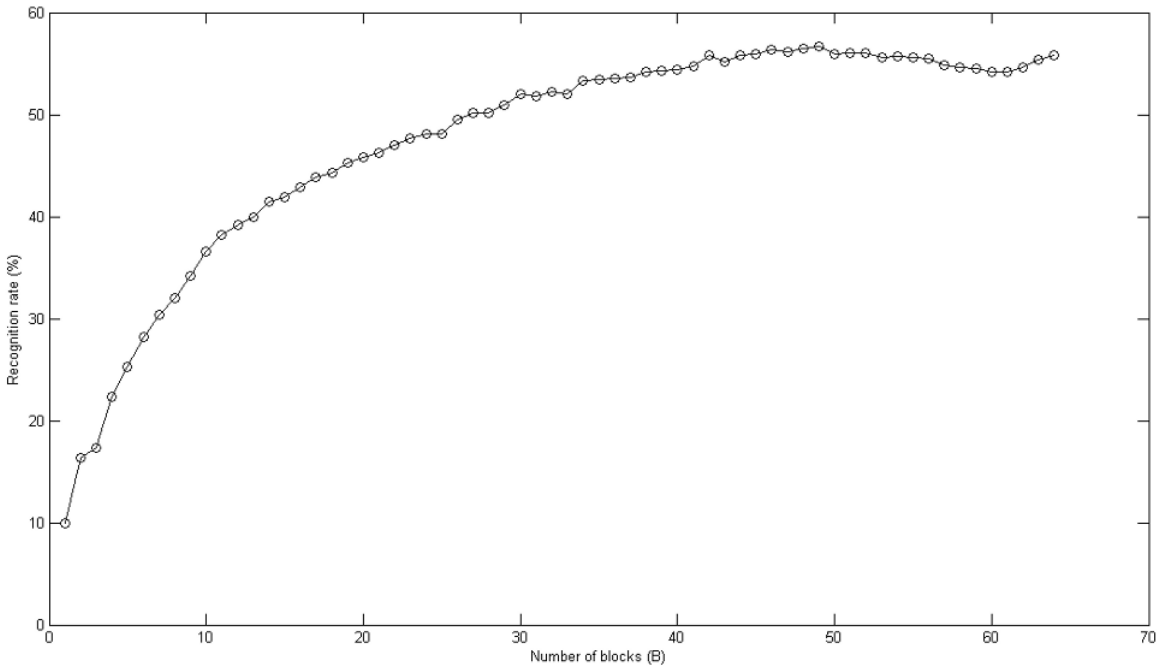

Figure 4.8: Confidence Weighting and Block Selection on 8x8 blocks

On 16x16 blocks, we have obtained accuracy of 65.49% with using EW. However, block selection does not work and for each additional block, recognition accuracy increases and the accuracy cannot reach the result of 65.49%. For 8x8 blocks, after the 49^{th} block, recognition rate begin to decrease, which means the remaining 15 blocks do not bring any additional information for classification and have negative influence. However, the accuracy reached when 49 blocks are used is lower than the accuracy of using EW.

4.2.4 Different Distance Metrics

Apart from L_2-norm for distance calculation in nearest neighbor classification, we have also conducted experiments with L_1-norm and cosine angle (COS) distance metrics. For some of the cases that provide highest recognition rates on the M2VTS database and the AR database, we also present recognition accuracies of other distance metrics in Table 4.13.

Although on the M2VTS database, other distance metrics do not increase recognition rates, on the AR database, L_1-norm and cosine angle provide slightly higher

Table 4.13: Accuracies using different distance metrics

	L_2-norm	L_1-norm	COS
M2VTS DCT	97.30%	96.62%	93.92%
M2VTS NNDA	96.96%	87.16%	91.55%
AR NNDA	85.90%	88.47%	89.10%
AR DCT	84.65%	85.53%	86.39%

recognition rates and the highest recognition rate is obtained by cosine angle for NNDA, which is 89.10%.

4.2.5 Comparison With Other Techniques

For closed set identification, we have compared our accuracies with some of the previously used baseline techniques which are implemented commonly.

The first set of algorithms that we have tried on our two databases is provided by CSU Face Identification Evaluation System [42]. It is a package that contains a standard PCA (Eigenfaces) algorithm, a combination of PCA and LDA algorithms and a Bayesian Intrapersonal/Extrapersonal Image Difference Classifier. Prior to these face recognition algorithms, a normalization is applied on face images as a preprocessing. This four step normalization consists of geometric normalization that lines up human chosen eye coordinates, masking that crops image using an elliptical mask such that only the face from forehead to chin and cheek is visible, histogram equalization and pixel normalization which is similar to our image domain normalization except it is applied on whole image instead of blocks. The recognition accuracies of these algorithms on both databases are presented in Table 4.14

In addition to CSU Face Identification evaluation system we have also conducted a set of experiments on our database in the following set up. A previously presented illumination correction algorithm which is proposed in [43], is applied on face images and global DCT and global PCA are applied on both databases. Recognition results are presented in Table 4.15.

The highest recognition rate we obtained on the M2VTS database is 97.30% and

Table 4.14: Accuracies of CSU Face Identification Evaluation System

	M2VTS	**AR**
PCA Euclidean	86.48%	22.15%
PCA Mahalinobis	88.17%	42.56%
LDA	100.00%	21.94%
Bayesian ML	91.89%	23.95%
Bayesian MAP	92.56%	27.84%

Table 4.15: Accuracies of Global DCT and PCA with illumination correction

	M2VTS	**AR**
DCT	93.58%	47.54%
PCA	89.53%	48.46%

only CSU Face Identification Evaluation System PCA + LDA algorithm provides higher recognition result higher than 97.30%, which is 100%. However, we have obtained the accuracy of 97.30% by using DCT which is computationally faster than both PCA and LDA, and also does not require training data. For the AR database, in which there is less amount of training data from each class, the highest accuracy obtained by CSU Face Identification Evaluation system is 42.56%. On the other hand, illumination correction + global PCA provide 48.46% accuracy on the AR database whereas the highest recognition rate we have obtained on the AR database is 89.10%.

4.3 Open Set Identification

Open set identification refers to face recognition process when it is unknown if a subject belongs to the database or not. So, open set identification first determines if the unknown face belongs to the database and then finds the identity of the subject from the database. Typically, if the score of the best match is not higher than a

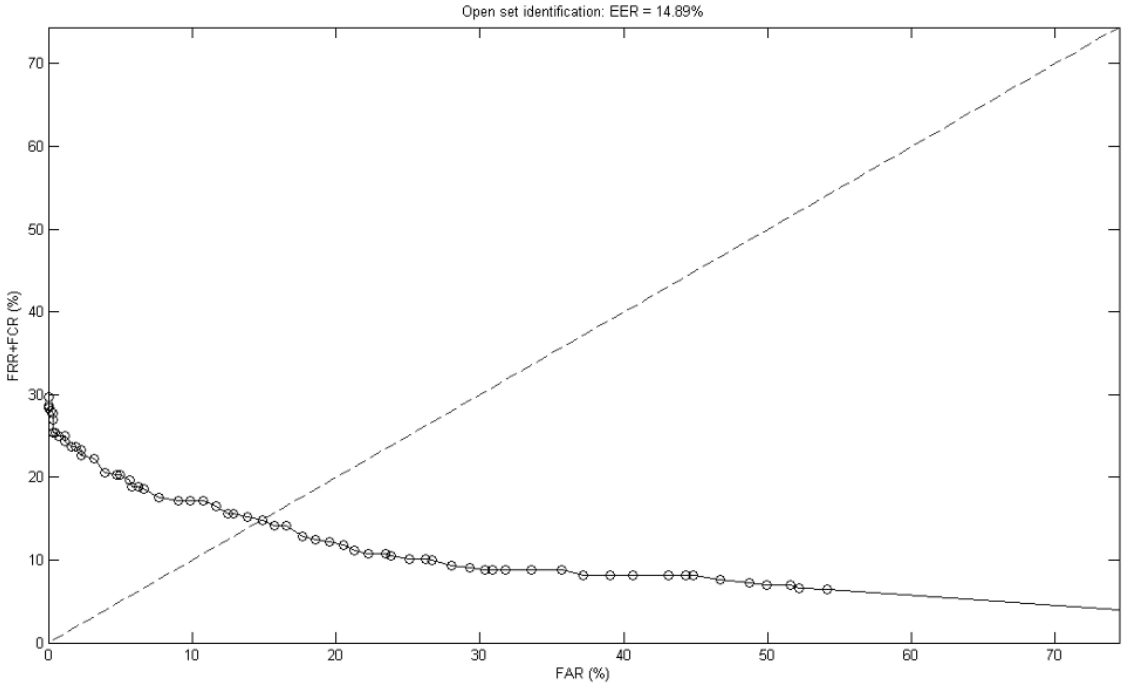

Figure 4.9: Open Set Identification Accuracy for DCT with norm division on the M2VTS

pre-determined threshold, the subject is rejected.

For some of the decision fusion experiments, in which the highest recognition rates are obtained for closed set identification, we conducted open set identification experiments. Although our databases were not designed for open-set identification experiments, we can use them in such experiments using the technique presented in [44]. ROC curve on false accept rate (FAR) versus false reject rate (FRR) plus false classification rate (FCR) (FAR vs. FRR+FRC) is plotted for these cases. For each database, we present open set identification performance of two cases.

The first case on the M2VTS database is the performance of DCT applied with ND which provides 97.30% close set recognition accuracy. In Figure 4.9, equal error rate (EER), where FAR = FRR + FRC is calculated as 14.89%.

The second case on the M2VTS database is the performance of NNDA applied with SVN and image domain normalization which provides 96.96% closed set recognition accuracy. In Figure 4.10, EER is calculated as 11.01%.

The first case on the AR database is the performance of NNDA applied with ND which provides 85.90% closed set recognition accuracy. In Figure 4.11, EER is

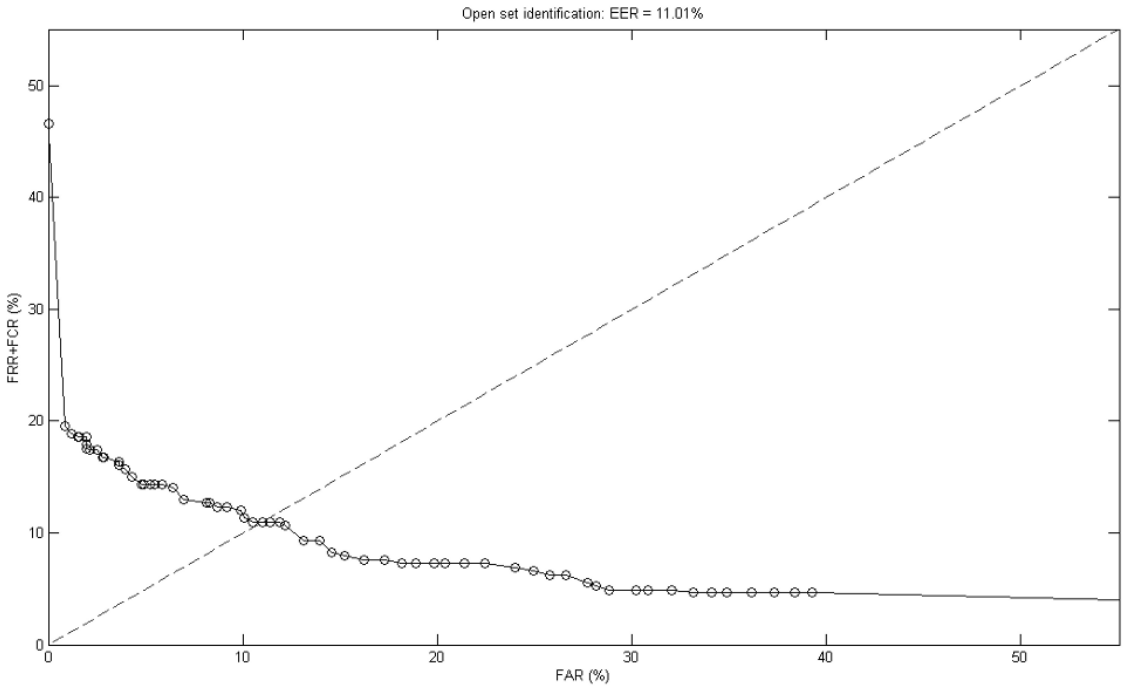

Figure 4.10: Open Set Identification Accuracy for NNDA with sample variance normalization and image domain normalization on the M2VTS database

calculated as 22.60%.

The second case on the AR database is the performance of DCT applied with SVN and image domain normalization which provides 85.90% closed set recognition accuracy. In Figure 4.11, EER is calculated as 22.04%.

4.4 Verification

Verification is the process of confirming or rejecting an individual's claimed identity. Unlike closed or open set identification, face verification deals with a two class problem, accept or reject.

For the same cases with open set identification, in which the highest recognition rates are obtained for closed set identification, we conducted face verification experiments. ROC curve on false accept rate (FAR) versus false reject rate (FRR) (FAR vs. FRR) is plotted for these cases.

The first case on the M2VTS database is the performance of DCT applied with ND which provides 97.30% close set recognition accuracy. In Figure 4.13, equal

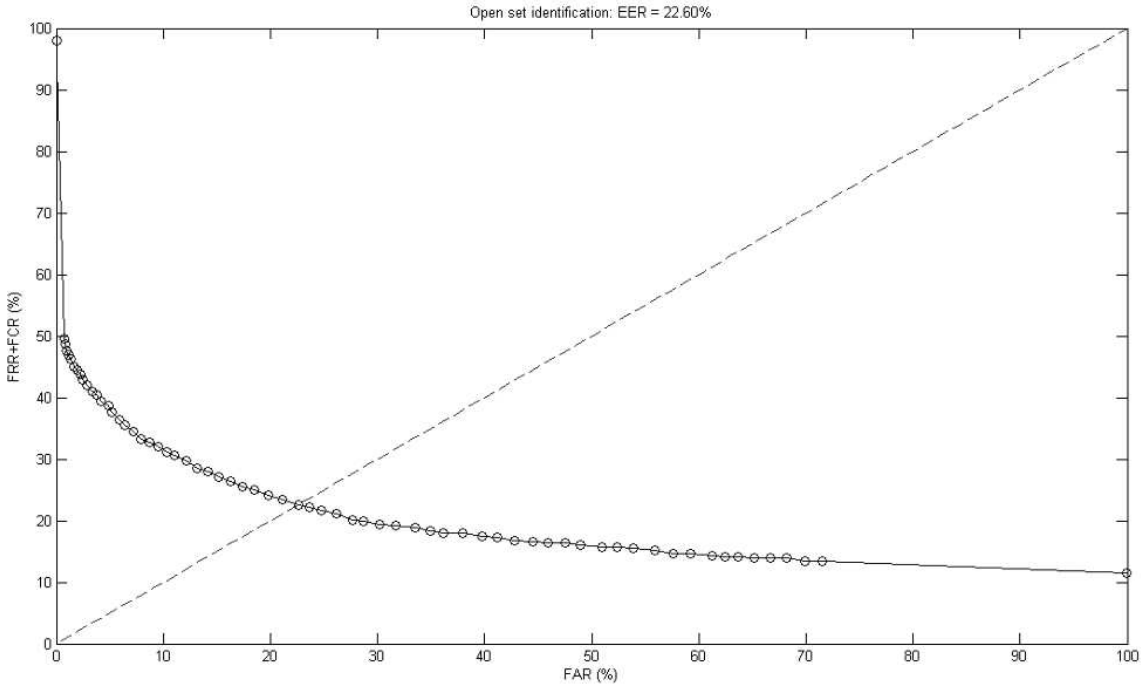

Figure 4.11: Open Set Identification Accuracy for NNDA with norm division on the AR database

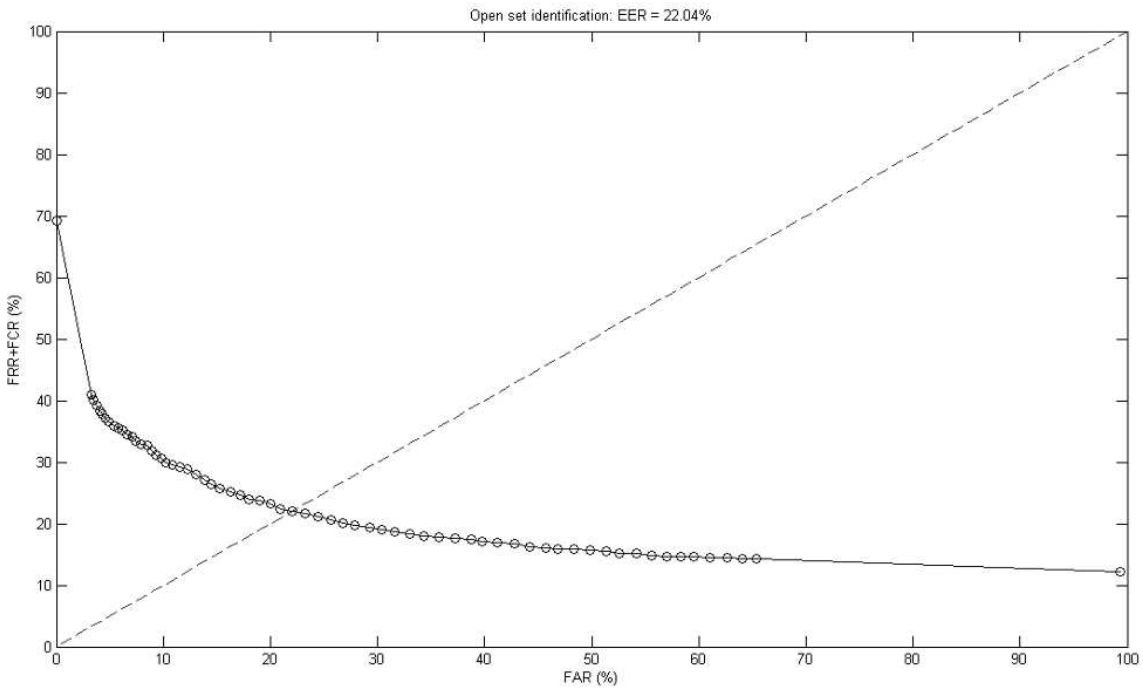

Figure 4.12: Open Set Identification Accuracy for DCT with sample variance normalization and image domain normalization on the AR database

error rate (EER), where FAR = FRR is calculated as 5.74%.

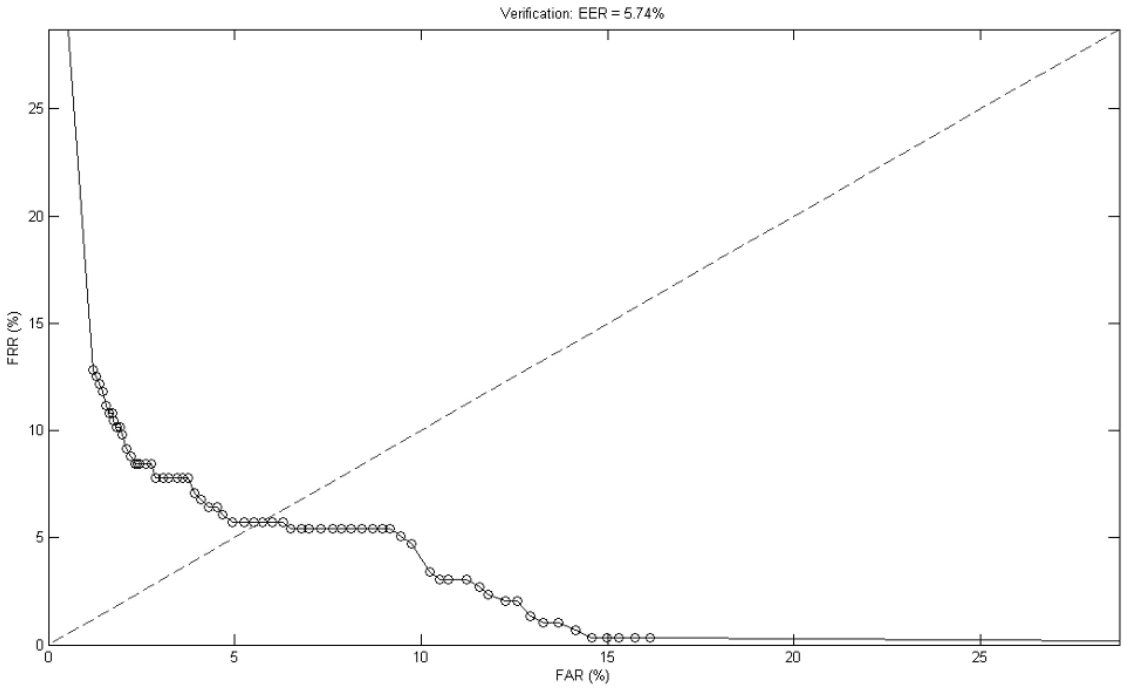

Figure 4.13: Verification Accuracy for DCT with norm division on the M2VTS database

The second case on the M2VTS database is the performance of NNDA applied with SVN and image domain normalization which provides 96.96% closed set recognition accuracy. In Figure 4.14, EER is calculated as 4.05%.

The first case on the AR database is the performance of NNDA applied with ND which provides 85.90% closed set recognition accuracy. In Figure 4.15, EER is calculated as 8.40%.

The second case on the AR database is the performance of DCT applied with SVN and image domain normalization which provides 85.90% closed set recognition accuracy. In Figure 4.16, EER is calculated as 5.97%.

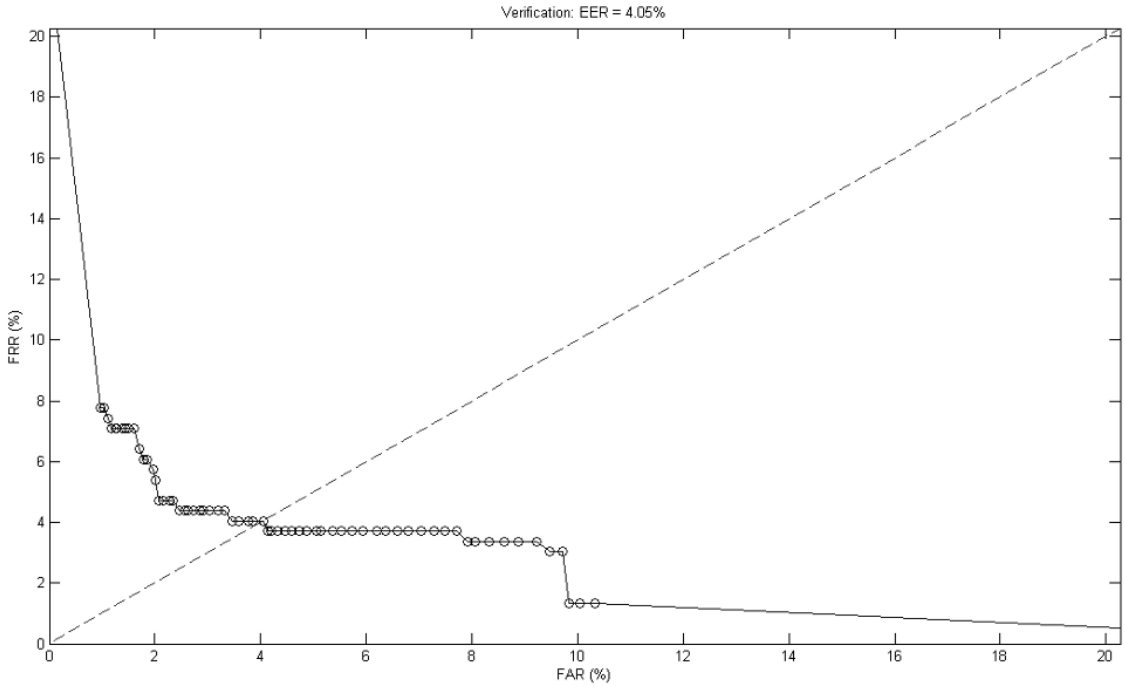

Figure 4.14: Verification Accuracy for NNDA with sample variance normalization and image domain normalization on the M2VTS database

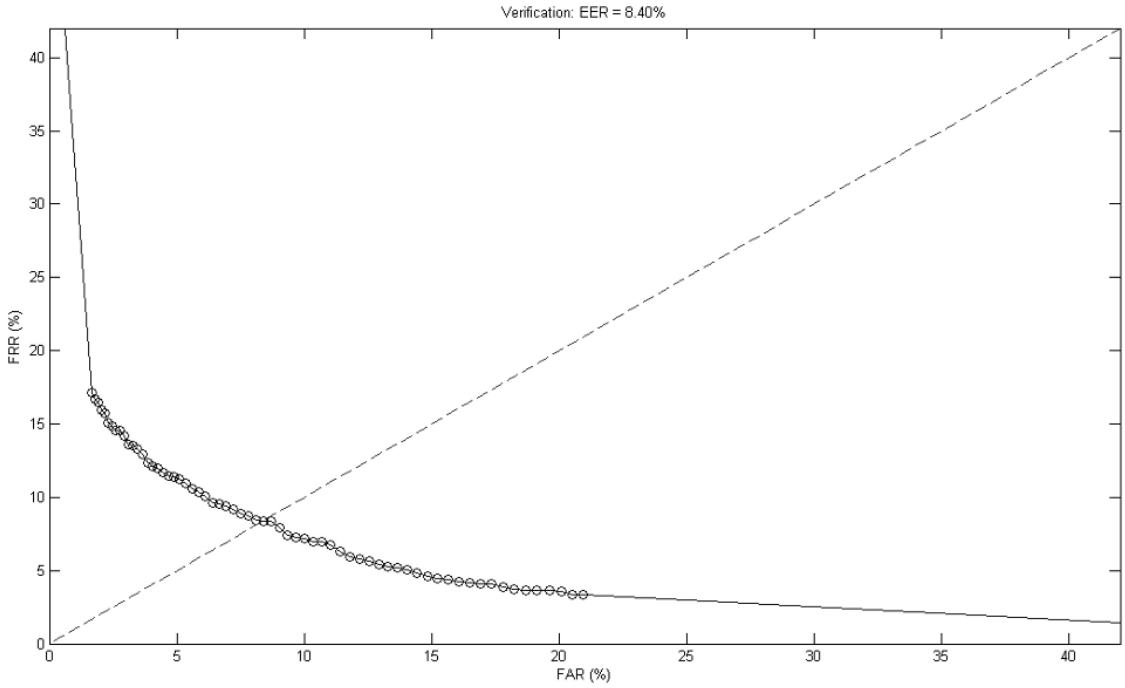

Figure 4.15: Verification Accuracy for NNDA with norm division on the AR database

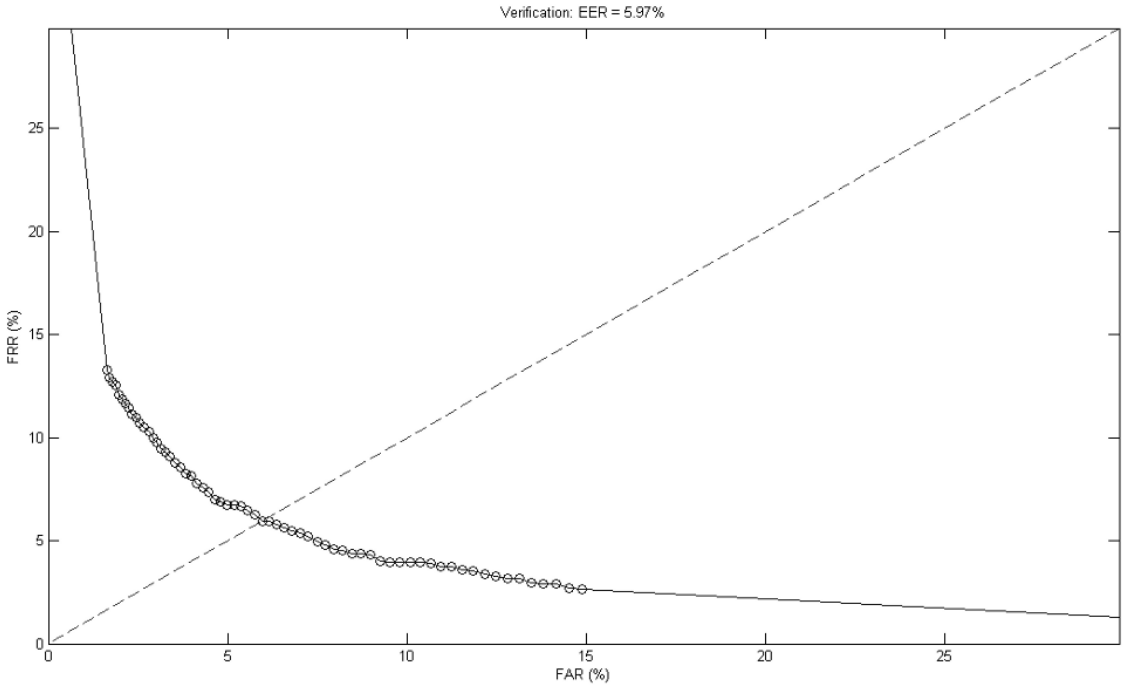

Figure 4.16: Verification Accuracy for DCT with sample variance normalization and image domain normalization on the AR database

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we have investigated different dimensionality reduction, normalization methods and decision fusion techniques for patch-based face recognition. Several experiments are conducted on two separate databases and recognition accuracies are presented. In addition to closed set identification, we have also performed open set identification and verification experiments using methods which had promising closed set identification accuracies.

One conclusion that can be made following several experiments is the superiority of patch-based recognition over global approaches. In patch-based face recognition, we have used non-overlapping blocks and extracted features using these independent blocks. By applying both feature fusion and decision fusion methods, we have outperformed previously proposed global methods. On M2VTS database, we have achieved a recognition rate of 93.45% by feature fusion and 97.30% by decision fusion. The only highest recognition rate that exceeds these two rates is the employement of PCA+LDA algorithm by CSU Face Identification Evaluation System which is 100%. However, the same method provides a recognition rate of 21.94% on the AR database, in which we reach recognition accuracies of 48.08% by feature fusion and 89.10% by decision fusion. We attribute the success of PCA+LDA algorithm on M2VTS database to the high number of training samples for each subject in M2VTS database. When there is not enough training sample for each subject, as in the AR database, PCA+LDA algorithm fails to classify face images. Apart from CSU Face Identification Evaluation System, global PCA and DCT algorithms enhanced by illumination correction provide 93.58% accuracy on the M2VTS database

and 48.46% accuracy on the AR database. We have also outperformed these two methods with our decision fusion methods.

For decision fusion, we have used weighted sum rule over class posterior probabilities of blocks. For choice of weights, we have proposed a novel methods which we name as score weighting. Also we have experimented with using validation accuracies for weight assignment. With both of these methods, we obtained recognition rates slightly higher than using equal block weights.

In addition to block weighting, we have also derived a method to assign weights to blocks of test images independently (or online), which we named as confidence weighting. This method aims to discard (or weight less) the face blocks that are occluded. As this information cannot be learned offline, we need to learn this online during testing. However by using confidence weighting and block selection using confidence weights, we could not improve the recognition accuracy obtained using equal weights. It appears it is very hard for the recognizer to not believe itself and give low confidence to its decisions, when its role is to give the best result in the first place.

We can categorize dimension reduction methods according to their dependency on training data. When there is less training data per subject, DCT, PCA, NPCA and NNDA perform better than LDA, APAC and NLDA. However, in the presence of enough number of training samples, LDA, APAC and NLDA may be superior at discriminating classes. Therefore, on the M2VTS database, LDA, APAC and NLDA perform better, providing higher recognition rates and on the AR database, due to lack of training data, highest recognition rates are obtained by DCT, PCA, NPCA and NNDA.

Influence of normalization methods depend on the nature of images. In the M2VTS database, normalization methods usually increase recognition rates as there are variations in illumination across sessions. Normalization methods strive to eliminate illumination changes and images of the same subject from different sessions become closer to each other. However, in the AR database, train and test images are taken in similar lighting conditions, so, normalization methods seem to slightly hurt the recognition process instead of improving. To illustrate this situation, we have performed face recognition on the AR database with a single training data per

subject. When normalization methods are applied, test images become closer to training images and recognition rates increase.

5.2 Future Work

As a continuation of this research, in the future, one can pursue some of the following avenues:

- Moving block centers so that each block corresponds to same location on the face for all images of all subjects.
- Using color information in addition to gray scale intensity values.
- More accurate distance to posterior probability conversion for nearest neighbor classification.
- Better dimensionality reduction techniques.
- More intelligent decision fusion methods suited to the problem, particularly better ways to estimate the weights in the weighted sum rule.

Appendix A

Feature Fusion Experiments

Table A.1: Feature fusion results on the M2VTS database for all normalization methods without image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	85.47%	88.18%	83.11%	**89.19%**	**89.19%**
PCA	83.78%	81.08%	78.04%	82.09%	84.46%
LDA	84.80%	76.35%	79.39%	75.68%	86.49%
APAC	86.15%	86.49%	82.77%	84.46%	**91.22%**
NPCA	83.78%	81.76%	78.38%	81.42%	84.46%
NLDA	87.16%	87.50%	78.04%	88.51%	**90.54%**
NNDA	85.47%	81.08%	83.11%	81.42%	85.81%

Table A.2: Feature fusion results on the M2VTS database for all normalization methods without image normalization on 8x8 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	73.99%	84.80%	74.66%	79.39%	77.70%
PCA	83.78%	71.62%	80.41%	59.46%	84.46%
LDA	82.77%	61.82%	75.68%	56.08%	84.46%
APAC	**88.18%**	77.36%	86.15%	73.99%	**90.54%**
NPCA	83.78%	72.30%	80.07%	65.54%	84.46%
NLDA	**87.16%**	81.76%	82.77%	75.00%	**87.84%**
NNDA	79.05%	53.04%	76.01%	38.18%	78.04%

Table A.3: Feature fusion results on the M2VTS database for all normalization methods with image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	87.16%	88.18%	83.11%	89.19%	87.16%
PCA	87.50%	86.15%	85.47%	86.82%	86.49%
LDA	84.46%	**92.23%**	83.78%	**91.89%**	88.51%
APAC	86.15%	**91.55%**	85.14%	**90.20%**	**90.88%**
NPCA	87.50%	86.82%	87.16%	84.80%	86.82%
NLDA	83.11%	82.43%	78.38%	82.77%	**93.45%**
NNDA	87.84%	87.16%	85.47%	85.47%	86.15%

Table A.4: Feature fusion results on the M2VTS database for all normalization methods with image normalization on 8x8 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	87.84%	84.46%	87.84%	79.39%	84.12%
PCA	89.19%	85.14%	89.86%	78.38%	86.49%
LDA	88.51%	87.50%	88.18%	85.14%	88.85%
APAC	**90.88%**	**91.89%**	**91.55%**	85.14%	**93.58%**
NPCA	88.85%	86.49%	89.86%	83.11%	86.15%
NLDA	89.19%	80.74%	88.51%	82.43%	**90.54%**
NNDA	89.53%	82.43%	89.86%	72.97%	89.86%

Table A.5: Feature fusion results on the AR database for all normalization methods without image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	41.35%	**46.15%**	41.28%	45.26%	43.91%
PCA	45.32%	**45.71%**	42.18%	45.19%	44.55%
LDA	31.09%	27.88%	24.36%	27.63%	31.35%
APAC	31.86%	26.22%	29.74%	24.87%	31.92%
NPCA	45.32%	**45.71%**	42.24%	44.74%	44.04%
NLDA	32.76%	35.58%	27.88%	35.19%	33.33%
NNDA	42.31%	**48.08%**	39.81%	**47.24%**	43.40%

Table A.6: Feature fusion results results on the AR database for all normalization methods without image normalization on 8x8 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	33.53%	42.18%	33.78%	36.35%	36.73%
PCA	**44.81%**	43.27%	42.31%	33.21%	**44.23%**
LDA	35.90%	30.06%	26.73%	24.68%	36.92%
APAC	38.27%	28.59%	37.88%	20.38%	39.94%
NPCA	**44.94%**	43.59%	41.79%	31.78%	**44.04%**
NLDA	38.33%	35.90%	38.33%	30.51%	39.87%
NNDA	21.47%	17.95%	20.38%	11.60%	21.67%

Table A.7: Feature fusion results results on the AR database for all normalization methods with image normalization on 16x16 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	44.62%	**46.15%**	43.97%	**45.26%**	**45.19%**
PCA	42.95%	41.92%	43.72%	42.05%	42.76%
LDA	30.19%	42.05%	29.29%	41.67%	36.86%
APAC	30.45%	36.92%	29.74%	35.51%	37.05%
NPCA	43.01%	41.86%	43.85%	42.69%	42.56%
NLDA	29.62%	37.76%	26.99%	37.82%	33.08%
NNDA	41.54%	43.72%	39.55%	42.56%	41.60%

Table A.8: Feature fusion results results on the AR database for all normalization methods with image normalization on 8x8 blocks

	NN	**ND**	**SVN**	**BMVN**	**FMVN**
DCT	**44.04%**	44.49%	**43.91%**	36.35%	**44.04%**
PCA	47.05%	43.40%	**47.50%**	40.19%	46.60%
LDA	42.24%	41.92%	39.94%	42.05%	44.10%
APAC	39.42%	44.36%	38.21%	33.91%	42.24%
NPCA	47.05%	34.10%	**46.99%**	41.09%	46.03%
NLDA	28.91%	37.76%	27.05%	32.95%	29.87%
NNDA	14.36%	**13.97%**	14.10%	10.51%	16.15%

Appendix B

Decision Fusion Experiments

Table B.1: Decision fusion results on the M2VTS database without any feature normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	96.28%	96.96%	96.28%	93.58%	96.62%
PCA	88.85%	88.51%	88.85%	88.18%	88.85%
LDA	85.81%	86.15%	85.81%	85.47%	85.47%
APAC	86.15%	88.18%	86.82%	87.16%	86.82%
NPCA	88.85%	88.85%	89.19%	88.51%	88.85%
NLDA	89.19%	89.53%	89.19%	89.53%	89.19%
NNDA	89.19%	89.19%	89.19%	89.53%	89.19%

Table B.2: Decision fusion results on the M2VTS database with norm division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	95.95%	96.96%	97.30%	96.96%	96.96%
PCA	88.51%	88.85%	88.85%	89.19%	88.85%
LDA	85.47%	84.80%	84.80%	83.78%	85.47%
APAC	91.22%	90.20%	90.54%	90.54%	90.88%
NPCA	88.51%	89.19%	88.85%	89.19%	89.19%
NLDA	92.57%	90.88%	92.91%	92.57%	92.91%
NNDA	89.19%	89.19%	89.19%	89.19%	89.19%

Table B.3: Decision fusion results on the M2VTS database with sample variance division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	95.27%	95.27%	95.95%	95.95%	95.95%
PCA	91.55%	92.23%	92.23%	90.54%	91.89%
LDA	84.12%	84.12%	84.80%	84.46%	84.46%
APAC	85.47%	86.49%	85.47%	85.81%	85.47%
NPCA	91.22%	91.89%	91.55%	90.54%	91.89%
NLDA	89.19%	90.20%	89.19%	89.53%	89.53%
NNDA	90.20%	89.19%	89.53%	89.53%	90.20%

Table B.4: Decision fusion results on the M2VTS database with block mean and variance normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	91.22%	93.24%	92.23%	91.89%	91.89%
PCA	91.55%	91.89%	90.88%	91.22%	90.88%
LDA	81.08%	81.76%	81.42%	81.76%	82.09%
APAC	86.49%	88.18%	87.16%	87.84%	87.16%
NPCA	92.57%	92.23%	90.54%	90.20%	90.54%
NLDA	89.19%	91.22%	89.53%	89.86%	89.53%
NNDA	96.28%	94.93%	93.92%	93.24%	95.27%

Table B.5: Decision fusion results on the M2VTS database with feature vector mean and variance normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	94.93%	95.95%	95.61%	95.61%	95.95%
PCA	91.55%	91.55%	91.55%	91.22%	91.55%
LDA	86.82%	86.49%	86.49%	85.81%	86.82%
APAC	88.51%	89.53%	89.19%	89.53%	88.85%
NPCA	91.55%	91.55%	91.55%	91.22%	91.55%
NLDA	91.89%	91.22%	91.55%	90.88%	91.89%
NNDA	91.55%	91.89%	91.89%	91.55%	91.89%

Table B.6: Decision fusion results on the M2VTS database without any feature normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	92.91%	94.26%	94.26%	94.26%	94.26%
PCA	90.54%	92.57%	91.55%	91.22%	91.55%
LDA	86.82%	90.20%	88.18%	90.20%	86.82%
APAC	87.50%	90.54%	88.51%	89.86%	88.18%
NPCA	91.22%	93.24%	91.55%	91.22%	91.55%
NLDA	87.84%	87.84%	88.85%	91.22%	88.85%
NNDA	93.92%	95.27%	94.93%	94.59%	94.93%

Table B.7: Decision fusion results on the M2VTS database with norm division on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	93.92%	95.27%	93.92%	94.59%	94.26%
PCA	90.88%	91.55%	91.22%	90.20%	91.89%
LDA	91.55%	93.24%	91.55%	91.55%	91.55%
APAC	90.20%	91.22%	90.88%	91.22%	90.88%
NPCA	91.22%	92.23%	91.22%	90.20%	92.23%
NLDA	90.54%	91.22%	91.22%	92.91%	90.20%
NNDA	92.91%	94.59%	94.26%	94.26%	94.93%

Table B.8: Decision fusion results on the M2VTS database with sample variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	93.92%	94.26%	96.28%	95.61%	94.59%
PCA	94.59%	92.57%	94.29%	93.24%	94.93%
LDA	86.15%	89.19%	88.85%	90.20%	87.50%
APAC	86.82%	90.88%	89.19%	88.15%	88.85%
NPCA	94.26%	92.91%	94.93%	92.91%	94.26%
NLDA	92.23%	90.54%	92.23%	92.93%	92.23%
NNDA	94.59%	96.96%	95.61%	95.61%	95.95%

Table B.9: Decision fusion results on the M2VTS database with block mean and variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	91.22%	93.24%	92.23%	91.89%	91.89%
PCA	90.20%	91.89%	91.89%	90.88%	91.55%
LDA	89.86%	91.55%	90.88%	91.55%	90.20%
APAC	87.16%	88.85%	87.84%	88.85%	87.50%
NPCA	90.88%	92.91%	92.57%	90.88%	92.57%
NLDA	87.84%	87.50%	87.84%	86.82%	87.84%
NNDA	93.58%	93.92%	94.59%	94.26%	94.59%

Table B.10: Decision fusion results on the M2VTS database with feature vector mean and variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	92.57%	93.24%	94.26%	92.57%	94.26%
PCA	91.22%	92.91%	91.89%	90.20%	91.55%
LDA	92.23%	93.92%	92.57%	93.24%	91.89%
APAC	89.19%	90.88%	89.53%	89.86%	89.53%
NPCA	91.55%	92.91%	91.89%	90.54%	91.89%
NLDA	87.84%	90.20%	91.22%	91.22%	90.54%
NNDA	92.91%	94.26%	94.26%	94.26%	94.93%

Table B.11: Decision fusion results on the AR database without any feature normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	74.58%	74.86%	76.74%	76.11%	76.11%
PCA	65.49%	65.90%	67.57%	65.63%	66.81%
LDA	55.35%	57.85%	64.24%	67.43%	61.18%
APAC	65.83%	66.60%	69.10%	68.96%	68.26%
NPCA	65.35%	65.97%	67.64%	65.90%	66.94%
NLDA	69.79%	70.28%	74.72%	77.01%	72.29%
NNDA	75.76%	76.60%	77.85%	78.82%	77.29%

Table B.12: Decision fusion results on the AR database with norm division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	75.90%	76.18%	77.57%	77.29%	76.94%
PCA	78.82%	79.58%	80.83%	81.25%	80.21%
LDA	66.32%	66.60%	69.79%	71.67%	68.61%
APAC	67.78%	70.21%	71.39%	70.90%	70.56%
NPCA	78.54%	79.86%	80.49%	81.04%	80.28%
NLDA	73.40%	76.74%	77.99%	79.86%	76.74%
NNDA	83.75%	83.75%	85.90%	85.97%	85.14%

Table B.13: Decision fusion results on the AR database with sample variance division on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	78.68%	79.10%	82.01%	82.29%	81.04%
PCA	82.36%	82.01%	83.75%	83.96%	83.54%
LDA	65.00%	66.39%	70.69%	72.22%	68.40%
APAC	66.04%	66.25%	69.10%	70.14%	68.26%
NPCA	82.36%	82.08%	83.40%	83.89%	83.68%
NLDA	75.07%	76.39%	79.17%	80.83%	77.64%
NNDA	78.96%	79.51%	80.97%	82.99%	79.72%

Table B.14: Decision fusion results on the AR database with block mean and variance normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	74.58%	74.86%	76.74%	76.11%	76.11%
PCA	77.64%	78.33%	79.93%	79.72%	79.03%
LDA	64.58%	64.65%	68.61%	69.72%	66.53%
APAC	64.24%	68.26%	69.65%	70.35%	67.92%
NPCA	77.43%	78.61%	79.38%	79.86%	78.89%
NLDA	72.57%	75.49%	77.22%	78.33%	75.90%
NNDA	83.33%	83.54%	85.86%	85.86%	84.58%

Table B.15: Decision fusion results on the AR database with feature vector mean and variance normalization on 16x16 blocks - without image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	74.03%	74.17%	76.39%	76.18%	75.56%
PCA	63.61%	64.58%	66.94%	67.36%	65.83%
LDA	56.60%	58.13%	63.68%	66.81%	61.39%
APAC	67.29%	68.13%	70.56%	71.39%	70.07%
NPCA	64.24%	64.58%	66.53%	67.01%	65.69%
NLDA	70.56%	71.74%	73.26%	74.72%	72.08%
NNDA	75.69%	75.14%	77.78%	78.19%	77.08%

Table B.16: Decision fusion results on the AR database without any feature normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	77.08%	78.13%	79.10%	77.01%	78.68%
PCA	72.85%	77.57%	76.94%	74.17%	76.25%
LDA	60.76%	64.86%	66.94%	69.58%	64.38%
APAC	62.15%	65.83%	68.26%	69.24%	66.11%
NPCA	72.99%	77.64%	76.94%	74.31%	76.46%
NLDA	63.06%	66.39%	70.63%	72.92%	67.92%
NNDA	77.92%	81.32%	80.90%	80.21%	80.07%

Table B.17: Decision fusion results on the AR database with norm division on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	75.90%	76.18%	77.57%	77.29%	76.94%
PCA	72.85%	76.25%	75.69%	73.06%	74.93%
LDA	65.00%	68.82%	68.75%	69.58%	67.43%
APAC	65.35%	69.24%	71.39%	72.01%	68.61%
NPCA	72.85%	76.18%	75.42%	73.40%	74.79%
NLDA	63.89%	68.54%	71.32%	75.00%	68.96%
NNDA	79.24%	81.81%	81.32%	81.39%	80.69%

Table B.18: Decision fusion results on the AR database with sample variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	82.71%	83.96%	84.65%	83.89%	83.82%
PCA	81.67%	84.58%	83.96%	82.78%	83.82%
LDA	62.08%	66.25%	67.57%	68.89%	65.83%
APAC	63.47%	66.67%	68.75%	69.72%	67.57%
NPCA	82.01%	84.79%	84.38%	82.99%	84.10%
NLDA	69.72%	72.99%	74.10%	75.97%	72.57%
NNDA	79.24%	82.92%	82.43%	82.01%	81.39%

Table B.19: Decision fusion results on the AR database with block mean and variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	74.44%	75.63%	75.63%	74.31%	76.39%
PCA	72.43%	76.04%	74.79%	72.08%	75.14%
LDA	63.40%	67.08%	67.50%	67.78%	65.76%
APAC	63.89%	67.99%	70.56%	71.94%	67.71%
NPCA	71.53%	75.49%	75.56%	73.47%	73.89%
NLDA	63.13%	67.36%	71.04%	73.96%	67.85%
NNDA	77.78%	81.18%	80.56%	80.56%	79.58%

Table B.20: Decision fusion results on the AR database with feature vector mean and variance normalization on 16x16 blocks - with image domain normalization

	EW	**SW**	**VAW**	**VAW^2**	**VAW$^{1/2}$**
DCT	76.74%	78.13%	79.86%	77.15%	78.82%
PCA	72.85%	77.22%	76.67%	74.65%	75.76%
LDA	62.29%	64.86%	66.81%	68.40%	64.86%
APAC	66.74%	70.07%	71.74%	73.40%	70.00%
NPCA	73.13%	77.22%	76.53%	73.68%	75.49%
NLDA	64.44%	66.88%	70.83%	73.54%	68.33%
NNDA	79.44%	82.01%	81.67%	81.32%	81.39%

Bibliography

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[2] Y. Koren and L. Carmel, "Visualization of labeled data using linear transformations," *IEEE Symposium on Information Visualization*, p. 16, 2003.

[3] J. Kittler, I. C. Society, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[4] H. Ekenel and R. Stiefelhagen, "Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization," in *Conference on Computer Vision and Pattern Recognition Workshop, 2006 (CVPRW '06)*, June 2006, pp. 34–34.

[5] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 137–143, 1997.

[6] P. Phillips, A. Martin, C. Wilson, and M. Przybocki, "An introduction evaluating biometric systems," *Computer*, vol. 33, no. 2, pp. 56–63, 2000.

[7] K. Messer, J. Matas, J. Kittler, J. Lttin, and G. Maitre, "XM2VTSDB: The Extended M2VTS Database," in *In Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.

[8] Business Wire, "Mexican government adopts faceit face recognition technology to eliminate duplicate voter registrations in upcoming presidential election," May 2000.

[9] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, December 2003.

[10] P. J. Phillips, R. M. Mccabe, and R. Chellappa, "Biometric image processing and recognition," in *European Signal Processing Conference.*, 1998.

[11] Bruner, J.S. and Tagiuri, R., *Person Perseption and Interpersonal Behavior.* Stanford University Press, 1954.

[12] M. D. Kelly, "Visual identification of people by computer," Ph.D. dissertation, Stanford, CA, USA, 1971.

[13] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, Jan 1990.

[14] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91.*, Jun 1991, pp. 586–591.

[15] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[16] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of Optical Society of America A*, vol. 14, pp. 1724–1733, 1997.

[17] P. Viola and M. Jones, "Robust real-time face detection," in *Eighth IEEE International Conference on Computer Vision, 2001*, vol. 2, 2001, pp. 747–747.

[18] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision*, Jan 1998, pp. 555–562.

[19] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[20] A. Lanitis, C. Taylor, and T. F. Cootes, "An automatic face identification system using flexible appearance models," *Image and Vision Computing*, vol. 13, pp. 393–401, 1995.

[21] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, Jun 2001.

[22] J. Alvarado, W. Pedrycz, M. Reformat, and K.-C. Kwak, "Deterioration of visual information in face classification using eigenfaces and fisherfaces," *Mach. Vision Appl.*, vol. 17, no. 1, pp. 68–82, 2006.

[23] C. Liu and H. Wechsler, "Independent component analysis of gabor features for face recognition," *IEEE Transactions on Neural Networks*, vol. 14, no. 4, pp. 919–928, July 2003.

[24] M. Bartlett, H. Lades, and T. Sejnowski, "Independent component representation for face recognition," in *SPIE Symposium on Electronic Imaging: Science and Technology*, 1998, pp. 528–539.

[25] B. Niu, Q. Yang, S. C. K. Shiu, and S. K. Pal, "Two-dimensional laplacianfaces method for face recognition," *Pattern Recognition*, vol. 41, no. 10, pp. 3237–3243, 2008.

[26] K. Hotta, "Robust face recognition under partial occlusion based on support vector machine with local gaussian summation kernel," *Image Vision Comput.*, vol. 26, no. 11, pp. 1490–1498, 2008.

[27] B. Li, C.-H. Zheng, and D.-S. Huang, "Locally linear discriminant embedding: An efficient method for face recognition," *Pattern Recognition*, vol. 41, no. 12, pp. 3813–3821, 2008.

[28] P. Mohanty, S. Sarkar, R. Kasturi, and P. Phillips, "Subspace approximation of face recognition algorithms: An empirical study," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 734–748, Dec. 2008.

[29] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Individual stable space: An approach to face recognition under uncontrolled conditions," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1354–1368, Aug. 2008.

[30] Y. Zhang, J. Tian, Z. He, and X. Yang, "MQI based face recognition under uneven illumination," *Advances in Biometrics*, pp. 290–298, 2007.

[31] R. Gottumukkal and V. K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recogn. Lett.*, vol. 25, no. 4, pp. 429–436, 2004.

[32] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 84–91.

[33] H. Ekenel and R. Stiefelhagen, "Local appearance-based face recognition using discrete cosine transform," in *13th European Signal Processing Conference (EUSIPCO 2005)*, September 2005.

[34] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *Int. J. Comput. Vision*, vol. 43, no. 3, pp. 167–188, 2001.

[35] I. Fodor, "A survey of dimension reduction techniques," Tech. Rep., 2002.

[36] M. Loog, R. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 762–766, 2001.

[37] X. Qiu and L. Wu, "Stepwise nearest neighbor discriminant analysis," in *International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh*, 2005, pp. 829–834.

[38] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: global versus component-based approach," in *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, vol. 2, 2001, pp. 688–694.

[39] R. P. W. Duin and D. M. J. Tax, "Classifier conditional posterior probabilities," in *SSPR '98/SPR '98: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*. London, UK: Springer-Verlag, 1998, pp. 611–619.

[40] P. Paclk, T. Landgrebe, D. M. J. Tax, and R. P. W. Duin, "On deriving the second-stage training set for trainable combiners." in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, N. C. Oza, R. Polikar, J. Kittler, and F. Roli, Eds., vol. 3541. Springer, 2005, pp. 136–146.

[41] A. Martinez and R. Benavente, "The AR face database," CVC, Tech. Rep., 1998.

[42] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper, "The CSU face identification evaluation system: Its purpose, features, and structure," in *ICVS*, 2003, pp. 304–313.

[43] U. Meier, R. Stiefelhagen, J. Yang, and A. Waibel, "Towards unrestricted lip reading," in *International Journal of Pattern Recognition and Artificial Intelligence*, 1999, pp. 571–585.

[44] H. Erdogan, "Evaluating open-set identification performance using closed-set data," Tech. Rep., in preparation.