Face Recognition Through Regret Minimization

by

Daniel Yule

B.Sc., University of Northern British Columbia, 2009

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
MATHEMATICAL, COMPUTER, AND PHYSICAL SCIENCES
(COMPUTER SCIENCE)

THE UNIVERSITY OF NORTHERN BRITISH COLUMBIA

May, 2011

Your file Votre référence
ISBN: 978-0-494-75163-3
Our file Notre référence
ISBN: 978-0-494-75163-3

Abstract

Face Recognition is an important problem for Artificial Intelligence Researchers, with applications to law enforcement, medicine and entertainment. Many different approaches to the problem have been suggested; most approaches can be categorized as being either Holistic or Local. Recently, local approaches have shown some gains. This thesis presents a system for embedding a holistic algorithm into a local framework.

The system proposed builds on the concept of Regional Voting, to create Weighted Regional Voting which divides the face images to be classified into regions, performs classification on each region, and finds the final classification through a weighted majority vote on the regions. Three different weighting schemes taken from the field of Regret Minimization are suggested, and their results compared. Weighted Regional Voting is shown to improve upon unweighted Regional Voting in every case, and to outperform or equal many modern face recognition algorithms.

Contents

List of Tables

List of Figures

Acknowledgement

First off, I would like to thank my friends for continuing to be so, even as I ignored them to work on this thesis. Or while I pretended to work on this thesis, and instead played video games.

I would like to thank my family, for all the encouraging (read: nagging) in this and everything else. If I have accomplished anything, it is only because my parents gave me the tools to succeed.

Naturally, I would like to thank my fiancé Darcy for reminding me that there are other things outside of this thesis, and also for putting up with a grumpy me.

Lastly, and most importantly, I would like to thank my supervisor Dr. Liang Chen, who I somehow managed to fool many years ago into thinking I knew what I was talking about. His continued support and effort on my behalf, even when I wasn't sure of myself has been absolutely amazing. I am absolutely convinced that I could not have a better supervisor, and my only regret about finishing this thesis is that it signals an end to my tenure as his graduate student. Dr. Chen, words cannot express how much I appreciate everything you've done.

Chapter 1

Introduction

1.1 Overview

Face Recognition has become a very important and popular field of Computer Science. It has wide ranging applications, from smart homes[69] to entertainment[56] to security[51] [42]. It is a problem that is easy to understand, and very difficult to solve. In 1966, a well known researcher by the name of Marvin Papert assigned an undergraduate student to work over the summer on solving the problem of computer vision, of which face recognition is a small part, thinking this student would be finished by the end of the summer. This student failed completely to accomplish much of anything, and a significant portion of Artificial Intelligence Researchers have spent over 40 years attempting to solve that summer problem[13]

Face recognition, in the broadest possible terms, is the process of assigning an identity to an image of a face. However, face recognition is divided into two categories of problems: verification and identification. Verification is a one to one problem, ie it attempts to determine if the image is of a particular person. Identification is a one to many problem, ie it attempts to find the identity of the person

in the picture. Although the system presented here is one for identification, I feel it could easily be extended to the verification problem as well.

In order to carry out this task, a working knowledge of digital image processing is required. An image is assumed to be composed of pixels. Each pixel corresponds to a point of colour in the image. An image is then an $h \times w$ matrix of pixel values, where h is referred to as the height of the image, and w is referred to as the width. Most face recognition systems in general, and the one outlined here in particular, assume that the image is grayscale. Thus, each pixel represents only the intensity of the gray, and can be represented by a single number. Thus, an image can be assumed to be composed of an $m \times n$ matrix of positive real valued numbers representing pixel intensities.

Visual approaches attempt to perform classification based entirely on the pixels that comprise the image. Other approaches use other biometric data for recognition, such as range data[2] [77] [43], thermal imaging[5], [84] [14] or 3D data[44], [71] [6].

1.2 Contribution

The major contribution of this thesis is a new system for face recognition that is based on adding weights to the already proven system of Regional Voting. Regional Voting has been shown to be very stable in the face of a noisy system, and the weighting algorithms proposed here improve on the already best in class results of Regional Voting, in some cases cutting the error rate in half.

The remainder of the thesis is structured as follows: Chapter 2 contains an overview of the field of face recognition and a survey of pertinent literature. Chapter 3 gives a detailed description of the proposed algorithm. Chapter 4 details the experiments performed to verify the performance of the system. Chapter 5 analyzes

the results of the experiments and Chapter 6 summarizes the concepts presented and suggests some future directions.

Chapter 2

Literature Survey

Most humans have little difficulty with face recognition. This ability is mostly developed by the age of one year [62]. Thus, we as humans assume it must be an easy problem to solve. A computer should simply replicate the process that humans go through to recognize a face.

However, psychological research has found strong indicators that face recognition is a specialized task [62]. Several case studies have been done on individuals who have lost the ability to recognize faces, but retain the ability to recognize all other classes of objects (for example, McNeil and Warrington[57] and Farah[27]). In fact, there is a marked decrease in the ability of humans to recognize faces when the faces are upside down[82], belonging to other species of primates[68] or even of members of other ethnic groups[58]. Although there is some evidence to support the idea that the latter two cases are caused by a lack of exposure to non members of species or ethnic group, McNeil and Warrington[57][28] showed that a subject who is unable to distinguish faces when right side up is better at distinguishing them when they are upside down, indicating that right side up face recognition is a specialized process.

Thus, it seems face recognition is more than a matter of matching objects that

are similar. Furthermore, given the lack of knowledge as to how precisely human face recognition works[62], it is very difficult to design an algorithm to replicate it. As the brain is made of hundreds of billions of neurons, it may also be that the mechanism that the brain uses is infeasible for computational use. So, modern face recognition systems attempt to find different approaches to performing the face recognition problem.

In order to create a system that is capable of recognizing human faces, several steps must be handled:

1. Capture the image of a face
2. Digitize the image
3. Locate the face
4. Normalize the location of the face
5. Pre-process the face data
6. Perform face recognition

Although all the steps are important, for the system proposed here, the first five steps are assumed to have already taken place. For an overview of each step, please see Gonzalez et al[33]. In particular, we assume that the images are grayscale, aligned based on pupil location, and cropped to a common size.

Even within the face recognition step, there are many problems to be dealt with. How can we deal with variation in lighting? How can we recognize that faces belong to the same person when they have different expressions? Can we be sure the system will still identify faces correctly when they are subject to normal cosmetic changes (facial hair change, glasses on or off), or when the face ages?

Generally speaking, the problem of Face Recognition has been approached in one of two ways: with holistic approaches or local approaches[91]. Holistic approaches use the entire image as input data. Local approaches attempt to identify salient features of the face and perform recognition based on those. Each has their own advantages and disadvantages.

2.1 Holistic Approaches

Holistic Approaches are so named because they incorporate the whole of the image data at once. In general, a holistic method treats the $h \times w$ matrix that represents the image as a vector of length hw. This vector can in turn be interpreted as a point in hw dimensional space.

Once the data is in vector form, any kind of pattern matching algorithm can be used, such as neural networks[59] or support vector machines[70]. Some approaches attempt to convert the image into a different domain, such as Gabor Wavelets [74]. However, the most common form of holistic algorithms used are based around dimension reduction.

A dimension reduction approach requires some training images for each face to be recognized. These are referred to as the gallery. Let $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \ldots \mathbf{g}_n\}$ be the gallery. Then our training data X can be viewed as a matrix containing n rows of hw vectors:

$$X = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \vdots \\ \mathbf{g}_n \end{bmatrix}$$

The first step for dimension reduction is to solve a generalized matrix problem

$$XP = B$$

where P is a linear projection function represented as a $hw \times k$ matrix, and B is a $n \times k$ matrix corresponding to the basis of a reduced k dimensional subspace of X. Each of the rows in B is assumed to correspond to the identity of one of the images in X.

If we have a new probe image y, we arrange it into a vector of length hw, and compute $y_p = yP$. Then, using some measure, the best match between y_p and the rows of B is found.

The idea is that an image contains redundant information. The process of projecting it into a lower dimensional space eliminates redundancy and highlights the important information carried by the image. This point will be illustrated through four examples of Holistic Matching that will be embedded in the proposed system: Principal Component Analysis, Fisher's Linear Discriminant, Spectral Regression and Locality Preserving Projections.

2.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique for determining a basis for a vector space that accounts for as much of the variance within that space as possible [3]. This approach was first applied by Sirovich and Kirby to the problem of image compression[75, 46]. Shortly thereafter, Turk and Pentland [80] reasoned that if the important information in a facial image could be compressed to a smaller size, perhaps that compressed data could be used to classify the face. They named their approach Eigenfaces, after the nature of the calculation that is required.

Let X be the training gallery, as defined above. Then the first step is to calculate the average face $\overline{\mathbf{g}} = \sum_{i=1}^{n} \mathbf{g}_i$ We then create a mean-centred version of X, labelled X_m, where

$$X_m = \begin{bmatrix} \mathbf{g}_1 - \overline{\mathbf{g}} \\ \mathbf{g}_2 - \overline{\mathbf{g}} \\ \mathbf{g}_3 - \overline{\mathbf{g}} \\ \vdots \\ \mathbf{g}_n - \overline{\mathbf{g}} \end{bmatrix}$$

We then calculate the covariance matrix C of X_m, which can be found to be

$$C = X_m' X_m \tag{2.1}$$

We are trying to maximize variance between data points. This can be formalized as attempting to find a P such that P optimizes

$$P = \arg\max_{P} \{ \| P'CP \| \}$$

This P can be found through eigenvectors, giving the approach its name. We find the eigenvectors ν_i and eigenvalues μ_i of the covariance matrix, which are the set of vectors and associated values which satisfy

$$C\nu_i = \mu_i \nu_i$$

However, this calculation is very large, and very time and memory consuming, resulting in the calculation of hw eigenvectors. Furthermore, only at most n eigenvalues will be non-trivial. Instead, we can calculate the eigenvectors α_i and

eigenvalues β_i associated with the matrix

$$D = X_m X_m' \tag{2.2}$$

of which there are only n. We arrange α_i in decreasing order according to β_i. We can recover the original eigenvectors ν_i by

$$\nu_i = \alpha_i \cdot g_i'$$

Using the notation introduced for the general case of dimension reduction, we let $k = n$ and set

$$P = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_n \end{bmatrix}$$

In order to classify a new face image $\mathbf{y}$, we simply find $\mathbf{y}_m = \mathbf{y} - \overline{\mathbf{g}}$ and then compute $\mathbf{y}_p = \mathbf{y}_m P$. We then find the identity in B most closely related to $\mathbf{y}_p$.

In their original paper, Pentland and Turk suggest treating the vectors as points in n dimensional space, and computing the simple Euclidean distance: $||\mathbf{y}_p - \nu_i||$. The classification is selected as the identity of the image that has the smallest Euclidean distance. This is the metric used in this thesis for this and other holistic classifiers. However, since the introduction of Eigenfaces, other techniques for selecting the closest identity have been suggested.

In 1997, Moghaddam and Pentland [60] suggested using the Principal Components, along with other information from the image space to create a probability distribution over the images in the gallery for each probe image. The classifica-

tion can be performed using a maximum likelihood estimator. At the time of its publication, this technique achieved the highest results on the FERET[64] dataset.

Li and Lu [50] propose that instead of simply finding the Euclidean distance between the probe image and each image in the gallery, a so called "feature line" should be drawn between the point in space corresponding to each pair of images belonging to the same person. Then, instead of classifying based on the distance between each image, the distance between the probe image and each feature line is used. Li and Lu reported that this approach performed 43.7% to 65.4% better than the standard Eigenface method on a custom dataset, and 81% better on the ORL[8] dataset.

It's clear from its long usage that PCA is a feasible approach to face classification. However, the principal components it finds only maximize the amount of variance between each image. At no point does PCA take into account the similarities and differences between faces belonging to the same person. This suggests that another approach, one which includes information about identity in the training process, might show an advantage over PCA.

2.1.2 Fisher's Linear Discriminant

The Fisherface technique was developed in 1997 by Belhumeur et al[9] to address the limitations of PCA analysis. Since the principal components only maximize variance between each face, regardless of identity, the criteria it uses for maximizing faces might be responding to environmental factors such as lighting change, differing expressions, eyewear or facial hair. The Fisherface approach finds components that separate the faces based on identity, thus ignoring the effects of environmental factors.

The Fisherface technique is based on Fisher's Linear Discriminant[29]. In Fisher's

original 1936 paper, he suggests a technique for differentiating between members of two classes based on maximizing the ratio between the differences of the class means and the standard deviations within species.

More formally, let X and X_c be defined as in the eigenface approach. However, instead of finding the covariance matrix, and maximizing based on that, we find two matrices, S_W and S_B, where S_W is called the within class scatter matrix and S_B is called the between class scatter matrix. We assume that we are classifying p separate identities of person. Thus, we can find the covariance of the images within each class. We call the matrix for each class c S_c, where c is the identity of some person to be classified. Using this we can calculate S_W as

$$S_W = \frac{1}{p}\sum_{c=1}^{p} S_c$$

The between class scatter matrix C_B can be found by

$$S_B = \frac{1}{p}\sum_{c=1}^{p}(\overline{\mu_c} - \overline{\mathbf{x}}) \cdot (\overline{\mu_c} - \overline{\mathbf{x}})'$$

where $\overline{\mu_c}$ is the mean for class c, and $\overline{\mathbf{x}}$ is the overall mean for all training data. So, we are looking for a projection P that maximizes the between class scatter, and minimizes the within class scatter. Formally, we want a matrix P such that

$$P = \arg\max_{P} \frac{|P'S_BP|}{|P'S_WP|} \tag{2.3}$$

It has been shown[9] that if we let

$$P = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{bmatrix}$$

where α_i corresponds to the ith generalized eigenvector solution of the equation

$$S_B \alpha_i' = \beta_i S_W \alpha_i'$$

for some eigenvalue β_i, then P maximizes equation (2.3).

This general approach for finding a projection works quite well, but has a problem when applied to face recognition. The matrix S_W is typically singular as the number of classes p is far far less than the number of dimensions hw. Thus, a projection P can be found that makes $|P' S_W P|$ exactly zero.

To get around this problem, Belhumeur et al[9] suggest first projecting the data X_m down to a smaller dimensional space using PCA, then reducing it further using Fisher's Linear Discriminant. So, we find our P as follows:

$$P = P_{pca} P_{fld} \tag{2.4}$$

$$P_{pca} = \arg\max_P \{P_{pca}' C P_{pca}\} \tag{2.5}$$

$$P_{fld} = \arg\max_P \frac{\left|P_{fld}' P_{pca}' S_B P_{pca} P_{fld}\right|}{\left|P_{fld}' P_{pca}' S_W P_{pca} P_{fld}\right|} \tag{2.6}$$

where C is found according to equation (2.1). The PCA projection can be configured to project into an $hw - p$ dimensional space, thus circumventing the possibility of

a singular S_W. The P that is found will be of rank at most $p-1$, so that the dimensionality of the projection will be $p-1$. Thus, if we have five classes, we will project into a four dimensional space; with twenty classes, we will have nineteen dimensions etc.

Once we have found our P, classification proceeds precisely as it does with eigenfaces: the original data X_m is projected by $B = X_C P$ and then compared with probe images $\mathbf{y}$ by projecting $\mathbf{y}$ according to yP.

The approach outlined above is the Fisherface technique in its simplest form, and will be the approach that is embedded into the proposed system. However, there are many variations.

Contemporaneous with Belhumeur et al, Etemad and Chellappa[26] proposed a technique for performing linear dimension analysis for face recognition, but instead of using the Euclidean distance between points in the feature space, they calculate a weight for each dimension in the feature space, and find the distance between a probe image and the gallery images as a weighted sum of the distance along each dimension. They also suggest that their approach can be combined with a wavelet transformation to perform classification based on multiple criteria. Their experiments were performed on an augmented ORL[8] dataset, and are quite impressive, getting a error rate of only 1%. However, no comparison is made to other approaches, and it is unclear exactly how many images are used for training for each individual, or whether the benefit was due to the pre-processing and augmentation applied to the dataset, or the algorithm itself.

Similar to the above is a more recent approach proposed by Yang et al[87] called "Fuzzy 2DFLD", which approaches the problem of face classification as a series of two class problems. For each identity, there are two possible classes: either the face belongs to that person or it does not. The Fuzzy 2DFLD approach uses Fisher's

linear discriminant for each classification problem. Then, the k nearest neighbours to each gallery image are found. The number of these neighbours that belong to the given class is used as a class membership weight. The class membership is calculated for each gallery image for each class. These weights are then used to modify the scatter matrix used for Fisherface, to project probe images as closely as possible to the class centre. This approach provides a $33\% - 50\%$ improvement over the standard Fisherface technique, when applied to Yale[85], ORL[8] or FERET[64] datasets.

In 2003, Bressan proposed a non-parametric extension of Fisherfaces[12], based on the observation that Fisher's linear discriminant makes an assumption of normality on the distribution of faces, which may not be valid. His experiments show that the non-parametric assumption does improve face recognition slightly, but not as much as it improves other related fields, such as letter recognition or gender classification. This may indicate that the assumption of normality within classes is not an invalid one.

Cai et al[16] suggest an approach that builds on Fisher's linear discriminant by including unlabelled face data to the pre-labelled training data required by Fisherface. The concept is based on Regularized Discriminant Analysis(RDA)[31]. In RDA, the discrimination optimization equation, equation (2.3), is modified to

$$P = \arg\max_{P} \frac{|P'S_BP|}{|P'S_WP + \alpha J(P)|} \tag{2.7}$$

where α is a scaling factor and $J(W)$ corresponds to the model error. Cai et al suggest using the $J(W)$ term to incorporate the structure of the unlabelled data. This approach yielded a significant benefit. Although they do not compare their approach to traditional LDA, it shows a large improvement for face recognition over

PCA when tested on the CMU PIE[81] dataset.

2.1.3 Spectral Regression

Fisherfaces is very effective at classifying multi-dimensional data, and has a high degree of success with face recognition. However, it suffers from at least two problems above those addressed in the various algorithms listed above. The first is that it requires the calculation of eigenvalues, not once but twice. This is a rather large computation, especially when performed on the entire covariance matrix, as is done for the PCA reduction step. The second problem is the PCA reduction step itself. Although not very much information is lost in the dimension reduction, some is. It may be valuable to try to retain that information in the discriminant analysis phase.

Spectral Regression Dimension Analysis (SRDA)[17, 18] is both a framework and an algorithm designed to reduce the computational complexity of not only LDA, but also many other dimension reduction techniques. SRDA builds on the framework of Graph Embedding[86] to present a more efficient LDA algorithm.

Yan et al[86] propose the generalized framework of Graph Embedding for solving dimension reduction problem. The framework can handle linear, kernel and tensor reductions, although only the linear approach is treated here.

The concept of graph embedding is to treat each image as a vertex in a graph. Thus, if there are n images, there are n vertices in the graph. The edges are represented by the $n \times n$ matrix of real numbers W, where the i, jth entry of W is the (possibly zero) edge weight between vertex i and vertex j. The purpose of the graph reduction model is to represent the vertices of the graph as a vector with dimension lower than hw. If B is an $n \times k$ matrix that is the reduction of X to a

lower dimensionality k, where

$$B = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}$$

then the optimal B is given by minimizing

$$\sum_{i,j} (b_i - b_j)^2 W_{ij} \tag{2.8}$$

That is, if vertex i and vertex j have a large weight, then b_i and b_j should be close together. It has been shown that these optimal bs can be found by finding solutions to the equation

$$\mathbf{b} = \arg\max_{\mathbf{b}} \frac{|\mathbf{b}'W\mathbf{b}|}{|\mathbf{b}'D\mathbf{b}|} \tag{2.9}$$

where D is a diagonal matrix whose entries are column sums (that is, $D_{ii} = \sum_j W_{ji}$) and that the solutions to this equation are the eigenvector solutions to

$$W\mathbf{b} = \lambda D\mathbf{b} \tag{2.10}$$

However, since the translation from X to B will be linear, we know that $PX = B$ Thus, we can reformulate the optimization given in equation (2.9) as

$$\mathbf{p} = \arg\max_{\mathbf{p}} \frac{|\mathbf{p}'X'WX\mathbf{p}|}{|\mathbf{p}'X'DX\mathbf{p}|} \tag{2.11}$$

and the associated eigenvector problem becomes

$$XWX'\mathbf{p} = \lambda XDX'\mathbf{p} \tag{2.12}$$

Cai et al[17] show that if

$$\mathbf{b} = X\mathbf{p} \tag{2.13}$$

is a solution to equation (2.10), then $\mathbf{p}$ will be a solution to the eigenvector problem in (2.12). Thus, if we can find B, we can find P, if we can determine the vectors $\mathbf{p}$ that are a solution to (2.13), we can avoid having to solve equation (2.12).

The advantage to avoiding equation (2.12) is that finding eigenvalues is a computationally expensive procedure, and furthermore, can't be done if XDX' is non-singular, which it is when the number of features in the image is larger than the number of images. So, solving equation (2.13) is more efficient, if the eigenvectors are easier to find. Note that D is always non-singular, so it will always be possible to perform this eigen decomposition.

This generalized framework can be made specific through the choice of W. Cai et al [17] provide examples of a W for several linear dimension reduction strategies, including LDA [9], Locality Preserving Projection[38] and Neighbourhood Preserving Embedding[37] In particular, W for LDA can be determined as

$$W_{ij} = \begin{cases} \frac{1}{n_c} & \text{if vertex } i \text{ and vertex } j \text{ both belong to the } c\text{th class} \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

where n_c is the number of samples from class c. From this, it is clear that $D = I$.

We can assume without loss of generality that the images in X are ordered according to identity. Then it is easy to see that W is the block diagonal matrix

given by

$$\begin{bmatrix} W^{(1)} & 0 & \cdots & 0 \\ 0 & W^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W^{(p)} \end{bmatrix}$$

where each $W^{(c)}$ is an $n_c \times n_c$ matrix. So, we can find the eigenvectors of $W\mathbf{b} = \lambda D\mathbf{b}$ by finding the union of the eigenvectors of each of the blocks, and padding them with zeros to get the appropriate length.

From examining equation (2.14) it can be seen that there will be an eigenvector $\mathbf{b}_i = [1, 1, 1, \ldots, 1]$ with associated eigenvalue 1. This is not a useful eigenvector, as the response of all data points is the same. So we take it along with the other $p-1$ eigenvectors associated with non-zero eigenvalues, and apply the Gram Schmidt algorithm. Then, we can remove the all ones eigenvector from the orthogonal basis, leaving us with a $p-1$ dimensional basis, similar to the LDA approach outlined in section 2.1.2.

If we let B be the basis found through the application of the Gram Schmidt process, then we can attempt to find a matrix P such that

$$B = XP$$

In particular, for each $\mathbf{b}_i$ in B, we must find a p_i such that

$$\mathbf{b}_i = X \cdot \mathbf{p}_i$$

However, such a p_i may not exist. So we can approximate $\mathbf{p}_i$ by finding the $\mathbf{p}$

that minimizes

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{j=1}^{n} (\mathbf{x}_j \mathbf{p} - \mathbf{b}_{ij}) \tag{2.15}$$

where $\mathbf{b}_{ij}$ is the jth element of vector $\mathbf{b}_i$ and x_j is the jth column of X. Equation (2.15) can be minimized by a least squares approximation technique. There are many such algorithms that can handle large scale least squares problems very efficiently, such as LSQR [67].

As the number of faces n is typically far smaller than the original dimensionality of the images hw, the minimization problem in equation (2.15) is ill posed. Thus, there are infinitely many solutions to this equation. In fact, if we introduce a regularizing condition, then we can find only the projective functions that would have been found via the original dimension reduction problem, before being analysed through graph spectral analysis. The regularized least squares optimization problem is given by

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{j=1}^{n} (\mathbf{x}_j \mathbf{p} - \mathbf{b}_{ij} + \alpha \|\mathbf{p}\|^2) \tag{2.16}$$

where α is a shrinkage parameter.

The advantage of this approach is that, while it is more complex and less direct, the number of computations performed is drastically reduced. Furthermore, there is no need to perform a reduction on the face image data prior to performing LDA analysis. Experimental results[17] have shown that performing spectral regression dimension analysis performs roughly on par with Regularized Discriminant Analysis [16], which outperforms LDA. However, Spectral Regression takes approximately 1/20th the time of RDA, implying that much more complex methodologies could be embedded in this framework, and even better results could be achieved, without too much of a time penalty.

2.1.4 Locality Preserving Projections

Another approach that uses the graph embedding model is Locality Preserving Projections[38] (LPP). Building on the framework outline above, the LPP graph is constructed by placing an edge between two vertices if they are "close." He et al[39] suggest two methods for determining "closeness:" k-nearest neighbours and ϵ distance. That is, W can be defined by either

$$W_{ij} = \begin{cases} \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\eta}\right) & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

or

$$W_{ij} = \begin{cases} \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\eta}\right) & \text{if } j \text{ is among the } k \text{ nearest neighbours of } i \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

where $\eta \in \mathbb{R}$ is a tuning parameter.

The goal is to minimize equation (2.8), which is

$$\sum_{i,j} (b_i - b_j)^2 W_{ij}$$

It can be shown that this equation is equivalent to

$$P'XLX'P \tag{2.19}$$

under the assumption that $B = PX$, and L is the so called "Laplacian Matrix" of W. L can be found by $L = D - W$ where D is a diagonal matrix of column sums of W (see section 2.1.3).

Similar to the Spectral Regression approach, equation (2.19) can be minimized by solving the generalized eigenvector problem

$$XLX'\mathbf{p} = \lambda XDX'\mathbf{p} \tag{2.20}$$

So, the projection matrix P is simply the collection of all eigenvectors $\mathbf{p}$ that satisfy equation (2.20).

However, like LDA, because the number of pixels is so much higher than the number of face images for training, XLX' is usually singular, which means that equation (2.20) cannot be solved. To get around this problem, as in LDA, principal component analysis is first used to reduce the number of dimensions, before attempting to solve the eigen problem.

Locality Preserving Projections have been shown to be extremely effective at solving the face recognition problem. He et al [39] show a dramatic improvement over PCA and LDA for face recognition on the CMU PIE[81], Yale[85] and MSRA datasets.

Since the publication of the Locality Preserving Projection paper, there have been numerous investigations into improving and applying LPP in various ways. In 2006, Cai at al[19] built on the idea of Locality Preserving Projection with Orthogonal Locality Preserving Projections (OLPP). OLPP employs a similar algorithm to standard LPP, constructing the same graph, minimizing the same function, and solving the same eigenproblem. However, once the eigenvectors have been found, the eigenvector corresponding to the smallest eigenvalue is used, and then the algorithm modifies the eigenproblem and re-solves it, finding another set of eigenvectors which are orthogonal to the first. Again, the eigenvector corresponding to the smallest eigenvalue is selected, and the process is repeated k times, yielding k eigenvectors.

The OLPP algorithm shows a performance increase over standard LPP, in the Yale [85], ORL [8] and CMU PIE [81] datasets. In some cases, OLPP cut the error rate for LPP in half. However, OLPP, like many dimension reduction techniques, is sensitive to changes in dimensionality. Unlike Fisherface, there is no way to determine what dimensionality should be used.

A separate technique for improving LPP suggested independently by Whittier and Qi [83] and by Zheng et al [92] is Supervised Locality Preserving Projection (SLPP). The standard approach to LPP does not take class labels into account. However, the graph construction step can easily be modified to take this information into account. Instead of placing an edge between two vertices if they are close by some metric, an edge is placed between two vertices if they are in the same class, and optionally also if they are close.

Zheng et al demonstrate SLPP showing a significant improvement over standard LPP [92]. However, the testing methodology described uses a custom face detection system, as well as gabor wavelets for classification, so direct comparisons with previous results cannot be made. Nevertheless, the results demonstrate the effectiveness of using class labels in the training step.

Yu Teng and Liu [88] suggest a similar idea called Discriminant Locality Preserving Projection (DLPP), but take the idea of including class labels a step further. They suggest modifying the objective function for LPP outlined in equation (2.8) to one inspired by a Fisher type approach, namely:

$$\frac{\sum_{c=1}^{p}\sum_{i,j=1}^{n_c}(\mathbf{b}_i-\mathbf{b}_j)W_I^c}{\sum_{i,j=1}^{p}(\mathbf{m}_i-\mathbf{m}_j)W_E} \tag{2.21}$$

where p is the number of individuals in the database, n_c is the number of images

belonging to individual c, $\mathbf{m}_i$ is the mean face of individual i and W_I^c and W_E are within class and between class weightings respectively.

This objective function can be solved by an eigenvector equation, and from there the algorithm proceeds as standard LPP.

Although this algorithm is shown to have improvements over LPP, the improvements are not as dramatic as those demonstrated by either SLPP or OLPP. However, it is impossible to know if this is because it is a weaker algorithm, or simply a result of different testing environments. It seems that DLPP should offer some improvement over SLPP, as DLPP not only minimizes within class scatter, like SLPP, but also maximizes between class scatter, but the literature is not clear on this point.

2.2 Local Approaches

Unlike the holistic approaches described above, local approaches do not treat the entire image as a pattern to be classified. Instead, they break the image up into smaller pieces, and perform classification based on these. There are a wide variety of local approaches, which differ not only in how the face is broken up, but how the pieces are compared, and how the overall classification is determined from the classification of the features.

Hidden Markov Model (HMM) based approaches use some technique to extract information about the forehead, eyes, nose mouth and chin, and then train a Hidden Markov Model [40] to recognize each individual. The HMM approach was first suggested by Samaria and Young in 1994 [72]. Their approach extracts strips of pixels corresponding to the important features, and trains the model based on this. Hidden Markov Models are designed for one dimensional data; to extend it to a two dimensional case is NP-hard. Thus, the strips are presented as one dimensional

data, which loses the vertical spatial relationships.

Nefian and Hayes [61] improved upon the speed of Samaria and Young's results by using a Discrete Cosine Transform on the strips of pixels corresponding to the original data. They achieve a slight improvement on the ORL [8] dataset, but see a more than 10x speed up of recognition rate.

Later researchers have employed further variations on the concepts outlined here. Hu and Liu [41] suggest a Hidden Markov Model based around the Fast Fourier Transform and the Partial Least Squares approach. Bicego et al [10] suggest using Haar Wavelets[25] instead of DCT, and achieve similar results, but are able to improve on these results through a clever model selection process. In fact, based on their experiments, HMM + Haar Wavelets achieves 100% recognition accuracy on the ORL dataset.

A popular method for finding features in a face is based on Scale Invariant Feature Transform (SIFT)[53]. SIFT, as its name implies, generates a large collection of features that are invariant to scale, rotation or translation. These features are extracted from each image, and compared. Images that have a large number of matching features are considered to belong to the same class.

SIFT was originally suggested for general object recognition, but has since been applied to Face Recognition by several researchers, first by Mohamed Aly in a term paper[7]. Aly directly follows the approach laid out for general object recognition, and achieves better accuracy than Fisher or Eigenfaces.

Bicego et al [11] and Luo et al[54] suggest grid and clustering approaches respectively. In essence, they suggest finding the features of local regions of the image, and classifying based on those. This not only improves the accuracy, but decreases the amount of computational power required to perform the calculation. Majumdar and Ward [55] suggest an approach inspired by Fisher's Linear Discriminant,

wherein features that have a high class discriminative power are selected. Geng and Jiang [32] outline two algorithms that modify the features themselves for improved accuracy.

Another common approach uses textures to perform some form of matching[76] [52] . One of the most well known approaches to using textures for matching is Local Binary Pattern analysis.

2.2.1 Local Binary Pattern

Local Binary Patterns (LPB) were first introduced by Ojala et al [66] in 1996, building on a previous idea proposed by Wang and He [36]. In its most basic form, LPB works as follows:

A 3×3 window is created around each pixel (except the ones on the outside edge of the image). A pattern is generated starting with the top leftmost pixel, and proceeding clockwise around the centre pixel. If the current pixel is greater than the centre pixel, then the pattern is a 1, if less, then the pattern is a 0. Thus, there are eight binary digits (one for each non-centre pixel in the window), which when concatenated together form an 8-bit integer. This 8-bit integer then represents the pattern surrounding the central pixel.

Although this approach shows some positive results for pattern classification, Ojala et al[65] propose some extensions to their original suggestion, to ensure that their classification is rotation and grayscale independent. Their suggestion is two-fold. First, instead of using a rectangular 3×3 window, use a circular window. The points that do not lie in the centre of a pixel can be estimated using interpolation. Then, since the pattern should be invariant to the selection of the first pixel in the window, all patterns that are the same except for rotation are classified as the same.

This leaves 36 distinct patterns. The second idea proposed in [65] comes out of

the observation that one type of pattern makes up about 90% of patterns observed in the data. This type of pattern, which they call a uniform pattern, is one in which there are two or fewer transitions from 0's to 1's or 1's to 0's. For example the patterns 00000000 and 00111000 are uniform, whereas the pattern 01100100 is not. Ojala et al suggest classifying all non-uniform patterns in the same class, thus using only uniform patterns for classification. Their justification is that the remaining 27 patterns don't appear often enough to learn anything useful about their probability.

Once the patterns have been found, they are used to construct a histogram, which can then be compared to perform matching. However, Ahonen et al [4] suggest a further improvement for face recognition. LBP as it is provides excellent information about micro-structures, but carries no information at all about the spatial relationship of the data. The suggestion is to break the image up into regions, find a histogram for each region, and then concatenate the histograms from each region together in order to perform global matching.

There several ways to compare histograms, including Histogram Intersection:

$$D(S, M) = \sum_{i,r} \min\{S_{i,r}, M_{i,r}\} \tag{2.22}$$

Log-likelihood:

$$L(S, M) = \sum_{i,r} S_{i,r} \log M_{i,r} \tag{2.23}$$

and Chi squared statistic (χ^2)

$$\chi^2(S, M) = \sum_{i,r} \frac{(S_{i,r} - M_{i,r})^2}{S_{i,r} + M_{i,r}} \tag{2.24}$$

where S and M are the histograms to be compared and $S_{i,r}$ is the frequency of pattern i in region r in histogram S. In their experiments, Ahonen et al found

that using a χ^2 statistic worked better than the other approaches, so that is the comparison that they used.

The approach outlined above was tested on the FERET[64] dataset, and achieved a high accuracy rate compared to the Eigenfaces technique, in some cases halving the error rate.

Like other approaches mentioned in this chapter, various improvements over the original strategy have been suggested. Zhang et al [89] convert the face classification problem into a two class problem by creating a classifier for each individual in the database, and classifying each image as either being of that individual or not. Then the Adaboost[30] algorithm learns the similarity between every face pair. This approach shows a competitive result on the FERET fa/fb partition.

Zhang et al [90] suggest combining Local Binary Batterns with Gabor Wavelets. In their approach, each region is convolved with its Gabor filters, to generate so called Gabor Magnitude Pictures (40 for each region). These Gabor Magnitude Pictures are then subjected to Local Binary Pattern analysis, and classification proceeds as normal. The inclusion of Gabor filters dramatically improves on the performance of Local Binary Patterns when tested on the FERET[64] dataset.

2.2.2 Volterrafaces

Volterrafaces is a very new technique, suggested by Kumar et al in 2009[48]. Although it could be used holistically, the authors apply it as a local approach, so that is what is followed here. Volterrafaces is based around the idea of the Volterra Series.

Volterra series can completely describe any non-linear translation invariant func-

tional $\aleph : H \to H$ which maps the function $x(t)$ to $y(t)$ with

$$\Im(x(t)) = y(t) = \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \ldots, \tau_n) x(t\tau_1) \ldots x(t\tau_n) d\tau_1 \cdots d\tau_n \tag{2.25}$$

where $h_n(\tau_1, \ldots, \tau_n)$ is referred as the Volterra Kernel of the functional.

Since digital images are only discrete approximations of continuous functions, we can use the discrete form of the equation:

$$\Im(x(t)) = y(t) = \sum_{n=1}^{\infty} \left(\sum_{q_1=-\infty}^{\infty} \cdots \sum_{q_n=-\infty}^{\infty} h_n(q_1, \ldots, q_n) x(tq_1) \ldots x(tq_n) \right) \tag{2.26}$$

However, if we want to perform any computations, we must approximate these infinite sums with

$$\Im^p(x(t)) = x(t) \otimes_p h(t) \tag{2.27}$$

where p denotes the number of terms being used in the approximation, $\otimes_p$ represents the convolution operator, and $h(t)$ stands in for all the different orders of kernels.

Under this approach, the goal is to approximate a functional $\aleph$ that can map images of faces to identities with a low error. We can evaluate the *goodness* of $\aleph$ at this task with a so called goodness functional defined as:

$$O(\Im^p) = \frac{\sum_{c_k \in c} \sum_{i,j \in c_k} \|\Im^p(\mathbf{x}_i) - \Im^p(\mathbf{x}_j)\|^2}{\sum_{c_k \in c} \sum_{m \in c_k, n \notin c_k} \|\Im^p(\mathbf{x}_m) - \Im^p(\mathbf{x}_n)\|^2} \tag{2.28}$$

where c is the set of all identities. Using equation (2.27) we can rewrite the above as

$$O(\Im^p) = \frac{\sum_{c_k \in c} \sum_{i,j \in c_k} \|\mathbf{x}_i \otimes_p K - \mathbf{x}_j \otimes_p K\|^2}{\sum_{c_k \in c} \sum_{m \in c_k, n \notin c_k} \|\mathbf{x}_m \otimes_p K - \mathbf{x}_n \otimes_p K\|^2} \tag{2.29}$$

where K is the Volterra Kernel that will be described shortly.

The convolution of the image and the kernel is easiest done if it is linear. Thus, we

transform X first, and then perform a linear product with the kernel to approximate $\Im^p$. The image $\mathbf{x}_i$ is represented in two dimensions. Then the transformed version of X, labelled A, is dependent on the order of the approximation. The Volterraface paper outlines first and second order approximations, and provides a framework for higher order approximations (although they are not usually necessary).

For a first order approximation, X is transformed into the matrix A by finding a $b \times b$ neighbourhood of each pixel in X, vectorizing each neighbourhood, and stacking the results to form a new matrix A. For a second order approximation, not only is each pixel in the neighbourhood converted into a vector, but each neighbourhood's vector includes the product of each pixel in the neighbourhood with every other pixel.

Now that we have transformed the input data, we can replace

$$\mathbf{x}_i \otimes_p K = A_i \cdot K$$

and so our objective function from equation (2.29) becomes

$$O(\Im^p) = \frac{\sum_{c_k \in c} \sum_{i,j \in c_k} \|A_i \cdot K - A_j \cdot K\|^2}{\sum_{c_k \in c} \sum_{m \in c_k, n \notin c_k} \|A_m \cdot K - A_n \cdot K\|^2} \tag{2.30}$$

or

$$O(\Im^p) = \frac{K' S_W K}{K' S_B K} \tag{2.31}$$

where

$$\begin{aligned} S_W &= \sum_{c_k \in c} \sum_{i,j \in c_k} (A_i - A_j)'(A_i - A_j) \\ S_B &= \sum_{c_k \in c} \sum_{m \in c_k, n \notin c_k} (A_m - A_n)'(A_m - A_n) \end{aligned}$$

It can be shown that equation (2.30) can be minimized by finding the eigenvector corresponding to the smallest eigenvalue of $S_B^{-1}S_W$. This eigenvector is taken to be K.

In order to classify an image, first the kernel is found on the training set. Any new images to be classified are convolved with the kernel and then compared to the convolved gallery. The image is then classified by nearest neighbour.

The authors of the Volterraface paper suggest that better results can be found by dividing the image into regions and performing Volterrafaces on each region separately. The overall class can be found by a vote over the all the regions. Through cross validation, the authors found that 8×8 patches work best. The authors also tried overlapping vs non-overlapping regions, as well as different sizes for the kernel.

Only the best results for all combinations of parameters in a dataset are reported. On the Yale [85], CMU PIE [81] and Extended Yale datasets, first order approximations were shown to outperform second order approximations, but both outperformed all other face classification systems tested (including PCA, Fisherface and LPP).

As of yet, there has been no work to extend the ideas presented in this paper, but such a promising algorithm should be investigated more fully in the near future.

2.3 Regional Voting

Unlike all of the above algorithms, the Regional Voting framework suggested by Chen and Tokuda [21] [22] [23] is not a specific face classification technique. Instead it is a framework for embedding any holistic matching technique.

In the Regional Voting framework, the image is broken up into non-overlapping regions. Each region is classified separately by the holistic classification scheme

being used. The overall classification is found by a simple majority over all the regions.

The idea to break an image into regions for facial recognition has been used many times by many researchers. As mentioned above, both Local Binary Patterns [4] and Volterrafaces[48] suggest breaking the image into regions to maintain spatial information. Breaking the image into regions has also been suggested for Eigenfaces[24, 78, 34], SIFT [11] as well as Gabor Wavelets[93][35], and Discrete Cosine Transform[1].

A major difference between the above systems and the Regional Voting framework suggested by Chen and Tokuda, aside from one being a framework. and the others being specific methodologies, is the method that is used to combine the regions together. Although Kumar et al in 2009[48] and Zhou et al in 2007 [93] both advocate a majority vote, all other systems simply concatenate the results from each region together and perform final classification based on that. In 1999, when Regional Voting was first suggested, local based approaches were just gaining notice, and there was no previous work on combining regions together with a simple vote[79].

More formally, the Regional Voting approach assumes a gallery of images $\mathcal{G} = \{g_1, g_2, \ldots g_n\}$ with associated identities $\mathcal{I} = \{i_1, i_2, \ldots i_n\}$, and a holistic face recognition system $H : \mathcal{P} \rightarrow \mathcal{I}$, where $\mathcal{P}$ is the set of all possible probe images of faces.

Each g_i in the gallery is divided k times vertically and ℓ times horizontally to create $k \times \ell$ non-overlapping regions. For a region r, we denote the region in image g_i by g_i^r and the region in every image in the gallery by $\mathcal{G}^r$. The holistic algorithm $\mathcal{H}$ is trained using $\mathcal{G}^r$ for each region r. We call this regionally trained algorithm $\mathcal{H}^r$.

To classify a probe image $p \in \mathcal{P}$, each region is classified separately. That is, for

Figure 2.1: Regional Voting Algorithm

for each image $g_i \in \mathcal{G}$ **do**
  Divide g_i k times horizontally and ℓ times vertically
end for
Let $\mathcal{R}$ be the set of regions that each image is broken into.
for each region $r \in \mathcal{R}$ **do**
  Train H on $\mathcal{G}^r$
end for
Classify the image by $\arg\max_{i \in \mathcal{I}} \left\{ \sum_r eq(H^r(p) = i) \right\}$

each region r, the result of $H(p^r)$ is found. The overall winner is taken to be the identity that has the most regional classifications. The algorithm is given in Figure 2.1.

This approach is very straightforward, but it yields very powerful results. Chen and Tokuda[23] show that their approach improves the face recognition accuracy of any holistic approach that is embedded into it on the Yale [85], ORL [8] and CMU PIE[81] datasets, typically halving the error rate. Their experiments show that a region size of between 4×4 and 5×5 usually yields the best results.

The reason suggested by Chen and Tokuda for the improvement is the increase in stability. They define stability as invariance to noise, and use two types of noise: uniform and concentrated. They have shown for both binary images [21] and grayscale using the Hamiltonian distance[22] that Regional Voting increases stability to both types of noise. The reason posited for the increase in stability is that Regional Voting is able to contain noise contamination to the regions affected, so that it takes widespread noise to change the classification of the system.

Chapter 3

Proposed Algorithm

The proposed algorithm builds on the success of Regional Voting by applying weights to each region. The idea is motivated by the observation that there should be some regions of the face that are more important than others. This concept has been exploited as early as the 1970's in some of the earliest research into face recognition[45]. More recently, Adaptively Weighted Sub-Pattern PCA [78] as well as weighted local Gabor wavelets[93] have been suggested. The system at hand is distinct from these approaches in that the weights are estimated independent of any human knowledge of the structure of the face or of the underlying algorithm.

To modify the Regional Voting Algorithm into a Weighted Regional Voting (WRV) algorithm, we assume that each region r has an associated weight $w^r \in [0, 1]$. Then the final classification step in Figure 2.1 becomes

$$\arg\max_{i \in \mathcal{I}} \left\{ \sum_r w^r eq(H_r = i) \right\}$$

Although it does not deal specifically with face recognition, the paper "Combining Classifiers: A Theoretical Framework" by Kittler et al[47] provides a strong theoretical foundation for why a weighted vote provides the best estimation of the

overall classification of several sub-classifiers. Under the assumption that the data provided to each classifier is independent (non-overlapping blocks) and each classifier selects one class for the input, then a weighted vote is the maximum likelihood estimator for the output of the combination of the classifier.

In their paper, Kittler et al suggest doing a search over the entire state space to find the best weights for each classifier. Although this is possible, for the case of face recognition, if the image is divided 15 times horizontally and 15 times vertically, there are 255 weights to be estimated, and no suggestion as to how to find these weights.

The main contribution of this thesis is the suggestion of several techniques for weight estimation. These weight estimation techniques are simple, and computationally feasible to implement.

A very simple approach to estimating the weights is to simply make them proportional to the accuracy of the system. That is, if $right(r)$ is the number of training images H is able to classify in region r then we can simply use

$$w_r = right(r)/n \tag{3.1}$$

where n is the total number of images in the gallery. However, there may be more sophisticated approaches to weight estimation. The concepts used are taken from Regret Minimization, so we take a brief detour before describing the weight estimation techniques.

3.1 Regret Minimization

Regret minimization is a technique borrowed from algorithmic game theory for making decisions in the face of uncertainty. Formally, regret minimization works as

Figure 3.1: Polynomial Weights Algorithm

Require: $w_1^a = 1$ and $p_1^a = 1/|A|$ for $a \in A$
  for each time t **do**
    for each action $a \in A$ **do**
      Let $w_t^a = w_{t-1}^a \cdot (1 - \eta L_{t-1}(a))$
      Let $p_t^a = w_t^a / W_t$ where $W_t = \sum_{a \in A} w_t^a$
    end for
  end for

follows:

Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of allowable actions. At each discrete time t, we assume there exists a function $L_t : A \to [0, 1]$, called the loss function, which assigns a loss for each action that we can take. Our goal then is to select actions to minimize the amount of loss we suffer.

There is a significant amount of theory surrounding this idea (see [20], [63] for example). Here we distill only what we need. It is fairly plain to see that selecting only one action all the time may not be the best approach. For example, when playing rock paper sissors, it is not a good strategy to always pick rock. But it is a good strategy to randomly choose between rock, paper and sissors. Thus, what we are looking for is a distribution over the actions, $p : [0, 1] \to A$ such that $\sum_{a \in A} p(a) = 1$

It can be shown the algorithms in Figures 3.1 and 3.2 minimize the expected loss, if the action at time t is selected according to distribution p^t. So, p_t^a is the amount of weight distribution p places on action a at time t. w_t^a is similar, but isn't normalized. The parameter η is a tuning paramter typically between 0 and 0.1. See [20] for a derivation of this proof and how to find η

To apply the above approach to face recognition requires a few changes and assumptions. Firstly, we don't have actions take take. Instead, we have regions. Each 'action' corresponds to selecting a particular region as the classifier. Secondly,

Figure 3.2: Exponential Weights Algorithm

Require: $w_1^a = 1$ and $p_1^a = 1/|A|$ for $a \in A$
  for each time t **do**
    for each action $a \in A$ **do**
      Let $w_t^a = w_{t-1}^a \mathrm{e}^{-\eta L_{t-1}(a)}$
      Let $p_t^a = w_t^a / W_t$ where $W_t = \sum_{a \in A} w_t^a$
    end for
  end for

there is no concept of time. Each t corresponds to a training image. Thirdly, the loss function is either 0 or 1: 0 if the classification is correct, 1 if it is not. Lastly, and most significantly, instead of selecting the classification based solely on one classifier, we will use p as a weight on each region and sum the weighted votes for each classifier to select our overall classification.

Essentially, I am adopting the weighting for actions from regret minimization, and applying them to Regional Voting. This means that I've lost the theoretical guarantees of minimization, but have gained the stability of Regional Voting. I will show empirically that this trade-off is worthwhile.

3.2 Estimating Regional Weights

As outlined in Section 3.1, I make several changes to the ideas of Decision Theory in order to apply them to Regional Voting. I show the adaptation of the Exponential Weights algorithm, as the derivation for Polynomial is similar.

Making the changes outlined in Section 3.1 to the algorithm given in Figure 3.2 yields the algorithm given in Figure 3.3, we see that the function $L_r : \mathcal{I} \to \{0, 1\}$ is defined as $L_r(g_i) = 0$ if region r of image g_i is classified as the identity of image g_i, and $L_r(g_i) = 1$ otherwise.

Examining the result of p_r after updating for every image, we see that w_r is only

Figure 3.3: Exponential Weights Algorithm

Require: $w_1^r = 1$ and $p_1^r = 1/|A|$ for $a \in A$
  for each gallery image $g_i \in \mathcal{G}$ **do**
    for each region $r \in \mathcal{R}$ **do**
      Let $w_{g_i}^r = w_{g_j}^r \mathrm{e}^{-\eta L_r(g_i)}$
      Let $p_{g_i}^r = w_{g_i}^r / W_{g_i}$ where $W_{g_i} = \sum_{g_i \in \mathcal{G}} w_{g_i}^r$
    end for
  end for

updated if the loss function is 1, or if the region is incorrectly identified. In that case, w_r is multiplied by $\mathrm{e}^{-\eta}$. So, if we count the number of times the weight for each region is updated, we can reduce the weight update process to

$$w_r = \mathrm{e}^{-\eta \cdot wrong(r)} \tag{3.2}$$

where $wrong(r)$ is the number of times that region r incorrectly classifies an image in the training set. We can derive the formulation of polynomial weighting similarly to arrive at the equation

$$w_r = (1 - \eta)^{wrong(r)} \tag{3.3}$$

I further note that since I am no longer using weights as a probability distribution, I can drop the requirement that $\sum_{r \in \mathcal{R}} p(r) = 1$, and instead set $p(r) = w_r$. There are now three possible weighting approaches, given in equations (3.1), (3.2) and (3.3).

Although I have used the notations $wrong(r)$ and $right(r)$, I have given no indication as to how to calculate these. Since we are attempting to fit a model to our output, the standard approach is cross validation. In particular, I use leave one out cross validation to estimate the weights, as well as finding the optimal parameter η. The process is as follows:

For each individual in the database, select one image as the probe image. Train

the holistic classifier in each region using the remaining images. Then attempt to classify each of the probe images and store the result. Repeat the process by selecting a distinct set of probe images from the gallery. Continue repeating the process until all of the training images have been used exactly once. Then check the accuracy of each region at classifying the probe images. This is used as input for $wrong(r)$ and $right(r)$.

Cross validation has the added bonus of allowing the system to estimate an optimal value for η. Once the estimated classifications for each image in each region is known, the weights can be estimated using different values of η. The η that results in the best classification rate for the training data is the η that is selected, and the corresponding weights used for the final classification.

Any of the three weighting equations introduced can be used as part of the algorithm. I will call equation (3.1) Direct Proportional weighting, equation (3.2) Exponential Weighting and equation (3.3) Polynomial Weighting. Thus, the final overall algorithm can be seen in Figure 3.4.

Figure 3.4: Weighted Regional Voting Algorithm

for each image $g_i \in \mathcal{G}$ **do**
  Divide g_i k times horizontally and ℓ times vertically
end for
Let $\mathcal{R}$ be the set of regions that each image is broken into.
for each region $r \in \mathcal{R}$ **do**
  Train H^r on $\mathcal{G}^r$
  for each image[a] $g_i \in \mathcal{G}$ **do**
    Classify $H^r(g_i)$
    Record the result
  end for
  Find w^r using a weighting equation and the results above
end for
Classify the image by $\arg\max_{i \in \mathcal{I}} \left\{ \sum_r w^r eq(H^r(p) = i) \right\}$

[a]As a timesaver, instead of taking out all images one at a time, one sample from all identities can be removed, and processed simultaneously.

Chapter 4

Experiments

Extensive experiments were carried out to validate the Weighted Regional Voting approach. For each dataset, the images were divided k times horizontally and k times vertically for $k = 1, 2, \ldots 20$. Four different holistic classifiers were used: Eigenfaces[80] (see section 2.1.1), Fisherfaces[9] (see section 2.1.2), SLPP[92](see section 2.1.4) and SRDA[17] (see section 2.1.3). Each of the holistic classifiers was trained on the gallery. Probe images were reduced using the linear reduction found during the training phase, and then classified by nearest neighbour classification. Each image to be classified was normalized by scaling the representative vector so that it had a unit sum. All gallery images were perturbed up to two pixels in each every direction, to account for possible misalignment issues. The closest match from among all the perturbed images was selected.

All four dimension reduction techniques were embedded using code from Deng Cai's website[15]. For the Eigenface approach, the eigenvectors corresponding to the largest 80% of the eigenvalues were used to calculate the projection function. For the Fisherface approach, the data was first reduced using PCA, and then reduced using Fisher. For the SRDA approach, the α selected for regularization was $\alpha = 0.01$. For

SLPP, cosine similarity was used to calculate the distances in the adjacency matrix. Again, the α selected for regularization was $\alpha = 0.01$.

In order to validate the Weighted Regional Voting approach, three different datasets were used: Yale database[85], the Olivetti Research Laboratory database[8] (ORL) and the Carnegie Melon University Pose, Illumination, and Expression database[81] (CMU PIE).

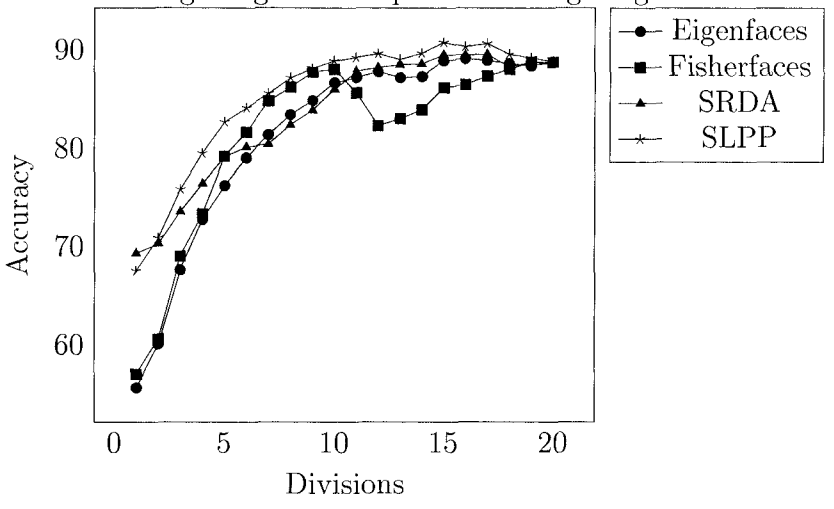
The Yale dataset comes unsurprisingly from Yale University and contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: centre-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink[85].

The ORL dataset comes from the now defunct Cambridge AT&T Laboratory, formerly Olivetti Research Laboratory. In this database, there are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement)[8].

The CMU PIE database was taken at Carnegie Melon University between October and December 2000. There are 41, 368 images of 68 people. For each person, there are 13 different poses, 43 different illumination conditions and four different facial expressions [81]. The experiments performed here, however, used only the five frontal poses (C05, C07, C09, C27, C29), for a total of 170 images per individual, except for six indivudals with only 169 images each.

The faces for all three datasets were normalized by manually finding the eye positions, scaling and translating the faces so that they were aligned on the eyes, and cropping the images to 64×64 pixels. This process was done prior to being

Figure 4.1: Embedding various holistic classifiers in different sized regions on Yale with 2 training images and exponential weighting



downloaded from Deng Cai's website[15].

In the first set of experiments, conducted on the Yale and ORL datasets, each of the four holistic classification techniques mentioned above were embedded in the Weighted Regional Voting framework, with different numbers of regions, from 1×1 to 20×20. That is, each image was first left undivided, then divided twice vertically, and twice horizontally, and then thrice each direction and so on up to 20 divisons vertically and 20 horizontally. For these experiments, exponential weighting was used (see equation (3.2)), to demonstrate the validity of weighting in general. The weights were estimated using cross-validation as outlined above. Each algorithm is evaluated being split into training and testing data randomly, in 50 different ways for each number of regional divisions. The same 50 splits were used for each regional division The results for the Yale dataset for 2 and 5 training images are given in Figures 4.1 and 4.2 respectively. The results for the ORL dataset for 2 and 5 training images are given in Figures 4.3 and 4.4.

Another set of experiments were run, to test the performance of the various sug-

Figure 4.2: Embedding various holistic classifiers in different sized regions on Yale with 5 training images and exponential weighting

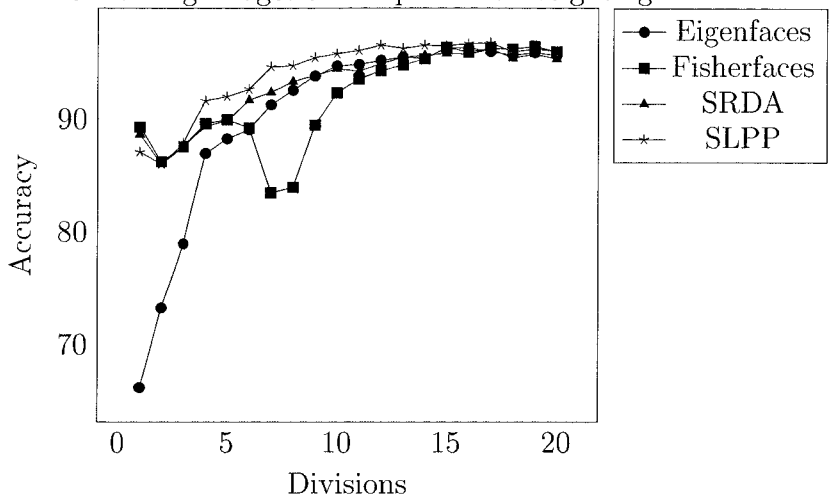


Figure 4.3: Embedding various holistic classifiers in different sized regions on ORL with 2 training images and exponential weighting

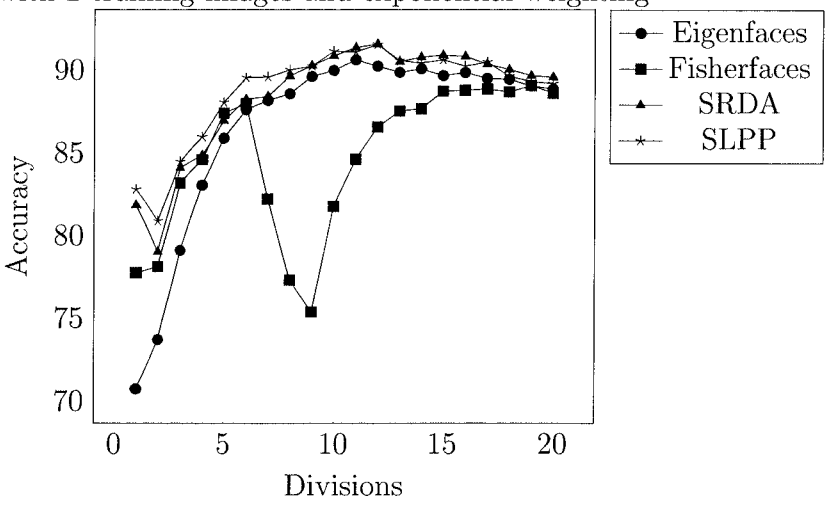
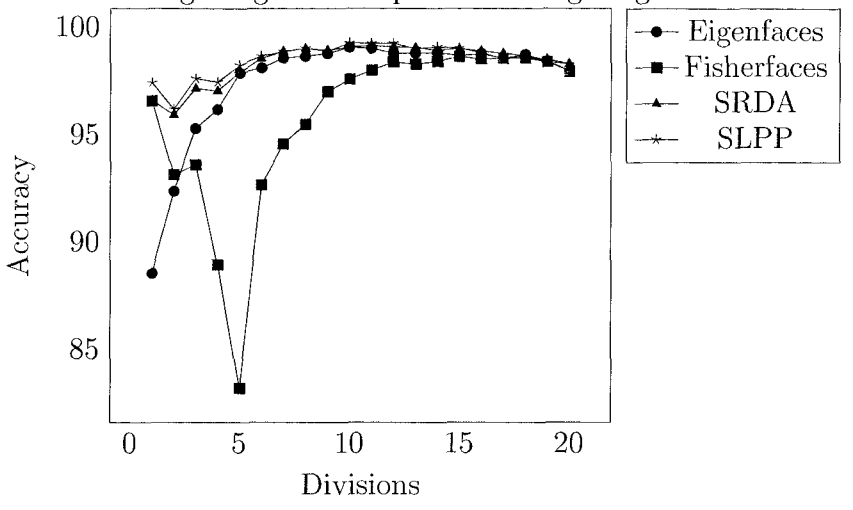


Figure 4.4: Embedding various holistic classifiers in different sized regions on ORL with 5 training images and exponential weighting



gested weighting schemes across various numbers of divisions. All three suggested weighting schemes – exponential weighting (see equation (3.2)), polynomial weighting (see equation (3.3)) and directly proportional weighting (see equation (3.1)) – as well as equal weighting were compared. Equal weighting corresponds to standard regional weighting[22] (see section 2.3). The weighting schemes were tested using by embedding SLPP. As above, each division was tested with the same 50 random splits for each dataset. The results for the Yale dataset for 2 and 5 training images are given in Figures 4.5 and 4.6 respectively. The results for the ORL dataset for 2 and 5 training images are given in Figures 4.7 and 4.8.

For comparison, the same tests were run, but instead of finding the best weights based on cross validation, the weights that gave the best results on the *testing* images was used. The results for the Yale dataset with two and five training images are given in Figures 4.9 and 4.10 respectively and The results for the ORL dataset with two and five training images are given in Figures 4.11 and 4.12 respectively.

More tests were done to compare the results of Weighted Regional Voting with

Figure 4.5: Comparing various weighting schemes in different sized regions on Yale with 2 training images and SLPP as the regional classifier

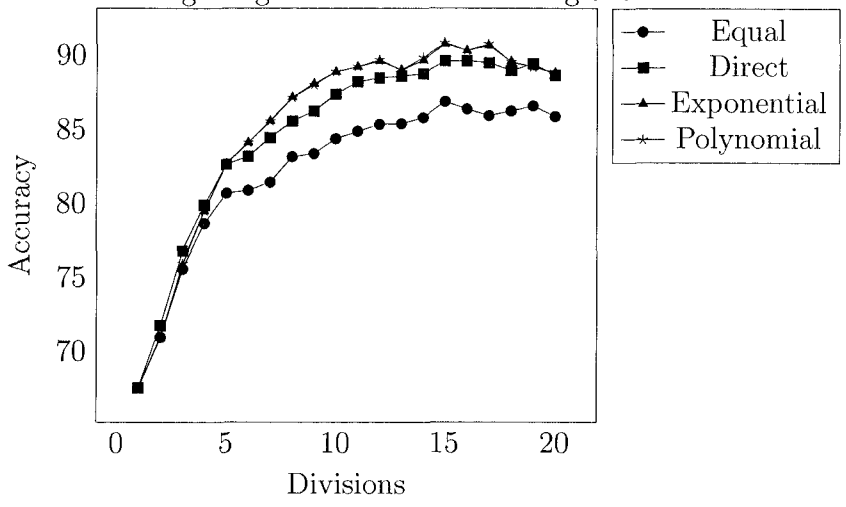


Figure 4.6: Comparing various weighting schemes in different sized regions on Yale with 5 training images and SLPP as the regional classifier

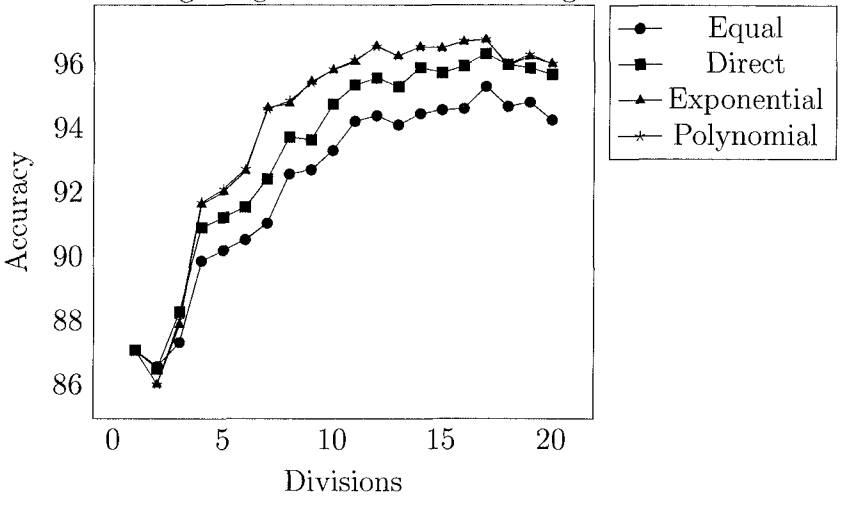


Figure 4.7: Comparing various weighting schemes in different sized regionson ORL with 2 training images and SLPP as the regional classifier

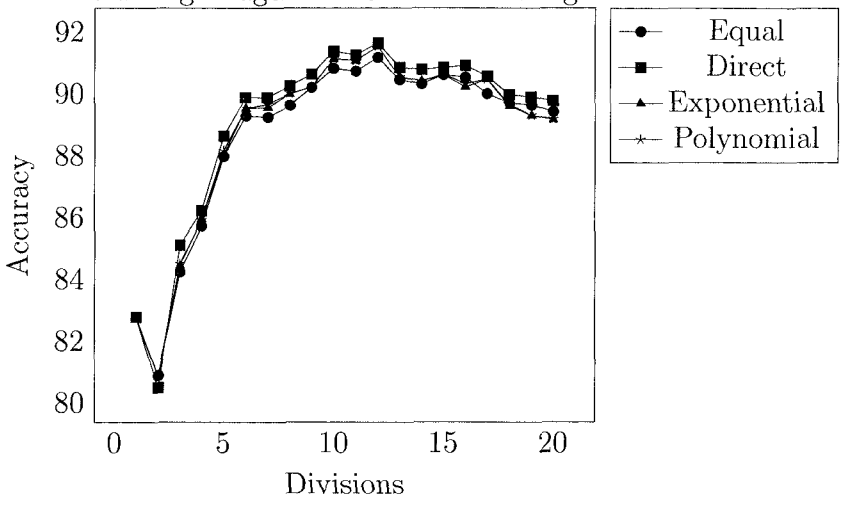


Figure 4.8: Comparing various weighting schemes in different sized regions on ORL with 5 training images and SLPP as the regional classifier

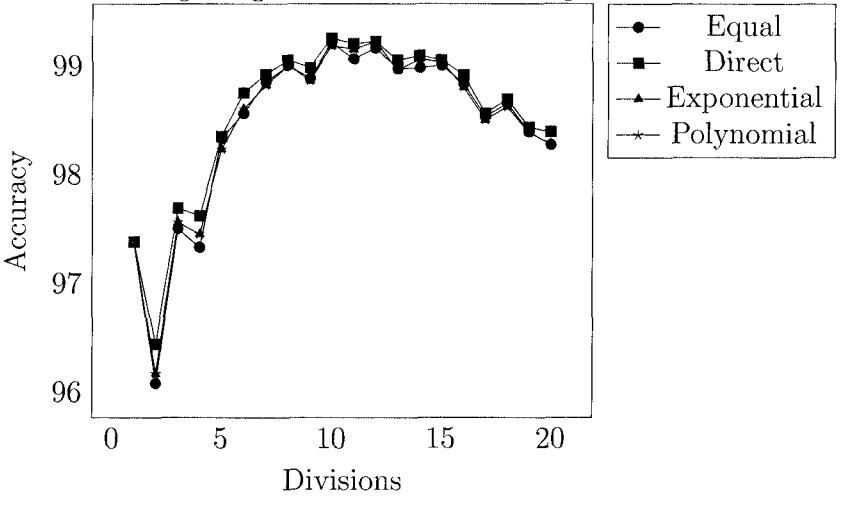


Figure 4.9: Comparing various weighting schemes in different sized regions with best weights on Yale with 2 training images and SLPP as the regional classifier

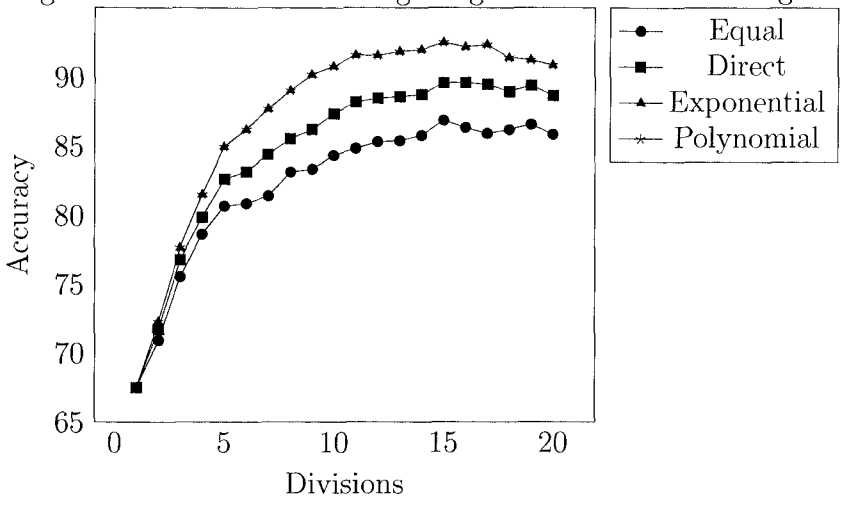


Figure 4.10: Comparing various weighting schemes in different sized regions with best weights on Yale with 5 training images and SLPP as the regional classifier n

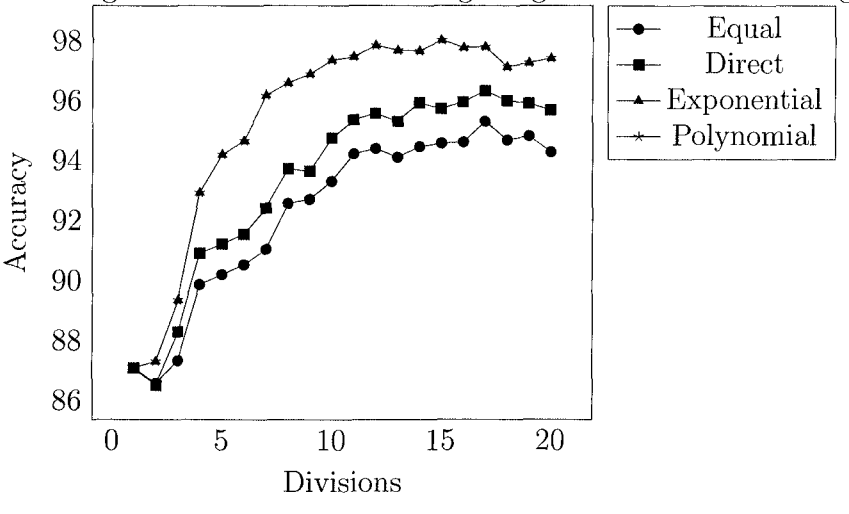


Figure 4.11: Comparing various weighting schemes in different sized regions with best weights on ORL with 2 training images and SLPP as the regional classifier

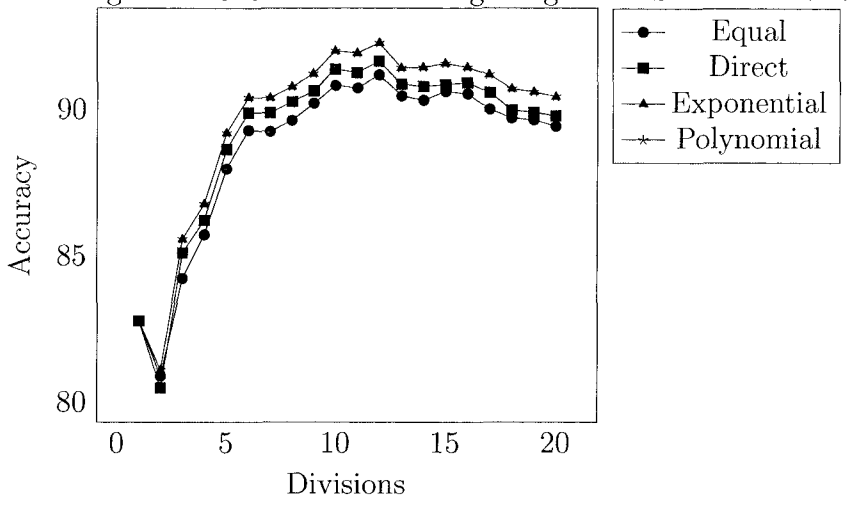
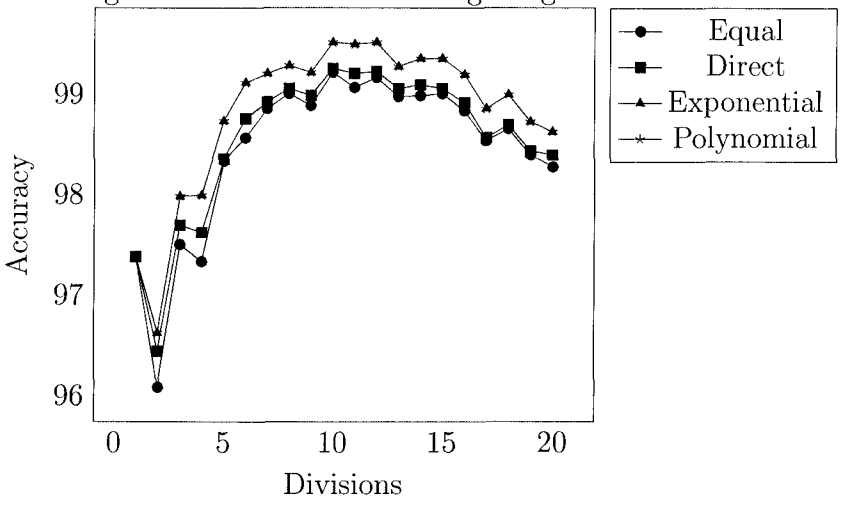


Figure 4.12: Comparing various weighting schemes in different sized regions with best weights on ORL with 5 training images and SLPP as the regional classifier



the results of some top face matching algorithms. The result of performing Weighted Regional Voting with 16×16 divisions is compared to performing the holistic approach with no regional divisions. The results are also compared to two local feature based approaches: Local Binary Patterns[4] (see section 2.2.1) and Volterrafaces[48] (see section 2.2.2). For the Yale and ORL datasets, the results over 50 random splits are compared using both 2 subjects for training and 5. For CMU PIE, only one random split was used, with 5 training subjects.

Each of the other algorithms was tested locally. The Local Binary Pattern algorithm was taken from the University of Oulu's Computer Science and Engineering website[73]. The Volterrafaces code provided by Ritwik Kumar on the MATLAB Central website[49] was modified to run on my face data. For Local Binary Patterns, the results using histogram intersection and a neighbourhood of 16 circular pixels with distance two from the centre are shown. For Volterrafaces, 8×8 patches were used, with each patch overlapping the next by four pixels, using a linear kernel of size 5×5. These parameters were chosen to yield the highest results across all datasets.[1]

The results for the Yale Dataset for 2 training subjects and 5 training subjects are given in Tables 4.1 and 4.2 respectively. The results for the ORL dataset with 2 and 5 training subjects are given in Tables 4.3 and 4.4. The results for the CMU PIE dataset for five training subjects are given in Table 4.5.

[1]The Volterra code was modified by Liang Chen to work with the the datsets I was using. The experiments using the Volterra and LBP algorithms on these data sets were carried out by Dr. Liang Chen and summaries presented in this thesis are based on results are provided courtesty of him.

Table 4 1 Comparison of various face classifiers on Yale Database with two training subjects with 16×16 divisions

	Polynomial Weighting	Exponential Weighting	Proportional Weighting	No Weights	Global
Alg	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
PCA	$10\ 90\% \pm 3\ 34\%$	$10\ 90\% \pm 3\ 34\%$	$11\ 97\% \pm 3\ 22\%$	$15\ 36\% \pm 3\ 07\%$	$44\ 40\% \pm 5\ 14\%$
Fisher	$13\ 54\% \pm 3\ 29\%$	$13\ 54\% \pm 3\ 29\%$	$12\ 87\% \pm 2\ 91\%$	$15\ 29\% \pm 2\ 62\%$	$43\ 02\% \pm 4\ 67\%$
SRDA	$10\ 53\% \pm 3\ 08\%$	$10\ 53\% \pm 3\ 08\%$	$10\ 79\% \pm 2\ 37\%$	$14\ 03\% \pm 2\ 82\%$	$30\ 71\% \pm 4\ 69\%$
SLPP	$9\ 70\% \pm 3\ 25\%$	$9\ 70\% \pm 3\ 39\%$	$10\ 40\% \pm 3\ 01\%$	$13\ 66\% \pm 3\ 04\%$	$32\ 50\% \pm 4\ 44\%$
LBP	-	-	-	-	$24\ 78\% \pm 6\ 19\%$
Volterra	-	-	-	-	$28\ 99\% \pm 3\ 91\%$

Table 4.2: Comparison of various face classifiers on Yale Database with five training subjects with 16×16 divisions

	Polynomial Weighting	Exponential Weighting	Proportional Weighting	No Weights	Global
Alg	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
PCA	$3.73\% \pm 2.08\%$	$3.67\% \pm 2.04\%$	$4.56\% \pm 2.16\%$	$5.98\% \pm 2.46\%$	$33.84\% \pm 3.38\%$
Fisher	$4.07\% \pm 2.12\%$	$4.07\% \pm 2.12\%$	$4.20\% \pm 2.07\%$	$5.40\% \pm 1.92\%$	$33.84\% \pm 3.16\%$
SRDA	$4.27\% \pm 2.24$	$4.24\% \pm 2.27\%$	$4.80\% \pm 2.07\%$	$6.31 \pm 2.17\%$	$11.38\% \pm 2.96\%$
SLPP	$3.29\% \pm 2.07\%$	$3.31\% \pm 2.08\%$	$4.07\% \pm 2.07\%$	$5.40\% \pm 2.36\%$	$12.93\% \pm 3.60\%$
LPB	-	-	-	-	$14.06\% \pm 2.97\%$
Volterra	-	-	-	-	$14.06\% \pm 2.81\%$

Table 4 3 Comparison of various face classifiers on ORL Database with two training subjects with 16×16 divisions

	Polynomial Weighting	Exponential Weighting	Proportional Weighting	No Weights	Global
Alg	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
PCA	10 12% ± 2 10%	10 17% ± 2 06%	9 74% ± 1 91%	10 12% ± 2 23%	29 29% ± 3 15%
Fisher	11 21% ± 2 25%	11 24% ± 2 27%	10 65% ± 2 44%	10 79% ± 2 40%	22 28% ± 2 82%
SRDA	9 16% ± 2 08%	9 19% ± 2 06%	8 92% ± 0 796%	9 24% ± 2 13%	18 19% ± 2 81%
SLPP	9 69% ± 2 22%	9 79% ± 2 27%	9 09% ± 2 14%	9 48% ± 2 14%	17 53% ± 3 04%
LBP	-	-	-	-	15 38% ± 2 73%
Volterra	-	-	-	-	24 46% ± 2 53%

Table 4 4 Comparison of various face classifiers on ORL Database with five training subjects with 16×16 divisions

	Polynomial Weighting	Exponential Weighting	Proportional Weighting	No Weights	Global
Alg	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
PCA	1 26% ± 0 929%	1 26% ± 0 929%	1 14% ± 0 866%	1 31% ± 0 916%	11 48% ± 2 26%
Fisher	1 49% ± 0 886%	1 51% ± 0 886%	1 35% ± 0 814%	1 45% ± 0 934%	3 45% ± 1 30%
SRDA	1 15% ± 0 814%	1 14% ± 0 800%	1 03% ± 0 796%	1 20% ± 0 742%	3 44% ± 1 19%
SLPP	1 20% ± 0 889%	1 22% ± 0 890%	1 09% ± 0 785%	1 17% ± 0 840%	2 62% ± 1 20%
LBP	-	-	-	-	3 74% ± 1 30%
Volterra	-	-	-	-	7 71% ± 1 83%

Table 4.5: Comparison of various face classifiers on CMU PIE Database with five training subjects with 16×16 divisions

	Polynomial Weighting	Exponential Weighting	Proportional Weighting	No Weights	Global
Alg	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
PCA	20 55%	20 55%	20 46%	23 21%	35 63%
Fisher	20 47%	20 47%	20 65%	22 46%	35 63%
SRDA	20 00%	20 00%	20 34%	22 44%	29 76%
SLPP	18 85%	18 85%	19 00%	21 38%	28 72%
LPB	-	-	-	-	37 71%
Volterra	-	-	-	-	19 37%

Chapter 5

Analysis

It is easy to see that embedding a holistic classifier into the Weighted Regional Voting framework improves the accuracy of the classifier. Figures 4.1, 4.2, 4.3 and 4.4 all show that as the number of regions increases, so too does the accuracy of the system. However, after a certain point, the regions become too small to be effective classifiers, and the accuracy begins to drop. This is precisely the pattern shown by standard Regional Voting[22].

Figures 4.5, 4.6, 4.7 and 4.8 demonstrate that Weighted Regional Voting outperforms standard Regional Voting, as soon as the number of regions is large enough for the weights to make a difference. On the ORL dataset, weighted voting and standard voting are approximately equally powerful. In particular, on the ORL dataset, directly proportional weighting performs the best. This is likely because standard Regional Voting is already so accurate on the ORL dataset that it is difficult to improve upon.

Since Weighted Regional Voting is dependent on the training data to estimate regional weights, it is instructive to examine the accuracy that can be achieved in the best possible case. Figures 4.9, 4.10, 4.11 and 4.12 demonstrate the accu-

racy of the system when weights are selected that are optimal for the testing set. An actual system would not be able to select this ahead of time, but the greatly increased accuracy demonstrates that better weight selection algorithms will yield better results.

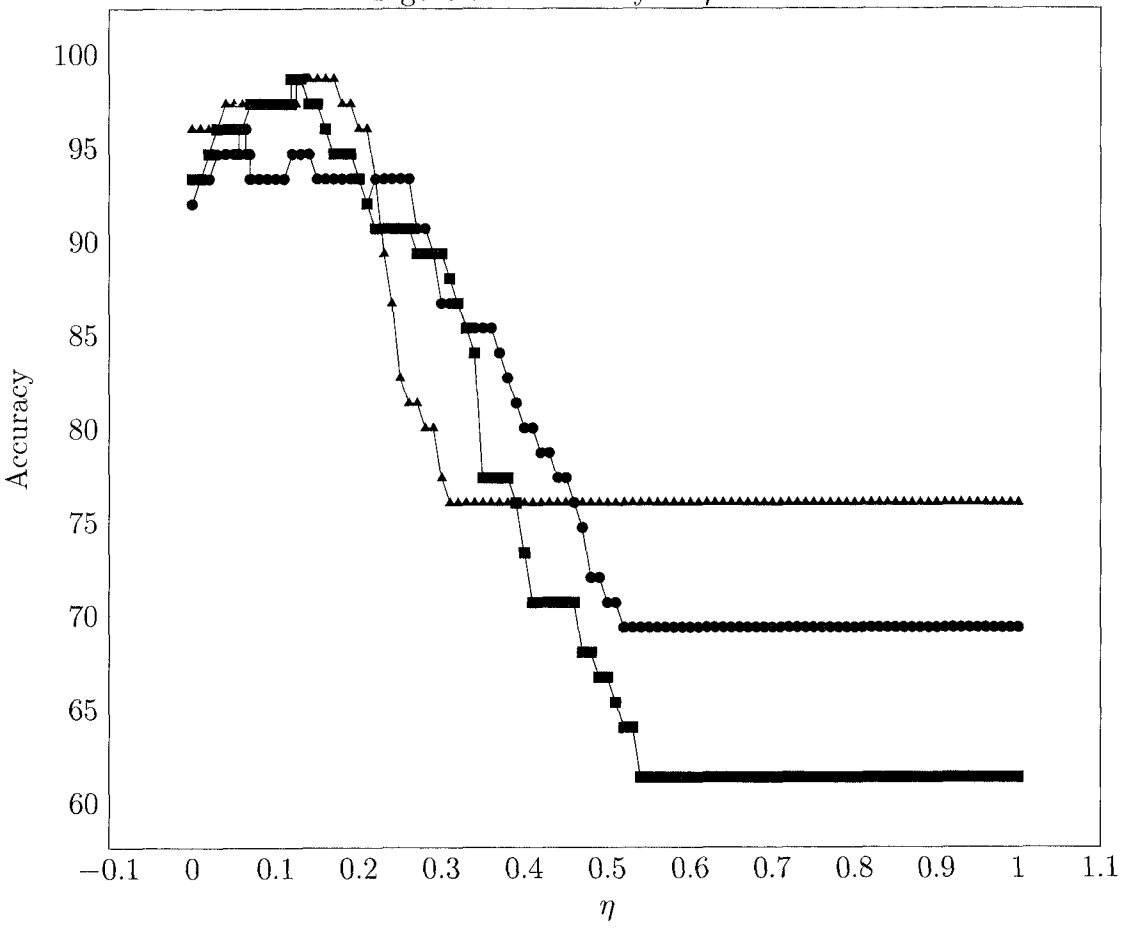
As Weighted Regional Voting embeds a holistic approach and makes it local, it is important to compare the results of Weighted Regional Voting with well known local approaches. Tables 4.1, 4.2, 4.3, 4.4 and 4.5 demonstrate the advantage of Weighted Regional Voting over Volterrafaces and Local Binary Patterns.

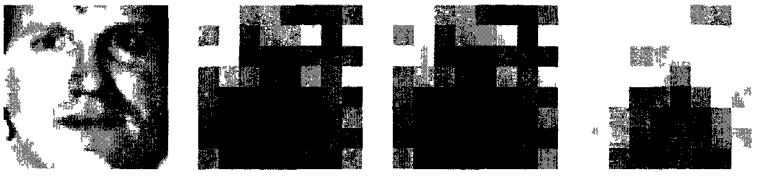
On the Yale dataset, both polynomial and exponential Weighted Regional Voting beats out standard Regional Voting. It is also clear from comparing the error rate that Weighted Regional Voting outperforms Volterrafaces and Local Binary Patterns. On the ORL dataset, the advantage is not as clear, but Weighted Regional Voting is certainly competitive with the other approaches listed, especially for the case where there are only two training images. On the PIE dataset, there is not enough data to perform statistical inference, but the advantage of Weighted Regional Voting over the other techniques is clear.

Figure 5.1 shows the relationship between η and the accuracy of the system for SLPP on the Yale dataset with 5 Training images at 16×16 with polynomial weighting for several different random splits. This shows that accuracy is highly dependent on a good selection of η, but also that different gallery images have different optimal choices for η. However, there are choices of η which will yield good results regardless of the gallery images.

From the graphs, it is clear to see that as the number of regions increases (and the sizes of the regions decreases correspondingly) that all dimension reduction techniques begin to perform about the same. However, SLPP appears to be the most stable, as it outperforms the others at every size.

Figure 5.1: Accuracy vs η





(a) Original Image (b) Exponential Weights (c) Polynomial Weights (d) Proportional Weights

Figure 5.2: Regional Weights for Yale at 16×16

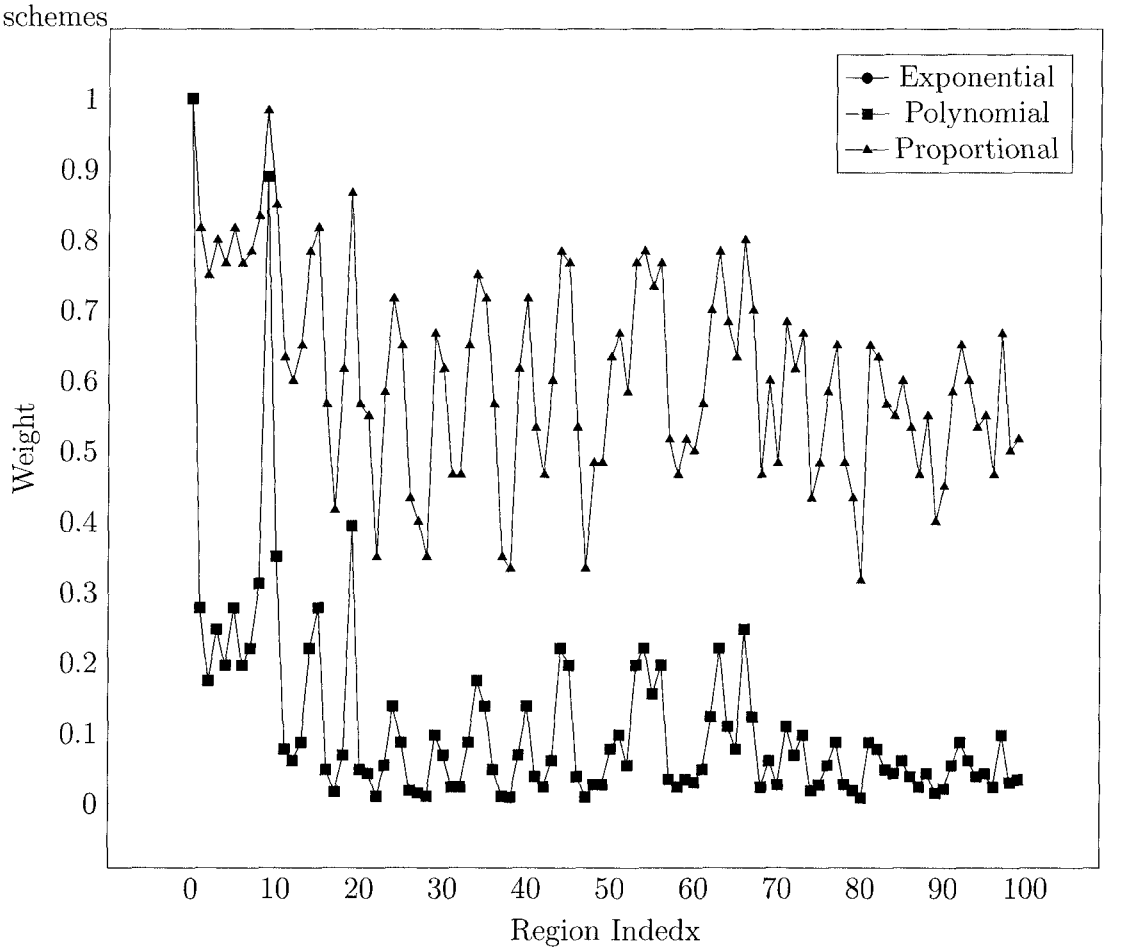
Since the system automatically determines the optimal weights for various regions, it is interesting to see what regions are weighted higher than others. Figure 5.2 shows the various regional weighting schemes when SLPP is embedded in Weighted Regional Voting for 16×16 divisions on the Yale dataset. Brighter colours correspond to a higher weight for that region.

Counter to intuition, the regions of the face that yield best accuracy are not the nose and mouth. In fact, those regions seem to have the lowest accuracy, and thus the lowest weights. The eyes, cheeks and forehead instead seem to be the regions that yield the best recognition accuracy. This demonstrates the advantages of using an automatic weight estimation technique instead of attempting to use human knowledge to combine various regions together.

Figure 5.3 graphs the numerical difference between the three suggested weighting techniques on the Yale dataset, with 5 training images and 10×10 divisions. From this figure, it is easy to see that polynomial and exponential weighting yield almost identical weights, whereas direct proportional weighting has more extreme variation in values, while keeping the same general shape. All weights have been scaled to fit between 0 and 1.

Lastly, Weighted Regional Voting has a very fast running time, at least during the testing phase. The time complexity is obviously dependent on the time complexity of the embedded algorithm. If c is the time of the holistic classifier, then the

Figure 5.3: The weighting of regions using each of the three suggested weighting schemes



time complexity of Regional voting is simply rc where r is the number of regions. Assuming a linear dimension reduction classifier, then the complexity of c is one matrix multiplication (with complexity of hwk where h and w are the height and width of the image and k is the reduced dimension), followed by a distance calculation between the probe and each image in the gallery (complexity hwn). So the total complexity is simply $O(rhw(k+n))$.

The situation for training is much more complex, and because of the disparity in complexities for different holistic algorithms, can't be found directly. But if we assume that t is the complexity of the classifier's training process, then we can provide an estimate of the training complexity. We will perform this trainign r times, once for each region. Then for leave one out cross validation, we will perform n training classifications, so the cross validation process requires $nrhw(k+n)$ operations. The brute force finding of the best parameter for η will try as many values for η as is desired. Let b be the number of values of η tested (this paper used 100,000,000,000,000). Each time required r operations to calculate the weights and another r to find th final vote. So, the total number of operations required is $O(rt + nrhw(k+n) + br)$.

Chapter 6

Summary

The field of Face Recognition is a very popular area for researchers. Many researchers have suggested varying approaches to the problem of Face Recognition. These approaches can be broadly categorized into being either holistic or local algorithms. Although initially holistic algorithms were in favour, in the last decade, local approaches have seen a rise in popularity and accuracy. This thesis attempts to provide a framework for embedding a holistic approach into a local algorithm.

Regional Voting has been shown to be a very stable framework for embedding holistic approaches. Regional Voting consists of dividing the images into non-overlapping regions, performing classification within each region, and classifying the result as the majority vote winner over all the regions.

Weighted Regional Voting, the contribution of this thesis, builds on the success of Regional Voting by estimating a weight for each region. Three different weighting schemes are suggested: one that is directly proportional to the accuracy of the region at classifying the images, and two that are borrowed from Regret Minimization: Exponential Weighting and Polynomial Weighting. These latter two approaches are dependent on parameter selection, but are more sophisticated in how they apply the

weights.

Extensive experiments were run to validate Weighted Regional Voting as a face classification method. Weighted Regional Voting was shown to improve the results of every holistic classifier embedded in it. Weighted Regional Voting was also shown to improve upon the results of standard Regional Voting. In particular, the Exponential and Polynomial weighting schemes achieved almost identical, top of the line results. Weighted Regional Voting was also compared to two current local approaches, and was shown to improve on their results.

Experiments testing the performance of the system under ideal conditions demonstrate that improving the algorithm used for weighting can improve the performance of the system even more. This suggests an avenue of future research: finding an improved weighting scheme. Either another approach from regret minimization, or one from machine learning, such as Adaboost should be investigated.

Furthermore, both Local Binary Patterns and Volterrafaces involve dividing the image into regions. More research could be done in applying a weighting mechanism to those two approaches.

Weighted Regional Voting provides an improvement over any current holistic approach, and is better or equivalently accurate with other local based approaches.

Bibliography

[1] Abdallah S Abdallah, A Lynn Abbott, and Mohamad Abou El-Nasr. A New Face Detection Technique using 2D DCT and Self Organizing Feature Map. *Engineering and Technology*, pages 15–19, 2007.

[2] Bernard Achermann, Xiac Jiang, and Horst Bunke. Face Recognition Using Range Images. *Computer Engineering*, pages 0–7, 1997.

[3] Nasir U. Ahmed and K. Ramamohan Rao. *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, January 1975.

[4] Timo Ahonen, Abdenour Hadid, and Matti Pietik. Face Recognition with Local Binary Patterns. In *ECCV 2004*, pages 469 -481, Berlin, 2004. Springer-Verlag.

[5] Ajmal Mian. *Comparison of Visible, Thermal Infra-Red and Range Images for Face Recognition* , volume 5414 of *Lecture Notes in Computer Science*, pages 807–816. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] L Akarun and A A Salah. 3D Face Recognition for Biometric Applications. *Image Rochester NY*, 2005.

[7] Mohamed Aly. Face Recognition using SIFT Features, November 2006.

[8] AT&T. The ORL Dataset. `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

[9] P Belhumeur, J Hespanha, and D Kriegman. Eigenfaces vs.Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[10] M Bicego, U Castellani, and V Murino. Using hidden Markov models and wavelets for face recognition. *Conference on Image Analysis and Processing*, 12:52–56, 2003.

[11] M Bicego, A Lagorio, E Grosso, and M Tistarelli. On the Use of SIFT Features for Face Authentication. *2006 Conference on Computer Vision and Pattern Recognition Workshop CVPRW06*, 00(c):35–35, 2006.

[12] M Bressan. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, November 2003.

[13] Rodney A Brooks. *Flesh and Machines: How Robots Will Change Us*. Pantheon Books, 2002.

[14] P Buddharaju, I Pavlidis, and I Kakadiaris. Face Recognition in the Thermal Infrared Spectrum. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 00(C):133–133, 2004.

[15] Deng Cai. Deng Cai's Website. `http://www.zjucadcg.cn/dengcai/`.

[16] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised Discriminant Analysis. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, October 2007.

[17] Deng Cai, Xiaofei He, and Jiawei Han. Spectral Regression for Efficient Regularized Subspace Learning. *IEEE 11th International Conference on Computer Vision (2007)*, L(05):1 8, 2007.

[18] Deng Cai, Xiaofei He, and Jiawei Han. SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):1–12, 2008.

[19] Deng Cai, Xiaofei He, Jiawei Han, and Hong-Jiang Zhang. Orthogonal Laplacianfaces for Face Recognition. *Image Rochester NY*, 15(11):3608–3614, 2006.

[20] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[21] Liang Chen and Naoyuki Tokuda. Robustness of regional matching scheme over global matching scheme. *Artificial Intelligence*, 144(1-2):213–232, March 2003.

[22] Liang Chen and Naoyuki Tokuda. A general stability analysis on regional and national voting schemes against noise–why is an electoral college more stable than a direct popular election? *Artificial Intelligence*, 163(1):47–66, 2005.

[23] Liang Chen and Naoyuki Tokuda. A unified framework for improving the accuracy of all holistic face identification algorithms. *Artif. Intell. Rev.*, 33(1-2):107–122, 2010.

[24] Songcan Chen and Yulian Zhu. Subpattern-based principle component analysis. *Pattern Recognition*, 37(5):1081–1083, 2004.

[25] Ronald A DeVore, Bjoern Jawerth, and Bradley J Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*,

38(2):719–746, 1992.

[26] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14(8):1724, 1997.

[27] M J Farah. Is face recognition 'special'? Evidence from neuropsychology. *Behavioural Brain Research*, 76(1-2):181–189, 1996.

[28] M J Farah, K D Wilson, H M Drain, and J R Tanaka. The inverted face inversion effect in prosopagnosia: evidence for mandatory, face-specific perceptual mechanisms. *Vision Research*, 35(14):2089–2093, 1995.

[29] Ronald Aylmer Fisher. The Use of Multiple Measures in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[30] Y Freund and R E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[31] Jerome H Friedman. Regularized Discriminant Analysis. *Journal of the American Statistical Association*, 84(405):165, 1989.

[32] Cong Geng and Xudong Jiang. *SIFT features for face recognition*, volume 0. IEEE, Beijing, China, 2009.

[33] Rafael C Gonzalez, Richard E Woods, and Barry R Masters. Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, 14(2):029901, 2009.

[34] R Gottumukkal and Vijayan K Asari. An improved face recognition technique

based on modular PCA approach. *Pattern Recognition Letters*, 25(4):429–436, 2004.

[35] M.H. Hassoun and P. Watta. Combining gabor features: Summing vs.voting in human face recognition. *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, pages 737–743, 2003.

[36] Dong-Chen He and L Wang. Texture features based on texture spectrum. *Pattern Recognition*, 24(5):391–399, 1991.

[37] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood Preserving Embedding. In *Tenth IEEE International Conference on Computer Vision ICCV05 Volume 1*, volume 2, pages 1208–1213. Ieee, 2005.

[38] Xiaofei He and Partha Niyogi. Locality Preserving Projections. *Advances in Neural Information Processing Systems 16*, 16(December):153–160, 2003.

[39] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[40] R A Howard. *Dynamic programming and Markov process*. MIT Press, 1960.

[41] Yegang Hu and Benyong Liu. Face Recognition Based on PLS and HMM. *Pattern Recognition 2009 CCPR 2009 Chinese Conference on*, pages 1–4, 2009.

[42] Y Ijiri and M Sakuragi. Security Management for Mobile Devices by Face Recognition. *7th International Conference on Mobile Data Management MDM06*, pages 49–49, 2006.

[43] Sina Jahanbin, Hyohoon Choi, Alan C Bovik, and Kenneth R Castleman. Three Dimensional Face Recognition Using Wavelet Decomposition of Range Images. In *ICASSP2007*, pages 3–6, 2007.

[44] I Kakadiaris, G Passalis, and G Toderici. 3D Face Recognition. *Handbook of Biometrics*, pages 1–10, 2008.

[45] Takeo Kanade. Computer recognition of human faces. *Interdisciplinary Systems Research*, 47, 1977.

[46] 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*, 12(1):103–108, 1990.

[47] J Kittler, M Hatef, R P W Duin, and J Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[48] R. Kumar, A. Banerjee, and B.C. Vemuri. Volterrafaces: Discriminant analysis using Volterra kernels. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 150–155. IEEE, June 2009.

[49] Ritwik Kumar. Volterrafaces Face Recognition System. `http://www.mathworks.com/matlabcentral/fileexchange/28027-volterrafaces-face-recognition-system`.

[50] S Z Li and J Lu. Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks*, 10(2):439–443, 1999.

[51] Xilai Li, Aihua Li, and Xiangfeng Bai. 3D face recognition for security and

defense engineering. *Pattern Recognition*, 7820:78201H–78201H–8, 2010.

[52] Yunfeng Li, Zongying Ou, and Guoqiang Wang. Face Recognition Using Gabor Features. pages 119 122, 2005.

[53] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91-110, 2004.

[54] J Luo, Y Ma, E Takikawa, S Lao, M Kawade, and B Lu. Person-Specific SIFT Features for Face Recognition. *Proc of ICASSP*, pages 593–596, 2007.

[55] A. Majumdar and R. K. Ward. Discriminative SIFT features for face recognition. In *Canadian Conference on Electrical and Computer Engineering*, pages 27–30, St. John's, NL, May 2009. IEEE.

[56] Mark Ingebretsen. The Brave New Intelligent Interface. *IEEE Intelligent Systems*, 25(4):4–8, July 2010.

[57] J E McNeil and E K Warrington. Prosopagnosia: a face-specific disorder., 1993.

[58] Christian A Meissner and John C Brigham. Thirty years of investigating the own-race bias in memory for faces: A meta-analytic review. *Psychology Public Policy and Law*, 7(1):3–35, 2001.

[59] Tom Mitchell. *Artificial Neural Networks*, chapter 4, page 414. McGraw Hill, Maidenhead, U.K., 1997.

[60] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[61] Ara V Nefian and Monson H Hayes III. Hidden Markov models for face recognition. In *Acoustics Speech and Signal Processing*, volume 5, pages 2721–2724. IEEE, 1998.

[62] Charles A Nelson. The development and neural bases of face recognition. *Infant and Child Development*, 10(1-2):3–18, 2001.

[63] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, Cambridge, UK, 1 edition, 2007.

[64] NIST. The Feret Database. `http://www.itl.nist.gov/iad/humanid/feret/`.

[65] T Ojala, M Pietikainen, and T Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[66] Timo Ojala, M Pietikainen, and David Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[67] Christopher C Paige and Michael A Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[68] Olivier Pascalis and Jocelyne Bachevalier. Face recognition in primates: a cross-species study. *Behavioural Processes*, 43(1):87–96, 1998.

[69] Alex Pentland and Tanzeem Choudhury. Face recognition for smart environ-

ments. *Computer*, 33(2):50–55, 2000.

[70] P Jonathon Phillips. Support Vector Machines applied to Face Recognition. In *Advances in Neural Information Processing Systems 11*, volume 2, pages 803–809. National Institute of Standards and Technology, {MIT} Press, 1999.

[71] Albert Ali Salah. *Biologically Motivated 3D Face Recognition.* PhD thesis, Bogazici University, 2007.

[72] F Samaria and S Young. Hmm-based architecture for face identification. 12(8):537–543, 1994.

[73] Oulu University Computer Science and Engineering Laboratory. Local Binary Pattern (LBP) implementation for Matlab. `http://www.cse.oulu.fi/MVG/Downloads/LBPMatlab`.

[74] Linlin Shen and Li Bai. A review on Gabor wavelets for face recognition. *Pattern Analysis and Applications*, 9(2-3):273–292, 2006.

[75] L Sirovich and M Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A Optics and image science*, 4(3):519–524, 1987.

[76] K Sobottka and Ioannis Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication*, 12(3):263–281, June 1998.

[77] Anuj Srivastava, Xiuwen Liu, and Curt Hesher. Face recognition using optimal linear components of range images. *Image and Vision Computing*, 24(3):291–299, 2006.

[78] Keren Tan and Songcan Chen. Adaptively weighted sub-pattern PCA for face recognition. *Neurocomputing*, 64:505–511, 2005.

[79] Naoyuki Tokuda and Liang Chen. Regional Voting versus National Voting - Stability of Regional Voting. In *Proceedings of International ICSC Congress on Computational Intelligence Methods and Applications*, Rochester, New York, 1999.

[80] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, January 1991.

[81] Carnigie Melon Univeristy. The CMU PIE datset. `http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261`.

[82] T Valentine. Upside-down faces: a review of the effect of inversion upon face recognition. *British journal of psychology London England 1953*, 79 (Pt 4)(4):471–491, 1988.

[83] Crystal Whittier and Xiaojun Qi. Supervised Heat Kernel LPP Method for Face Recognition, 2006.

[84] Lawrence B Wol, Diego A Socolinsky, and Christopher K Eveland. Face Recognition in the Thermal Infrared. *New York*, 2005.

[85] Yale. The YALE Dataset. `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

[86] S Yan, D Xu, B Zhang, and Hong-Jiang Zhang. Graph Embedding: A General Framework for Dimensionality Reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, volume 2,

pages 830 837. Ieee, 2005.

[87] Wankou Yang, Jianguo Wang, Mingwu Ren, and Jingyu Yang. Fuzzy 2-Dimensional FLD for Face Recognition. *Journal of Information and Computing Science.*, 4(3):233–239, 2009.

[88] W Yu, X Teng, and C Liu. Face recognition using discriminant locality preserving projections. *Neurocomputing*, 70(3):239–248, 2006.

[89] Guangcheng Zhang, Xiangsheng Huang, Stan Z Li, Yangsheng Wang, and Xihong Wu. Boosting Local Binary Pattern (LBP)-Based Face Recognition. In *on Biometric Recognition*, volume 3338/2005, pages 179–186. Springer, 2004.

[90] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. *Tenth IEEE International Conference on Computer Vision ICCV05 Volume 1*, 1:786–791, 2005.

[91] W Zhao, R Chellappa, P J Phillips, and A Rosenfeld. Face recognition: A Literature Survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.

[92] Z Zheng, F Yang, W Tan, J Jia, and J Yang. Gabor feature-based face recognition using supervised locality preserving projection. *Signal Processing*, 87(10):2473–2483, 2007.

[93] Jie Zou, Qiang Ji, and George Nagy. A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing*, 16(10):2617–2628, 2007.