Face Recognition Committee Machine: Methodology, Experiments and A System Application

Tang Ho-Man

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. Michael R. Lyu

©The Chinese University of Hong Kong
July 2003

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.

Abstract of thesis entitled:

Face Recognition Committee Machine:
Methodology, Experiments and A System Application
Submitted by Tang Ho-Man
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in July 2003

Face recognition has raised much attention since 1990 due to its wide applications such as access control, surveillance and multimedia search engine. Numerous algorithms proposed by researchers are claimed to have satisfactory results in past years. However, these algorithms are evaluated under different face databases with various pre-processing approaches. Thus, these algorithms are not compared objectively. In this thesis, we implement five state-of-the-art algorithms under the same framework: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching, (4) Support Vector Machines and (5) Neural Networks. We provide a thorough comparison of their performances with four standard well-known face databases: ORL, Yale, AR and HRL.

Currently, there is no unique face recognition algorithm that can handle all kinds of variations such as different poses, illumination conditions, expressions and glasses. Besides, accuracy of current algorithms is still not satisfactory. Combining results of algorithms in various face recognition fields is a key to success. We therefore propose a framework that integrates various face recognition algorithms as experts in a committee machine. We call this machine as Face Recognition Committee Machine (FRCM). We define result, confidence and weight for each expert, and explain the ensemble of the experts' results. In addition, we design two architectures for FRCM: Static Structure and Dynamic Structure. We also propose a feedback mechanism to adjust the weights of the experts according to their performance. The dynamic architecture and the feedback mechanism help adapting different environments in real situations.

Furthermore, we implement the FRCM in a face recognition system to demonstrate our work. It provides face tracking, detection and recognition in real-time for verification and identification applications. In addition, we propose a distributed system solution to solve the time and storage overhead of committee machine, which can be further applied to light-weight device such as PDA or mobile phone.

摘要

自 1990 年以來，由於存取控制、監視系統及多媒體搜索系統被廣泛應用，人臉辨識引來了極大關注。在過去幾年，研究員們提出了許多均聲稱有令人滿意的結果的算法。但是，這些算法皆被測試在不同的人臉數據庫之下並以各種各樣的方法預先處理，因此各算法都不能客觀地作比較。在這份論文中，我們開發了五種目前最先進的算法：（1）特徵臉、（2）費沙臉、（3）彈性圖匹配、（4）支撐向量機及（5）神經網絡在同一框架之內，並以四個知名的標準人臉資料庫：ORL、耶魯、AR 和 HRL 作客觀比較。

由於目前並沒有一個人臉辨識算法能處理面部上如不同姿勢、燈光、表情及有否佩戴眼鏡等變化，加上現有算法的準確性仍未令人滿意，結合各樣領域的人臉辨識算法是一個成功之法。我們提出了一個為各樣人臉辨識算法結合的一個框架，稱作人臉辨識委員機（FRCM）。我們為每個專家定義三個主要元素：結果、信心和比重並展示如何結合各專家的結果。我們提出兩種不同結構的委員機：靜態結構及動態結構以適應真實情況的不同環境。除此以外，我們提出回應機制使動態結構能根據專家們的表現以調整各專家的比重來開發出一個能適應不同環境的人臉辨識系統。

此外，我們開發一個以 FRCM 為基礎的人臉辨識系統來展示我們的研究成果。這系統提供實時人臉跟蹤、人臉檢測和人臉辨識作為身份識別及身份確認等應用。我們亦提出一個分散式系統方案來解決因應用委員機所帶來的時間及儲存問題，這方案更可進一步應用於輕量型手提設備如個人數碼助理及流動電話。

Acknowledgement

I would like to take this opportunity to express my gratitude to my supervisor, Prof. Michael R. Lyu, for his generous guidance and patience to me in the past two years. I am also grateful for the time and valuable suggestions that Prof. Irwin King has given. The inspiring advice from Prof. Lyu and Prof. King are extremely essential and valuable. Without their effort, I will not be able to strengthen and improve my research and papers (published in ICASSP 2003, ICIAP 2003 and CAIP 2003).

I would also like to show my gratitude to the Department of Computer Science and Engineering, the Chinese University of Hong Kong for the provision of the best equipment and pleasant office environment for high quality research.

Special thanks should be given to my project partner Harvest Jang who worked with me in my research. He has given me encouragement and supports. And I would like to give my thanks to my fellow colleagues, M.L. Ho, K.C. Chan, T.C. Lau, Sam Cheng, Joe Lau, Annie Chiu, Jacky Ma, Nicky Ng, C.H. Law, W. Hung, Edith Ngai and Pat Chan. They have given me a joyful and wonderful time in my research. These two years should not have been that fruitful without them.

This work is dedicated to my family for the support and patience

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Security is gaining importance in recent years. People look for more secure methods to protect their valuable information. Three main types of authentication approach are commonly used: (1) password authentication, (2) card key authentication and (3) biometric authentication. Currently, we need a PIN to get cash from ATM; a password to access a computer or internet services and a key to unlock our door. However, these measures are not secure. For examples, password can be guessed easily as people probably pick ones that are easy to remember like child's name or favorite sport; card key can also be lost or snooped easily. Only biometric characteristics can not be borrowed, stolen or forgotten. Users can not pass their characteristics to other. All these prove that biometric is the most secure authentication approach among the three security measures.

In [40], Liu mentioned that biometrics measure individuals' unique physical or behavioral characteristics to recognize or authenticate their identities. The common physical biometrics include fingerprint, hand geometry, palm geometry, retina, iris and facial characteristics. Different technologies may be appropriate for different applications and a comparison for some common biometrics are presented in Table 1.1 [2].

Table 1.1: Comparison of biometrics

	Retina	Iris Scan	Fingerprints	Face	Voice
Accuracy (error rate)	Very High	Very High	High	High	Medium
Ease of use	Low	Medium	High	Medium	High
Barrier to attack	Very High	Very High	High	Medium	No
Uses personally distinct char.	Very high	Very high	High	High	Medium
User friendliness	Medium	Medium	Medium	High	Medium
Long-term stability	High	High	High	Medium	Medium

Among the biometric technologies, retina and iris are the most secure measures. However, they are not user friendly and they sacrifice personal privacy. In [49], Pentland mentioned that although fingerprint, retina and iris recognition are appropriate for bank transactions and entry into secure areas, they have the disadvantage of being intrusive, both physically and socially. The reason is that people do not recognize each other by retina scans and these types of identifications feel intrusive. Instead, face is the most user friendly measure and most natural because this is how human recognize other people. People are likely to be more comfortable with systems that use similar means of recognition.

Face recognition provides a convenient way to recognize a person from a group of people. It recognizes a person by simply taking an image with video camera. User no longer needs to scan fingerprint or iris. Instead, the user just needs to stand in front of the camera. This personal identification approach is more user friendly than the intrusive approaches mentioned. Therefore, we concentrate on face recognition in this thesis due to its wide acceptance and user friendliness.

1.2 Face Recognition

Face recognition is a recognition process that analyzes facial characteristics of a person. It consists of two main phases: enrollment phase and recognition phase. In enrollment phase, images of different people with known identities are taken to form a face database. These images are then stored as templates to train a face recognition system. In recognition phase, we classify recognition into two modes: [2]

- Identification: Similar to answer the question "Who is this?" (One-to-many relation). A test image of a person is taken, which is then compared with other templates. The template image with highest similarity would be chosen as the recognized image. The person's identity would be the recognized image's identity accordingly.
- Verification: Similar to answer the question "Is this person who she/he claim to be?" (One-to-One relation). A test image with claimed identity from the person is compared with the templates of that identity. The system authenticates that identity if the two images match.

Face recognition can be applied to various applications such as access control, surveillance and multimedia search engine. In a literature survey [68], Zhao mentioned some typical face recognition applications which is listed in Table 1.2.

Access control system is one of the major verification applications in security, which securely protects any important data or places from intruder. Whenever a person enters this system, an image of the person is taken. Only authorized user can get access to the system. Conversely, intruder who is not in the face database would be rejected.

Apart from the security application, face recognition can be applied in other fields like multimedia search engine. Fast growing on multimedia technology and Internet technology enables searching for multimedia data like video possible. However, information retrieval within vast amount of multimedia data is

Table 1.2: Typical face recognition applications

Areas	**Specific Applications**
Biometrics	Drivers' Licenses, Immigration, National ID, Passports
Information Security	Desktop Logon, Application Security
Law Enforcement and Surveillance	CCTV Control, Portal Control
Smart Cards	Stored Value Security, User Authentication
Access Control	Facility Access, Vehicular Access
Multimedia Database	Video Indexing, Human Search Engine

still a challenging task. With face recognition and video segmentation technology, we can find videos of a particular person easily by simply providing an image of that person to the search engine. All related videos like news clips would be retrieved easily.

1.3 Contributions

Our research work has the following contributions:

- Face recognition has raised much attention since 1990. The increase is due to its wide applications such as access control, surveillance and multimedia search engine. Numerous algorithms proposed by researchers are claimed to have satisfactory results in past years. However, these algorithms are evaluated under different face databases with various pre-processing approaches. Thus, these algorithms are not compared objectively. Therefore, we implement five state-of-the-art algorithms under the same framework: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching (EGM), (4) Support Vector Machines (SVM) and (5)

Neural Networks (NN). We provide a thorough comparison of their performances with four standard well-known face databases: ORL, Yale, AR and HRL.

- Currently, there is no unique face recognition algorithm that can handle all kinds of variations such as different poses, illumination conditions, expressions and glasses. Besides, accuracy of current algorithms is still not satisfactory. Combining results of algorithms in various face recognition fields is a key to success. We therefore propose a framework that integrates various face recognition algorithms as experts in a committee machine. The committee machine consists of the five algorithms mentioned above. We call this machine as Face Recognition Committee Machine (FRCM). We define result, confidence and weight for each expert and explain the ensemble of the experts' results. (published in ICASSP 2003 titled "Face Recognition Committee Machine" [55])

- We design two architectures for FRCM: Static Structure and Dynamic Structure. We also propose a feedback mechanism to adjust the weights of the experts according to their performance. The dynamic architecture and the feedback mechanism help adapting different environments in real situations, which are useful in the development of a robust face recognition system. (published in ICIAP 2003 titled "Face Recognition Committee Machines: Dynamic Vs. Static Structures" [56])

- We implement the FRCM in a face processing system to demonstrate our work, which consists of two modules: Face Detection Module and Face Recognition Module. It provides face tracking, detection and recognition in real-time for verification and identification applications. We apply FRCM in the Face Recognition Module to achieve better performance. (published in CAIP 2003 with co-author Harvest Jang titled "A Face Processing System Based on Committee Machine: The Approach and Experimental Results" [35])

- We propose a distributed system solution for the time and storage overhead of committee machine, which can be further applied to light-weight device such as PDA or mobile phone to enable face recognition applications feasible in the future .

1.4 Organization of this Thesis

The thesis is organized in the following way:

- Chapter 1 (this chapter) gives a brief introduction of biometrics. It compares the advantages and disadvantages of different biometric technologies. It also introduces face recognition and shows some face recognition applications. Moreover, it outlines the contributions and the organization of this thesis.

- Chapter 2 is a review of the previous related work. It first gives a brief review on committee machine and shows the two major classifications of committee machine in the literature. It also explains the five face recognition algorithms related to our research work. Furthermore, it describes some current commercial products in the market to show the state-of-the-art technologies.

- Chapter 3 describes the FRCM with Static Structure. We call this machine as Static Face Recognition Committee Machine (SFRCM). It first gives an overview of the SFRCM. It then defines result, confidence and weight for each expert and describes how each value can be obtained from the experts. In addition, it explains the ensemble of the experts' results.

- Chapter 4 states the problems of SFRCM and describes another FRCM with Dynamic Structure. We call this as Dynamic Face Recognition Committee Machine (DFRCM). It explains the use of a gating network to change the weights of the experts dynamically. Moreover, it describes a feedback mechanism to update the weights of the experts according to their performance.

- Chapter 5 describes the implementation of the FRCM in a face recognition system in details. It first gives the overall system architecture, then it explains the two main modules in the system: face detection module and face recognition module. Furthermore, it states the time and storage overhead of committee machine and shows how distributed system can be applied to solve the problem.

- Chapter 6 evaluates the SFRCM, DFRCM and the five expert algorithms with four well-known face databases. It introduces these face databases and explains the experimental procedures. Finally, it analyzes the experimental results of the FRCMs and compares the results with the experts to show the improvement.

- Finally, Chapter 7 is a conclusion of the thesis. It summarizes the achievements of the research work.

□ **End of chapter.**

Chapter 2

Literature Review

In this chapter, we first give a brief review on committee machine. We introduce two main types of committee machines in the literature. Then, we explain five well-known face recognition algorithms in details which are related in our research. They are: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching (EGM), (4) Support Vector Machines (SVM) and (5) Neural Networks (NN). Our proposed Face Recognition Committee Machine (FRCM) is composed of the algorithms above. Finally, we show some leadership commercial face recognition products which make use of the above algorithms to demonstrate the effectiveness of the algorithms.

2.1 Committee Machine

In recent years, the committee machine, an ensemble of estimators, has proven to give more accurate results than the use of a single predictor [7]. The basic idea of committee machine is to train a committee of classifiers and combine the individual predictions to achieve improved generalization performance. A key assumption for committee machine is that classifiers are independent in their predictions. Therefore, the prediction errors may not be overlapped and so combining the predictions from the classifiers may cancel the errors out to some degree to make the improvement [58].

Recently, researchers have applied the committee machine in various fields. Disorntetiwat used simple ensemble-averaging model on neural network in financial forecasting problem [13]. Mao applied bagging, boosting and basic ensembles of neural networks for OCR problems [42]. Su and Basu used the gating network on the image deblurring problem [54]. In face recognition, Gutta *et al.* used an ensemble of Radial Basis Function (RBF) network and a decision tree in the face processing problem [26] [27]. Huang *et al.* formulated an ensemble of neural networks for pose invariant face recognition [32].

Different approaches of committee machine are proposed by researchers within the last ten years. We can generally classify them into two different structures, static structure and dynamic structure:

2.1.1 Static Structure

This is generally known as an ensemble method. Input data is not involved in combining the committee experts. It is generally believed that a good ensemble should consist of a set of individual classifiers which are individually accurate and are loosely correlated in making errors. By using different kinds of classifiers with various features, or training classifiers with different training data set, we can obtain classifiers for the ensemble [42]. The final decision is obtained by majority voting or averaging from the results of the classifiers. Examples of static structure include basic ensemble, bagging and boosting:

- Basic ensemble: Basic ensemble is a collection of M neural networks with identical structure which are trained on the same data set with various initial weights.

- Bagging ensemble: Bagging (Bootstrap aggregation) [6] is a statistical re-sampling technique for generating various training data set for each classifier in the ensemble. Each training data set is created by randomly drawing N data points with replacement from the original training data set with N data points. This means that some data points will appear more than once in a given new data set and some will not appear at all. Using this technique, the resulting networks are less correlated than the Basic ensemble.

- Boosting: Boosting [51], [17], [18], is another statistical re-sampling technique for generating a serious of classifiers. The training set used for each member of the series is chosen based on the performance of the earlier classifiers in the series. In Boosting, examples that are incorrectly predicted by previous classifiers in the series are chosen more often than examples that were predicted correctly. Therefore, Boosting tries to create new classifiers that can be able to predict examples for which the current ensemble's performance is poor [46].

2.1.2 Dynamic Structure

Input is directly involved in the combining mechanism that employs an integrating unit to adjust the weight of each expert according to the input. It is often known as Mixture of Experts (ME) [34]. In 1991, Jacobs *et al.* proposed a learning procedure for systems composed of separate networks. The complete training data is partitioned into different subsets and various experts are trained on each subset. Jacobs proposed the use of a gating network and a selector to select output from the networks' results.

Figure 2.1 gives a overview of the system. Each expert is a feed-forward neural network and all the n experts in the com-

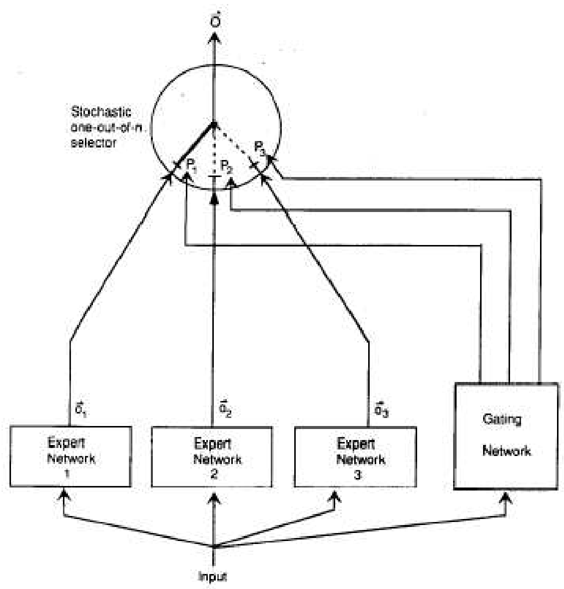

Figure 2.1: System of experts and gating network

mittee machine receive the same input. The gating network is also a feed-forward neural network which receives the same input as the experts. It determines which expert is responsible for the different subsets in the input space.

2.2 Face Recognition Algorithms Overview

In the field of face recognition, numerous face recognition methods were proposed by researchers within these few years. For almost all the techniques, the success depends on how to solve two problems: representation and matching [67]. In the followings, we would give a review on five well-known face recognition algorithms and they are the experts used in our committee machine. The algorithms are: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching, (4) Support Vector Machines and (5) Neural Networks. They use different approaches to represent and to match faces in recognition. We will give explanation on the way they represent and match the faces respectively.

Among the five methods , Eigenface [52] is the most popular one due to its effectiveness. It makes use of Principal Component Analysis (PCA) to find a feature space for projection of face images. A similar approach, Fisherface [5] was proposed later which makes use of Fisher's Linear Discriminant (FLD) instead of PCA. Apart from template matching approaches, Elastic Graph Matching (EGM) [39] was proposed to take into account the human facial features by extracting the features with Gabor wavelet transform. Recently, Support Vector Machines (SVM) [62] has gained a wider acceptance in face recognition. Based on the statistical theory by Vapnik, several SVM algorithms on face recognition [25], [12] were developed by researchers and were proved with impressive results. Neural Networks (NN) have been widely used for pattern classification in a variety of fields. In face recognition, Neural Networks were proven having high accuracy in recognizing human face.

2.2.1 Eigenface

Eigenface [52] was first proposed by Sirovich and Kirby in 1987 as an application of Principal Component Analysis (PCA). In this approach, faces are represented by a small number of eigenvectors which contain the major variations in the faces. They created approximate reconstruction of face images with a weighted sum of the eigenvectors obtained from PCA. Pentland and Turk refined the method by adding preprocessing and procedures of face detection in 1991 [60], [61]. They also used the eigenvectors, known as eigenfaces, for the reconstruction of faces.

At elementary level, the image of a face can be expressed as a one dimensional column vector of concatenated rows of pixels:

$$X = [x_1, x_2, \ldots, x_n]^T, \tag{2.1}$$

where n is the total number of pixels in the image. To compare two images, the simplest method is to compare the images pixel by pixel. However, n would be too large for such comparison. To cite an example, n would be 10000 for a small image with 100×100 pixels. The comparison is time-consuming and not efficient. Moreover, not all pixel values are important for comparison because most of them do not represent the characteristic of a face. Therefore, a dimensional reduction technique, Principal Component Analysis, is employed in Eigenface to keep the representation of image compact and efficient for comparison.

Principal Component Analysis

Principal Component Analysis, also known as the Karhunen-Loeve (KL) expansion in communication theory [37], [41] or the Hotelling transform in image processing [29], is a well-known approach for feature extraction. It is a method of identifying patterns of data, and expressing the data in a way to highlight their similarities and differences. The basic idea of PCA is to take advantages of the redundancy existing in the original set for representing the set in a more compact way, so that the dimension of the image can be greatly reduced. For any given number n of terms, the mean square error (MSE) between the approximation and the original pattern is minimal [24], [53].

Let $\{x_k, 1 \leq k \leq n\}$ be a random sequence of n-dimensional pattern vectors with covariance matrix C, which contains values of variance between the corresponding elements of the sequence. For any real symmetric matrix C, there exists an $n \times n$ unitary matrix W, which reduces C to its diagonal form Λ:

$$W^T C W = \Lambda, \tag{2.2}$$

or

$$CW = W\Lambda, \tag{2.3}$$

where W^T is the conjugate transpose of W. The matrix Λ contains the eigenvalues $\lambda_k, k = 1 \ldots n$ of C:

$$\lambda = Diag\{\lambda_1, \lambda_2, \ldots, \lambda_n\}. \tag{2.4}$$

Equation 2.3 is the set of eigenvalue equations:

$$Cw_k = \lambda_k w_k \qquad k = 1, \ldots, n, \tag{2.5}$$

where $\{\lambda_k\}$ and $\{w_k\}$ are the eigenvalues and eigenvectors of C respectively.

Original vector X can be transformed by:

$$Y = W^T X, \tag{2.6}$$

so that elements of the transformed sequence y_k are orthogonal. Equation 2.6 is known as KL Transform. The reverse transform of KL Transform is as follow:

$$X = WY = \sum_{k=1}^{n} y_k \phi_k. \tag{2.7}$$

From the set of eigenvectors found, the eigenvector with the highest eigenvalue is the principal component of the data set. Figure 2.2 shows a two dimensional data and the principal component of the data. Notice that the major variations of the data is on the principal component.

The KL transform packs the maximum average energy in $d \leq n$ samples of X. Therefore, the MSE of the approximation of the input data X, $\hat{X}$, using only d samples is minimized if $\hat{x_k}$ is reconstructed from the eigenvectors w_k with the d largest eigenvalues, i.e.:

$$\hat{x_k} = \sum_{k=1}^{d} y_k w_k, \tag{2.8}$$

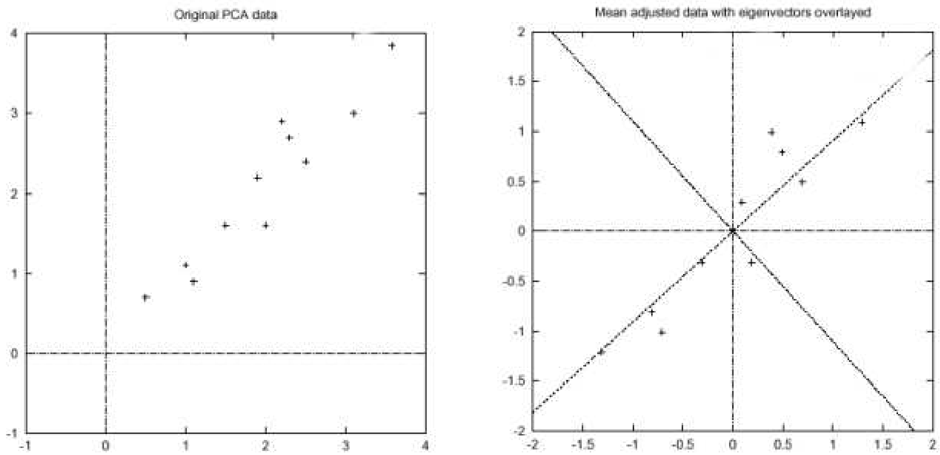

Figure 2.2: Original data (left) and principal component (right)

where eigenvalues λ_k corresponding to vectors w_k which satisfy

$$\lambda_1 \geq \lambda_2 \geq \ldots \lambda_d \geq \ldots \lambda_n. \tag{2.9}$$

We can reduce the dimension of the data in Figure 2.2 to one by using only the one principal component. Figure 2.3 shows the transformed data and the compressed data.

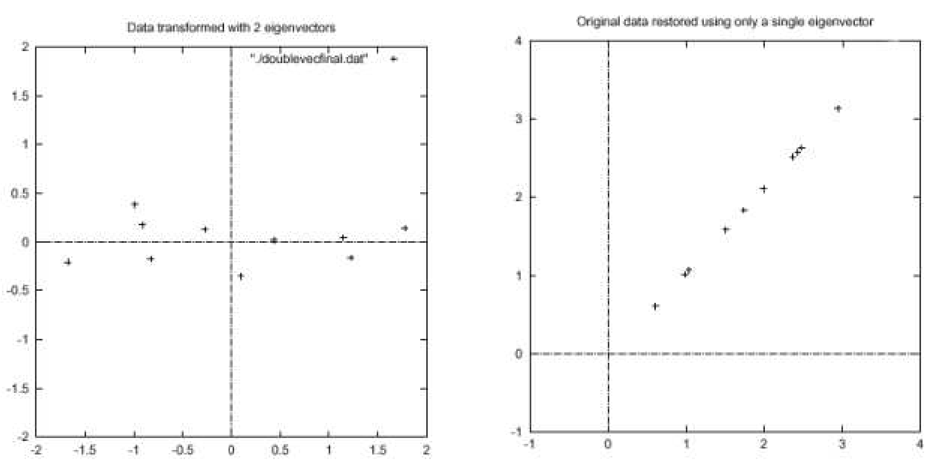

Figure 2.3: Transformed data (left) and compressed data (right)

Eigenface works by finding the eigenvectors and eigenvalues of the covariance matrix C from the training set m images $\{X_1, X_2, \ldots, X_m\}$. To find the covariance matrix, we have to first find the average face Ψ of the set which is defined by:

$$\Psi = \frac{1}{m} \sum_{i=1}^{m} X_i, \tag{2.10}$$

and then calculate the covariance matrix as follow:

$$C = \frac{1}{m} \sum_{i=1}^{m} (X_i - \Psi)(X_i - \Psi)^T. \tag{2.11}$$

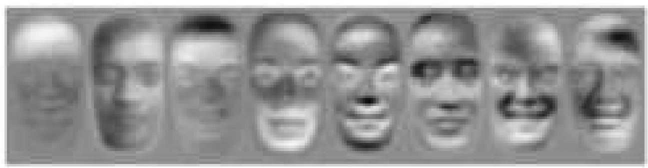

Figure 2.4: Standard eigenface

We can rank the eigenvectors (eigenface) with the associated eigenvalues according to the usefulness in characterizing the variation and then select d eigenvectors, say 50 eigenvectors, for the estimation of image. Figure 2.4 shows some eigenfaces. The eigenfaces form a linear transformation matrix W_{pca} to project any input images X to a lower dimensional vector space, known as the face space. The transformation matrix is chosen to maximize the determinant of the total scatter matrix S_T (the covariance matrix C) of the centered samples, i.e., to find the largest eigenvalue eigenvectors by:

$$W_{pca} = arg \max_{W} |W^T S_T W|, \tag{2.12}$$

and the transformation is defined as:

$$Y = W_{pca}(X - \Psi), \tag{2.13}$$

where $Y = [y_1, y_2, \ldots, y_d]^T$ is the feature vector of the image that contains weights y_k to describe the contribution of each eigenface in representing the input face image X. As d is much smaller than the number of pixels n ($50 \ll 10000$), comparison between images would be efficient. Recognition of face can be used only the most important features instead of all the pixel values of an image.

2.2.2 Fisherface

Fisherface [5] was suggested by Belhumeur *et al.* in 1997. It is similar to Eigenface that both methods are template matching method which makes use of projection of images into a feature space. However, Fisherface uses Fisher's Linear Discriminant (FLD) [16], [69], a class specific method, instead of Principal Component Analysis (PCA).

Fisher's Linear Discriminant

In PCA, the projection W_{pca} is best for reconstruction of images from a low dimensional basis. However, the projection does not make use of between-class variance. The projection may not be optimal from discrimination for different classes. In FLD, the projection maximizes the ratio of the between-class scatter to that of the within-class scatter. It tries to "shape" the scatter in order to make it more reliable for classification [5]. Let the between-class scatter matrix be defined as [14]:

$$S_B = \sum_{k=1}^{c} m_i(\Psi_k - \Psi)(\Psi_k - \Psi)^T, \tag{2.14}$$

where c is the number of classes and m_i is the number of samples in class T_i. The within-class scatter matrix is defined as:

$$S_W = \sum_{i=1}^{c} \sum_{X_k \epsilon C_i} (X_k - \Psi_i)(X_k - \Psi_i)^T, \tag{2.15}$$

where Ψ_i is the mean image of class C_i. The optimal projection W_{opt} is chosen as the matrix with orthonormal columns, which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix. It is defined as follows:

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}. \tag{2.16}$$

As there are at most $c-1$ non-zero generalized eigenvalues, the upper bound on the number of eigenvalues d is $c-1$, where c is the number of classes.

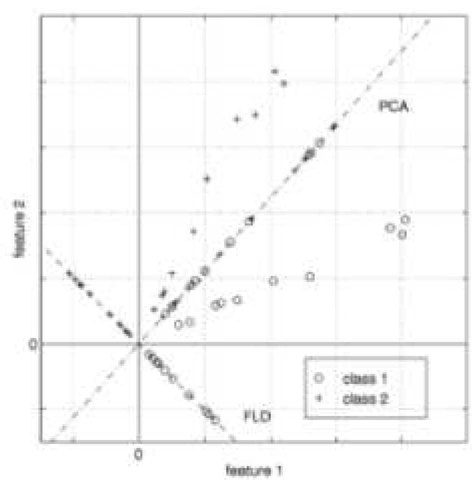

Figure 2.5: Comparison of PCA and FLD

Figure 2.5 is a comparison of PCA and FLD for a two-class problem in which the samples from each class are randomly perturbed in a direction perpendicular to a linear subspace. In the figure, both PCA and FLD reduce the dimension by projecting the points from $2-D$ to $1-D$. Note that when comparing the projections in the figure, PCA smears the class together that the samples in the projected space are no longer linearly separable. Although PCA achieves larger total scatter, FLD achieves greater between-class scatter and thus the classification is simplified.

In face recognition problem, there is a difficulty that the within-class scatter matrix $S_W \in \Re^{nxn}$ is always singular. Peter in [5] proposed the Fisherface method which combines the PCA and FLD to overcome the problem. Image set data are first projected to a lower dimensional space by PCA so that the resulting within-class scatter matrix S_W is non-singular. FLD is then applied to reduce the dimension to $c-1$. The optimal projection is defined by:

$$W_{opt}^T = W_{fld}^T W_{pca}^T, \tag{2.17}$$

where

$$W_{fld} = arg \max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}. \tag{2.18}$$

Fisherface is good at handling variation in lighting and expression as it optimizes the classification ability of the projected samples. Therefore, it achieves high recognition accuracy in general cases. A comparison of PCA and FLD in face recognition is provided in [70] which demonstrates the classification ability of FLD.

2.2.3 Elastic Graph Matching

Elastic Graph Matching has been proposed since 1989. Buhmann *et al.* first proposed object recognition using hierarchically labelled graphs [8], [39] which is applied to face recognition. It is a simple implementation of the dynamical link architecture. In the matching, gray level images are represented by a resolution hierarchy of local Gabor filters. Each facial feature is extracted by Gabor Wavelet Transformation on the fiducial points such as eyes, nose and mouth as a jet. A face is represented by an image graph G consisting of N nodes of jets and any test image graph G^I is compared to all modal graphs G^M by a cost function. The identity of the lowest cost modal graph is the recognized identity.

Later, Wiskott [64] presented a system which is the implementation of the Elastic Graph Matching in 1997. In the system, he proposed an image graph extraction approach named the bunch graph which is constructed from a small set of sample image graphs. The idea for the bunch graph is to find the fiducial points in new faces with a general representation which covers a wide range of possible variations in the appearance of faces.

Gabor Wavelet Transformation

In Elastic Graph Matching, the representation of facial feature is based on Gabor Wavelet Transformation. Gabor wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function. The Gabor wavelet is used because it can extract the human face feature well. The wavelet filters can be generated by a family of Gabor kernels φ which is defined as

$$\varphi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp(\frac{-k_j^2 x^2}{2\sigma^2})[\exp(i\vec{k}_j\vec{x}) - \exp(\frac{-\sigma^2}{2})], \qquad (2.19)$$

The kernels are in the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Gaussian envelope function. In [64], a discrete set of 5 different frequencies, index $\nu = 0, 1, \ldots, 4$ and 8 orientations, index $\mu = 0, 1, \ldots, 7$. The wave vector is formulated by:

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\varphi_\mu \\ k_\nu \sin\varphi_\mu \end{pmatrix}, k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \varphi_\mu = \mu\frac{\pi}{8}, \tag{2.20}$$

with index $j = \mu + 8\nu$ and $\sigma = 2\pi$. Figure 2.6 shows the 40 Gabor filters used in the transformation, which is composed of 5 different frequencies and 8 orientations.

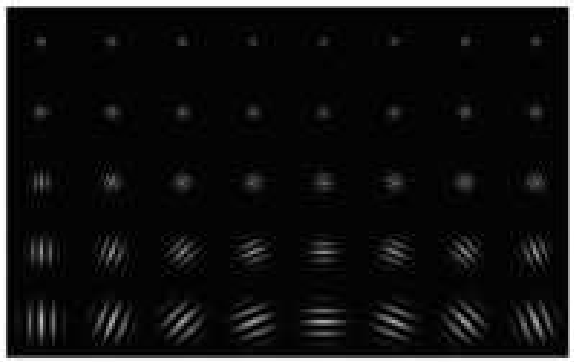

Figure 2.6: Gabor filter of 5 frequencies and 8 orientations

Gabor wavelet transformation is a process of convolution of the image with Gabor filters. As shown in Figure 2.7, a jet J describes a small patch of gray values in the image X around a given pixel. It is defined as the set J_i of 40 complex coefficients obtained from one image point and can be written as

$$J_i = a_j exp(i\varphi_j), \tag{2.21}$$

with magnitudes $a_j(\vec{x})$, which slowly varies with position, and phase $\varphi_j(x)$, which rotates at a rate approximately determined by the spatial frequency or wave vector $\vec{k}_j$ of the kernels. A collection of jets, together with the relative location of the jets form an image graph in the right.

To recognize a new test image, we first extract the test image graph of the image and then compare the image graph to all modal graphs with the cost function, which is defined as

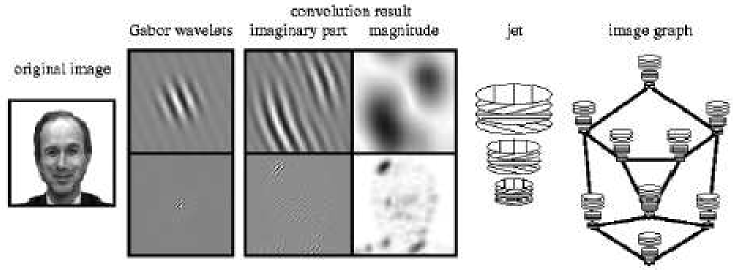

Figure 2.7: Gabor wavelet transformation

$$C_{total}(G^I, G^M) = \lambda S_e(G^I, G^M) - S_v(G^I, G^M), \qquad (2.22)$$

where S_e is edge comparison function and S_v is vertex similarity function. λ is rigidity coefficient to control the relative importance of S_e and S_v in the matching, S_e is defined as:

$$S_e(G^I, G^M) = \frac{1}{E}\sum_e(\Delta\vec{x}_e^I - \Delta, \vec{x}_e^M)^2, \qquad (2.23)$$

where E is the number of edges and $\Delta\vec{x}_e$ are the distance vectors used as labels at edge e. S_v is defined as:

$$S_v(G^I, G^M) = \frac{1}{N}\sum_n \frac{J_n^I \cdot J_n^M}{J_n^I J_n^M}, \qquad (2.24)$$

where N is the number of nodes and J_n are the jets at nodes n. With the cost function, the training set image with minimum cost would be the best match in comparison.

Elastic Graph Matching is well-known in the field due to its robustness in face recognition. As it applies local feature extraction which captures the local facial feature instead of global one like Eigenface and Fisher, it is more adaptive to different variations.

2.2.4 Support Vector Machines

Support Vector Machines [62], [10] were invented by Vapnik *et al.* in the late seventies, which is based on Structural Risk Minimization principle from statistical learning theory. It aims to minimize an upper bound on the expected generalization error. Recently, SVM has been applied to face detection and face recognition. Osuna E. used SVM on face detection in 1997 [48]. Guo and Kim used SVM on face recognition in 2000 and 2001 respectively [25], [38]. Dihua used SVM on facial component extraction and face recognition in 2002 [12].

For classification, SVM takes the training set images with identification:

$$S_n = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \tag{2.25}$$

as input of dimension n from probability distribution $P(x, y)$ where $x_i \in \Re^n$ represents the feature vector of image i and $y_i \in \{-1, +1\}$ represents the class label [9]. For linearly separable data, SVM looks for a separating hyperplane which separates the data with the largest margin. The decision rules is defined as:

$$h(x) = \begin{cases} +1 & \text{if } w * x + b \geq 0 \\ -1 & \text{otherwise,} \end{cases} \tag{2.26}$$

where w is normal to the hyperplane and b is a threshold. To find w and b, we need to solve a quadratic programming problem. To minimize the Lagrangian (the primal-problem) with Lagrange multipliers $\alpha_i, i = 1, \ldots, l$:

$$L_p \equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i (x_i \cdot w + b) + \sum_{i=1}^{l} \alpha_i. \tag{2.27}$$

The Lagrangian can be expressed into duel problem, which is to maximize:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j, \tag{2.28}$$

with conditions:

$$w = \sum_i \alpha_i y_i x_i, \tag{2.29}$$

$$\sum_i \alpha_i y_i = 0. \tag{2.30}$$

As shown in Figure 2.8, the samples closest to the hyperplane h are called Support Vectors (circled in the figure). The distance between the support vectors is called margin. According to which sides a feature vector x_i lies on the hyperplane, the sample is classified to either class $+1$ or -1.

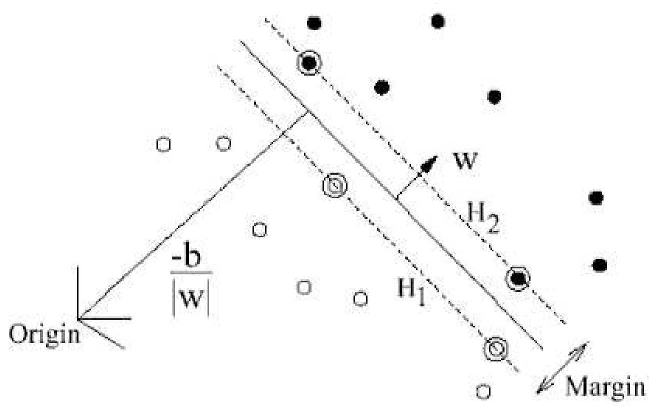

Figure 2.8: Separating hyperplane

For linearly non-separable data, Boser *et al.* suggested to project the data into a higher order space, possibly of infinite dimension where a hyperplane may exists by mapping the data:

$$x\epsilon\Re^I \;\mapsto\; \Phi(x)\epsilon\Re^h, \tag{2.31}$$

with kernel function K:

$$K(u,v) = \Phi(u) \cdot \Phi(v). \tag{2.32}$$

The kernel functions still keep the SVM solvable. We list some classical kernels in Table 2.1 [4]. They differ in the way of mapping the data from low dimension to high dimension.

Table 2.1: Common SVM kernels

Kernels	**Formula**
Linear	$K(u,v) = u \cdot v$
Sigmoid	$K(u,v) = tanh(au \cdot v + b)$
Polynomial	$K(u,v) = (1 + u \cdot v)^d$
RBF	$K(u,v) = exp(-a \parallel u - v \parallel^2)$
Exponential RBF	$K(u,v) = exp(-a \parallel u - v \parallel)$

2.2.5 Neural Networks

Artificial Neural Networks (simply refers to Neural Networks) has been widely used for pattern classification in a variety of fields including signal processing, speech recognition, image recognition, and character recognition. For face recognition, several face recognition systems have been proposed. Aizenberg proposed neural networks with multi-valued neuron [1]. Meng proposed face recognition using Radial Basis Function (RBF) neural networks [44]. They are proven having high accuracy in recognizing human face.

The idea of neural networks is to mimic human brain architecture so that computer can solve the same type of problems as human brain. It consists of the following elements:

- Processing units (artificial neurons)
- Weighted interconnection (neurons connections)

- Activation rule to propagate signals through network
- Learning rule to specify how weights are adjusted

Neural networks is an information processing system composed of interconnected network of artificial neurons. Each neuron is linked to certain of its neighbors with varying weights. The neural network is trained to learn from experience to solve different problems. In other words, neural network is a cellular system that can acquire, store and utilize experiential knowledge [67].

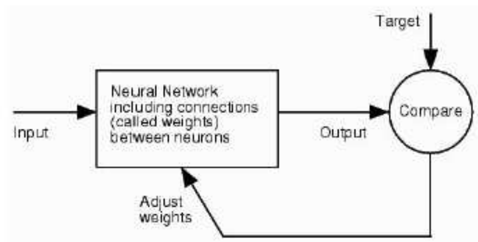

Figure 2.9: Neural network training

As shown in Figure 2.9, each neuron calculates its activity locally on the basis of the activities of the cells to which it is connected. The weights are changed according to some transfer functions that determine the cell's output, given the input. The main objective of neural network training is to minimize the errors at the output. An efficient method commonly used is the back-propagation algorithm. The principal advantages of back-propagation are simplicity and reasonable training speed. It is well suited to pattern recognition problem.

The back-propagation algorithm [11] is given in Figure 2.10. The training is based on a gradient descent in error space, where the error is defined as:

$$E = \sum_p E_p, \tag{2.33}$$

1. Initialize all synaptic weights w to small random values.
2. Present an input from the class of learning examples (input/output pattern) and calculate the actual outputs.
3. Specify the desired outputs and evaluate the local error ε for all layers.
4. Adjust the synaptic weights to minimize ε.
5. Present another input pattern corresponding to the next learning example (repeat step 2).

Figure 2.10: Back-propagation algorithm

where E_p is the error of each input pattern and

$$E_p = \frac{1}{2}\sum_i (Target_i - Input_i)^2. \tag{2.34}$$

We can adjust the weights in step 4 corresponding to the gradient of error:

$$\Delta w = -\eta \nabla E, \tag{2.35}$$

where η is a constant scaling factor defining the step-size of training. Once the neural network is trained, it can classify any incoming feature vectors effectively and accurately.

2.3 Commercial System and Applications

Several face recognition commercial products are available in the market. We will give a brief introduction on currently four market leaders. However, for proprietary concerns, their techniques are not published though their systems may have been initially based on the published algorithms such as Eigenface and Elastic Graph Matching etc. We can only get the systems' information from the advertisement and their websites [15], [71], [59] and [63].

2.3.1 FaceIT

FaceIT [15] (from identix) is an award-winning facial recognition software engine that allows computers to rapidly and accurately detect and recognize faces. It is the top performer in the Facial Recognition Vendor Test (FRVT) in 2002. FaceIT combines all technologies in face recognition including detection, tracking, segmentation and recognition.

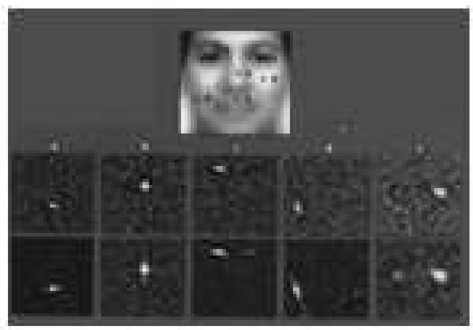

Figure 2.11: LFA and facial representation

FaceIT uses Local Feature Analysis (LFA), a mathematical technique developed by Visionics Corporation, to represent facial images in terms of local statistically derived building blocks (shown in Figure 2.11). It maps an individual's identity into a complex mathematical template, named as faceprint, to match and compare with others in the face database.

2.3.2 ZN-Face

ZN-Face [71] (from ZN Vision Technologies AG) uses Elastic Graph Matching to extract facial feature. The integrated ZN-Face camera takes a picture of the person's face in front of the console and converts it into the biometric template.

For calculating the facial information, an elastic grid, is put over the face to extracts around 700 characteristic features as

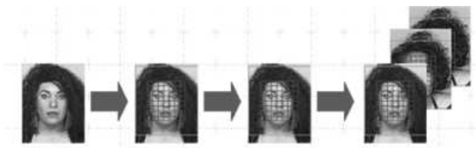

Figure 2.12: Graph matching technology

shown in Figure 2.12. At every node within the graph, a feature vector is calculated from the specific local information as shown in Figure 2.13. The facial similarities between the live feature and the graph model are established on the basis of the weighted sum of node similarities with a total of 1, 700 facial features used.

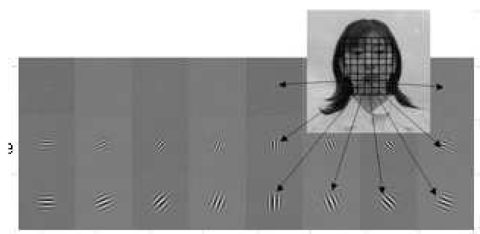

Figure 2.13: Wavelet features extracted in face image

2.3.3 TrueFace

TrueFace [59] (from Miros, Inc) uses neural network technology, which learns how to compare faces from its own experience. It is used by the MR. Payroll Corp. system in its checking-cashing machines and is popular with financial organizations such as Bank of America, Kroger and Wells Fargo [19].

It works best for accommodating the variations of the face image such as variations in head orientations and in lighting conditions. It uses features from every part of the face to determine the similarity between two face images, not just the distances and angles between the eyes, nose and mouth. However, the details of how the features are extracted are not published.

2.3.4 Viisage

FacePASS [63] (from Viisage Technology, Inc), the Viisage's patented face-recognition (FR) technology, is originally developed at the Massachusetts Institute of Technology (MIT). It uses eigenfaces which map characteristics of a person's face into a multi-dimensional face space. Using a sophisticated algorithm based on PCA developed at the MIT's Media Lab, the system translates the characteristics of a face into a eigenface for both identification and verification.

Figure 2.14: FacePASS recognition

To conclude, we have reviewed several important algorithms related to our research in this chapter. We have given a brief overview of committee machine and have explained the five major face recognition algorithms in details. Finally, we provide several face recognition products in the market to show the current technologies.

□ **End of chapter.**

Chapter 3

Static Structure

3.1 Introduction

Currently, there are numerous face recognition algorithms proposed by researchers. However, there is no unique face recognition algorithm that can handle all kinds of variations such as different poses, illumination conditions, expressions and glasses. Besides, accuracy of current algorithms is still not satisfactory. Their performances are still not better than other biometrics such as fingerprint and iris scanning. Combining results of algorithms in various face recognition fields is a key to success. Therefore, we propose a framework that integrates various face recognition algorithms as experts in a committee machine called Face Recognition Committee Machine (FRCM).

In previously work, Gutta *et al.* [28] proposed the use of neural network ensemble to solve the facial analysis problem. He used several neural network classifiers (radial basis functions, RBF) and inductive decision tree (DT) to combine the outputs from different networks. Huang *et al.* [32] built view-specific eigenfaces and used the extracted features to train different view-specific neural networks for pose invariant face recognition.

All the above work used homogeneous experts in committee machine, i.e., they used experts of same nature (either RBF or neural networks). The results obtained from homogeneous experts are highly co-related and thus committee machine may not have great improvement on accuracy. Instead, we employ five heterogeneous experts in FRCM, they are: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching, (4) Support Vector Machines and (5) Neural Networks, which are reviewed in Chapter 2. We are the first to employ heterogeneous experts in committee machine on face recognition. All the experts are different in nature. Eigenface and Fisherface are template matching approaches; EGM is a graph matching approach; SVM and Neural Networks are machine learning approaches. As these experts are different in the representation of face feature and the classification of the feature, their recognition results are independent to each others. Therefore, committee machine can help to achieve better accuracy.

We name our FRCM with Static Structure as Static Face Recognition Committee Machine (SFRCM). In the following sections, we give a brief introduction on SFRCM, and then describe the three important elements for each expert of SFRCM: (1) result, (2) confidence and (3) weight. Finally, we explain the ensemble of the experts' results.

3.2 Architecture

SFRCM consists of the five experts for face recognition, and a voting machine to combine the outputs of the experts. Figure 3.1 shows an overview of SFRCM. To recognize an input face image, the image is given to the five experts for recognition individually. We define three major elements in SFRCM: (1) result r, (2) confidence c and (3) weight w. Each expert gives its recognition result, i.e., the identity/acceptance of the face image, together with the confidence of the expert on that result, to the voting machine. The voting machine collects all the results and confidences from the experts, and then assembles them by

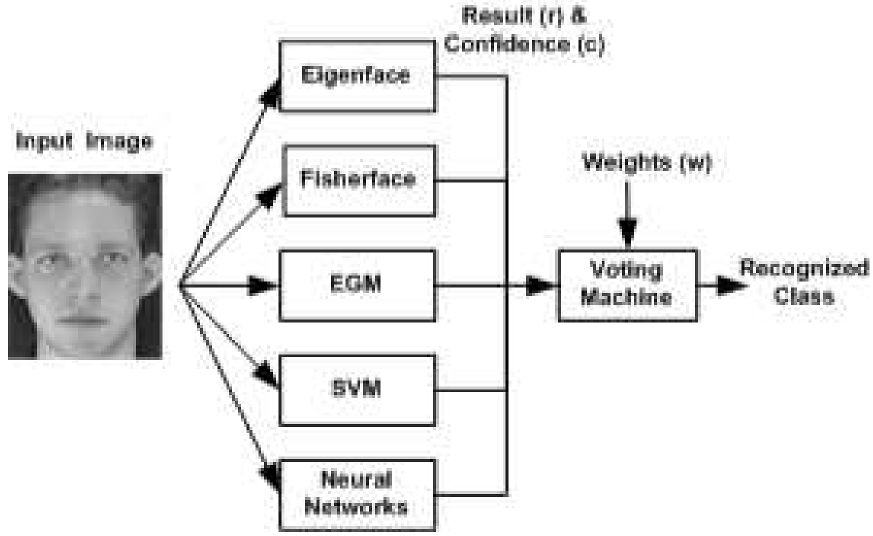

Figure 3.1: SFRCM architecture

ensemble averaging. In the ensemble, weights (derived from the performance of the experts) are given to the experts, which are fixed for all images in this Static Structure.

3.3 Result and Confidence

Due to heterogeneous nature of the experts, it is tough to assemble their results and confidences, especially to normalize their confidences. To solve this problem, we have to ensure that:

- Each expert gives one recognition result for an input image.
- Confidence of result should be normalized and within the range [0, 1].

Face recognition consists of two major modes: face identification and face verification. Identification classifies the input image while verification determines the acceptance of the input image's identity. Verification differs from identification that a threshold for each expert is necessary to determine the acceptance of that claimed identity. In the followings, we define separate methods for both modes to find the result and confidences for each expert in SFRCM .

3.3.1 Eigenface, Fisherface, EGM

As Eigenface, Fisherface and EGM are template matching approaches that each enrollment image (in compressed form or graph representation) is stored in the face database as templates for recognition, we can apply similar techniques to find their results and confidences.

Identification

We employ K nearest neighbor classifier for identification. It labels an unknown input image with the majority of the K nearest neighbors. We choose five nearest neighbor classifier in SFRCM. Five nearest training set templates in the database with shortest Euclidean distance for Eigenface/Fisherface, or highest graph similarity for EGM with the input image are selected in the majority voting. We define vote v as a vector of size J for J classes in the voting. Among the five neighbors, a vote is given to the class of each neighbor. The final result of expert i, r_i, in identification is defined as the class j with the highest votes v in J classes:

$$r_i = \arg\max_j (v(j)), \tag{3.1}$$

and the confidence of the result is defined as the number of votes that the final class obtained divided by K:

$$c_i = \frac{v(r_i)}{K}. \tag{3.2}$$

Verification

For verification, an input image with a given identity is compared to the templates of that identity in the face database. K nearest neighbor classifier is no longer used, instead, majority voting is used to determine the acceptance of the identity. The number of templates of that claimed identity whose Euclidean distance/graph similarity is within the selected threshold $N_{threshold}$ is counted. If $N_{threshold}$ is larger than half of the total number of templates N_{total}, the expert authorizes the user's identity; otherwise the expert rejects the identity. The result is defined as:

$$r_i = \begin{cases} 1 & \text{if } N_{threshold} \geq \frac{N_{total}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

and the confidence c_i is defined as:

$$c_i = \frac{N_{threshold}}{N_{total}}. \tag{3.4}$$

3.3.2 SVM

As SVM was originally developed for two-class classification, it can only separate two classes instead of multiple classes. We use "one-against-one" approach to recognize multiple classes. We construct ${}_JC_2$ (i.e., $\frac{J(J-1)}{2}$) SVMs to recognize an input image in J classes. Each one is trained with two different classes among the J classes.

Identification

We use majority voting for all the SVMs in identification. The image is tested against each SVM. The class j with the highest votes in all SVMs is selected as the recognition result r_i. The confidence is defined as:

$$c_i = \frac{v(r_i)}{J-1}, \tag{3.5}$$

where $J-1$ is the maximum number of votes a class could obtained.

Verification

Similar to identification, SVM determines the acceptance of the face pattern by matching its identification result of the input image and the user's identity. SVM authorizes the identity if it matches the identification result; otherwise SVM rejects that identity. The confidence of verification is defined as the same as identification (Equation 3.5).

3.3.3 Neural Networks

For Neural Networks, we choose a binary vector of size J for the target representation of J different classes. The target class is set to 1 and the rest are set to 0 in the target representation. We use back-propagation to train the Neural Networks with the target vector. Figure 3.2 shows n target vectors of J classes. In recognition, an input image is given to the network for prediction and an output vector of the prediction results is provided.

Identification

For identification, the class j with the largest output value o_j in the output vector is chosen as the identification result:

$$r_i = \arg\max_j(o_j), \tag{3.6}$$

and this output value is chosen as the confidence on the result.

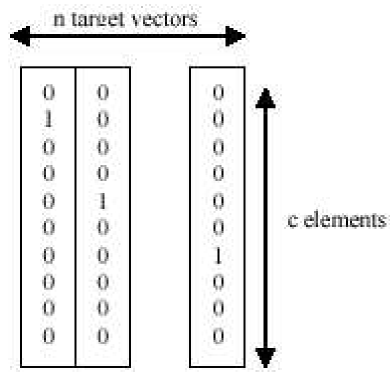

Figure 3.2: Neural networks target representation

Verification

For verification, the value of the claimed identity in the output vector is used to determine the acceptance. If the value is larger than 0.5, the Neural Network accepts the identity and rejects otherwise. The confidence is chosen as the output value.

$$r_i = \begin{cases} 1 & \text{if } o_j \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

3.4 Weight

In SFRCM, weights are determined by the average performance of the experts, which are evaluated in the experimental testing for individual algorithms on four face databases: ORL, Yale, AR and HRL face database. The details of the experiments and the results will be given in Chapter 6 (Table 6.8). The performance p_i of expert i is defined as:

$$p_i = \frac{n_i}{t_i}, \tag{3.8}$$

where n_i and t_i are the total number of correct recognition and trail in the experiments for expert i respectively. In defining the weights of SFRCM, we notice that if we simply use the performance as weight, the difference in performance of the experts makes little changes in the overall ensemble results. Therefore, we use an exponential function to amplify the difference and thus the weight of an expert i (w_i) is defined as:

$$w_i = \exp(\alpha p_i), \tag{3.9}$$

where α is a scaling factor. As $0 \leq p \leq 1$ and exponential function $\exp(x)$ grows slowly for $0 \leq x \leq 1$, the scaling factor α is used to scale up the weight of each expert such that $\alpha p_i > 1$. The weights of the experts are then normalized to ensure that $0 \leq \hat{w}_i \leq 1$ and summed to one by:

$$\hat{w}_i = \frac{w_i}{\sum_{i=1}^{5} w_i}. \tag{3.10}$$

As the weights $\hat{w}$ are scaled in the exponential function, the difference of the normalized weights between the experts is significant even for little difference in the experts' performance.

3.5 Voting Machine

In the voting machine, we apply ensemble averaging approach to assemble the results obtained from the experts. The reason for using ensemble averaging instead of majority voting is that it can achieve better accuracy [58]. In the ensemble, we take two major considerations:

- Low confidence result should not contribute much in the final result.
- Low performance expert should give lower weight on its result.

Based on the above considerations, we apply (1) confidence c as a weighted vote in the voting machine, to avoid low confidence result of an expert that affects the ensemble result significantly. Apart from confidence, we also give (2) weight w to expert's result in the voting machine. It is possible that expert has high confidence on its result, however, its result is wrong in most cases. The use of weight further reduces this kind of high confidence result to affect the ensemble result. The voting machine assembles the results by calculating the score s of each class j as follows:

$$s_j = \sum_{i=1}^{5} \hat{w}_i * c_i, \forall j \in r_i. \tag{3.11}$$

With this definition of the score, only expert with high performance on average and high confidence on its result would contribute most in the final score. The final decision $\hat{r}$ is defined as the class j with the highest score:

$$\hat{r} = \arg\max_j (s_j). \tag{3.12}$$

To conclude, we have described the Static Face Recognition Committee Machine and have explained its architecture in this chapter. We have also defined the confidence, result and weight of each expert in SFRCM. In SFRCM, input image is not involved in the determination of weights. Therefore, the experts apply same weights for all input images. In next chapter, we state two problems of SFRCM, and propose another committee machine which is based on SFRCM to solve these problems.

□ **End of chapter.**

Chapter 4

Dynamic Structure

4.1 Introduction

In Chapter 3, we propose a Static Face Recognition Committee Machine. In its voting machine, input image is not involved in the determination of weight for each expert. The experts have same weights for all input images once after training. However, we find that this SFRCM has two major drawbacks:

- Fixed weights under all situations: The weights of the experts are fixed no matter which images are given to SFRCM. However, algorithms may perform dramatically different under various situations. For example, global comparison methods like Eigenface and Fisherface achieve good results under normal situation but perform badly under strong lighting variation; EGM has better performance with different lighting and pose variation. It is undesirable to fix the weights for the experts under all kinds of variations. Instead, we should give different weights for each expert under various variations. For example, we should give higher weight for EGM and lower weight for Eigenface in strong light environment. At current state, we do not include an expert specialized on certain variations such as Illumination Cone [22], [20], [21] for illumination variation. This kind of algorithm has excellent performance under specific environment but performs poorly under normal situation. It is necessary to give higher weight for the expert in its specialized situation and lower weight vice versa. Therefore, different weights for experts in various situations are necessary to build a committee machine for handling all kinds of situations.

- No update mechanism for weights: The weights of the experts can not be updated once the system is trained. In SFRCM, the weights are determined by the performances of the experts in preliminary experiments. In addition, SFRCM does not provide a mechanism to update the weights after system training. However, the performance of the experts may vary under different situations. It is necessary to have a mechanism to update the weights accordingly, which is particularly important for a real-time recognition system because we would like to keep training the system at any time for the best performance.

To overcome these drawbacks, we develop another committee machine, named Dynamic Face Recognition Committee Machine (DFRCM), which can dynamically select different weights according to an input image. Besides, we have introduced an important component in DFRCM, a gating network, to determine the situations of the input image and to give the corresponding weights to the experts. Moreover, we have included a feedback mechanism in DFRCM such that the weights can be updated anytime after system training.

4.2 Architecture

The architecture of DFRCM is similar to that of SFRCM. It is composed of the five face recognition experts and a voting

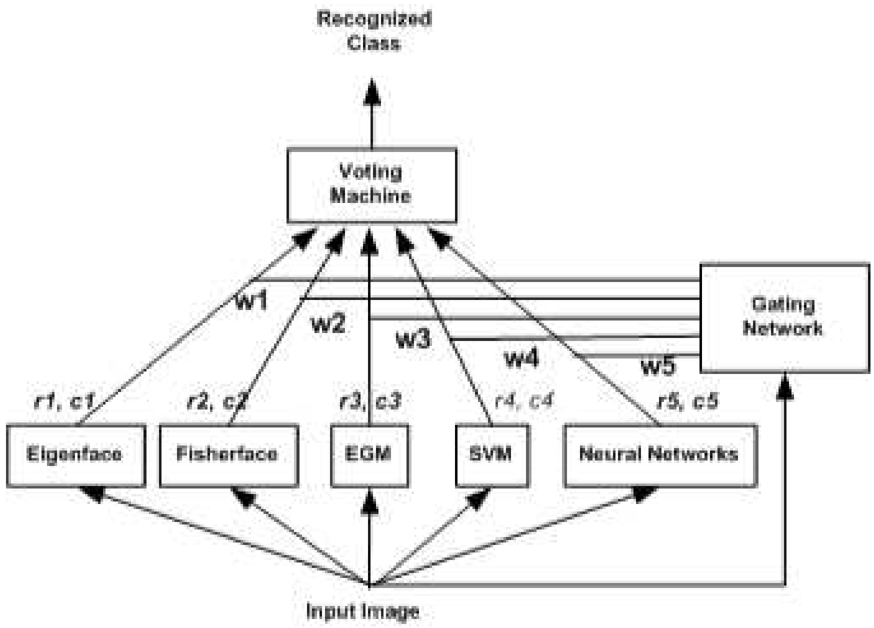

Figure 4.1: DFRCM architecture

machine. The definition of result, confidence and weight for the experts are applied to DFRCM. Figure 4.1 shows the overall DFRCM architecture. Similar to SFRCM, an input image is given to the five experts and each expert reports its result and confidence for the ensemble. Note that the image is also given to the gating network to determine the image's current situation, i.e., input is involved in the determination of the weights. The gating network gives corresponding weights of different experts to the voting machine according to the image's situation. Then, the voting machine collects all the results, confidence and weights and calculates the score of each class. It takes the class with maximum score as the final recognition result.

4.3 Gating Network

The gating network is consisted of a neural network that accepts an input image and determines suitable weights for the experts.

It first figures out the current situation of the input image and then assigns specific weights of that image's situation to the experts. For example, the gating network accepts an input image and determines that the image is taken under strong lighting condition. It will then assign the lighting condition's weights to the experts. At current status, we use several face databases [23] [43] including faces with various scale, rotation, illumination to model images under different situations. Instead of using a single set of average performance in SFRCM, we train the experts with these databases and store their performances on each database separately. Table 4.1 shows the databases used, the situation of the images taken and the approximate number of images for each database.

Table 4.1: List of face databases

Face Database	Situations	Number of images
ORL	Normal	400
Yale	Various expressions	165
AR Face	Various occlusions	2500
Umist	Various poses	560
HRL	Various lightings	1370
Feret	Most variations	1200

In the training, we select 50 images from each database randomly as the training set data to train the gating network. The dimension of the training set images is first reduced by PCA to 50 for efficient training and recognition. Similar to the Neural Networks training method mentioned before, we use a binary vector of size 6 for the target representation. Once the gating network is trained, it can identify any images and can determine the images 's weights accordingly.

4.4 Feedback Mechanism

For both SFRCM and DFRCM, each expert is trained independently on different face databases in the training phase. For SFRCM, the expert's performance, a measure of the weight for the expert, is the overall performance of the expert in all the face databases. On contrary, the performance is tested for different face databases and is stored separately for DFRCM. The performance p_i for expert i on database j is defined as:

$$p_{i,j} = \frac{n_{i,j}}{t_{i,j}}, \tag{4.1}$$

where $n_{i,j}$ is the total number of correct recognition and $t_{i,j}$ is the total number of trail for expert i on face database j.

In DFRCM, we propose a feedback mechanism to keep updating the performance $p_{i,j}$. The overall mechanism for the training, testing and recognition phase of DFRCM is given in Figure 4.2.

1. Initialize $n_{i,j}$ and $t_{i,j}$ to 0 $\forall i, j$.
2. Train each expert i on different database j.
3. while TESTING
   (a) Determine j for each test image.
   (b) Recognize the image in each expert i.
   (c) if $t_{i,j}! = 0$ then calculate $p_{i,j}$,
   (d) else Set $p_{i,j} = 0$.
   (e) Calculate $w_{i,j}$.
   (f) Determine ensemble result.
   (g) if FEEDBACK then update $n_{i,j}$ and $t_{i,j}$.
4. end while

Figure 4.2: Overall feedback mechanism

The total number of correct recognitions and trails for all experts are first initialized to 0 on each database. In the training phase, we train the experts on each database using the training set data. In the testing phase, which list of images and identities are given, the gating network determines the database j of the test image. Each expert recognizes the image and then gives its result and confidence. If the total number of trail for the experts is not 0, we calculate the performance $p_{i,j}$. Otherwise, we set the performance to 0. The performance is normalized to obtain the weight $w_{i,j}$ for each expert. We can determine the ensemble result with the given weights, results and confidences from the gating network and experts in the voting machine. We update the total number of correct recognition and trail for the current testing if feedback is required. In the recognition phase, the procedure is similar to the testing phase except that we do not have feedback because the correct result for the test image is unavailable.

In conclusion, we have described Dynamic Face Recognition Committee Machine (DFRCM) in this chapter. We point out two problems of SFRCM and provide solutions to these problems, i.e., use of dynamic structure such that input image is involved in the determination of the weights. We propose the use of gating network to assign the weights for the experts. Besides, we propose the feedback mechanism to provide a way for updating the weights of the experts after system training. The two solutions help DFRCM to recognize images under different environments with greater improvement.

□ **End of chapter.**

Chapter 5

Face Recognition System

5.1 Introduction

We describe our implementation of FRCM, face recognition system, in this chapter. The system integrates face detection and face recognition techniques to provide real-time face recognition. It consists of two main modules: (1) face detection module and (2) face recognition module. As face detection is another major research area, we would only give a brief introduction on the face detection module and mainly focus on the face recognition module in this thesis. We apply FRCM in the system to provide robust, reliable face recognition. However, there are two major problems of committee machine: time overhead and storage overhead. Committee machine take more time and storage than using only one algorithm in recognition, which grow linearly with the number of expert used. Besides,face recognition application on small hand-held device like PDA and mobile phone with small computational capability and storage is not practical for current technology. We therefore propose a distributed framework for the system to tackle these time and storage overhead problems, and more importantly, to enable face recognition on small hand-held device possible.

5.2 System Architecture

We provide architecture and implementation details of the system in this section. As we mainly focus on face recognition in this thesis, background information and implementation of face detection would not be explained in details. Instead, we only give a brief introduction of the face detection module.

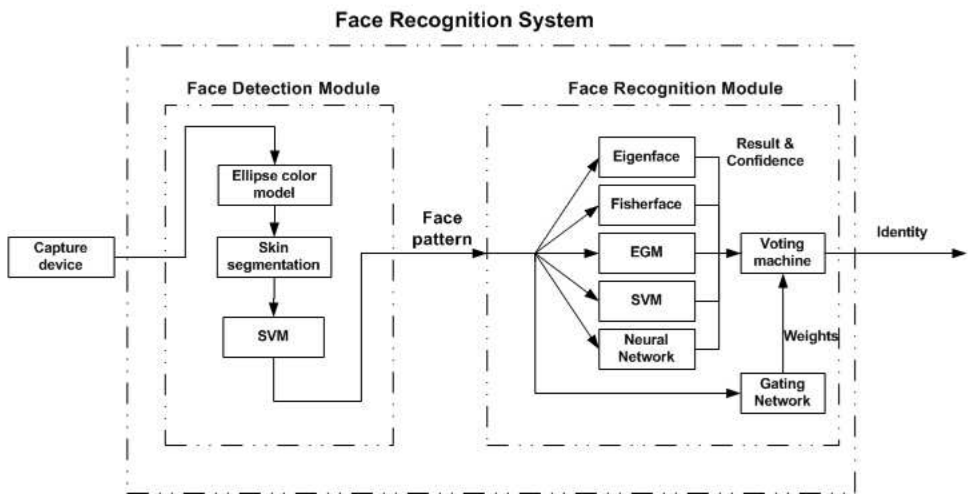

Figure 5.1: Overall system architecture

Figure 5.1 shows the architecture of the face recognition system. The system is composed of two main modules: face detection module and face recognition module. The detection module receives video data of a user from capture device, and then starts face tracking according to the skin color. Any possible face regions in the video are extracted for validation. SVM is then applied to determine whether the face region contains face pattern or not. The face pattern found acts as the interface between the two modules. The face recognition module recognizes the face pattern by the FRCM, which consists of the five face recognition experts together with a gating network and a voting machine, and reports the final identity of the user.

5.2.1 Face Detection Module

Face detection is often the first step in applications such as video surveillance, image database management and especially face recognition. Face pattern from video camera, i.e., the capture device, is the key input for the face recognition. In our implementation, the face detection module consists of three submodules: (1) tracking, (2) skin segmentation and (3) face validation. The tracking module applies condensation algorithm [33] to track the motion of the user according to his skin color. The skin segmentation module extracts any possible skin color candidates and then passes the candidates to the face validation module, which checks whether the candidates contain face pattern or not. A detail survey of face detection from Yang *et. al* can be found at [66] and the approach of detecting faces in color images is given at [31].

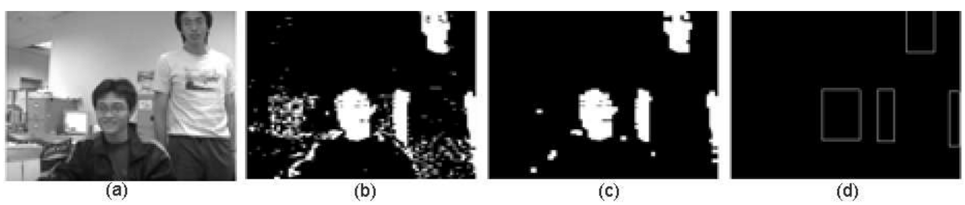

Figure 5.2: Skin segmentation process

Once the user is ready for recognition, the video camera captures a color still image. Figure 5.2 shows the overall skin segmentation process. The still image from the capture device is first transformed into skin color model that is most suitable for face detection (a). The image is converted into a binary image according to the skin color (b). Morphological operation is applied to reduce any noisy components (c). The face candidates found are then passed to the validation module that is a SVM in the implementation (d). It is well suitable for the validation as it can test the candidate efficiently and accurately [57].

The face pattern found from the candidates is then histogram equalized to reduce brightness variation and is passed to the face recognition module to recognize the user's identity.

5.2.2 Face Recognition Module

We implement the two FRCMs mentioned in the previous two chapters for the face recognition module, which consists of the five recognition algorithms as submodules: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching, (4) Support Vector Machines and (5) Neural Networks, together with the gating network and the voting machine to provide dynamic weighting in DFRCM.

The system allows user to select any experts in the face recognition module. Face pattern from the face detection module is transmitted to the selected experts for recognition, and the gating network to determine the situations of the face pattern. The results of the submodules are then passed to the voting machine that assembles these results to arrive a uniform decision of the identity.

In terms of implementation, the system is written in Visual C++ under Windows platform. Table 5.1 lists the implementation details of the experts. For SVM and Neural Networks, special thanks should be given to Chang *et al.* [10] and Mitchell [45] who provide the library with source code which help greatly in the system development.

In EGM implementation, we define 12 fiducial points for the face which is shown in Figure 5.3. The points include important features of a face like the eyes, nose and mouth. At current state, the fiducial points are not located automatically by the system. Instead, user has to locate the fiducial points manually. Therefore, we do not include the EGM in real-time recognition now. EGM is used only in the offline processing for experimental

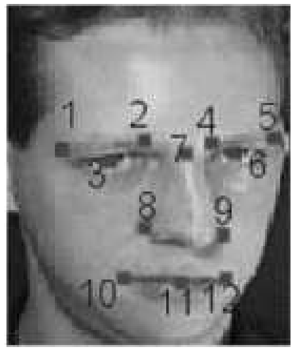

Figure 5.3: EGM fiducial points

usage.

Table 5.1: Implementation details of the experts

Expert	Detailed Description
Eigenface	Use 50 eigenvectors and 5 nearest neighbor for classification
Fisherface	Use 5 nearest neighbor for classification
EGM	Use 40 Gabor filters (8 orientation & 5 frequency)
	12 fiducial points (shown in Figure 5.3) located manually
SVM	Employ LIBSVM [10] as SVM impelmentation Use polynomial kernel function
Neural Networks	Employ Feed forward back-propagation network [45]
	Fisher projection used as feature vector

5.3 Face Recognition Process

We describe the face recognition process of the system step by step in this section. The first step in face recognition is enrollment, i.e., to collect face patterns from user and to train the face recognition algorithms for recognition. The images from the capture device are subjected to histogram equalization to reduce brightness variation. The experts in FRCM are trained to create different models for the enrollment images provided. Once the FRCM is ready for recognition, the system provides two modes of recognition: identification and verification to provide variety of applications.

5.3.1 Enrollment

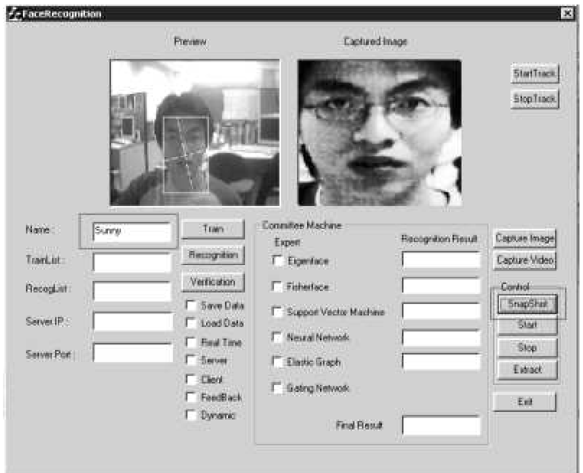

Figure 5.4: System user interface

In enrollment phase, each user takes eight images and provides his identity. Figure 5.4 shows the user interface of the system. The tracking module tracks the user who is in front of the video camera until he is ready for taking a snapshot by clicking the "Snapshot" button. The left window of the system gives a preview of the video with a square tracking the face of the user. The window on the right shows the snapshot taken after histogram equalization.

Figure 5.5: Different enrollment images

For the eight snapshots, users should provide different variations in expression, rotation and scale. Examples of some enrollment images are shown in Figure 5.5. By providing enough variations, the system is more robust because it has acquired the information of the variations already. Once the system collects all the enrollment images, it is ready to train the experts individually. The training process creates a model for each expert which stores the information of the enrollment images. For examples, Eigenface and Fisherface stores the compressed enrollment images; Neural Networks stores the weights of the neurons. All the information would be saved for further recognition.

5.3.2 Recognition

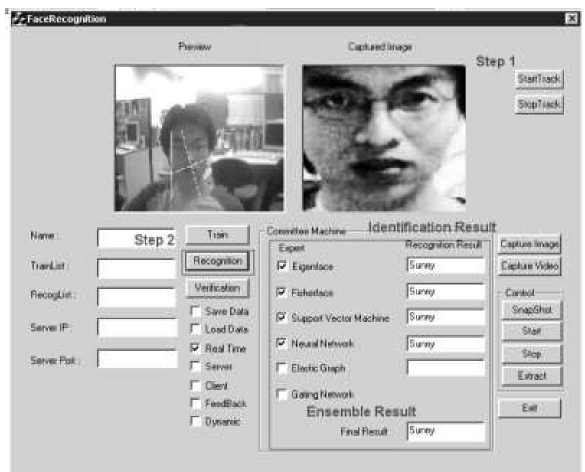

Figure 5.6: User identification

After the system training, the system is able to recognize users in real-time, which provides not only one-to-one verification but also one-to-many identification. Users can select either static or dynamic structure for recognition. Besides, the feedback mechanism mentioned previously allows users to keep updating the weighting (i.e., performance of the expert) of the

Figure 5.7: User verification (real identity)

system. All these features enable users to develop a reliable and robust face recognition system.

Identification

Identification is the first mode of recognition with numerous security applications such as video indexing, surveillance and portal control. It works by recognizing the identity of the user. In our system, the face detection module captures a face pattern of the user and passes it to FRCM for identification. The system allows user to select the experts employed in identification. Each selected expert identifies the face pattern and reports its identification result on the system window (as shown in Figure 5.6). The final result, the ensemble result of experts, is reported at the bottom of the window.

Verification

Verification is the second mode of recognition. It verifies the identity from user with variety of security applications, specifically on access control system and user authentication like desk-

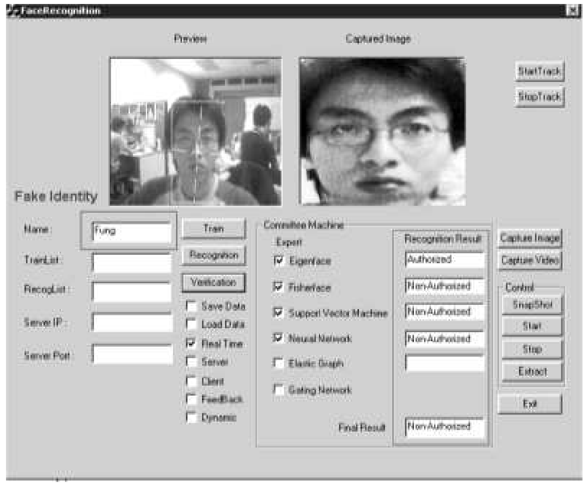

Figure 5.8: User verification (fake identity)

top logon. In verification mode, an identity is given by the user (shown in Figure 5.7) for verification. The system reports "Authorized" or "Non-Authorized" according to the experts and the ensemble result of FRCM. Figure 5.8 shows an invader with fake identity to demonstrate the rejection of invader access. The average running time for face detection and recognition on a 1400 MHz desktop machine is listed in Table 5.2:

Table 5.2: Average system running time

Process	**Time**
Face detection	1.5s
Face tracking	0.5s
Face recognition	1s

5.4 Distributed System

As face recognition is becoming more mature and popular nowadays. It is desirable to apply face recognition application on some mobile devices like notebook, Personal Data Assistant

(PDA) and especially mobile phone in the near future. However, face recognition requires large memory storage and high CPU power which is not available on most mobile devices at current status. Besides, the use of committee machine further increases the memory and time requirement. Therefore, we propose a distributed face recognition system to solve the above problems.

5.4.1 Problems

The three main problems with the use of committee machine on mobile devices are listed as follows:

- Memory limitation
- CPU power limitation
- Time and storage overhead of committee machine

Huge memory is needed to store a number of models for different algorithms, which is proportional to the number of users and images in the system. To give a rough idea of the necessary memory, Table 5.3 lists the storage of the five algorithms in four face databases. SVM requires the largest memory as it stores most information from all the training set images as support vectors. Other methods, except neural network, require less memory but they still require several megabytes. Most PDA and mobile phones still have limited memory nowadays. Therefore, it is not feasible to have such face recognition application on the machines.

Apart from the memory requirement, processing power is another important consideration. It is desirable to have a real-time response from the system. However, most mobile devices do not have high processing power to cope with the need. Take PDA as an example, the most advanced model for Pocket PC now only

Table 5.3: Storage requirements for different algorithms

Algorithm	**ORL(40)**	**Yale(15)**	**AR(130)**	**HRL(5)**
Eigenface	5.0MB	5.0MB	5.5MB	15.0MB
Fisherface	4.0MB	1.5MB	13.5MB	0.5MB
EGM	1.5MB	0.5MB	4.5MB	1.0MB
SVM	38.0MB	14.0MB	122.0MB	14.0MB
Neural Networks	32.0KB	13.0KB	106.0KB	6.0K

has 400 MHz processor (X-Scale processor). For other models, most popular processors have 33 MHz, 66 MHz or 200 MHz processing power, which is not enough for real-time face recognition, especially for face detection. Therefore, a high processing power device is necessary for face recognition.

Although committee machine helps in improving the accuracy of recognition, it has drawback that it increases the time and storage requirements linearly with the number of expert used. Time required for the FRCM is:

$$T_{FRCM} = \sum_{i=1}^{n} T_i, \tag{5.1}$$

where T_{FRCM} and T_i are the time for FRCM and the expert i respectively.

5.4.2 Distributed Architecture

Distributed system would be a solution to solve the storage and processing limitation on mobile devices. We propose a client-server approach for the implementation of FRCM. Whenever the client receives a face pattern from the face detection module, it distributes the image into different experts in the server for recognition. The experts can be located in a single server or distinct servers. Once the experts finish recognition, the corresponding results and confidences are sent to the voting machine of the client. Final decision would be made and shown to user.

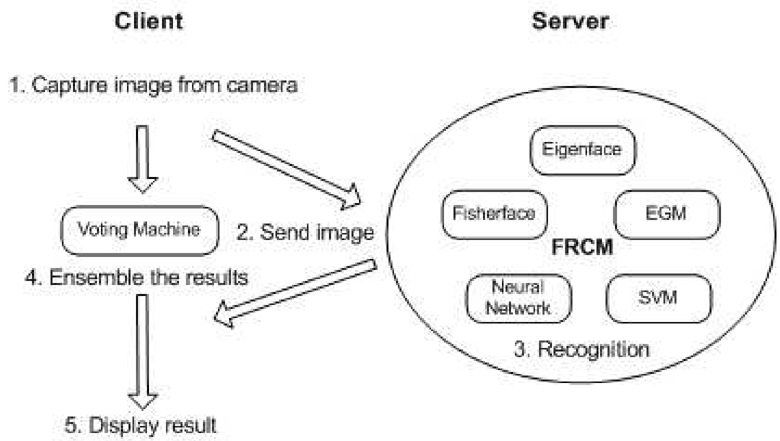

Figure 5.9: Distributed face recognition system architecture

With the distributed architecture, client does not need high processing power and memory as the most CPU and memory consuming tasks are processed in the server. Client only needs to: (1) capture an image, (2) send the image and (4) ensemble the results, as shown in Figure 5.9. For PDA or mobile phone, face detection is not feasible due to the limitation memory and processing power. However, it is still acceptable to just capture the face as the size of the screen is not large. User can fit his face on the screen of PDA and mobile phone easily. Without the need for face detection, it is feasible to apply face recognition with this distributed architecture even on mobile devices as these devices only need to take pictures and to do simple calculation only.

Table 5.4: Processing time (S: Startup, R: Recognition)

Machine for Testing	**Time (S+R)**	**Time (R)**
PIV 1400 MHz(Desktop)	13s	1s
PII 300 MHz (Notebook)	93s	2s
PII 300 MHz Client + PIV 1400 MHz Server	16s	2s

We have implemented the client in a way that several threads are created to handle different experts in FRCM. Client can connect to distinct servers (one server for an expert) to speed up the processing time. We adopt an experimental testing for the distributed system on a notebook (client) and a desktop (server). Besides, we test the system on the notebook and the desktop machine separately for comparison. The processing power of the machines are 1400 MHz and 300 MHz respectively. Table 5.4 lists the processing time. The result shows that there is no big difference in the processing time for recognition on the desktop and the notebook computer. The time differs by one second only. Although we can not tell the improvement of the distributed architecture over the non-distributed one in this experiment, the difference would be significant if PDA or mobile phone is used instead of the notebook. From the experiment, we can still see the improvement in startup time for loading the models of the experts. Moreover, if distinct servers are used, the time required for the distributed FRCM is further reduced because the servers can recognize the face pattern concurrently. The overall time required for the distributed system is:

$$T_{FRCM} = T_{cs} + \max_i T_i + T_{sc}, \tag{5.2}$$

where T_{cs} and T_{sc} are the time for transmission of the face pattern between client and server, which are relatively small when compared to the recognition time T_i.

Currently, we have not implemented client program on PDA or mobile phone. We have tested the distributed architecture on a notebook only. However, due to the limitation power of PDA and mobile phone, it is in fact not feasible to implement the face recognition system solely on these machines. Therefore, the distributed architecture is a solution to make it possible. Besides, we believe the proposed distributed system architecture can greatly increase the recognition time for PDA and mobile phone as their processing power is much lower than that of the notebook computer.

5.5 Conclusion

In this chapter, we describe our implementation of the FRCM (Static and Dynamic Structure) in the face recognition system to demonstrate its effectiveness. We describe the implementation details of the system, including the system architecture, face recognition process. We also raise some potential problems of the feasibility of face recognition application on mobile devices.

For the system architecture, the system consists of two main modules: (1) face detection module and (2) face recognition module. The modules combine the most state-of-the-art technologies in face tracking, face detection and recognition to provide real-time face recognition for identification and verification applications like access control system, surveillance and portal control.

Furthermore, we state the three main problems of applying face recognition on mobile devices: (1) memory limitation, (2) CPU power limitation and (3) time and storage overhead of committee machine. To tackle these problems, we propose the distributed system architecture which enables face recognition application on limited memory and CPU power mobile devices feasible. The reason is that the client is thin which does not involve the most time and storage consuming tasks in the face recognition process. We evaluate its feasibility through experiment testing on a desktop and a notebook computer. To conclude, the architecture provides a solution for a practical face recognition system on mobile devices in the future.

□ **End of chapter.**

Chapter 6

Experimental Results

6.1 Introduction

In this chapter, we present our experimental results on SFRCM, DFRCM and the five algorithms. We evaluate their performances with the four face databases: (1) ORL Database, (2) Yale Database, (3) AR Database and (4) HRL Database. The databases contain variations in expression, occlusion, glasses or non-glasses and illumination, which is well known in the field of face recognition.

In the following section, we briefly introduce each database by providing some snapshots of the databases and characteristic descriptions. Then, we explain our face pre-processing technique applied on these databases, and we describe our experimental procedure, cross validation testing and experimental setup. We give detailed experimental results and an evaluation of the results. Finally, we compare the five algorithms and make a conclusion for the results.

Figure 6.1: Snapshot of ORL database

6.2 Database

6.2.1 ORL Face Database

First experiment is performed on ORL face database from AT&T Laboratories Cambridge [47]. The images are gray-scale with a resolution of 92×112 pixels. The database contains 400 images, including 40 distinct people, each with 10 images that vary in position, rotation, scale and expression. The images are taken under constant lighting condition. Figure 6.1 shows a snapshot of 7 people.

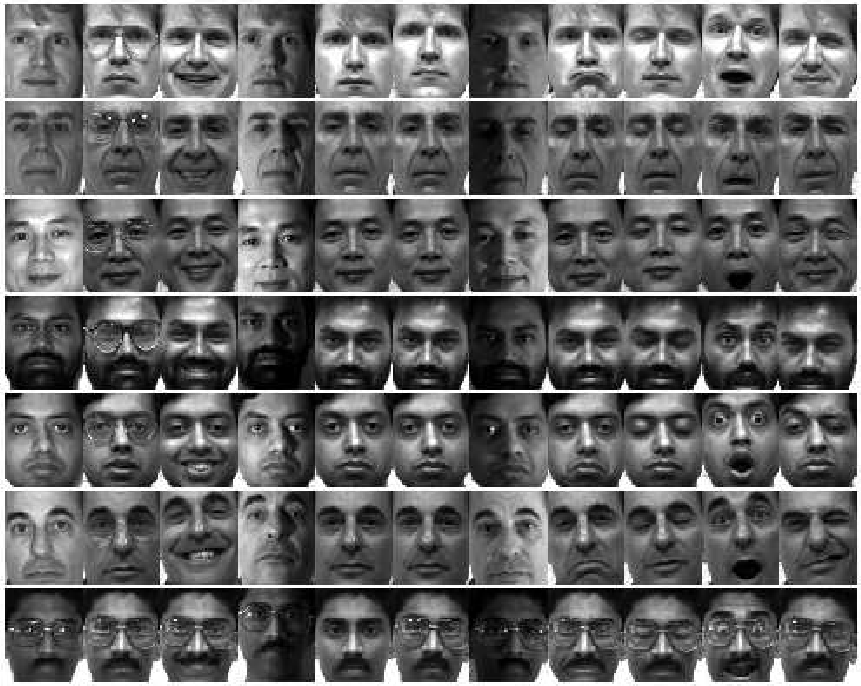

Figure 6.2: Snapshot of cropped Yale database

6.2.2 Yale Face Database

Second experiment is performed on Yale face database from Yale University [65]. The database contains 165 images, including 15 distinct people, each with 11 images that vary in both expression and lighting. The images are gray-scale and are cropped to a resolution of 92×112 pixels for testing. Figure 6.2 shows a snapshot of 7 people in the database.

6.2.3 AR Face Database

Third experiment is performed on AR face database from the Computer Vision Center (CVC) at the U.A.B. [43], [3]. It contains over 2,500 color images corresponding to 131 people's faces (75 men and 60 women). The images feature frontal view faces with different facial expressions, illumination conditions, and

Figure 6.3: Snapshot of AR database

occlusions (sun glasses and scarf). The pictures were taken at the CVC under strictly controlled conditions. No restriction on wear (clothes and glasses), make-up and hairstyle were imposed to participants. The images are gray-scale and are cropped to a resolution of 92×112 pixels for testing. A snapshot of 8 people in the database is shown in Figure 6.3.

6.2.4 HRL Face Database

The final experiment is performed on the HRL face database which is created by Hallinan at the Harvard Robotics Laboratory [30]. Each image contains a person held his/her head steady while being illuminated by a dominant light source. The person is centered in the sphere with light source changes at various longitude and latitude directions. Figure 6.4 gives a snapshot of

Figure 6.4: Snapshot of HRL database

the 3 people in the database.

6.3 Experimental Details

6.3.1 Pre-processing

Figure 6.5: Snapshot of original AR database

We do not use the images from the above databases in the experiments directly. Instead, the images of the databases require pre-processing as the original Yale, AR and HRL face databases

Figure 6.6: Snapshot of original HRL database

are not ready for experiments immediately. The reasons are as follows:

- The images may include simple background.
- The face region in the images is not located properly.
- The images have different size.

Figure 6.5 and Figure 6.6 show some original images from the AR and HRL face database. The background has to be removed since it would deteriorate the performance of the experts. It is not desirable that the experts takes background into account in the recognition. Therefore, pre-processing is necessary to remove the background and to crop the images such that the images contain faces only. Besides background removal, the cropped face images have to be resized to a standard size (92×112 pixels in our experiments). The cropped images are not histogram equalized and are used for the later experiments.

Face Region Extraction

We propose a face region extraction to get the face region of each image in the face databases. The basic idea of the extraction approach is to use the face boundary for locating the face region. We have made an important assumption in the algorithm that the background of the images is simple. We describe our

Figure 6.7: Snapshot of the original Yale database

approach below by using the right image from the snapshot of Yale databases (shown in Figure 6.7) as an illustration. Figure 6.8 shows a binary image and a face region extracted on the Yale image.

1. Apply median filter to reduce the noise in the background.
2. Apply Sobel filter for edge detection.
3. Convert the image to a binary image.
4. Apply horizontal and vertical projection.
5. Find the face boundary by selecting the horizontal and vertical lines passing through a region threshold.
6. Obtain the center of the face region.
7. Crop the face region and resize it to the standard size.

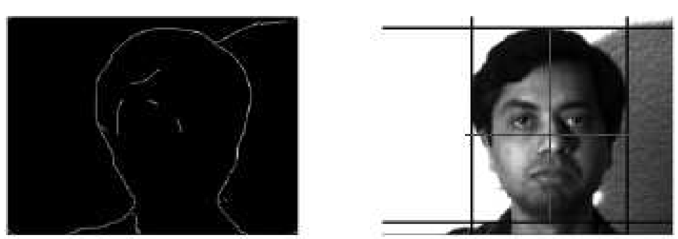

Figure 6.8: Binary image (left) and face region (right)

6.3.2 Cross Validation

In our experiments, we apply the cross validation evaluation approach to test the performance of the experts and the FRCMs. Cross validation is a standard tool in statistics by Janssen *et al.* [36] to help evaluating performance of different algorithms. The simplest kind of cross validation is known as the holdout method. Firstly, we partition the whole data set into a training set and a testing set. We further divide the training set into two subsets: (1) a training set to estimate the model and (2) a validation set to test the performance of the model [50]:

- Training set: It is used to train the experts of the FRCMs. We employ the training set images to train the models of each expert separately.
- Validation set: It is used to estimate the performance of the experts. We test the performance of each expert with the validation set data. The performance is then applied to calculate the weights of the experts in DFRCM.
- Testing set: To evaluate the performance of the experts and FRCMs.

We adopt the holdout method to evaluate the performance of the FRCMs and the experts on AR and HRL face database. However, the evaluation can have a high variance as it may depend heavily on the selection of training set and test set. Therefore, the evaluation may be significantly different depending on how the division is made.

K-Fold Cross Validation

K-fold cross validation is one way to improve over the holdout method. We adopt this approach on the ORL and Yale test. We divide the data set into K equal size subsets, and the holdout method is repeated K times. We train the experts K times, each time leaving out one of the subsets from training and test for the remaining subset. Figure 6.9 describes how we divide the data in each experiment.

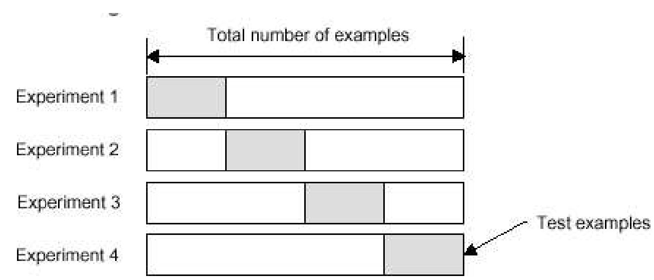

Figure 6.9: K-Fold cross validation

The advantage of this method is that every image in the face database is eventually used for training and testing the experts. However, we have to train and test the experts K times, which means the computation time of the evaluation is K times longer than the original one.

6.3.3 System details

The experiments are conducted on a 1400 MHz desktop machine with Windows platform. Distributed system is not used in these experiments. Different thresholds are selected empirically for template matching approaches (Eigenface, Fisherface & EGM) to accept or to reject a template for different databases. Templates not passing the thresholds are not counted in the voting. The thresholds for different databases are listed in Table 6.1.

Table 6.1: Threshold for template approaches

Threshold	Eigenface	Fisherface	EGM
ORL	8	0.4	0.6
Yale	12	0.3	0.6
AR	9	0.085	0.6
HRL	10	0.13	0.6

6.4 Result

6.4.1 ORL Result

In ORL face testing, the images are partitioned into 10 subsets (S_1, S_2, ..., S_{10}). Each subset includes one image of each subject. We apply the K-Fold Cross Validation approach to evaluate the performance of the five experts, DFRCM and SFRCM. The data set is partitioned as follows:

1. Six subsets for training
2. Three subsets for validation
3. One subset for testing

For example, to test subset 1, we use subsets 2-7 for training and subsets 8-10 for validation. The experiment is conducted for 10 trials. Table 6.2 lists the results of the 10 subsets. Each row presents the performance of the experts, DFRCM and SFRCM on subset i respectively.

From the results, we notice that SVM is the best expert among the five on average, which achieves 95.5% accuracy. The other experts Fisherface and NN also get good results on average (93.8% and 91.5% respectively). Besides, Eigenface and EGM obtain relatively lower performance (80.3% and 81.5% respectively) than the others.

Table 6.2: ORL result

S	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
1	82.5%	90.0%	90.0%	92.5%	**97.5%**	**92.5%**	**92.5%**
2	85.0%	**100.0%**	92.5%	**100.0%**	97.5%	**100.0%**	**100.0%**
3	87.5%	**100.0%**	72.5.0%	**100.0%**	92.5%	**100.0%**	**100.0%**
4	75.0%	92.5%	85.0%	**95.0%**	87.5%	**100.0%**	**95.0%**
5	72.5%	**97.5%**	80.0%	90.0%	87.5%	**90.0%**	**97.5%**
6	82.5%	90.0%	82.5%	**97.5%**	87.5%	**95.0%**	**92.5%**
7	80.0%	**92.5%**	75.0%	**92.5%**	90.0%	**97.5%**	**92.5%**
8	77.5%	87.5%	77.5%	**95.0%**	87.5%	**95.0%**	**90.0%**
9	75.0%	90.0%	77.5%	**97.5%**	92.5%	**100.0%**	**97.5%**
10	85.0%	**97.5%**	82.5%	95.0%	95.0%	**95.0%**	**97.5%**
Average	80.3%	93.8%	81.5%	**95.5%**	91.5%	**96.5%**	**95.5%**

DFRCM and SFRCM reach better or equal performance (96.5% and 95.5% respectively) than all the other experts in the testing, which demonstrates that our proposed approaches of combining the classifiers do work on improving the accuracy of individual classifiers.

To investigate the improvement of FRCMs, we provide the details of the underlying data (including individual expert result and its confidence) in subsets 1, 4 and 7 of Table 6.3. We show the difference in the classification results of DFRCM, SFRCM and the highest accuracy expert (Bolded in the subset). The results demonstrate that with the use of confidence and weight, poor results from some experts would not affect the final ensemble result significantly.

In subset 1, both DFRCM and SFRCM obtain lower accuracy (92.5%) than NN (97.5%). The FRCMs fail to improve the accuracy in this subset. In image 0 and 22, NN classifies the images correctly. However, as there are two experts (Eigenface and SVM) classify image 0 as class 15, and three experts

Table 6.3: Detailed ORL result

		Recognized Class/Confidence						
S	I	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
1	0	15/0.40	23/0.60	12/0.20	15/1.00	**0/0.50**	**15/0.40**	**23/0.24**
	22	37/0.60	37/0.60	22/0.80	37/1.00	**22/0.33**	**37/0.56**	**37/0.43**
	34	14/0.40	14/0.80	34/0.40	20/1.00	**38/0.49**	**20/0.37**	**14/0.34**
4	12	12/0.80	12/1.00	3/0.40	**4/1.00**	0/0.74	**12/0.34**	**12/0.43**
	38	38/0.60	38/0.80	38/0.80	**21/1.00**	38/0.59	**38/0.49**	**38/0.60**
7	15	15/0.80	15/1.00	2/0.20	**0/1.00**	15/0.46	**15/0.44**	**15/0.55**
	25	25/0.40	**27/0.80**	10/0.40	25/1.00	25/0.47	**25/0.45**	**27/0.32**
	30	30/0.40	**22/0.60**	30/0.60	30/1.00	37/0.42	**30/0.43**	**30/0.27**
	31	35/0.20	31/0.40	17/0.40	**1/1.00**	25/0.26	**31/0.34**	**1/0.17**
	34	26/0.60	**18/0.60**	34/0.40	34/1.00	5/0.23	**34/0.39**	**18/0.24**
	36	36/0.80	36/0.60	36/0.80	**27/1.00**	36/0.33	**36/0.38**	**36/0.46**

(Eigenface, Fisherface and SVM) classify image 22 as class 37, the final score of class 15 and 37 for DFRCM is higher than that of class 0 and 22 respectively. For SFRCM, as Fisherface is the best expert on average (87.6%, shown in Table 6.8), SFRCM gives higher weight to the results of Fisherface and thus SFRCM classifies image 0 as class 23 instead of class 15. Besides, both FRCMs classify incorrectly because only EGM gets the correct result for image 34 with low confidence(0.4).

In subset 4, none of the experts obtains 100% accuracy but DFRCM achieves it. SVM is the best among the five experts in this subset (95%), which fails in classifying image 12 and 38. The result of this subset shows the effectiveness of committee machine in improving the accuracy. FRCMs obtain correct results because other experts classify these images correctly.

In subset 7, both Fisherface and SVM obtain 92.5% accuracy while DFRCM achieves 97.5%. The improvement is due to the difference in nature of the heterogeneous experts in committee machine. The experts may get different correct result sets in classification even if they achieve same accuracy. FRCMs do increase the overall accuracy by combining the experts' correct results.

To conclude, although committee machine can not ensure that it can improve classification accuracy in all situations (e.g., it fails in subset 1), it increases the accuracy in most cases. We will show more results in the followings to show its effectiveness.

6.4.2 Yale Result

In Yale face testing, the partition method is similar to the ORL face testing. The images are partitioned into 11 subsets. Each subset includes one variation (expression/lighting) of each subject. We also apply the K-Fold Cross Validation approach to evaluate the performance of the five experts, DFRCM and SFRCM. However, four validation subsets are used instead of three. The experiment is conducted for 11 trials.

Table 6.4 lists the results of the experts, DFRCM and SFRCM in Yale testing respectively. Special attentions should be given to subset 4 and 7 in the testing, the performance for all experts drops significantly (lower than 40%) except EGM. The main reason for the non-satisfactory results is that Yale database contains variations in strong left and right lighting in subset 4 and 7 respectively (shown in Figure 6.10). These variations are not included in the training of subset 4 and 7 before.

EGM works relatively better in the light testing than other algorithms, which achieves 66.7% and 93.3% in left and right lighting respectively. This is due to the use of Gabor wavelet transformation of fiducial points, and the edge information in the graph. These kinds of representation is invariant to lighting variations and thus EGM has better performance in this Yale testing.

Table 6.4: Yale result

S	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
1: centerlight	40.0%	73.3%	**100.0%**	93.3%	60.0%	**93.3%**	**86.7%**
2: glasses	73.3%	**93.3%**	80.0%	86.7%	86.7%	**86.7%**	**93.3%**
3: happy	73.3%	86.7%	**93.3%**	86.7%	**93.3%**	**86.7%**	**93.3%**
4: leftlight	26.7%	40.0%	**66.7%**	26.7%	40.0%	**46.7%**	**40.0%**
5: noglasses	93.3%	**100.0%**	**100.0%**	**100.0%**	93.3%	**100.0%**	**100.0%**
6: normal	86.7%	**93.3%**	80.0%	86.7%	**93.3%**	**93.3%**	**93.3%**
7: rightlight	26.7%	40.0%	**93.3%**	20.0%	26.7%	**53.3%**	**46.7%**
8: sad	66.7%	**93.3%**	**93.3%**	**93.3%**	86.7%	**93.3%**	**93.3%**
9: sleepy	80.0%	**93.3%**	86.7%	**100.0%**	**93.3%**	**100.0%**	**93.3%**
10: surprised	**73.3%**	53.3%	26.7%	66.7%	46.7%	**60.0%**	**53.3%**
11: wink	93.3%	86.7%	86.7%	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Average	66.7%	77.6%	**82.4%**	78.2%	74.5%	**83.0%**	**81.2%**
Nolighting	75.6%	85.9%	83.0%	**90.4%**	83.7%	**90.4%**	**89.6%**

DFRCM achieves better performance (83.0%) than all the experts, while SFRCM obtains 81.2% which is slightly lower than EGM. Without the lighting variations, DFRCM achieves 90.4% while SFRCM obtains 89.6%, which is comparable to the ORL testing.

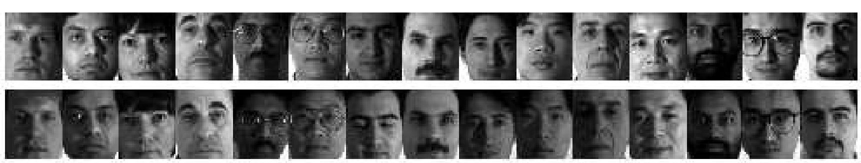

Figure 6.10: Left and right lighting images

6.4.3 AR Result

In the AR face testing, we use the holdout method to evaluate the performance of the experts and FRCMs. We do not apply the K-fold cross validation as in the ORL and Yale face testing because the number of images in AR face is more than that of ORL and Yale databases. K-fold cross validation takes extremely long time to train and to test the experts K times. Besides, it is hard to partition the images into equal subsets like ORL and Yale for the K-fold evaluation. In the holdout method, the images is chosen carefully to ensure that training set (780 images), validation set (260 images) and testing set (390 images) include enough variations within the subject. The experiment is conducted for 1 trial.

Table 6.5: AR result

Performance	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
Validation	38.1%	86.2%	35.4%	55.4%	59.2%		
Testing	28.7%	**89.2%**	58.7%	59.7%	76.4%	**89.2%**	**86.4%**

Validation and testing results of the experts and FRCMs are listed in Table 6.5. In AR testing, Eigenface has the lowest accuracy (28.7%). The reasons may due to the large variations in lighting, expression, sunglasses and occlusion existing in AR face database. Eigenface may takes these variations as the major principal components in PCA, which does not include class information for PCA projection. This shows that Eigenface is not reliable to handle face recognition in large variations.

Fisherface, the class specific approach, achieves much better performance (89.2%) than other experts. It is because Fisherface's FLD projection considers the between-classes and within-classes variations, which is well suitable in large class classification. Compared to Eigenface, FLD projection takes the variations as within-class variations, thus it can preserve the classification ability even for large variations.

For EGM, the main reason for the non-satisfactory accuracy (58.7%) is that it is hard to locate the fiducial points of the test face images, which are occluded by the comforter and sunglasses. The points can not be located properly even we specify the fiducial points manually. Therefore, EGM's accuracy is poor because of incorrect position of the fiducial points.

SVM has relatively low accuracy (59.7%) on AR testing. Compared to ORL and Yale testings, its performance drops significantly. The main reason for the drop is incorrect classification of the hyperplane. There are a large number of classes (130 people) in this testing. As SVM works by finding a hyperplane for two class problem, it may work poorly on a large number of classes even if "one-against-one" approach is used for multiple classes. Besides large classes problem, the large variation on the images, especially occlusion and lighting variations, is another major reason for the low accuracy.

Both DFRCM and SFRCM achieve better results than most of the experts, which obtain 89.2% and 86.4% accuracy respectively. DFRCM obtains same result as Fisherface in the testing. It is because Fisherface achieves superior performance than all other experts in validation. DFRCM thus gives higher weight to Fisherface than the other experts. With the use of weight, low performance experts (in extreme case like this AR testing) would not affect the overall ensemble results significantly. For SFRCM, as average performance of the experts is employed, it has lower accuracy than that of Fisherface. However, SFRCM's performance is still better than the other four experts.

6.4.4 HRL Result

We also apply the holdout method to evaluate the performance of the experts and FRCMs in the HRL face testing. Similar to AR face testing, we select the face images in training set (150 images), validation set (96 images) and testing set (97 images) carefully to ensure that each data set contains enough lighting variations. The experiment is conducted for 1 trial.

Table 6.6: HRL result

Performance	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
Validation	79.2%	81.3%	87.5%	75.0%	83.3%		
Testing	80.4%	89.7%	86.6%	82.5%	**90.7%**	**94.8%**	**90.7%**

Table 6.6 lists validation and testing results of the experts and FRCMs. In HRL testing, all experts have similar performance in validation, which is different from AR testing. They achieve over 80% accuracy in the testing. Therefore, comparable weights are given to all experts, and the results of FRCMs would not depend on one expert only.

Although HRL database has strong lighting variations, the variations are included in the training set. The experts have acquired enough knowledge for these variations, which is different from the Yale testing. Template based approaches (Eigenface, Fisherface & EGM) have good performance in this testing because most lighting variations are stored as templates. Fisherface and NN are better algorithms than the others in this testing. The main reason for the high accuracy of NN is that it is more easy to classify few classes than large classes. Compared to AR testing (130 classes), HRL testing has much fewer classes (5 classes).

DFRCM achieves 94.8% accuracy which is better than other experts again. SFRCM obtains 90.7% in the testing which is comparable to NN, which is the highest accuracy expert in this testing.

6.4.5 Average Running Time

Table 6.7 shows the average running time (in seconds) of the experts and FRCM on the four face databases. We simply use

Table 6.7: Average running time

Database (no.)	Eigen	Fisher	EGM	SVM	NN	FRCM	FRCM/Image
ORL (40)	2.1s	1.5s	16.3s	6s	1.4s	27.3s	0.68s
Yale (15)	0.9s	0.2s	6.5s	0.6s	0.3s	8.5s	0.57s
AR (390)	21s	48s	123s	118s	56s	366s	0.94s
HRL (97)	20s	1s	54s	3s	1s	79s	0.81s

FRCM for DFRCM and SFRCM as their running time is very close. We do not use the proposed distributed system for evaluation in these experiments, so the processing time of FRCM is the sum of the experts' processing time instead of the maximum time of the individual algorithm. Despite of this time overhead, FRCM uses less than a second to recognize an image on average, which is acceptable for real-time face recognition applications. If time is critical in a real application, we can apply the distributed architecture in Chapter 5 with distinct servers to further reduce the execution time.

Among the five experts, EGM requires longer processing time due to Gabor Wavelet Transformation, whose convolution takes much time as each face pattern is processed with 40 Gabor filters. Nevertheless, the time is still acceptable. For the others, the processing time is satisfactory for any real-time application.

6.5 Discussion

We discuss the five algorithms in terms of their advantages and disadvantages in this section. Generally speaking, they work by extracting face features from an image and then classify the face pattern according to the features. Eigenface and Fisherface project the entire face image into a feature space for comparison. In contrast, EGM takes local features of a face like eyes, mouth into account for recognition. For machine learning approaches, SVM looks for a separating hyperplane which separates the face pattern with largest margin. Neural Networks classifies face pattern by minimizing the prediction error through adjusting the weight of neurons. The five algorithms classify face patterns with different methodologies. We list some advantages and disadvantages of each algorithm to provide a thorough comparison.

6.5.1 Advantages

Eigenface and Fisherface are global approaches that take entire face image as a 2D array of pixels for comparison. Both methods are quite similar as Fisherface is a modified version of Eigenface. They find an orthonormal basis face space based on the common face features of the training set images, and then project these images into the face space for comparison. The difference is the way of projection. Eigenface uses PCA which works better with dimension reduction while Fisherface uses FLD which works better for classification. Therefore, Eigenface is a practical approach for face recognition due to its efficiency and simplicity. Fisherface is best for classification as it can deal with inter-class as well as inner-class variations.

EGM is a local approach of face recognition which is based on the fiducial points of an image but not the entire image like Eigenface and Fisherface. This is suitable for face recognition because it extracts important features from the face as criteria for comparison. Besides, Zhang [67] mentioned that the use of Gabor wavelet features as the output of bandpass filters is suitable for recognition. As the features are closely related to derivatives and are therefore less sensitive to lighting changes. Its effectiveness is demonstrated in the Yale's leftlight and rightlight tests in the experiment.

SVM and Neural Networks are proven accurate in general pattern classification and are widely used in various fields. Moreover, they are not template matching approaches which store enrollment images for comparison on each image with the test image. In contrast, they classify test image based on their models and thus have fast recognition speed.

6.5.2 Disadvantages

For Eigenface and Fisherface, there is a high correlation between the training data and the recognition data. Their accuracy depends on how the training data are obtained because they take the whole image as comparison. The accuracy would decrease greatly with varying light intensity, scale and orientation of the face image. Therefore, pre-processing of the image is required in order to achieve satisfactory result. Besides accuracy, both methods are difficult to expand because they use projection on face space which is formed by the training data. Whenever new subjects are added, the face space and the feature vectors may need to be recalculated, which is undesirable for a face recognition system.

EGM applies local facial features in recognition. However, it is difficult to find the fiducial points automatically and accurately. If the points are located manually, it is time-consuming for the location process. Moreover, the fiducial points may not be located for cases in AR testing where faces are occluded by glasses and clothes. In such situation, the performance drops significantly due to the incorrect location of the fiducial points. Although Gabor Wavelet Transformation helps achieving high accuracy, the transformation takes long processing time as each image is convoluted with all Gabor filters.

SVM and Neural Networks have fast recognition time, but they take time to train the models. SVM takes long time to solve the complex quadratic programming problem, which is dependent on the number of training samples and may last for several days. Neural Networks takes shorter training time than SVM. However, it may not find an optimal solution as SVM does. The training may be trapped by local minimum when reducing the error during training. Thus, its accuracy is lower than that of SVM.

6.6 Conclusion

We have performed a comprehensive testing on the five experts and the FRCMs in this chapter. Table 6.8 summarizes the average results of the four testings for a thorough comparison. In the four experiments, DFRCM achieves accuracy higher or equal to the accuracy of the best expert in the committee. The use of dynamic weighting enables the committee machine to choose the best expert in extreme case which is demonstrated in AR testing, or gives higher weighting to expert with better performance. Under normal situation which each expert performs with comparable performance, DFRCM can generally obtain higher performance by assembling experts' results. Moreover, DFRCM has higher accuracy than SFRCM in all the experiments (over 90% on average), which shows dynamic structure has better performance than static structure.

Table 6.8: Average overall results

Database	Eigen	Fisher	EGM	SVM	NN	DFRCM	SFRCM
ORL	80.3%	93.8%	81.5%	**95.5%**	91.5%	**96.5%**	**95.5%**
Yale	66.7%	77.6%	**82.4%**	78.2%	74.5%	**83.0%**	**81.2%**
AR	28.7%	**89.2%**	58.7%	59.7%	76.4%	**89.2%**	**86.4%**
HRL	80.4%	89.7%	86.6%	82.5%	**90.7%**	**94.8%**	**90.7%**
Average	64.0%	**87.6%**	77.3%	79.0%	83.3%	**90.9%**	**88.5%**

For the five algorithms, they have various performances on the four experiments: (1) SVM is the best classifier in the ORL testing; (2) EGM handles the expression and strong lighting variations well in the Yale testing; (3) Fisherface shows its effectiveness in classification of 130 subjects in AR testing and (4) Neural Networks gives better accuracy in HRL testing. From the experimental results, we conclude that Fisherface is the best classifier among the five experts due to its effectiveness in classification. It generally obtains high accuracy in the experiments. Besides, SVM and Neural Networks are also good classifiers in face recognition. EGM is especially good in handling different expression. Eigenface is the weakest classifier as it simply compresses images without taking other information like class information and facial features for recognition.

□ **End of chapter.**

Chapter 7

Conclusion

In this thesis, we have studied the five well-known face recognition algorithms: (1) Eigenface, (2) Fisherface, (3) Elastic Graph Matching, (4) Support Vector Machines and (5) Neural Networks. We have also made a thorough comparison of the algorithms by evaluating their performances on four standard face databases. From the experiments, we conclude that Fisherface is the best algorithm in general cases due to its effective classification ability. It achieves high accuracy constantly. The machine learning approaches, Support Vector Machines and Neural Networks, are also good algorithms, which obtain good results in most cases. However, Eigenface and Elastic Graph Matching are poor algorithms.

Apart from the comparison of the algorithms, we have also presented a framework named Face Recognition Committee Machine (FRCM) that integrates different face recognition algorithms as experts in a committee machine. FRCM is composed of these five algorithms, which assembles their results to achieve better accuracy. We define three basic elements for the ensemble:

- Result: The result of expert, which refers to an identity of a face in identification, and whether authorized or non-authorized in verification.

- Confidence: The confidence of expert on its result, which is a measure of the result's reliability.
- Weight: The weight of expert's result in ensemble. Different weight is given to each expert according to its performance. The better the performance, the higher the weight.

Moreover, we have proposed two structures for FRCM: Static Structure (SFRCM) and Dynamic Structure (DFRCM). The weights are fixed for all the experts in SFRCM. However, fixed weights are not suitable for face recognition application. It is because user may situates under various environments where the algorithms may have dramatic different performance. Therefore, we propose DFRCM that applies a gating network to determine the weights of the experts according to the environments. DFRCM helps developing a robust face recognition system by providing separate weights for experts in different environments. Furthermore, we have designed a feedback mechanism to enable continuous update of the weights according to their recent performances.

We have implemented the FRCM in a face recognition system to demonstrate the effectiveness of our work. The system consists of a face detection module and a face recognition module. The modules combine the most state-of-the-art technologies in face tracking, face detection and face recognition to provide real-time face recognition for identification and verification applications. Besides, we state the main problems of applying face recognition on mobile devices: (1) memory limitation, (2) CPU power limitation and (3) time and storage overheads of committee machine. Therefore, we propose a distributed system architecture to tackle these problems.

To conclude our work, we contribute on the followings:

- We compare five well known face recognition algorithms and evaluate their performances on four standard face databases.

- We propose a framework that integrates heterogeneous face recognition algorithms as Face Recognition Committee Machine (FRCM). We show the results of the algorithms can be assembled in terms of result, confidence and weight.
- We design two architectures for FRCM: SFRCM and DFRCM, which adapt different environments in real situations. Besides, we propose a feedback mechanism to adjust the weight of an individual algorithm according to its performance.
- We implement a face recognition system for real-time face recognition to demonstrate our work. We also propose a distributed system architecture to apply face recognition application on mobile devices.

□ **End of chapter.**

Bibliography

[1] I. Aizenberg, N. Aizenberg, C. Butakov, and E. Farberov. Image recognition on the neural network based on multi-valued neurons. In *Proc. 15th International Conf. on Pattern Recognition 2000*, volume 2, pages 989–992, 2000.

[2] G.-J. Ann. Authentication: From password to biometrics.

[3] Ar database. Database available at:
`http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html`.

[4] N. Ayat, M. Cheriet, and S. C.Y. Kmod- a new support vector machine kernel for pattern recognition. application to digit image recognition. In *ICDAR*, pages 1215–1219, Seattle,USA, Sept 2001.

[5] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 19, pages 711–720, July 1997.

[6] L. Breiman. Bagging predictors. In *Machine Learning*, volume 2, pages 123–140, 1996.

[7] L. Breiman. Some infinite theory for predictor ensemble. Technical Report 577, Statistics Department, UC Berkeley, Auguest 2000.

[8] J. Buhmann, J. Lange, and C. von der Malsburg. Distortion invariant object recognition by matching hierarchically

labeled graphs. In *Proc. IJCNN. International Joint Conf. on Neural Network, 1989*, pages 155–159, 1989.

[9] J. Burges, Christopher. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*, volume 2, pages 121–167, 1998.

[10] C.-C. Chang and C.-J. Lin. Libsvm – a library for support vector machines. Software available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[11] P. Cristea. Neural networks back propagation algorithm, March 1999. http://www.dsp.pub.ro/articles/nn/nn21/slide1.html.

[12] X. Dihua, I. Podolak, and L. Seong-Whan. Facial component extraction and face recognition with support vector machines. In *Proc. Fifth IEEE International Conf. on Automatic Face and Gesture Recognition, 2002*, pages 76–81, May 2002.

[13] P. Disorntetiwat and C. Dagli. Simple ensemble-averaging model based on generalized regression neural network in financial forecasting problems. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 477–480, 2000.

[14] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley, June 1973.

[15] Identix. http://www.identix.com.

[16] R. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[17] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteen National Conference*, pages 148–156, 1996.

[18] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 15(5):771–780, September 1999. http://citeseer.nj.nec.com/freund99short.html.

[19] R. Gary. Biometrics in human services. User Group Newsletter, May 1998.
http://www.dss.state.ct.us/digital/news08/bhsug08.htm.

[20] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: generative models for recognition under variable pose and illumination. In *Proc. Fourth IEEE International Conf. on Automatic Face and Gesture Recognition, 2000*, pages 277–284, 2000.

[21] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 23, pages 643–660, June 2001.

[22] A. Georghiades, D. Kriegman, and P. Belhurneur. Illumination cones for recognition under variable lighting: faces. In *Proc. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998*, pages 52–58, 1998.

[23] D. B. Graham and M. Allinson, Nigel. Characterizing virtual eigensignatures for general purpose face recognition. *Face Recognition: From Theory to Application*, pages 446–456, 1998.

[24] M. A. Grudin. *A Compact Multi-Level Model for the Recognition of Facial Images.* PhD thesis, Liverpool John Moores University, Nov 1997.

[25] G. Guo, S. Li, and K. Chan. Face recognition by support vector machines. In *Proc. Fourth IEEE International Conf. on Automatic Face and Gesture Recognition 2000*, pages 196–201, 2000.

[26] S. Gutta, J. Huang, B. Takacs, and H. Wechsler. Face recognition using ensembles of networks. In *IEEE Proc. 13th International Conference on Pattern Recognition*, volume 3, pages 50–54, August 1996.

[27] S. Gutta, J. R. J. Huang, P. Jonathon, and H. Wechsler. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. In *IEEE Trans. on Neural Networks*, volume 11, pages 948–960, July 2000.

[28] S. Gutta and H. Wechsler. Network ensembles for facial analysis tasks. In *IEEE Proc. International Joint Conference on Neural Networks*, volume 3, pages 305–310, 2000.

[29] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441,498–520, 1933.

[30] Hrl database. Database available at: `http://www1.cs.columbia.edu/~belhumeur`.

[31] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):696–706, May 2002.

[32] F. J. Huang, H. J. Zhang, T. Chen, and Z. Zhou. Pose invariant face recognition. In *IEEE Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 245–250, 2000.

[33] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *IJCV*, 29:5–28, 1998.

[34] R. A. Jacobs, M. I. Jordan, J. N. Steven, and E. H. Geoffrey. Adaptive mixtures of local experts. In *Neural Computation*, volume 3, pages 79–87, 1991.

[35] K.-F. Jang, H.-M. Tang, M. Lyu, and I. King. A face processing system based on committee machine: The approach and experimental results. In *Proceedings of The 10th International Conference on Computer Analysis of Images and Patterns (CAIP 2003)*, 2003.

[36] P. Janssen, P. Stoica, T. Soderstroms, and E. P. Model structure selection for multivariable systems by cross-validation methods. *International Journal of Control*, 47:1737–1758, June 1998.

[37] K. Karhunen. Uber lineare methoden in der wahrsccheilichkeitsrechnung. *Annales Academiae Scientiarum Fennicae, Seried A1: Mathematica-Physica*, 37:3–79, 1947.

[38] K. Kim, J. Kim, and K. Jung. Recognition of facial images using support vector machines. In *Proc. 11th IEEE Signal Processing Workshop on Statistical Signal Processing 2001*, pages 468–471, 2001.

[39] M. Lades, J. Vorbruggen, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. In *IEEE Trans. on Computers*, volume 42, pages 300–311, March 1993.

[40] S. Liu and M. Silverman. A practical guide to biometric security technology. *IEEE Computer Society, IT Pro - Security*, 2001.

[41] M. Loeve. Probability theory. Technical report, Van Nostrand, Princeton, 1955.

[42] J. Mao. A case study on bagging, boosting and basic ensembles of neural networks for ocr. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, volume 3, pages 1828–1833, 1998.

[43] A. Martinez and R. Benavente. The ar face database. Technical Report 24, Computer Vision Center, June 1998.

[44] J. E. Meng, S. Wu, J. Lu, and L. T. Hock. Face recognition with radial basis function (rbf) neural networks. In *IEEE Trans. on Neural Network*, volume 13, pages 697–710, May 2002.

[45] T. Mitchell. Source code from chapter 4 of machine learning. Software available at: `http://www-2.cs.cmu.edu/~tom/faces.html/`.

[46] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[47] Orl database. Database available at: `http://www.uk.research.att.com/facedatabase.html`.

[48] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[49] T. Pentland, A.and Choudhury. Face recognition for smart environments. *Computer*, 2:50–55, Feb 2000.

[50] H. S. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, Inc., 1999.

[51] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[52] L. Sirovich and M. Kirby. A low-dimensional procedure for the characterization of human faces. In *J. Opt. Soc. Amer. A*, volume 4, pages 519–524, 1987.

[53] L. I. Smith. A tutorial on principal component analysis, Feb 2002. http://kybele.psych.cornell.edu/˜edelman/Psych-465-Spring-2003/PCA-tutorial.pdf.

[54] M. Su and M. Basu. Gating improves neural network performance. In *Proc. IEEE Conf. on IJCNN '01*, volume 3, pages 2159–2164, 2001.

[55] H.-M. Tang, M. Lyu, and I. King. Face recognition committee machine. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, volume 2, pages 837–840, 2003.

[56] H.-M. Tang, M. Lyu, and I. King. Face recognition committee machines: Dynamic vs. static structures. In *Proceedings of The 12th International Conference on Image Analysis and Processing (ICIAP 2003)*, 2003.

[57] T. Terrillon, M. Shirazi, M. Sadek, H. Fukamachi, and S. Akamatsu. Invariant face detection with support vector machines. In *Proc. 15th International Conf. on Pattern Recognition 2000*, pages 210–217, 2000.

[58] V. Tresp. Committee machines. In *Handbook on Neural Network Signal Processing*. CRC Press, 2001.

[59] Sol universe limited. http://www.soluniverse.com.hk/TrueFace.htm.

[60] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[61] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[62] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.

[63] Viisage. http://www.viisage.com/facialrecog.htm.

[64] J. Wiskott, L.and Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 19, pages 775–779, July 1997.

[65] Yale database. Database available at: `http://cvc.yale.edu/frames.html`.

[66] M. Yang, D. Kreigman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, Jan 2002.

[67] J. Zhang, Y. Yong, and M. Lades. Face recognition: eigenface, elastic matching, and neural nets. In *Proc. IEEE*, volume 85, pages 1423–1435, 1997.

[68] W. Zhao. Face recognition: A literature survey. Technical report, National Institute of Standards and Technology, 2000.

[69] W. Zhao, R. Chellappa, and N. Nandhakumar. Discriminant analysis to principal components for face recognition. In *Nara*, pages 336–341, Japan, April 14-16 1998.

[70] W. Zhao, R. Chellappa, and N. Nandhakumar. Empirical performance analysis of linear discriminant classifiers. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 164–169, Santa Barbara, CA, 1998.

[71] Zn vision technologies ag. http://www.zn-ag.com.