AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY
Faculty of Information and Natural Science
Department of Information and Computer Science

**Jing Wu**

# Online Face Recognition with Application to Proactive Augmented Reality

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, May 25, 2010

| | |
|---|---|
| Supervisor: | Professor Erkki Oja |
| Instructor: | Markus Koskela, D.Sc. (Tech.) |

| AALTO UNIVERSITY SCHOOL OF<br>SCIENCE AND TECHNOLOGY | ABSTRACT OF THE<br>MASTER'S THESIS |
|---|---|

| | |
|---|---|
| **Author:** | Jing Wu |
| **Name of the Thesis:** | Online Face Recognition with Application<br>to Proactive Augmented Reality |
| **Date:** | May 25, 2010 **Number of pages:** ix + 64 |
| **Department:** | Department of Information and Computer Science |
| **Professorship:** | T-61 Computer and Information Science |
| **Supervisor:** | Professor Erkki Oja |
| **Instructor:** | Markus Koskela, D.Sc. (Tech.) |

Recently, more and more researchers have concentrated on the research of video-based face recognition. The topic of this thesis is online face recognition with application to proactive augmented reality. We intend to solve online single-image and multiple-image face recognition problems when the influence of illumination variations is introduced.

First, three machine learning approaches are utilized in single-image face recognition: PCA-based, 2DPCA-based, and SVM-based approaches. Illumination variations are big obstacles for face recognition. The next step in our approach therefore involves illumination normalization. Image preprocessing (AHE+RGIC) and invariant feature extraction (Eigenphases and LBP) methods are employed to compensate for illumination variations. Finally, in order to improve the recognition performance, we propose several novel algorithms to multiple-image face recognition which consider the multiple images as query data for subsequent classification. These algorithms are called MIK-NN, MMIK-NN and Kmeans+Muliple K-NN.

In conclusion, the simulation experiment results show that the LBP+$\chi^2$-based method efficiently compensates for the illumination effect and MMIK-NN considerably improves the performance of online face recognition.

Keywords: online face recognition, feature extraction, classification, machine learning, illumination normalization, recognition accuracy

# Acknowledgments

This Master's thesis has been completed in Content-Based Image and Information Retrieval group in Department of Information and Computer Science at Aalto University School of Science and Technology.

First, I would like to express my sincere gratitude to my instructor, Doctor Markus Koskela, for invaluable guidance, encouragement and providing important feedback throughout the thesis. He shared his inside-views with me that facilitated me to gain empirical knowledge. I am also grateful to my supervisor, Professor Erkki Oja, for being supportive, cooperative, proofreading the manuscript and giving invaluable comments on it.

Second, I want to thank Docent Jorma Laaksonen for trusting in me for summer training, which gave me the opportunity to work on this thesis. Many thanks also go to all my colleagues for their technical support and advice during the work, especially Zhirong Yang and Prem Raj Adhikari.

Finally, big thanks to my family and boy friend Marko Medjeral for always supporting me with belief and love. With their love, I am ready for any challenge.

Espoo, May 25th 2010

Jing Wu

# CONTENTS

# Abbreviations and Notations

| | |
|---|---|
| FR | Face Recognition |
| ML | Machine Learning |
| PR | Pattern Recognition |
| AR | Augmented Reality |
| PCA | Principal Component Analysis |
| LDA | Linear Discriminant Analysis |
| EBGM | Elastic Bunch Graph Matching |
| 2DPCA | Two-dimensional Principal Component Analysis |
| SVM | Support Vector Machine |
| HE | Histogram Equalization |
| AHE | Adaptive Histogram Equalization |
| GC | Gamma Correction |
| GIC | Gamma Intensity Correction |
| RGIC | Region-based Gamma Intensity Correction |
| LBP | Local Binary Pattern |
| K-NN | K nearest neighbors |
| MIK-NN | Multiple-image K-nearest neighbors |
| MMIK-NN | Modified Multiple-image K-nearest neighbors |
| $I$ | $M \times N$ image matrix |
| $\mathbf{I}$ | $1 \times MN$ image vector |
| $\mathbf{\Psi}$ | Mean image of a training set |
| $\mathbf{\Psi}'$ | Mean image in the MPEG-7 face descriptor |
| $\mathbf{\Phi}$ | Mean-subtracted image |
| $\mathbf{u}$ | Eigenfaces in PCA |
| $\omega$ | Principal Components of PCA |
| $\mathbf{\Omega}$ | Feature vector of an image $I$ |
| $V$ | The basis matrix in the MPEG-7 face descriptor |

| | |
|---|---|
| $\mathbf{U}$ | $N$-dimensional unit column vector |
| $X$ | Feature matrix in 2DPCA |
| $\bar{I}_{ij}$ | Canonical face image |
| $\gamma$ | Gamma parameter |
| $I_{ij}^{'}$ | Illumination normalized image |
| $\tilde{I}$ | Discrete Fourier transform (DFT) of an image $I$ |
| $\Phi$ | Phase spectrum of the image |

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# Introduction

During the past several decades, automatic face recognition (FR) has seen significant advances. Two major reasons contributing to the advances [1] are the rapid development of the available FR techniques and numerous applications to law enforcement, information security and commercial areas. For example, access control and video surveillance are two classical applications of law enforcement, desktop logon is a typical application for information security that can use face recognition to control access to the computer, and virtual reality is widely adopted for commerical entertainment. Moreover, research in FR has progressed due to the availability of a variety of facial image databases including the facial recognition technology (FERET) database, the CMU PIE database, and the ORL database [2] [3] [4], and systematic evaluation methods of the performance of FR algorithms, for instance the Face Recognition Vendor Tests (FRVT) 2002 and the Face Recognition Grand Challenge (FRGC) [5] [6].

The goal of FR is to automatically identify one or more persons from still images or video frames. The FR problem can be categorized into two approaches: still-image-based FR and video-based FR. Still-image-based FR often deals with controlled static images taken under fixed illumination conditions. This thesis, however, focuses on online face recognition which is a research branch of video-based FR. Unlike FR from still images, FR from video processes uncontrolled video frames for automatic FR in real time, which poses a large scope of technical challenges. These challenges include illumination and pose variations, motion and occlusion, high dimensionality of image data, and low quality of the video material. However, video-based FR also benefits from its nature in the following ways. First, a sequence of video frames contains abundant temporal information which indicates that the identities of the faces in the frames remain consistent over periods of time. Inspired by this idea, we used a new concept of online FR, multiple-image FR that takes an image sequence as query data. Second, video-based FR allows the face recognition system to learn and update the information about the subjects in real time [7].

Face recognition is an interdisciplinary research subject which borrows and builds upon ideas from image processing, machine learning and pattern recognition. Machine learning (ML) is a research field that focuses on the study of algorithms which support computers to learn from the seen examples to determine the class membership of unseen examples [8]. Pattern recognition (PR) is a closely related field which focuses on designing methods to classify data into different classes [9]. Face recognition is an application of the research in ML and PR. A three-dimensional (3D) face is characterized by a two-dimensional facial image and each person's face appears in the 2D image as a pattern which carries vast information about the face. With image processing techniques, and machine learning and pattern recognition algorithms, the pattern can be described by its features that should be distinctive among other patterns and invariant to illumination and pose variations. At this stage, online FR can be generalized to two steps: training and matching. A set of training features are used to train the models and query features are matched to the closest class among the training set.

This thesis is designed to solve online single-image FR and multiple-image FR problems when the influence of illumination variations is introduced, using ML and PR methods as well as image processing techniques. The focus of this thesis is, however, on online multiple-image FR.

## 1.1 Motivation for research

Our research in online FR is not only motivated by the technical challenges posed by the research problem, but also by its application to Augmented Reality (AR) which is one of the research branches of the Urban contextual information interfaces with multimodal augmented reality (UI-ART) project funded by Multidisciplinary Institute of Digitalisation and Energy (MIDE) programme at Aalto University School of Science and Technology.

### 1.1.1 UI-ART project background

The UI-ART project aims to devise novel sorts of contextual interfaces to information represented by multimodal augmented reality (AR). Motivated by Virtual Reality, AR supplements the reality by integrating $3D$ virtual objects into the real world so that the user would see the real and virtual worlds simultaneously [10]. In UI-ART, multimodal AR is the muscle which incorporates multiple modalities, for instance visual, auditory and human gesture modalities. Multimodal AR is a non-intrusive technique of characterizing the physical real-world environment with virtual auxiliary elements. It can augment the information associated to the physical scene from the reality by adding some explanatory visual or auditory data such that a better perception of the scene will be obtained, for example what is the inner structure of a building in the scene, the detailed information of the people in the scene, and how to reach a particular place. Therefore it is critical for AR to fit the virtual information seamlessly to the physical word while the information is easily perceived and will not disturb the original space perception. Another problem is retrieving contextual information in the way that the context-dependent search engine should efficiently return the information that the specific user is interested in the particular context.

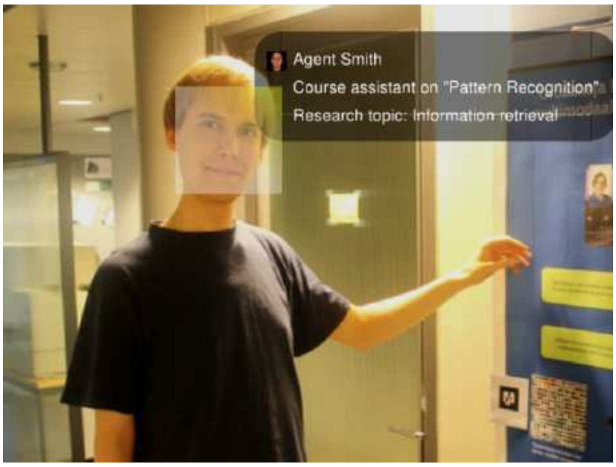
Agent Smith
Course assistant on "Pattern Recognition"
Research topic: Information retrieval

Figure 1.1: A screenshot from virtual laboratory guide shows how online face recognition works. This picture is originally from [11].

### 1.1.2 Applicability of Face Recognition

Online face recognition is a task considered in one of the applications in the UI-ART project, which is the *virtual laboratory guide* [11]. Virtual laboratory guide is an AR guide to a visitor at a department in an university. It navigates the visitor to find out more information about the researchers and research projects and presents the virtual information about the people and objects on a see-through display. Two display devices are currently used for the guide: a wearable near-to-eye display with an integrated gaze tracker, and a hand-held Sony Vaio computer with virtual see-through display. Online face recognition can then facilitate a user who is equipped with a display device to perceive augmented information about the people nearby. For example, when the user walks the corridors of Adaptive Information Research Center, the see-through display will show the presence of persons in reality and more importantly augmented information, such as the person's name, research activities, publications, teaching, office hours and links to the person's external web pages, including personal homepages, Facebook and LinkedIn [11]. Figure 1.1 is a screenshot from the see-through display that illustrates that the system has recognized Agent Smith's face and is showing augmented information about Smith, for instance that he is a course assistant on "Pattern Recognition" and interested in research in information retrieval.

## 1.2 Contributions of the Thesis

The work done for this thesis contributes to the project in the following ways:

- Investigation of the use of online single-image face recognition methods and illumination normalization techniques.
- Development of novel yet simple methods for online multiple-image face recognition that support recognition of image sequences from video frames.
- A set of experiments with the proposed algorithms in order to select the most reliable ones for online FR systems.

## 1.3 Thesis outline

Chapter 1 introduces the topic of this thesis, the motivation of our research and the major contributions of this thesis work. In Chapter 2, we review the previous work accomplished in the field, argue the distinction between our work and the previous works, and demonstrate the basic theory of online face recognition. Chapter 3 describes the algorithms of online single-image face recognition. Chapter 4 introduces the illumination variation problem and further proposes several normalization methods. Chapter 5 presents the idea of online multiple-image face recognition, and describes additional novel algorithms.

Chapter 6 gives a brief description of the experiments and draws the conclusions from the results. Finally, Chapter 7 summaries the content of this thesis and discusses possible further work in this research.

# CHAPTER 2
# Face Recognition: Brief Overview

The history of face recognition (FR) research as an engineering field dates back to the 1960s [12]. Since then, numerous algorithms have been developed for still-image-based FR [1], such as Principal Component Analysis (PCA) [13], Linear Discriminant Analysis (LDA) [14], Elastic Bunch Graph Matching (EBGM) [15] and Support Vector Machines (SVM) [16]. The research of still-image-based FR is a well established field. Moreover, ever since psychological studies have recently shown that the facial dynamics are considerably useful for face recognition especially in poor image quality scenarios [17], the research in video-based FR has been boosted.

## 2.1 Still-image-based face recognition

Our online single-image face recognition system is implemented using still-image-based algorithms. In [13], PCA was referred to as Eigenfaces. It projects an image to a lower dimensional subspace defined by a set of orthonormal basis vectors that account for the maximum variance of the projected images. The projected images are represented by sets of principal components which are used for classification. In our work, the MPEG-7 face descriptor is used to extract the features and it differs from [13] in the way it generates the set of basis vectors using both an original and a flipped face images [18]. Later it has been shown that SVM can achieve higher accuracy than Eigenfaces for face recognition [16]. SVM was originally designed as a binary classifier but it has been extended to perform multi-class classification with e.g. one-against-one strategy. In our method, the features of the images are extracted using the MPEG-7 face descriptor and two multi-class classifiers (one-against-one and one-against-all) are incorporated into the SVM method. Recently some researchers have proposed a 2DPCA method for efficient feature extraction. 2DPCA projects the image using the basis vectors of the covariance matrix constructed by the original image matrices [19]. 2DPCA method is also integrated into our online face recognition system.

Illumination variations are big obstacles for face recognition. In recent years, a variety of approaches has been proposed as solutions to the illumination variation problem in face recognition contexts. In general, those approaches fall into three categories: face modeling [20] [21], preprocessing and normalization [22] [23], and invariant feature extraction [24] [25]. The disadvantage of face modeling-based approaches is the requirement of training images under varying lighting conditions or 3-D shape information [26]. This disadvantage leads to the limitation of its application in practical face recognition systems. The early works that were done for preprocessing and normalization used image preprocessing algorithms to compensate for uniform illumination, for example histogram equalization (HE), gamma correction (GC), logarithm transform, etc. Those methods inevitably could not perform well when nonuniform illumination is introduced. In [23], region-based HE and gamma intensity correction (RGIC) methods were presented to cope with nonuniform illumination variations. In our method, we use the combination of adaptive histogram equalization (AHE) and RGIC where AHE was used to enhance an image's local constrast [22]. Moreover, local binary pattern (LBP) is a widely accepted algorithm for extracting features invariant to illumination variations in face recognition [24]. However, in [27] it was proven that distance transform (DT) could perform better than Chi-square statistics in LBP-based methods. We compare LBP with Chi-square statistics and LBP with DT on our datasets to verify the performance of the LBP-based method. Eigenphases [25] which have been proposed to efficiently recognize faces under illumination variations is also utilized in our experiments.

Multiple-image face recognition is a new subject in face recognition field and there are few related published studies. In [28], a framework of the Active Appearance Model is constructed to embrace the image variation over a sequence to improve the recognition performance. Lately, in [29] distinct approaches which used a sequence of images for robust recognition were studied and it has been proven that the classification task could be solved based on Kullback-Leibler divergence by considering it as a statistical hypothesis testing task. We have developed three novel but simple methods for multiple-image face recognition, including multiple-image $k$-nearest neighbors (MIK-NN), modified multiple-image $k$-nearest neighbors (MMIK-NN) and $K$-means with multiple $k$ nearest neighbors (K-means + Multiple K-NN). MIK-NN directly operates on the consecutive images with multiple $k$-nearest neighbors classifiers and uses the majority rule strategy to find out the most frequent class in the classification decisions of the image sequence. MMIK-NN modifies the idea of MIK-NN to integrate diverse images into the image sequence for higher recognition accuracy. Kmeans + Multiple K-NN applies clustering to obtain diverse images in the image sequence.

## 2.2 Online face recognition

Online face recognition is a real-time recognition task which aims to recognize incoming faces in video frames. Similar to face recognition based on static images, online face recognition can be implemented by using a machine learning-based approach. Each person's face is a pattern which contains large amount of information of the face, for example the location and size of the facial components such as eyes, nose, mouth and so on. The goal of online face recognition is to identify the class membership for an unknown facial pattern given some known facial patterns in video frames. In such a case, we need a large set of $N$ reference images $I_1, I_2, I_3, ..., I_N$, called a *training set*. The reference images are also called training images. Each image contains one known person's face.

In face recognition, a three-dimensional ($3D$) face is often treated as a two-dimensional ($2D$) image. The main idea of machine learning is to build up a model for estimating the identity of the persons who appear in the new face images. The model can be formulated as a function $y(x)$ determined based on the training images during the *training phase*, also known as *learning phase* [8]. Once the model has been trained using the training examples, it can then be used to determine the identity of new face images. The new face image is usually denoted as the *query image* or the *test image*. Typically, it is crucial to *preprocess* the input image to remove redudant or irrelevant information and transform it into reduced form. The reduced form is termed a *feature* representation that should be distinctive for subsequent classification and robust to changes in image scale, illumination and pose. This stage is often called *preprocessing* or *feature extraction*. The query image should also be preprocessed with the same feature extraction [8].

An online single-image face recognition system generally consists of six components: *face detection*, *face tracking*, *face alignment*, *illumination normalization*, *feature extraction* and *face recognition*. This idea originally comes from [30]. As sketched in Figure 2.1, *face detection* segments the face regions from the background by finding the locations and sizes of all faces in the video frames. *Face tracking* is required to track the detected faces since the subjects most likely move their heads. Face detection and tracking are able to determine which are the faces and which are non-faces and roughly estimate the location and size of each detected face. Face alignment is thereby introduced to achieve more accurate positioning of the detected faces. The alignment of the detected faces in the video can be done by using geometrical transforms. Then after the detection and tracking of the location of facial components, the face image is normalized in terms of geometrical properties, including size and pose [30]. Changes in lighting are big obstacles for face recognition, and the face image has to be further normalized in terms of illumination.

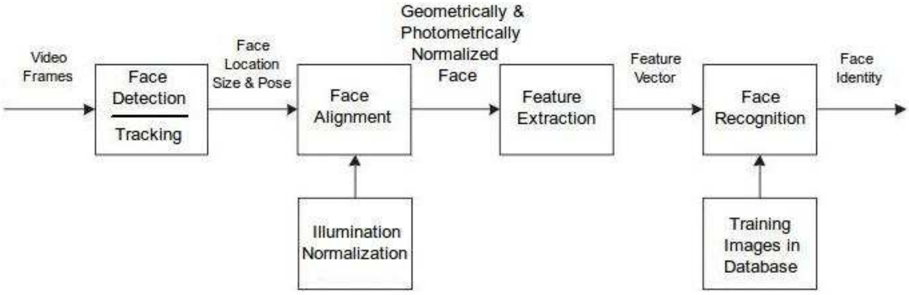
Video Frames
Face Detection
Tracking
Face Location Size & Pose
Face Alignment
Geometrically & Photometrically Normalized Face
Feature Extraction
Feature Vector
Face Recognition
Face Identity
Illumination Normalization
Training Images in Database

Figure 2.1: Architecture of online single-image face recognition, adapted from [30].

After geometrical and photometrical normalization, the features of the query face image are extracted by *feature extraction*. The features should be distinctive so that the face of the person in the query image would be easily distinguished from the faces of other persons. The features are normally represented as a feature vector. In the online single-image face recognition system, face detection, face tracking, face alignment, illumination normalization and feature extraction components operate approximately simultaneously.

Before *face recognition*, a training set which contains diverse images of the persons should be selected and processed with feature extraction. The corresponding training feature vectors are used for training the models. After the training phase, an one-to-many matching method will be performed on the extracted feature vector of the query face image during online face recognition. The method compares the feature vector of the query image against the feature vectors associated with the training set to find out the best match. If a sufficiently good match is found, the system outputs that the identity of the person represented in the query image is the identity of the best matched training image. Otherwise it indicates an unknown face. The functionality of recognizing unknown faces is not available in our current online face recognition.

Since the central topic of this thesis is face recognition, no further discussion of face detection, tracking and alignment will be included in the following chapters. The prior detection, tracking and alignment are therefore always assumed.

# CHAPTER 3 Online Single-image Face Recognition

As online single-image face recognition is an application of machine learning, we can therefore tackle it by using unsupervised and supervised learning approaches. Unsupervised learning is a machine learning approach in which a model is developed to fit the observed data [31]. The training data consists of a set of input objects with unknown target outputs [31]. Unsupervised learning methods often address three problems, including *clustering*, *density estimation* and *dimension reduction* which is used in our face recognition system to efficiently extract the features of the face images [9]. Supervised learning is, on the other hand, a machine learning technique for creating a model to predict the output pattern given the training data [8]. The training data consists of both the input objects and the desired outputs. In general, supervised learning involves two tasks, *regression* and *classification*. Regression requires the model to predict continuous variables, while classification needs the model to predict a class label of the input object. Classification is frequently used in pattern recognition, such as digit recognition and face recognition [8]. Unsupervised learning is often distinguished from supervised learning by the fact there is only unlabelled data used in unsupervised learning.

In this work, two unsupervised learning algorithms have been implemented for online single-image face recognition. They are principal component analysis (PCA) [13] and two-dimensional PCA (2DPCA) [19]. Support vector machines (SVMs) [16], a classical supervised learning algorithm, has also been used in our single-image face recognition system.

## 3.1 PCA-based approach

Principal component analysis (PCA) is a well known unsupervised learning technique used for dimension reduction and feature extraction [13]. The main idea in using PCA for face recognition is to project the face image $\mathbf{I}(\mathbf{I} \in R^D)$ onto a subspace of lower dimensionality $D^{'}(D^{'} < D)$ spanned by a set of basis vectors which correspond to the maximum variance of the projected images. In mathematics, it has been proven that the basis vectors are orthonormal and they are the eigenvectors of the covariance matrix of the set of face images [8]. In face recognition, the eigenvectors are often referred to as *Eigenfaces* and the lower dimensional subspace is called the *face space*. Each projected image in the set is then a linear combination of the eigenfaces that associate with the largest eigenvalues. Therefore, those eigenfaces can be considered to characterize the maximum variance of the projected images. Each original image is approximated by its project image. In our single-image online face recognition system, PCA is incorporated in the MPEG-7 face descriptor which has been widely adoped as a visual descriptor [18]. Based on PCA, MPEG-7 represents a facial image by a linear combination of a set of 48 basis vectors derived from the eigenfaces of the covariance matrix constructed by the training images.

### 3.1.1 Calculating Eigenfaces

A 2D image of size $N$ by $M$ pixels can be represented as a vector $M \times N$ by concatenating each row or column into a row or column vector. Let the training set of face images be $\{\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3, ...\mathbf{I}_n\}$, and each image to be in the vector form. The mean image of the training set is denoted as

$$\mathbf{\Psi} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_i.$$

The training images are then normalized to have zero mean so that they describe how each image differs from the mean image

$$\mathbf{\Phi}_i = \mathbf{I}_i - \mathbf{\Psi}, \quad i = 1, 2, 3....n.$$

The goal of PCA is to seek an optimal set of orthonormal eigenvectors of the covariance matrix of the set of training images. The covariance matrix is defined as

$$C = \frac{1}{n}\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{\Phi}_i^T.$$

Eigenfaces $\mathbf{u}_i$ $(i = 1, 2, 3...n)$ are the possible eigenvectors of the covariance matrix $C$. Note that the number of eigenfaces is the same as the number of nonzero eigenvalues of $C$, which is equal to the number of images in the training set. In practice, $n^{'}$ $(n^{'} < n)$ eigenfaces are sufficient for identification so that eigenfaces $\mathbf{u}_i$ $(i = 1, 2, 3..., n^{'})$ associated to the first $n^{'}$ largest eigenvalues are selected [13].

An image $\mathbf{I}$ is then projected onto the face space by a linear combination operation

$$\omega_i = \mathbf{u}_i^T(\mathbf{I} - \mathbf{\Psi}), \quad i = 1, 2, 3....n^{'},$$

where $\omega_i$ are the principal components of PCA that weight the contribution of each eigenface in representing the original face image [13]. The set of principal components form a vector $\mathbf{\Omega}^T = [\omega_1, \omega_2, \omega_3, ..., \omega_{n'}]$, which is usually called the *weight vector* or the *feature vector*.

### 3.1.2 MPEG-7 face descriptor

As mentioned above, our online face recognition system uses the MPEG-7 face descriptor to extract and describe the features of the facial images. Before feature extraction, a face image should be geometrically normalized. The normalized image is obtained by scaling the original image to 56 by 46 pixels such that the centers of the two eyes in each image are at (24st row, 16st column) and (24th row, 31st column) [18].

The construction of the basis vectors used in the MPEG-7 face descriptor is similar to that in PCA. After normalization, each image is represented as a one-dimensional face vector. The face vectors of the training set are then applied to generate 48 eigenfaces corresponding to the 48 largest eigenvalues. The training set is not from our database but is defined by the International Organization for Standardization. The MPEG-7 face descriptor represents a face image by the features of both the flipped face image and the original image. Since a face is symmetrical, the flipped face image can also be used to describe the face appearing in an image. In this case, an image can be approximated by a linear transformation of the features associated with both the original image and the flipped image. In addition, a weight is introduced to the feature of the flipped image to distinguish between the original image and the flipped image [18].

Let $\mathbf{I}$ be the original image, $\mathbf{I}^{'}$ be its flipped image and $\mathbf{u}_j$ $(j = 0, 1, 2, ...47)$ the 48 eigenfaces. Then the features of an input face image are formulated as

$$\omega_j = \frac{[\omega_j(\mathbf{I}) + c\omega_j(\mathbf{I}^{'})]}{1 + c},$$

where c is the weight assigned to the feature vector of the flipped image. We can denote $\omega_j(\mathbf{I}^{'})$ in two ways, $\mathbf{u}_j^T\mathbf{\Phi}^{'}$ and $\mathbf{u}_j^{'T}\mathbf{\Phi}$. Here, $\mathbf{u}_j^{'}$ is the flipped eigenface of $\mathbf{u}_j$, $\mathbf{\Phi}$ is the original image normalized to have zero mean, and $\mathbf{\Phi}^{'}$ is the flipped image normalized to have zero mean. Thus, the above equation can be further deduced as

$$\begin{aligned}\omega_j &= \frac{[\omega_j(\mathbf{I}) + c\omega_j(\mathbf{I}^{'})]}{1+c} \\ &= \frac{\mathbf{u}_j^T\mathbf{\Phi} + c\mathbf{u}_j^{'T}\mathbf{\Phi}}{1+c} \\ &= \frac{(\mathbf{u}_j + c\mathbf{u}_j^{'})^T\mathbf{\Phi}}{1+c} \\ &= \mathbf{v}_j^T\mathbf{\Phi}, \quad j = 0, 1, 2, ..., 47,\end{aligned}$$

where $V = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{47}]$ is the basis matrix in the MPEG-7 face descriptor, which is specified in Annex A in [18]. In fact, the MPEG-7 face descriptor extracts the features of an image by a linear transformation of the original eigenfaces and the flipped eigenfaces.

After the projection of a one-dimensional face vector $\mathbf{I}$ onto the face space, the features are described by the feature vector $\boldsymbol{\Omega^T} = [\omega_0, \omega_1, \omega_2, ..., \omega_{47}]$ where $\mathbf{\Omega}^T = V^T(\mathbf{I} - \mathbf{\Psi}^{'})$, and the mean image $\mathbf{\Psi}^{'}$ is defined in Annex A in [18]. There are additional normalization and quantization steps for the feature vectors before they are applied to recognition:

$$\boldsymbol{\omega}_j = \begin{cases} -128, & \text{if } \boldsymbol{\omega}_j/Z < -128, \\ 127, & \text{if } \boldsymbol{\omega}_j/Z < 127, \\ \boldsymbol{\omega}_j/Z, & \text{otherwise,} \end{cases}$$

where the normalization constant $Z = 16384$. In the context of online single-image face recognition, each training image is transformed into its feature vector by using the basis vectors from the MPEG-7 face descriptor and is then normalized and quantified.

### 3.1.3 Face recognition

The test images are also transformed into feature vectors and the feature vectors are further processed with normalization and quantization steps. Let the normalized feature vector be $\mathbf{\Omega}^T = [\omega_0, \omega_1, \omega_2, ..., \omega_{47}]$. The vector is then used to search for which face class in the training set the test image belongs to. A common way to determine it is to find the face class $k$ that minimizes the Euclidean distance of the feature vector and the $k$th training image.

$$d(\mathbf{\Omega}, \mathbf{\Omega}_k) = \|\mathbf{\Omega} - \mathbf{\Omega}_k\|^2,$$

where $\mathbf{\Omega}_k$ is the feature vector describing the $k$th training image. In fact, the face recognition task is regarded as a classification task. The classification technique used here is nearest neighbor classification [32] – a query image is assigned to the class of the training image which is nearest to the query image in the face space. Hence, suppose the training feature vectors in the database are $\mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3, ....\mathbf{\Omega}_n$ (where $n$ is the number of training face images) and each of the training images is assigned to the class $c_k$. Given a test image $\mathbf{\Omega}$, if

$$d(\mathbf{\Omega}, \mathbf{\Omega}_l) = \arg\min_j d(\mathbf{\Omega}, \mathbf{\Omega}_j), \quad j = 1, 2, 3..., n,$$

and $\mathbf{\Omega}_l \in c_k$, then

$$\mathbf{\Omega} \in c_k.$$

In order to have a clear picture of a online single image face recognition system based on PCA, the implementation steps are provided below:

1. Training step: obtain the feature vectors of the training images by using the MPEG-7 face descriptor and store them in the database.

2. Online feature extraction step: detect the new incoming image and project it onto the face space.

3. Online face recognition step: compare the feature vector of the new face image with all the feature vectors in the database, and find the closest facial class which is the identity of the new image.

PCA has been widely adopted in many areas of pattern recognition and computer vision. However, it has the weakness that there are two major assumptions it is mainly based on [33]. First, a linear projection can efficiently reduce the dimensionality of the data. Second, the projected data retains most of the information of the original data. These assumptions are not always met, therefore more advanced techniques are needed.

## 3.2 2DPCA-based approach

Two-dimensional principal component analysis (2DPCA) is a novel method developed for image feature extraction. It was proposed to extract image features computationally more efficiently than PCA [19]. Inspired by the idea of PCA, 2DPCA characterizes the features of an image by using the orthonormal eigenvectors derived by the covariance matrix of a set of 2D image matrices. Unlike PCA, there is no need to transform the image matrix into a vector before feature extraction in 2DPCA.

### 3.2.1 Basic idea

Let $I$ be an image matrix of size $M$ by $N$ and $\mathbf{U}$ be an $N$-dimensional unit column vector. The idea of 2DPCA is to construct a good projection of $I$ onto $\mathbf{U}$ while maximizing the variance of the projected image. The $M$-dimensional projected vector $\mathbf{\Omega}$ is a simple linear transformation

$$\mathbf{\Omega} = I\mathbf{U}.$$

Essentially, the vector $\mathbf{U}$ corresponds to the eigenfaces in PCA. Thus, a good projection stems from the selection of the projection vector $\mathbf{U}$. The total scatter of the projected image is introduced to help us to choose the optimal vector $\mathbf{U}$. The total scatter of the projected image can be described by the trace of the covariance matrix of the projected feature vectors. Let $C_u$ be the covariance matrix of the projected feature vectors of the training images and $tr(C_u)$ be the trace of $C_u$ which can be written as

$$\begin{aligned} C_u &= E(\mathbf{\Omega} - E\mathbf{\Omega})(\mathbf{\Omega} - E\mathbf{\Omega})^T \\ &= E[(I - EI)\mathbf{U}][(I - EI)\mathbf{U}]^T. \end{aligned}$$

Thus, the trace of $C_u$ is

$$\begin{aligned} tr(C_u) &= \mathbf{U}^T[E(I - EI)^T(I - EI)]\mathbf{U} \\ &= \mathbf{U}^T G_t \mathbf{U}, \end{aligned}$$

where $G_t$ is the covariance matrix of the training images. Similar to the analysis of PCA, the maximum of $tr(C_u)$ is obtained when the projection vector $\mathbf{U}$ is the orthonormal eigenvector of $G_t$ associated with the largest eigenvalue [8]. In general, only one optimal projection direction is not sufficient [19]. Therefore, a set of eigenvectors, $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, ..., \mathbf{U}_D$, corresponding to the first $D$ largest eigenvalues is selected to define the projection direction,

$$\arg\max tr(C_u) = \mathbf{U}_i, \quad i = 1, 2, 3, ..., D.$$

### 3.2.2 Feature extraction and face recognition

Consequently, it is necessary to operate an experiment of generating the best set of eigenvectors, see Chapter 6 for details. After the construction of the maximum projection direction, the 2D image $I$ is projected onto $\mathbf{U}$ and the projected vector $\mathbf{\Omega}$ is written as

$$\mathbf{\Omega}_i = I\mathbf{U}_i, \quad i = 1, 2, 3, ..., D.$$

A set of projected feature vectors, $\mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3, ..., \mathbf{\Omega}_D$, is yielded by the linear transformations of $\mathbf{U}$. Each vector is a principal component of the image $I$. Another difference of 2DPCA and PCA is reflected on the fact that each principal component of 2DPCA is a vector whereas that of PCA is a scalar. A scalar can not contain as much information as a vector. Therefore, we can come to the conclusion that the projected data contains more information of input data in 2DPCA compared to PCA. In other words, 2DPCA tends to project more information of the original data. In [19], researchers have observed that the dimensionality of the 2DPCA feature vector is always much higher than in PCA. They further provided the solution that the dimensionality can be reduced by implementing PCA after 2DPCA.

The set of principal component vectors can be represented as an $M \times D$ feature matrix $X$. The projection should be performed on the whole training set. Suppose there are $n$ training images, the projections of these $n$ images yield a set of feature matrices $X_1, X_2, X_3, ...X_n$ and each feature matrix is of the form $X_i = [\mathbf{\Omega}_1^i, \mathbf{\Omega}_2^i, \mathbf{\Omega}_3^i, ..., \mathbf{\Omega}_D^i]$. After detection and feature extraction of the new image, the nearest neighbor classifier is employed again for classification. Let the feature matrix of the image be $X_j = [\mathbf{\Omega}_1^j, \mathbf{\Omega}_2^j, \mathbf{\Omega}_3^j, ..., \mathbf{\Omega}_D^j]$, the Euclidean distance is utilized again for determining the nearest facial class the new image belongs to [19],

$$d(X_j, X_i) = \sum_{k=1}^{D} \left\| \mathbf{\Omega}_k^j - \mathbf{\Omega}_k^i \right\|^2 .$$

The criterion of choosing the nearest neighbor is adopted to determine the minimum distance between the new image and one of the training images, and identification is the way to assign the new image to the nearest facial class. The implementation of 2DPCA based online single-image face recognition is fairly similar to that of PCA based, thus we will not discuss it here.

## 3.3 SVM-based approach

Support Vector Machine (SVM), a supervised learning method, was originally designed as a binary classifier [34]. Some researchers have recently proposed that it can also be an effective approach for face recognition [16]. A binary SVM constructs a hyperplane that separates two classes in the feature space and maximizes the distance between the hyperplane and either class. The hyperplane is known as the optimal separating hyperplane (OSH) and the distance is termed the margin [34]. SVM has also been extended to tackle the multi-classification problem where several binary SVMs are trained to predict the class label of a query image [35].

### 3.3.1 Feature Representation

Before the implementation of the SVM classifiers, it is necessary to detect the face images and extract the features. The MPEG-7 face descriptor is used again here for feature extraction. Thus, the training images are all projected onto a lower dimensional face space. For the training phase, it is indispensable to incorporate the class information of the training images. Clearly, the difference of the training data between SVMs and PCA lies on the class labels. In other words, the training data of PCA consists only of the weighted feature vectors while the training data of SVM contains not only the feature vectors but also the associated class information.

### 3.3.2 Binary SVM Classifier

Let us first discuss the linearly separable case. Suppose the training set in the matrix form $\Omega = [\mathbf{\Omega}_1, \mathbf{\Omega}_2, ...\mathbf{\Omega}_n]$ contains $n$ training feature vectors that belong to two classes in an $N$-dimensional space,

$$\Omega = \left\{(\mathbf{\Omega}_i, l_i) | \mathbf{\Omega}_i \in R^N, l_i \in \{-1, +1\}\right\}_{i=1}^{n},$$

where $l_i$ is the class label or the class indicator, either -1 or +1, indicating the class to which the feature vector $\mathbf{\Omega}_i$ belongs. The goal of linear SVM is to find a linear hyperplane which partitions those feature vectors into the negative class and the positive class with maximum margin. This hyperplane is written as

$$\mathbf{w}^T \Omega + b = 0,$$

where $\mathbf{w}$ is a vector orthogonal to the hyperplane. Simply using geometry, the margin is observed as $\frac{2}{\|\mathbf{w}\|}$. Therefore, the task of maximizing the geometric margin can be considered as the task of minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. The OSH should satisfy the following constraints so that the training data is linearly separable:

$$\begin{cases} \mathbf{w}^T \mathbf{\Omega}_i + b \geq +1, & \text{for } l_i = +1, \\ \mathbf{w}^T \mathbf{\Omega}_i + b \leq -1, & \text{for } l_i = -1. \end{cases}$$

The constraints can be formulated into one set of inequalities [34]

$$l_i(\mathbf{w}^T \mathbf{\Omega}_i + b) \geq 1, \quad i = 1, 2, 3, ...n.$$

The standard formulation of SVMs can then be described as a minimization problem under constraints as follows. Find a vector $\mathbf{w}$ and a parameter $b$ such that $\frac{1}{2}\|\mathbf{w}\|^2$ is minimized, subject to $l_i(\mathbf{w}^T \mathbf{\Omega}_i + b) \geq 1$.

Solving the above optimization problem is equal to solving the dual problem where the Lagrange multipliers $\alpha_i$ are associated with each constraint

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}} \{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [l_i(\mathbf{w}^T \Omega - b) - 1]\}.$$

The solution of the optimization problem defines a linear OSH

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i l_i \boldsymbol{\Omega_i},$$

$$b = l_i - \mathbf{w}^T \boldsymbol{\Omega_i}.$$

The linear OSH formulates a classification function, which can be written as

$$f(\Omega) = \text{sign}(\mathbf{w}^T \Omega + b).$$

When the training data is not linearly separable, slack variables $\xi_i$ should be introduced. The OSH which encompasses both the separable and non-separable cases is then considered as an optimization problem

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i,$$

where $C$ is a predefined parameter and the optimization problem is subject to the constraints [16] [34]

$$l_i(\mathbf{w}^T \boldsymbol{\Omega}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ i = 1, 2, 3, ...n.$$

In practice, the non-linear case should also be considered where the OSH is nonlinear. With SVM, the non-linear case is tackled by first mapping the data vectors $\Omega$ to a high dimensional feature space $\varphi(\Omega)$ and then constructing an OSH in the feature space

$$\mathbf{w}^T \varphi(\Omega) = 0,$$

where $\mathbf{w} = \sum_{i=1}^{n} \alpha_i l_i \varphi(\boldsymbol{\Omega_i})$. By defining a *kernel function* as

$$K(\Omega, \boldsymbol{\Omega_i}) = \varphi^T(\Omega)\varphi(\boldsymbol{\Omega_i}),$$

the OSH is defined as

$$\sum_{i=1}^{n} \alpha_i l_i K(\Omega, \boldsymbol{\Omega_i}) = 0.$$

SVM is inherently used for binary classification, but several methods have been proposed to extend SVMs to solving multi-classification problems, such as the one-against-all method and the one-against-one method [35]. In many papers, it has been stated that one-against-one usually achieves higher accuracy in recognition due to the unbiased nature of one-against-all. On the other hand, compared to one-against-all, one-against-one is computationally more expensive [35] [36]. We also performed experiments comparing these two methods, see Chapter 6 for details.

### 3.3.3 Multi-class Classifier

The one-against-all classifier constructs $k$ SVM models to separate one class from the rest of the classes, where $k$ is the number of classes in the training set [35]. The $i$th SVM model is trained so that the images in the $i$th class are assigned with positive labels (+1) and the images of the remaining classes with negative labels (−1). The $i$th model is then used to predict a class label for an new image $I$. If the predicted label is +1, then the vote of the $i$th class will be incremented by one. Those steps are conducted for each model. Eventually, $I$ is classified to the class with the largest number of votes. Figure 3.1 (a) shows an example of the one-against-all method. There are four facial classes $A, B, C, D$ so that the SVMs produce four models. In each model, one of classes is labeled as the positive class and the rest of classes as the negative class. For instance, the first model illustrates that class $A$ is assigned with positive label (+1) and classes $B, C, D$ with the negative label (−1). Then, a new test image is compared with four models in turns, and each time it will be assigned to a class and the corresponding class will gain one vote. If eventually $A$ has the majority of the vote, the coming test face belongs to $A$.

The one-against-one classifier constructs $k(k-1)/2$ SVM models to classify between each pair of classes [35]. Each $i$th SVM model is obtained by training with images from two classes $i$ and $j$. Between these two classes, one is a positive class and the other is a negative class. The “Max Wins Voting” strategy is employed to discover to which class the test image belongs. The vote for $i$th class will be increased by one if the SVM model determines the image $I$ belongs to the $i$th class. Otherwise, the $j$th class will gain this vote. As a result, the test image $I$ is assigned to the class with the largest number of votes. Figure 3.1 (b) illustrates the one-against-one scheme. Four facial classes $A, B, C, D$ lead to that 6 SVMs models need to be constructed. Each model consists of two classes in which one is labeled as the positive class and

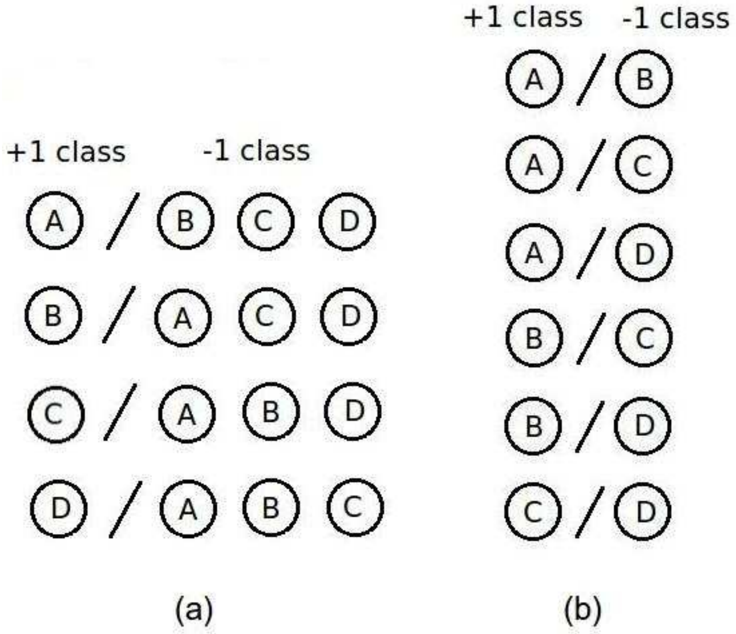

Figure 3.1: (a) illustrates the idea of the one-against-all classifiers. (b) depicts the one-against-one scheme.

the other as the negative class. For example, on the topmost SVM, class $A$ is the positive class while class $B$ is the negative one. When a new face image is detected, its feature vector will be processed by all 6 models. The class which wins the majority voting is the identity of the new face image.

### SVM-based Online Single-image Face Recognition

1. Feature extraction phase: The MPEG-7 face descriptor is used to extract the feature vectors of the training images. In addition, the class information is also provided to these training images.

2. Offline training step: The one-against-one approach is used to train the system to learn all the SVMs models between each possible pair of classes.

3. Online face recognition step: the incoming face image is detected and its features are represented as a feature vector by using the MPEG-7 descriptor. The system will compare the feature vector with each established SVM model and the class which has the largest number of votes is the identity of the incoming face.

# CHAPTER 4
# Illumination Normalization

Recognition of faces is a complex task for a computer. Nowadays, the state-of-the-art algorithms have achieved the recognition accuracy of about 90% when there is minimal variation in the face images [37]. However, when variations in pose or illumination are introduced, humans' ability to recognize faces is remarkable compared to computers which often achieve only poor recognition performance.

The performance of face recognition is heavily subject to varying illumination and pose. Illumination variations can yield large variability in facial appearance, as illustrated in Figure 4.1. We cope with the illumination effect by using three methods: the combination of adaptive histogram equalization (AHE) and region-based gamma intensity correction (RGIC) [22] [23], Eigenphases [25], and local binary pattern (LBP) [24].

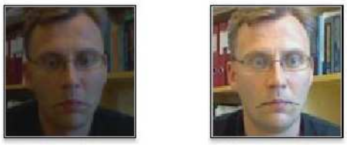

Figure 4.1: The same person can appear fairly different under varying lighting conditions.

In the architecture of an online face recognition system, illumination normalization as photometrical normalization should be performed after face alignment, as illustrated in Figure 2.1 in Chapter 2. In order to find out the best illumination normalization approach, the proposed algorithms have been implemented and experimented with. The comparison results are discussed in

Chapter 6.

## 4.1 AHE+RGIC

Histogram Equalization (HE) [38] is a fundamental image processing technique used to enhance the contrast of an image. Let $I_{ij}$ denote the intensity values of the pixels in any input face image captured under unknown lighting conditions. In HE, a transform function $T$ is applied on intensity levels $I_{ij}$ to obtain intensity levels $I^{'}_{ij}$ in the output image so that the histogram of the output image is equalized to be approximately constant:

$$I^{'}_{ij} = T(I_{ij}).$$

In fact, the transformation maps each pixel with intensity $I_{ij}$ in the input image into a pixel with the intensity value $I^{'}_{ij}$ in the output image [38]. This achieves an equalized histogram on which the intensity levels of the equalized image are better distributed. The equalized image shows that this method enables areas of low local contrast to gain a higher contrast. HE works effectively on global contrast enhancement, but it is less effective when the contrast characteristics vary across the image [26]. Later, Adaptive Histogram Equalization (AHE) was developed to overcome this drawback of HE by operating on small regions of the image, rather than on the entire image [22]. In this case, multiple histograms are obtained and equalized so that each region's constrast is enhanced respectively. Finally, the multiple histograms are combined to redistribute the intensity values of the image. Consequently, AHE is capable of improving an image's local contrast.

Gamma Intensity Correction (GIC) was developed to compensate for global brightness changes in a face image. The idea of GIC is based on traditional gamma correction (GC) which uses the gamma parameter $\gamma$ to control the overall brightness of an image displayed accurately on a computer screen [38]. GC is typically defined by a power-law transformation

$$\begin{aligned} I^{'}_{ij} &= G(I_{ij}, \gamma) \\ &= c{I_{ij}}^{1/\gamma}, \end{aligned}$$

where $I_{ij}$ is the intensity value of the pixels in any input face image, $I^{'}_{ij}$ is the pixel value of the resulting gamma-corrected image and $c$ is a positive constant. If an image is not properly corrected by GC, it can appear either bleached out or possibly too dark [38]. Thus, the method involves the problem of selecting an optimal $\gamma$. GIC was proposed to address the problem of choosing the value of $\gamma$ [23]. It assumes a predefined canonical face image $\bar{I_{ij}}$ is captured under natural lighting conditions. Any face image $I_{ij}$ will be processed by GIC so that its output image $I'_{ij}$ is denoted as

$$I'_{ij} = cI_{ij}^{1/\check{\gamma}},$$

where $\check{\gamma}$ is obtained by the following optimization equation which minimizes the difference between the corrected image and the predefined canonical image:

$$\check{\gamma} = \arg\min \sum_{i,j} (I'_{ij} - \bar{I_{ij}}).$$

The optimization process achieves that the corrected image is fairly approximate to the predefined canonical image. Intuitively, the global brightness of the corrected image is similar to that of the canonical face image. From the above equations, we can come to the conclusion that GIC corrects for global brightness changes which is the uniform case we discussed previously. Its weakness, however, is then that it can not cope with the non-uniform case where the lighting changes differently in different sides of the face.

The concept of Region-Based GIC (RGIC) is therefore developed to overcome the effects of side lighting. Unlike GIC, RGIC employs a local scheme where the face is divided into regions and gamma intensity correction operates on each region separately. In an ideal case, RGIC should segment a face into regions according to the face structure, for example the famous Candide model [39]. The Candide model is, nevertheless, a complex division of a face that was not implemented in this work. Instead, a simpler segmentation strategy was designed such that a face is divided into four regions according to the horizontally and vertically symmetric lines. Figure 4.2 depicts the partition of faces into four regions.

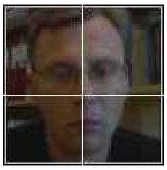

Figure 4.2: Four-region segmentation scheme divides the face into four even regions.

In the case of four regions, four different optimal values for $\gamma$ are generated to correct the brightness changes of each region. The predefined canonical image is specified as the mean face image of all training images, as shown in Figure 4.3.

In this work, we use the combination of AHE and RGIC, where the input image is first processed by AHE and further by RGIC. To facilitate the imple-

Figure 4.3: The mean face of the training set is used as the predefined canonical image.

mentation, any input face is divided into four regions based on the same rule as in RGIC. The algorithm can be described as follows:

1. For any detected face image, divide it into four regions along with the horizontally symmetric line and the vertically symmetric line.
2. For each region, apply AHE and combine them into an image.
3. For each region, calculate the optimal value for the gamma parameter $\gamma$ and use the optimized $\gamma$ to correct the brightness of the corresponding region.

## 4.2 Eigenphases

The phase information of an image has been proven to carry most of the intelligent information of the image [40] [41]. Recently, some researchers presented that the performance of face recognition could be substantially improved if the phase information of an image is only used and their results showed that phase information is invariant to illumination variations [25].

Let $I(m,n)$ be a $2D$ discrete input image of size $M \times N$ and $\tilde{I}(k,l)$ be its Discrete Fourier Transform (DFT) which can be denoted as follows:

$$\tilde{I}(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) e^{\frac{-i2\pi km}{M}} e^{\frac{-i2\pi ln}{N}},$$

where $i = \sqrt{-1}$. The polar form of $\tilde{I}(k,l)$ can be expressed as

$$\tilde{I}(k,l) = \mid \tilde{I}(k,l) \mid e^{i\Phi(k,l)}$$

$$\mid \tilde{I}(k,l) \mid = \sqrt{\mathrm{Re}[\tilde{I}(k,l)]^2 + \mathrm{Im}[\tilde{I}(k,l)]^2}$$

$$\Phi(k,l) = \arctan(\frac{\mathrm{Im}[\tilde{I}(k,l)]}{\mathrm{Re}[\tilde{I}(k,l)]}),$$

where the magnitude is $\mid \tilde{I}(k,l) \mid$ and the phase is $\Phi(k,l)$. In [25], PCA was implemented on the phase spectra of the training images and the results showed that the principal components of the phase spectra, which are termed *Eigenphases*, are significantly tolerant to illumination variations and also robust to occlusions.

## 4.3 Local Binary Pattern

Local binary pattern (LBP) defines an operator which is a computationally efficient local image texture descriptor [42]. The LBP operator has been widely used in various applications, including face and texture recognition [24] [43]. The LBP-based face recognition method was developed based on the fact that face images can be composed into lots of primitive micropatterns (such as edges, spots, corners, etc) which are invariant to monotonic gray-level variations.

The LBP operator was originally defined as an invariant local texture descriptor. The operator encodes an input image by thresholding the pixel values of its local $3 \times 3$-neighborhood with the center pixel value and histograms the resulting binary patterns. The histogram is used as a texture descriptor which extracts the micropatterns of the image. Figure 4.4 illustrates how the LBP operator encodes local micropatterns into a feature histogram. Let any pixel in an image be $(x_c, y_c)$ and its intensity value be $g_c$. The LBP operator takes an $3 \times 3$-neighborhood of the pixel $(x_c, y_c)$ and thresholds the pixel values of the 8-neighbors with the value of the center pixel $g_c$, and finally outputs an 8-bit binary number used for constructing the LBP coded image and its corresponding histogram. The thresholding process for every center pixel value $g_c$ is defined as follows:

$$s(g_i, g_c) = \begin{cases} 1, & \text{if } g_i \geq g_c, \\ 0, & \text{if } g_i < g_c, \quad i = 0, 1, 2..., 7. \end{cases}$$

By assigning a binomial factor $2^i$ for each indicator $s(g_i, g_c)$ [44], a unique LBP code (new value of $g_c$ 213 in Figure 4.4) can be computed which characterizes the spatial structure of the local image texture,

$$LBP(x_c, y_c) = \sum_{i=0}^{7} s(g_i, g_c) 2^i,$$

and the histogram of a LBP coded image $I_{LBP}(x_c, y_c)$ can be defined as

$$H(x_c, y_c) = \sum_{i=0}^{7} s(g_i, g_c).$$

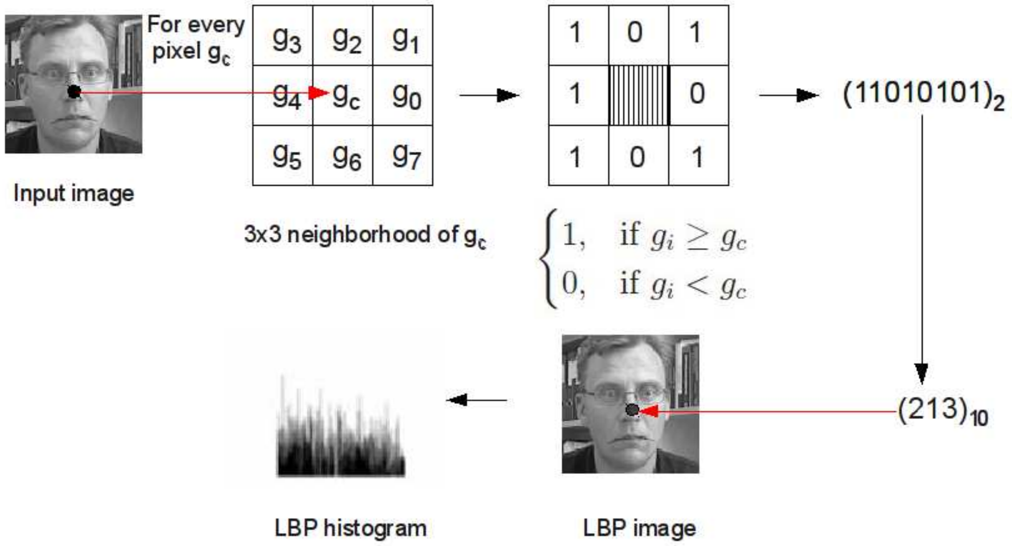

Figure 4.4: An example of the LBP operator.

Later an extension of the original operator was developed to use local neighborhoods of different sizes [44]. It utilizes the techniques of circular neighborhoods and bilinear interpolation. The circular neighborhoods technique defines a local neighborhood as a set of $P$ sampling points evenly distributed on a circle of radius $R$, and bilinear interpolation is required when a sampling point is not located in the center of a pixel. See Figure 4.5 as an example of different circular neighborhoods. Typically, the notation of the neighborhoods is defined as $(P, R)$.

*Uniform patterns* was introduced as another extension of the original operator [44]. The definition of uniform patterns determines a LBP as an uniform pattern if it contains at most two bitwise transitions from 0 to 1 or vice versa when the circular binary string is used. It has been proven by Ojala *et al.* that 90% of all patterns are uniform with the $(8, 1)$ neighborhood scheme and around 70% with the $(16, 2)$ neighborhood scheme. Thus considering the computation of the LBP histogram, a new strategy is designed by assigning each uniform pattern to a separate bin and the non-uniform patterns to a single bin. In addition, it should be noted that uniform patterns hold a vast majority due to the fact that most frequent uniform binary patterns existing in

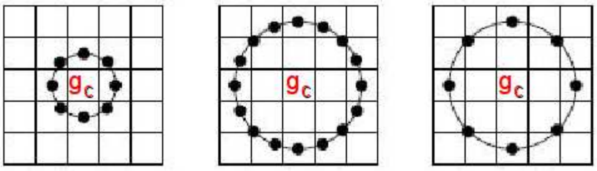

Figure 4.5: The circular (8,1), (8,2) and (16,2) neighborhoods, adapted from [45].

an image correspond to primitive features, for example edges, spots, flat areas and so on [44]. The LBP operator is therefore denoted as $LBP^{u^2}_{P,R}$ where the subscript specifies the used operator is in a circular $(P, R)$ neighborhood and the superscript represents that only the uniform patterns are available.

The LBP method can also be used efficiently for face representation [45]. In order to retain spatial information for efficient face description, the LBP-based method divides an input image into local regions $R_0, R_1, R_2, ...R_{m-1}$ from which the texture descriptor for each region is extracted independently, and then all the possible descriptors are combined into a global histogram to represent the face in the image, as depicted in Figure 4.6. The global histogram is also termed *spatially enhanced histogram* as it consists of both the appearance information and spatial relations of each region in the image. Correspondingly, the spatially enhanced histogram is defined as

$$H(x_c, y_c, j) = \sum_{p=0}^{P} s(g_p, g_c) I(x_c, y_c, j), \quad j = 0, 1, 2, ..., m-1,$$

where

$$I(x_c, y_c, j) = \begin{cases} 1, & \text{if } (x_c, y_c) \in R_j \\ 0, & \text{if } (x_c, y_c) \notin R_j. \end{cases}$$

According to [24], Chi square statistic ($\chi^2$) outperforms other different dissimilarity measures in LBP-based face recognition. The training images are encoded by LBP and their histograms are used as the reference data. Once a face image is detected by the online face recognition system, the image should also be processed by LBP and its output is taken as a query histogram. $\chi^2$ is used to measure the histogram distance between each training histogram and the query histogram. The nearest neighbor classifier is employed to find the

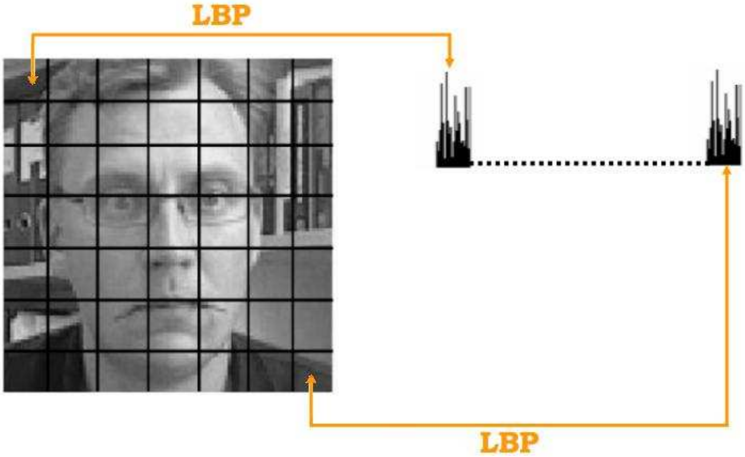

Figure 4.6: A facial image is divided into 7 × 7 windows. The local features of each region are extracted and a histogram is constructed for the region. The resulting 49 histograms are combined to form a single histogram which globally describes the face in the image. This figure is adapted from [24].

closest training histogram among the reference data, according to

$$\chi^2(p,q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i},$$

where $p$ and $q$ are the training histogram and the query histogram.

However, [27] presents criticisms of LBP-based face recognition based on two aspects. First, it is arbitrary to segment the face image with a regular grid, Second, they state that partitioning the descriptors into grid cells leads to aliasing and loss of spatial resolution. As the solution of these problems, it is proposed in [27] that the similarity measure should be reconsidered. Distance transforms method is a similarity measure that aims to take each LBP pixel code in a training image $X$ and check whether a similar code locates at a nearby position in a query image $Y$,

$$D(X,Y) = \sum_{(i,j)\in Y} \omega(d_X^{k_Y(i,j)}(i,j)),$$

where $k$ is any possible LBP code value and $d^k$ is the distance transform image calculated from a set of sparse binary images $b^k$ that are transformed from a LBP coded image of $X$. Each $b^k$ defines the pixel locations where the corresponding LBP code value $k$ appears. Each pixel of $d^k$ reflects the distance between the query image $Y$ pixel and the nearest image $X$ pixel where $k$ exists in both pixels. Thus it is easy to see that $k_Y(i,j)$ defines the code value of pixel $(i,j)$ of an image $Y$. $\omega()$ is a penalty function defined by the user which penalizes the inclusion of a pixel at a given spatial distance from the nearest matching code in image $X$ [27]. In our experiments, two distance metrics are applied to LBP-based face recognition to provide a comparison and to verify the performance of each method.

# CHAPTER 5 Online Multiple-image Face Recognition

In the previous chapters, we focused on the study of single image face recognition systems. Even with state-of-the-art algorithms and normalization methods, this approach inevitably leads to some amount of errors. The attention, thereby, is now turned to improving the recognition accuracy with other means. This gave us a motivation for developing multiple-image face recognition capable of decreasing the probability of recognition errors. It has been recently proven that recognition based on multiple images of a person can significantly improve performance [5] [6]. Multiple-image recognition is an exclusive property of online face recognition which assumes the identities of the faces in the video frames over periods of time remain the same. It deals with multiple images input which means the test data usually consists of a sequence of images rather than a single image, as depicted in Figure 5.1. Therefore, the multiple-image recognition problem can be formulated as a task that takes an image sequence from an unknown individual as query data and determines their class membership. In such a case, multiple-image face recognition can be considered as an extension of single image face recognition.

We have designed three different methods for the recognition system. In addition, performance comparison experiments of these three methods were performed, and the conclusion is made from the results that MMIK-NN clearly performs better than the other methods.

## 5.1 Multiple-image K-Nearest Neighbors

Multiple-image K-Nearest Neighbors (MIK-NN) uses several K-NN methods at the same time on the multiple images from the video frames and then determines the identity of the image sequence. Supposing there are $M$ video frames

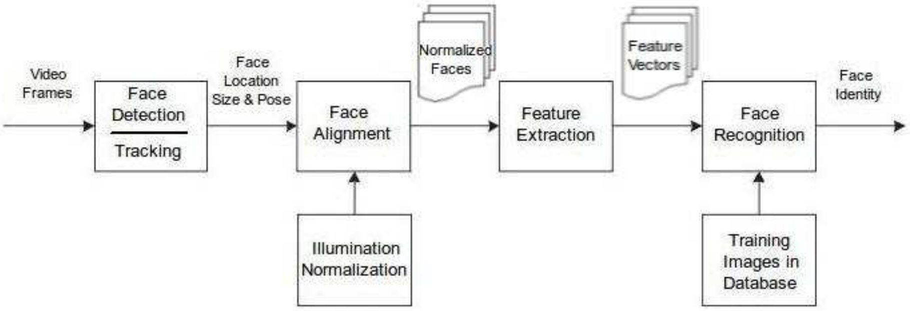

Figure 5.1: Architecture of Online Multiple-image Face Recognition.

$\{I_1, I_2, I_3, ..., I_M\}$, it is probable that the consecutive images $\{I_1, I_2, I_3, ..., I_N\}$ share the same class membership. Therefore, it is crucial to constantly segment a sequence of $N$ images from the video frames. One of the goals of the method is then to obtain the optimal value for $N$. Meanwhile, there is a need to group the $N$ sequential images together. The method consists of two major steps: 1) segmentation and grouping and 2) matching. The "sliding window" idea is applied to grouping in the following 3 steps:

Assume a sliding window of the size $N$ and the number of images in the dataset to be $P$,

1. Starting from the first image, the first $N$ images are included in the window. Those $N$ images are then grouped into the first subset, named as "Subset 1".

2. Sliding the window to the next $N$ images, the second image is the first object in the window and the $(N + 1)$th image is the last one in the "Subset 2".

3. The window is slided until the last image in the dataset belongs to the "Subset X". Note that the total number of the subsets is equal to $(P - N + 1)$.

Figure 5.2 (a) explains how to build up the subsets. During the experiments, a few different values for $N$ are selected. From the experiment results, we see that the value of $N$ has an impact on the recognition performance. See Chapter 6 for details.

Before recognition, the training images and the detected images are normalized and processed with feature extraction. Correspondingly, a set of training feature vectors and the feature vectors of the detected images are obtained. Suppose there are $P$ feature vectors associated with the original detected images, we first use sliding window method to group $N\,(N \ll P)$ consecutive neighboring vectors as a query subset. Multiple K-NN methods simultaneously operate on each vector to calculate the Euclidean distances between the vector and each of the training vectors. All possible classification decisions are collected for this query subset. In the later experiments, 1NN, 3NN, and 5NN are implemented. Based on the classification results from 1NN, 3NN and 5NN, *majority rule* is used to find the final identity of the query subset. Majority rule is a widely used voting strategy where the class having the majority of assignment of the images from classification decisions is the class membership of the image sequence. Consequently, in each subset, the most frequent person name appearing in the results is the final identity, see Table 5.1. *Algorithm 1* is described below

**Algorithm 1** MIK-NN

```
Input: $M$ training feature vectors $\Omega$, $P$ feature vectors $\hat{\Omega}$ of new video frames
Output: The identity of new video frames $D$
for $i = 1$ to $P - N + 1$ do
    sliding window such that $\hat{X}_i \leftarrow (\hat{\Omega}_i, ..., \hat{\Omega}_{i+N-1})$
    for $n = 1$ to $N$ do
        $\hat{\omega}_n \leftarrow \hat{X}_i^n$
        for $j = 1$ to $M$ do
            $c_j \leftarrow$ Euclidean distance $(\hat{\omega}_n, \omega_j)$
        end for
        1NN, 3NN and 5NN on $c$ such that $C_n \leftarrow c$
    end for
    $D_i \leftarrow \max C$
end for
return $D$
```

| Subset for Jorma | 1NN | 3NN | 5NN |
|---|---|---|---|
| Image 1 | Jorma | He | Jorma |
| Image 2 | Jorma | Markus | Jorma |
| Image 3 | Mats | Jorma | Jorma |

Table 5.1: Majority rule; an example of Subset 1 showing that the recognized person is Jorma with a maximum number of votes 6.

The table illustrates that mismatching can happen with higher possibility in the single face recognition than in multiple face recognition. In addition, the results from the experiments support this observation, see Chapter 6.

## 5.2 Modified MIK-NN

The first method proceeds by outputting the recognition result of a subset composed of a number of consecutive images. The method is developed to achieve better recognition accuracy due to the fact that the consecutive images should share the same identity. Nevertheless, this assumption might not work well when there are large amount of pose and facial appearance variations in the images. This phenomenon most likely happens in video-based face recognition, since the subject continuously moves his or her head and changes face expressions in the video. For example, in a video sequence some of the images are taken with the subject's frontal face pointing to the camera, and some of the images are captured with mostly side faces. Diverse facial appearances and poses will then manifest in images so that the images with the frontal faces are often correctly recognized but the accurate recognition of the rest of the pictures is more difficult, as shown in Figure 5.2 (b). Morever, it is redundant to recognize consecutive images which share the same pose and appearance. Therefore, the variability of facial images should be taken into account in the new method.

Our second method, which is called modified MIK-NN (MMIK-NN), is developed to give the solution to the problem by combining more diverse images into the subset, so that the correct recognition results from easily recognized images could balance the erroneous results and thereby improve the final decision. The essence of the approach is the random permutation of the sequence of the recorded images. Figure 5.2 (c) depicts how the permutation works in the new method. The recognition performed on the subsets from (a) would likely return an accurate result, whereas the results from (b) may not be as accurate as in (a). If the images are preprocessed with the random permutation step, the system includes new neighboring images in the subsets, for example images 04000000.bmp, 04000004.bmp and 04000212.bmp. We assume here the number of the images in each subset is 3. The rest of steps remain the same as the previous method. Hence, after illumination normalization and feature extraction on the detected video frames, *Algorithm 2* is designed to perform modified multiple-image face recognition.

Though the experiment results indicate that the modified method outperforms the first method, it introduces a delay problem as the system requires time to permutate the order of the features associated with the detected images.

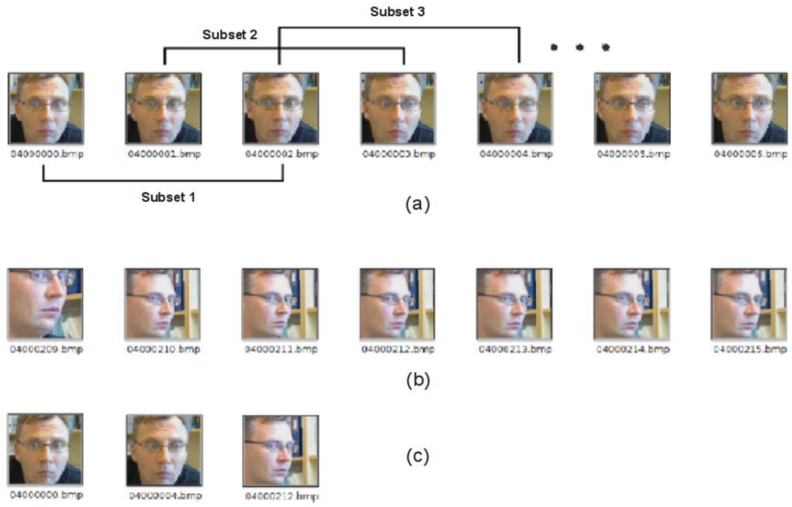

Figure 5.2: Let us assume there are 3 images in each subset. (a) shows the procedure of grouping the subsets starting from Subset 1 and so forth, given image 04000000.bmp is the first image taken. Moreover, those images can be correctly recognized. (b) exhibits a series of images considered as difficult to recognize. (c) gives an example of a combination of easily recognized and difficult images.

## 5.3 K-means+Multiple K-NN

The idea of multiple-image face recognition has been extended to group more diverse images in a video sequence for classification. Previously, the selection of these images was accomplished using random permutation and the "sliding window". However, it could also be done using clustering technique to obtain an image sequence in which various facial appearances exist. Let $M$ video frames be $\{I_1, I_2, I_3, ..., I_M\}$. We can summarize these video frames using clustering method and obtain the images $\{I_1, I_2, I_3, ..., I_N\}$ which characterize the possible poses and facial appearances. K-means, as a classical clustering technique, is used in this method. Before clustering, it is also necessary to divide each person's images into $M$ subsets due to the large number of the images stored in the database.

**Algorithm 2** MMIK-NN

```
Input: $M$ training feature vectors $\Omega$, $P$ feature vectors $\hat{\Omega}$ of new video frames
Output: The identity of new video frames $D$
random permutation $\hat{\Omega}' \leftarrow \hat{\Omega}$
for $i = 1$ to $P - N + 1$ do
    sliding window such that $\hat{X}_i \leftarrow (\hat{\Omega}'_i, ..., \hat{\Omega}'_{i+N-1})$
    for $n = 1$ to $N$ do
        $\hat{\omega}'_n \leftarrow \hat{X}^n_i$
        for $j = 1$ to $M$ do
            $c_j \leftarrow$ Euclidean distance $(\hat{\omega}'_n, \omega_j)$
        end for
        1NN, 3NN and 5NN on $c$ such that $C_n \leftarrow c$
    end for
    $D_i \leftarrow \max C$
end for
return $D$
```

### 5.3.1 K-means clustering

K-means is a well known method for solving the clustering problem [46] [47]. The method splits a set of data points into $K$ clusters where each data point is assigned to the cluster with the nearest cluster center. The typical way to implement K-means is to use an iterative refinement technique as follows. Initially, the $K$ centers are randomly selected. Then every data point is assigned to the cluster where that point is closest to the centroid. On each iteration, the centroids will be recalcuated and assignment of each point to the nearest centroid will be performed again. The loop will be executed until a stopping criterion is satisfied, for example when there is no further change in the centroids. In fact, the stopping criterion is obtained when K-means minimizes the within-cluster distance represented as an objective function as follows,

Given a set of data points $(x_1, x_2 ... x_n)$, K-means partitions $n$ data points into $k$ sets $(k < n)$ $S = \{S_1, S_2 ... S_k\}$, where the objective function is of the form

$$J = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - c_i\|^2,$$

where $c_i$ is the geometric centroid of the data points in $S_i$.

On one hand, the assumption that the algorithm will converge to the global optimum is by no means always met [48]. It also does not achieve the so called "local optimum", since a discrete assignment is used rather than a set of continuous parameters. One the other hand, the result depends on the assignment of initial clusters such that different assignment could return different clustering results. Due to the fast implementation of the algorithm, K-means clustering is usually executed multiple times with different starting assignments. Despite of these limitations, the algorithm is frequently used due to its easy implementation and fast computation.

### 5.3.2 Algorithm

After K-means clustering, each cluster is able to represent one pose for an individual, and with the nearest neighbor method, $K$ images nearest to their cluster centroids are selected as the most possible pose images. Subsequently, recognition with multiple K-NN is again applied to $K$ images in each subset. In an online setting, when recording the images the system uses K-means on 10 images. For example, the value of $K$ is selected as 3. Thus, 10 images are assigned to 3 clusters. And with the nearest neighbor method, 3 images are found which resemble the centers of the 3 clusters. At last, multiple K-NNs are implemented on those 3 images to search for the identity for the subset. In order to achieve more reliable recognition performance, the permutation of the order of the images is utilized. Since the experimental result shows that this method might not be reliable, the algorithm is not provided in pseudocode form.

**Algorithm 3**

Suppose there are $P$ images detected and tracked in the video

1. Illumination normalization and feature extraction, $P$ feature vectors associated with the original $P$ images.

2. Permute the order of the $P$ feature vectors.

3. Divide the permutated vectors into $W$ subsets.

4. For each subset, classify the vectors into $K$ clusters in which each vector is assigned to the cluster with the closest mean.

5. Find the nearest neighbor of each centroid of the clusters, and the vector found could describe the representative image with particular pose or facial appearance. For $K$ clusters, there will be $K$ vectors selected.

6. Perform multiple K-NN on the $K$ vectors for identification and find out the final identity.

7. Repeat Step 3,4,5 until the person is identified repeatedly.

# CHAPTER 6 Experiments

This chapter focuses on performing simulation experiments of the algorithms discussed in the previous chapters and on discussing of the results generated in the Matlab® environment [49]. It contributes to selecting the most reliable and promising algorithms for online face recognition systems.

## 6.1 Face Database

In order to thoroughly evaluate the performance of the algorithms used in this thesis, two image databases were used in the experiments. The images were originally from video streams so that they were captured in different lighting conditions and backgrounds. The first database (called *database* 1) consists of 412 images from 8 subjects from which 120 images were manually chosen for training and the rest were used as test images. In the *training set*, there were 15 images from each subject. Furthermore, the test images were divided into two sets: the *normal test set* where frontal faces appear in most of the images and the *difficult test set* in which the faces in the images are terribly tilted or rotated. Figure 6.1 depicts the partition of the 412 images and Figure 6.2 gives an example of Markus's 15 images which characterize the appearance of Markus in the video frames. In online face recognition, only a portion of all the images in the database are selected for training. Therefore a trade off is introduced when selecting the number of the training images. As we know, the more training images, the better performance a method might achieve, but the method might not generalize to other test sets. Thus 30% of the images in the database 1 were used for training. Figures 6.3 and 6.4 illustrate the various images in the different test sets. Intuitively, the images in the normal test set should be accurately recognized while the images in the difficult test set might not be as easily correctly identified.

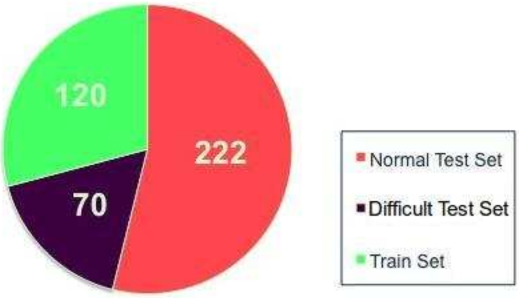

Figure 6.1: The database 1 is divided into three sets: the training set with 120 images, the normal test set with 222 images and the difficult test set with 70 images.

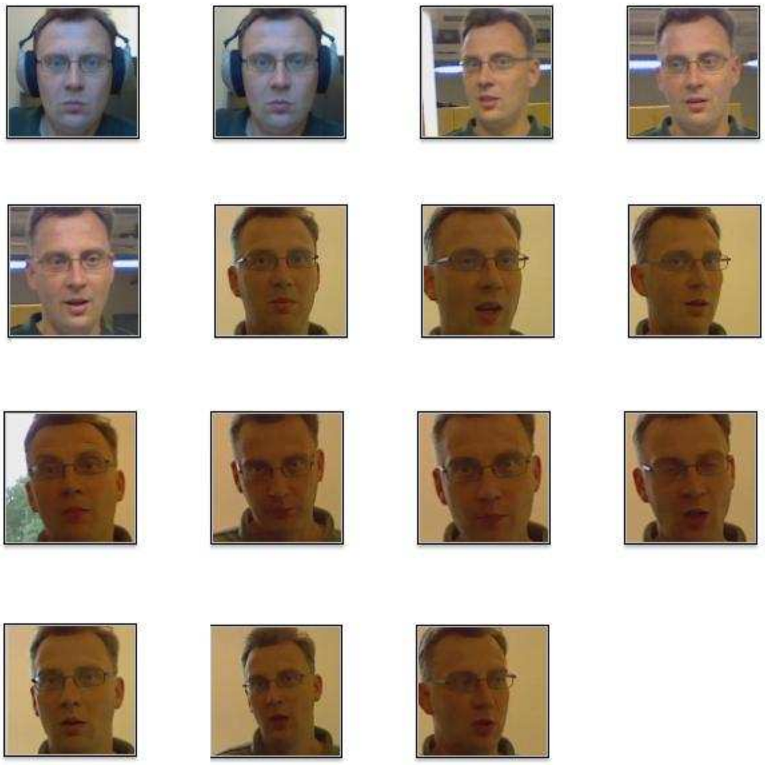

Figure 6.2: 15 training images for the subject "Markus" in the training set.

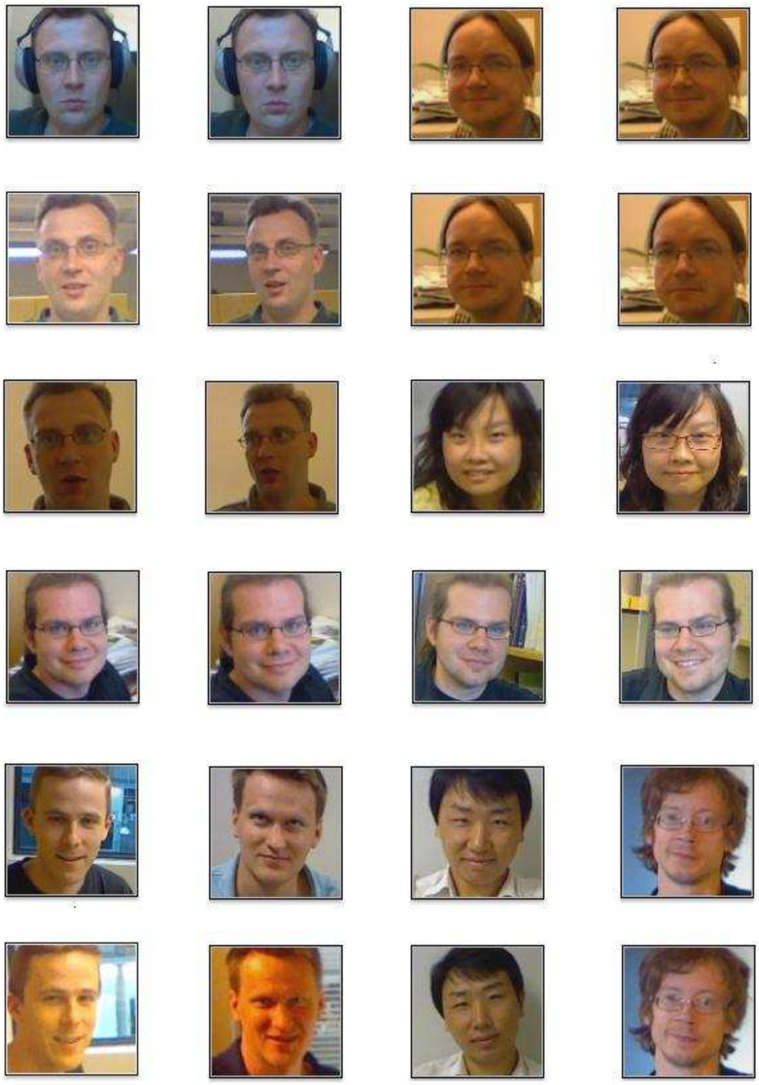

Figure 6.3: Most of the images in the normal test set capture the frontal faces of the subjects.

In order to evaluate the performance of the algorithms with a database of a larger size, *database* 2 that contains 7090 test images from 6 subjects was built up later. For the convenience of calculating the recognition accuracy, we manually combined the images of the same person together into 6 distinct subsets. The number of images for each subject was then as shown in Figure 6.5. As we mentioned in Chapter 1, online face recognition needs to deal with uncontrolled video frames so we can not control how many images to take

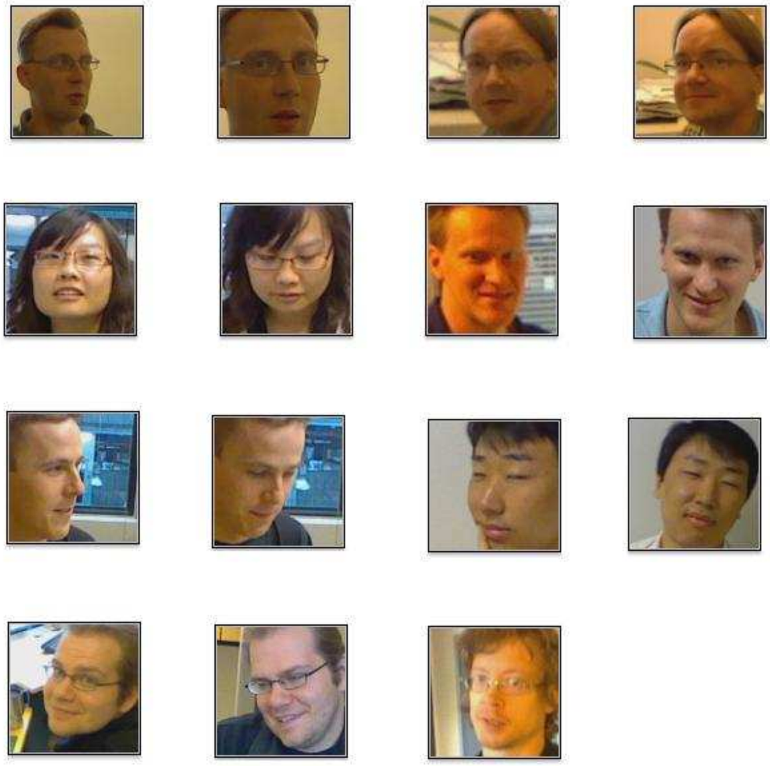

Figure 6.4: Sample images illustrate mostly terribly tilted or rotated faces, and occlusions also exist in some images.

from each person, which might lead to the biased situation as in database 2 that the images of one subject would be the majority. In addition, the images in this dataset have more variability than those in database 1 due to the fact that the images were captured intermittently in the various backgrounds over a long period of time. Figure 6.6 shows a few sample images of the subject "Markus". Compared to database 1, database 2 clearly contains more diverse views of the same face as well as more varying illumination conditions. For example, some images were taken in a dark place while other images were captured in a relatively bright background, and more different expressions and poses are present in the images. To measure how well the face recognition algorithms can generalize to this database, the training set from database 1 was employed also here but we added 30 images of the subjects "Antti" and "Magazine man" who is a test subject, whose images were captured directly from a magazine photo. 7090 images are used as test images.

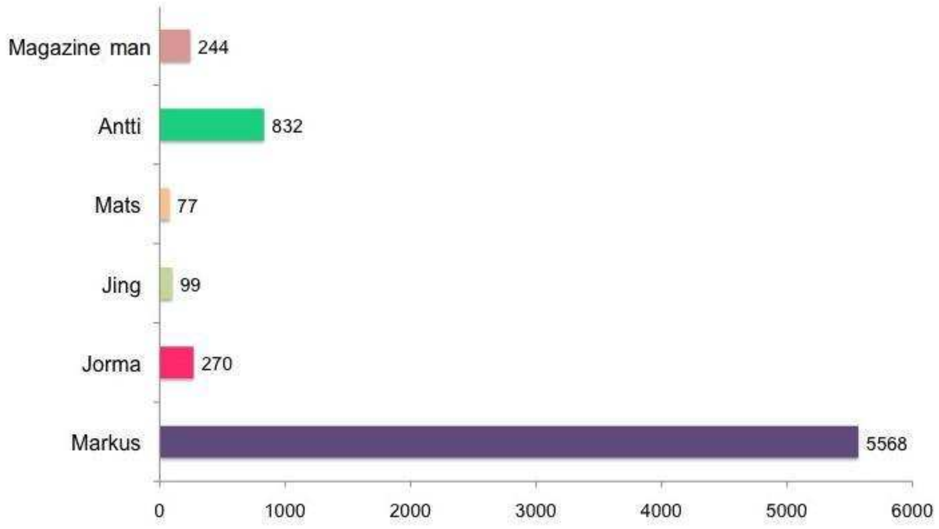

Figure 6.5: The number of images for different subjects in database 2. The subject Markus has the majority number of images and Mats has the smallest number of images.

## 6.2 Single-image face recognition experiments

For PCA-based and SVM-based face recognition, the image features have been extracted using the PicSOM system which incorporates the MPEG-7 face descriptor. PicSOM is an image retrieval system based on the self-organizing map (SOM) [50]. According to the requirements of the MPEG-7 face descriptor, it is necessary to geometrically normalize the input images to align the faces appearing in the images before feature extraction. Therefore we used face detection and captured the face images with 100 by 100 pixels so that the centers of the two eyes in the geometrically normalized image are located at (42st row, 36st column) and (42th row, 64st column).

In the experiments with PCA-based face recognition, nearest neighbor classification was performed on the two test sets. To implement SVM-based face recognition, a new library for SVM was included in Matlab [51]. SVM-based face recognition was then performed as follows:

1. Normalization step: Normalize all the features generated by PicSOM to the range $[0, 1]$.

2. Kernel selection step: Use radial basis function (RBF) as the kernel model: $K(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}$.

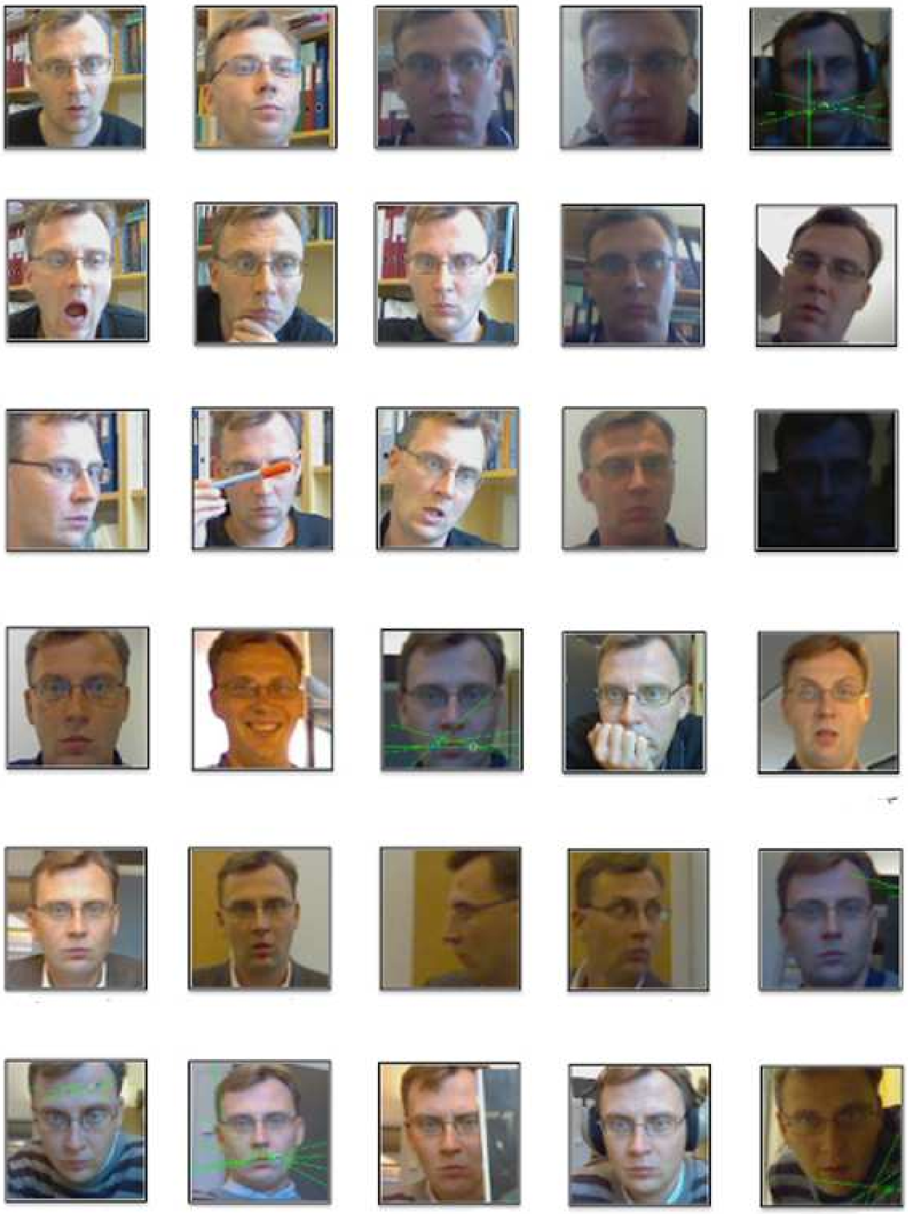

Figure 6.6: Sample images of Markus in database 2.

3. Training and validation step: Divide the training set into 2 subsets, one with 56 images is used for training and the other with 64 is for testing. Apply validation on these two subsets with one-against-all method and one-against-one method separately to find two sets of best parameters $C$ and $\frac{1}{\sigma^2}$.

4. Training and test step: Apply the best parameters to the whole train set to get optimal model and predict to which class the test data belongs [52].

Since the MPEG-7 face descriptor can not be used in 2DPCA-based face recognition, it is crucial to utilize the sufficient number of eigenvectors. Here we used 5-fold cross-validation to find out the optimal number of eigenvectors. The training set was divided into 5 subsets of equal size and each time one subset was chosen as the test set while the rest of subsets were used for training. The cross-validation accuracy was calculated as the percentage of images that have been accurately recognized. As the number of eigenvectors varies, the accuracy differs slightly in Table 6.1. Thereby we chose a middle value, 48 eigenvectors, for the 2DPCA-based method.

| | The number of eigenvectors | | | |
|---|---|---|---|---|
| | 40 | 48 | 50 | 55 |
| cross-validation accuracy (%) | 91.34 | 91.5 | 91.5 | 91.73 |

Table 6.1: Cross-validation accuracy for the 2DPCA-based method.

The number of errors and the recognition accuracy of the three algorithms are shown in Table 6.2 where $1vs1$ and $1vsall$ represent the one-against-one and one-against-all methods. Figure 6.7 further illustrates the performance of these algorithms. The recognition accuracy is defined as

$$Recognition\ Accuracy = \frac{N_{\mathbf{correct}}}{N} * 100\%,$$

where $N$ is the total number of images in a test set and $N_{\mathbf{correct}}$ is the number of images correctly recognized in that test set.

From the table, it can be observed that SVM with one-against-one method achieves the largest accuracy and that the results are fairly good even for the difficult test set. Before the experiments, we anticipated that the supervised way should outperform and that the recognition accuracy of the difficult test set would not be good. After the analysis of the classification result, the main reason for the high recognition accuracy could be that the number of images

| | Normal Test Set | | Difficult Test Set | |
|---|---|---|---|---|
| | # of errors | Accuracy (%) | # of erros | Accuracy (%) |
| PCA | 7 | 96.84 | 4 | 94.2 |
| 2DPCA | 25 | 88.73 | 8 | 88.5 |
| SVM + $1vs1$ | 3 | 98.64 | 3 | 95.7 |
| SVM + $1vsall$ | 11 | 95.04 | 10 | 85.7 |

Table 6.2: Recognition accuracy of PCA-based, 2DPCA-based, and SVM-based methods.

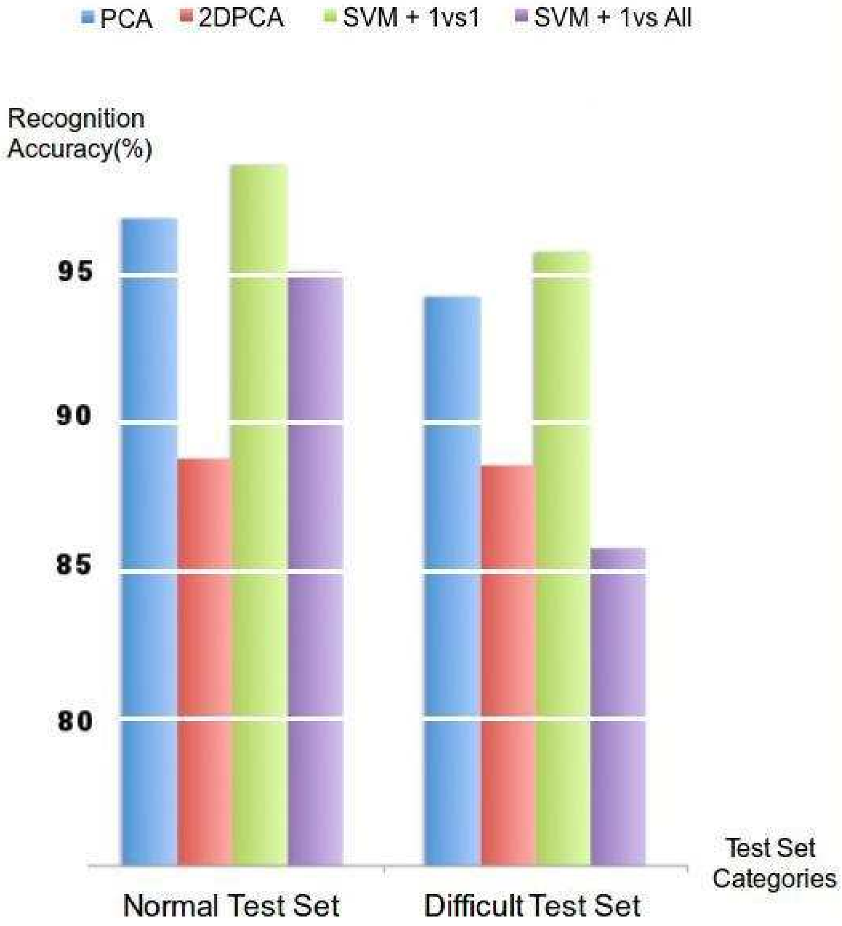

Figure 6.7: Illustration of the performance of PCA-based, 2DPCA-based, and SVM-based methods.

in the database is rather small and more importantly each individual had an unique gesture when we were taking pictures of them. This could be observed from Figures 6.3 and 6.4 where the subjects appearing in the images tend to have gestures exclusive for them. Therefore, we can come to the conclusion that gesture information has an effect on the recognition process which makes each person easier and more accurate to be recognized. Furthermore, we also noticed that the performance of the 2DPCA-based method was not better than that of the PCA-based method. Also it is due to the fact that the PCA-based method was implemented by mainly using the MPEG-7 face descriptor which uses 48 standardized eigenvectors to extract the features of the images that the recognition accuracy is guaranteed. The classification results also indicate that some images taken in abnormal lighting condition could not be accurately classified, as seen in Figure 6.8. Since varying illumination and pose have an impact on the performance of face recognition, further research should focus on compensating for varying illumination and pose. Finally, a larger test database should be constructed to obtain more convincing results.

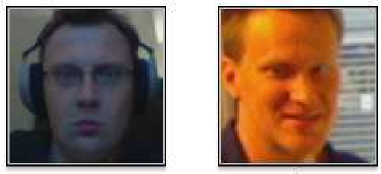

Figure 6.8: Two images that were taken in bad lighting conditions could not be correctly recognized.

## 6.3 Illumination normalization experiments

The illumination normalization experiments started with the implementation of the AHE+RGIC method. The input images were initially divided into four regions along with the horizontally and vertically symmetric lines. Each region was then processed with adaptive histogram equalization (AHE) and region-based gamma intensity correction (RGIC) in which the optimal value for $\gamma$ was determined by using the golden section search method [53]. The recognition experiments were carried out using the PCA-based and the SVM with one-against-one methods.

In the experiments of Eigenphases method, the phase spectra of all the images were extracted and the training phase spectra were used for generating the eigenvectors. The optimal number of the eigenvectors was 50. Therefore, the phase spectra of the images were projected onto the subspace spanned by these 50 eigenvectors and the nearest neighbor classifier performed the classification task.

In the LBP-based method, 14 × 14 pixels was selected as the optimal size of a small region for the 100 × 100 pixel images based on the conclusion of [24]. The input images were divided into several regions of 14 × 14 pixels. For each region, LBP encoded every pixel in this region by thresholding its circular neighborhood 8 × 2 and a local histogram was then constructed for each region. At last, the global histogram was constructed by combining the local histograms together. The global histogram can also be denoted as a histogram vector which was used for classification with two distinct distance metrics, Chi square statistic and distance transform.

The experiments with different illumination normalization methods were performed on two databases. In the first experiment, the recognition accuracies were calculated for each test set. Table 6.3 exhibits the recognition results for five methods. In AHE+RGIC+SVM, SVMs with the one-against-one method were used. Figure 6.9 plots the comparison results.

| | Normal Test Set | Difficult Test Set |
|---|---|---|
| AHE+RGIC+PCA | 99.10 | 90.00 |
| AHE+RGIC+SVM | 95.40 | 91.43 |
| Eigenphases | 32.88 | 32.86 |
| LBP+$\chi^2$ | 100.00 | 100.00 |
| LBP+DT | 100.00 | 100.00 |

Table 6.3: The results of different illumination normalization methods used with database 1.

One obvious finding is that the LBP-based method achieves perfect recognition accuracy. This high accuracy can be attributed to the small size of dataset 1. Another interesting finding is that the Eigenphases approach did not work well, unlike the statement in [25]. The possible explanation is that the Eigenphases method does not suit well to our database since it only uses the phase spectra of the images and the magnitude spectra are completely discarded. The magnitude spectra, however, is shown to be tolerant to pose variations [54]. As we know, large amounts of pose information exist in the images in our database and the Eigenphases method thus could not perform well. On the other hand, the experiments in [25] were operated on mostly frontal face images taken under 21 illumination variations from the CMU PIE illumination dataset. The relatively few changes in pose in the images ac-

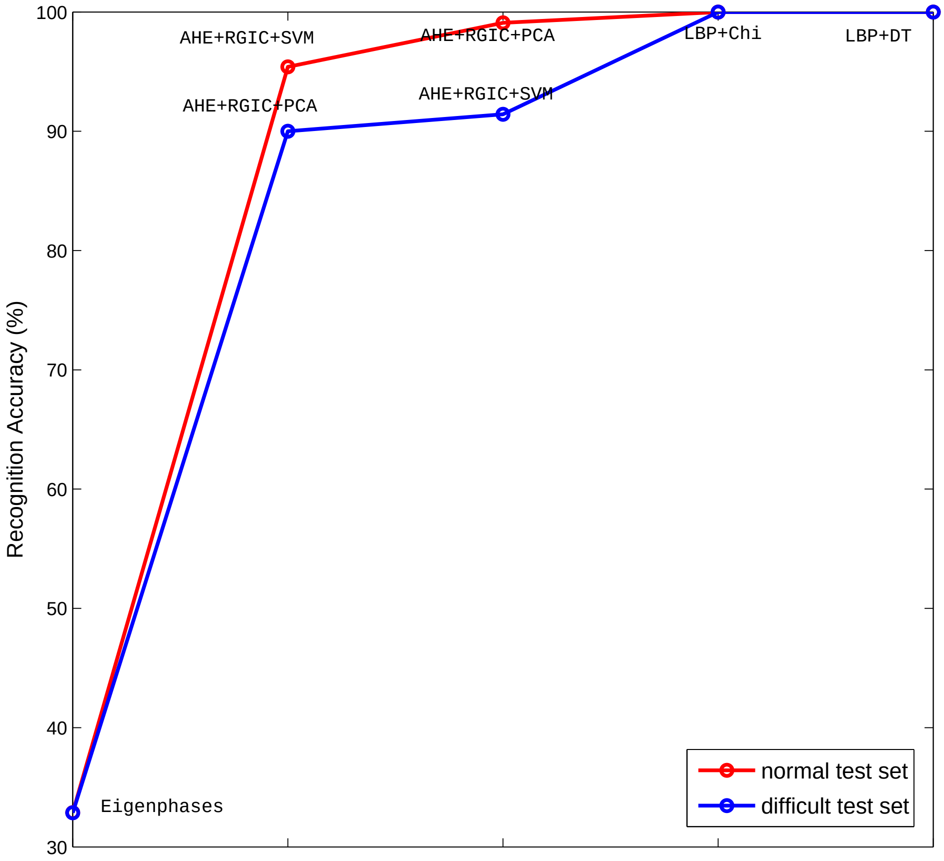

Figure 6.9: Recognition results of the distinct illumination normalization methods used with database 1.

count for why the Eigenphases method outperforms in [25]. Finally, we also observed that AHE+RGIC+PCA excels AHE+RGIC+SVM for recognition in the normal test set whereas AHE+RGIC+SVM is able to recognize more difficultly recognizable face images. It is hard to determine which one is better on average.

Subsequently, database 2 was used in the next experiment to verify the performance of the LBP-based method. AHE+RGIC+PCA was used as a baseline for comparison with the LBP-based method. Similarly, the recognition accuracy for each subset was calculated and the mean accuracy was further obtained based on the percentage of images accurately recognized among the whole test set,

$$Mean\ Accuracy = \frac{\sum_{i=1}^{6} N^i_{\mathbf{correct}}}{N} * 100\%,$$

where $N^i_{\mathbf{correct}}$ is respectively the number of images correctly recognized in each subset and $N$ is the total number of images in database 2. Table 6.4 illustrates the final results in which AHE+RGIC stands for AHE+RGIC+PCA.

| | Markus | Jorma | Jing | Mats | Antti | Magazine | Mean |
|---|---|---|---|---|---|---|---|
| AHE+RGIC | 52.07 | 67.41 | 77.78 | 15.58 | 83.77 | 99.59 | 57.97 |
| LBP+$\chi^2$ | 62.11 | 51.11 | 95.96 | 67.53 | 71.75 | 99.59 | 64.64 |
| LBP+DT | 63.57 | 54.82 | 95.96 | 66.23 | 72.48 | 99.59 | 66.00 |

Table 6.4: The results of illumination normalization methods used with database 2.

The tabular results show that the LBP+$\chi^2$ and LBP+DT methods both outperform the AHE+RGIC+PCA method. This could be interpreted by the intrinsic property of LBP-based face recognition. LBP-based face recognition is an invariant feature extraction method in which the micropatterns of the image are encoded by LBP into a feature histogram which is robust to illumination variations. An interesting result is that although the LBP+DT method slightly outperforms LBP+$\chi^2$, LBP+DT is computationally much more expensive. For example, the elapsed time to execute LBP+DT on Antti subset was 14209 seconds whereas LBP+Chi on the same subset took only 31.6 seconds. The conclusion can be made that LBP+$\chi^2$ could reliably operate in online face recognition. However, as the LBP-based face recognition method has been implemented only recently, the AHE+RGIC method was applied to online multiple-image face recognition.

## 6.4 Multiple-image face recognition experiments

In the experiments of multiple-image face recognition, only database 2 was utilized. All the images were initially preprocessed with the AHE+RGIC technique and the MPEG-7 face descriptor extracted features. For each subject, Table 6.5 illustrates its possible feature vectors grouped into a corresponding feature matrix.

For multiple-image k-nearest neighbors (MIK-NN) experiment, for each subject every $N$ sequential feature vectors are grouped together from its feature matrix by using the "sliding window" method. 1NN, 3NN and 5NN were subsequently used in these $N$ feature vectors and there were $3 \times N$ classification

| Subject | # of Images | Feature matrix Name |
|---|---|---|
| Markus | 5568 | face4Markus (5568*48) |
| Jorma | 270 | face4Jorma (270*48) |
| Jing | 99 | face4Jing (99*48) |
| Mats | 77 | face4Mats (77*48) |
| Antti | 832 | face4Antti (832*48) |
| Magazine | 244 | face4Magazine (244*48) |

Table 6.5: For each subject, the corresponding feature vectors are generated and saved into a feature matrix.

decisions. The majority rule was used to find out the facial class with the majority of the votes. Table 6.6 shows the recognition accuracy for each subject and the mean accuracy for the whole testset with different values of $N$. Figure 6.10 shows the obvious result that the recognition accuracy of MIK-NN generally increases when the value of $N$ is increased.

From both the tabular and plot results, the subject Magazine man gains the best recognition result. It is possibly because the Magazine man's training images were selected directly from database 2 and the training images characterized all the possible images in the database for Magazine man. For the Jing subset, the recognition accuracy is also almost 100% on average, and the reason could be that Jing is the only female that the gender information contributes to the high accuracy. The reason why the recognition accuracy for Mats subset is very low is probably because the subset is relatively small. However, if we look back to Table 6.4, the accuracy for the Mats subset was better when the LBP-based method was used, which suggests us that the LBP-based method can perform well even with small image sets. On the other hand, compared to the mean accuracy of single-image face recognition in Table 6.4, MIK-NN method obtained slightly higher accuracy.

The modified multiple-image k-nearest neighbors (MMIK-NN) method is designed to group diverse images in the image sequences for classification. Therefore in the experimental stage, the permutation of the feature matrices was required so that the grouping of $N$ feature vectors can be implemented randomly. 1NN, 3NN and 5NN were then implemented on the randomly selected $N$ feature vectors in the same fashion. Similarly, majority rule was used to output the final identity for those $N$ images. In order to obtain reliable results, the experiments were performed 10 times for the images of every subject. Table 6.7 illustrates the recognition accuracy for each subject and the mean accuracy for the whole dataset with different values for $N$. Figure 6.11 clearly shows that the recognition accuracy of MMIK-NN generally increases

| N | Markus | Jorma | Jing | Mats | Antti | Magazine | Mean |
|---|---|---|---|---|---|---|---|
| 3 | 63.50 | 74.63 | 91.75 | 16.00 | 84.85 | 100.00 | 67.56 |
| 4 | 64.80 | 74.53 | 95.83 | 14.86 | 85.04 | 100.00 | 68.65 |
| 5 | 64.83 | 74.81 | 95.86 | 12.33 | 85.63 | 100.00 | 68.72 |
| 6 | 65.14 | 74.34 | 100.00 | 13.89 | 85.61 | 100.00 | 69.02 |
| 7 | 65.31 | 74.62 | 100.00 | 16.90 | 85.96 | 100.00 | 69.24 |
| 8 | 65.42 | 76.43 | 100.00 | 17.14 | 86.67 | 100.00 | 69.48 |
| 9 | 65.50 | 75.95 | 100.00 | 15.94 | 86.40 | 100.00 | 69.48 |
| 10 | 66.11 | 77.01 | 100.00 | 19.12 | 86.63 | 100.00 | 70.06 |

Table 6.6: Accuracy result of MIK-NN.

when the value for $N$ increases.

By comparing Table 6.6 and Table 6.7, we can observe that the accuracies for Markus, Jorma and Antti subsets change dramatically after the use of MMIK-NN. Figure 6.12 and Figure 6.13 also support the conclusion that MMIK-NN performs better. Furthermore, the comparison of the mean accuracies in Table 6.4 and Table 6.7 also accounts for that MMIK-NN method significantly improves the performance. We attribute this to the reason that MMIK-NN tries to find out various images in one sequence that contribute to better recognition performance. An interesting finding in Table 6.7 is that the accuracy of the Mats subset has firstly increased and then decreased as the value for $N$ was increasing. This suggests that the optimal value for $N$ could be 5.

| N | Markus | Jorma | Jing | Mats | Antti | Magazine | Mean |
|---|---|---|---|---|---|---|---|
| 3 | 76.06 | 91.04 | 98.14 | 13.87 | 91.83 | 100.00 | 78.94 |
| 4 | 79.27 | 94.76 | 98.96 | 17.57 | 94.55 | 100.00 | 81.97 |
| 5 | 81.76 | 96.54 | 100.00 | 18.08 | 97.25 | 100.00 | 84.34 |
| 6 | 83.86 | 97.58 | 100.00 | 16.67 | 98.21 | 100.00 | 86.12 |
| 7 | 85.61 | 98.86 | 100.00 | 16.34 | 98.91 | 100.00 | 87.62 |
| 8 | 87.17 | 98.78 | 100.00 | 14.86 | 99.15 | 100.00 | 88.85 |
| 9 | 88.35 | 99.39 | 100.00 | 14.78 | 99.64 | 100.00 | 89.86 |
| 10 | 89.47 | 99.46 | 100.00 | 12.94 | 99.85 | 100.00 | 90.75 |

Table 6.7: Accuracy result of MMIK-NN.

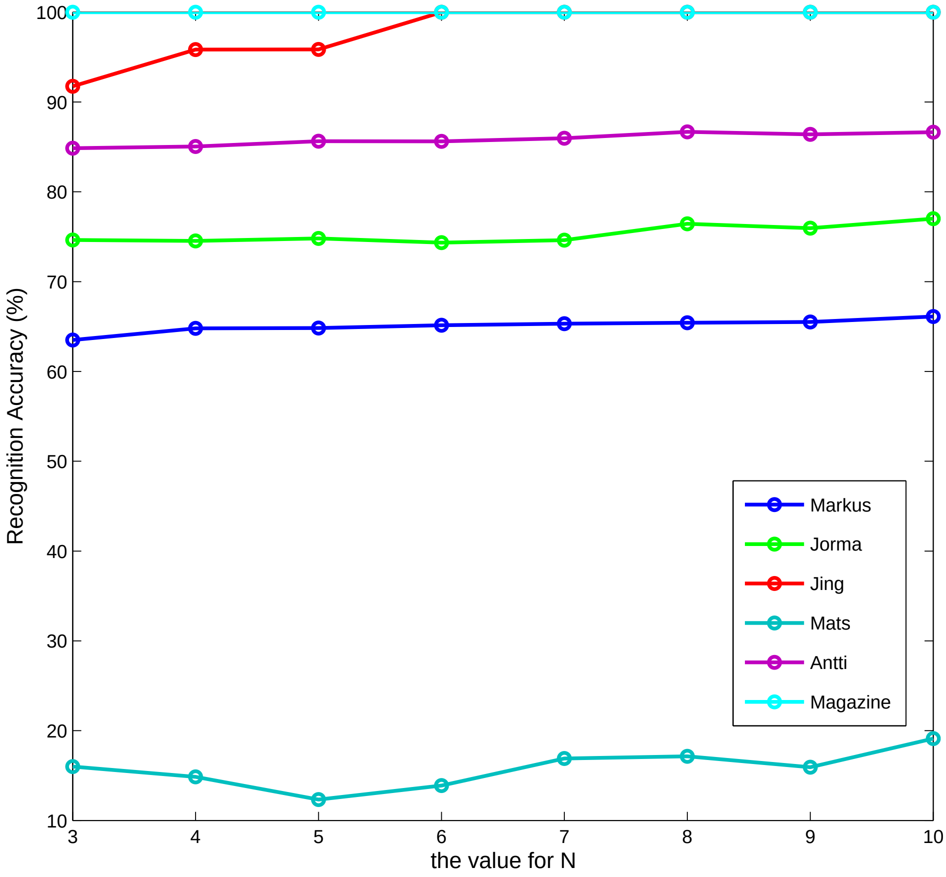

Figure 6.10: Recognition accuracy of MIK-NN.

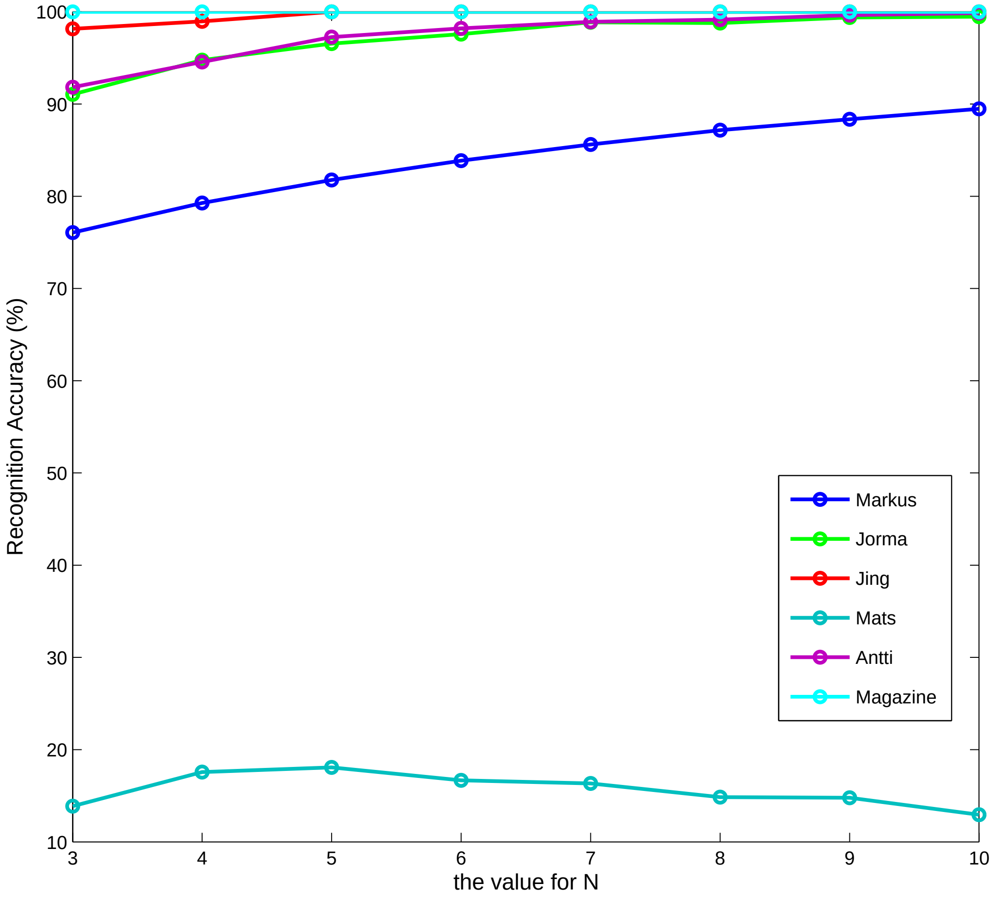

Figure 6.11: Recognition accuracy of MMIK-NN.

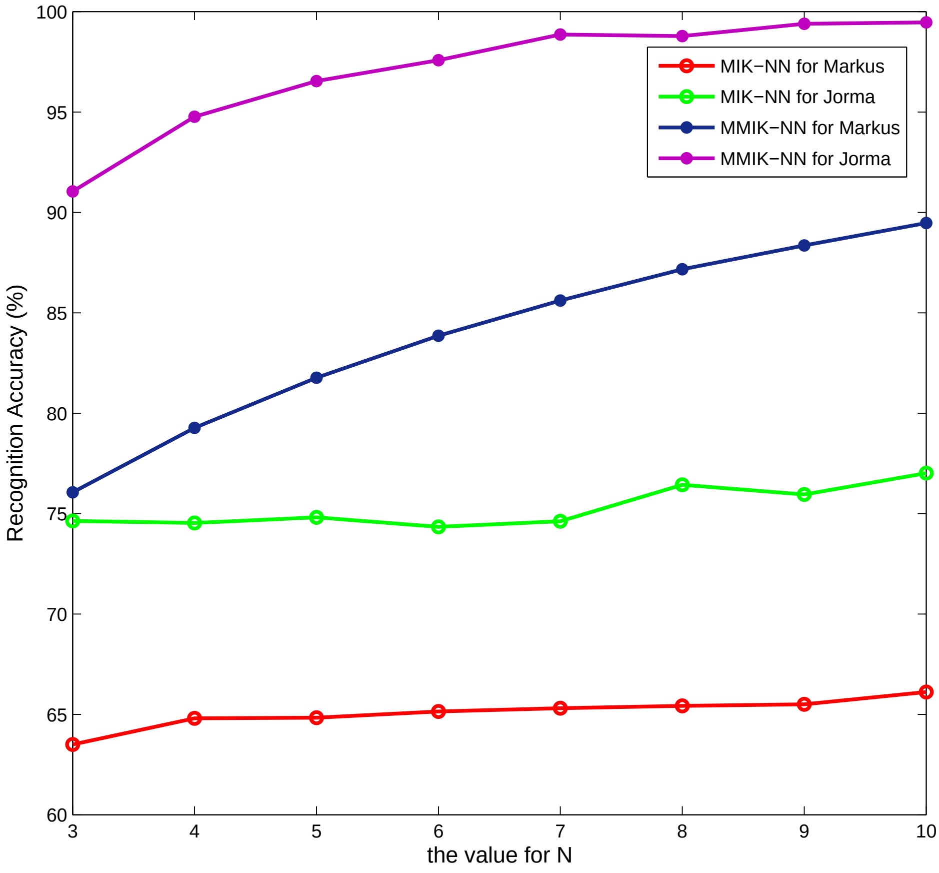

Figure 6.12: Comparison results of MIK-NN and MMIK-NN for Markus and Jorma.

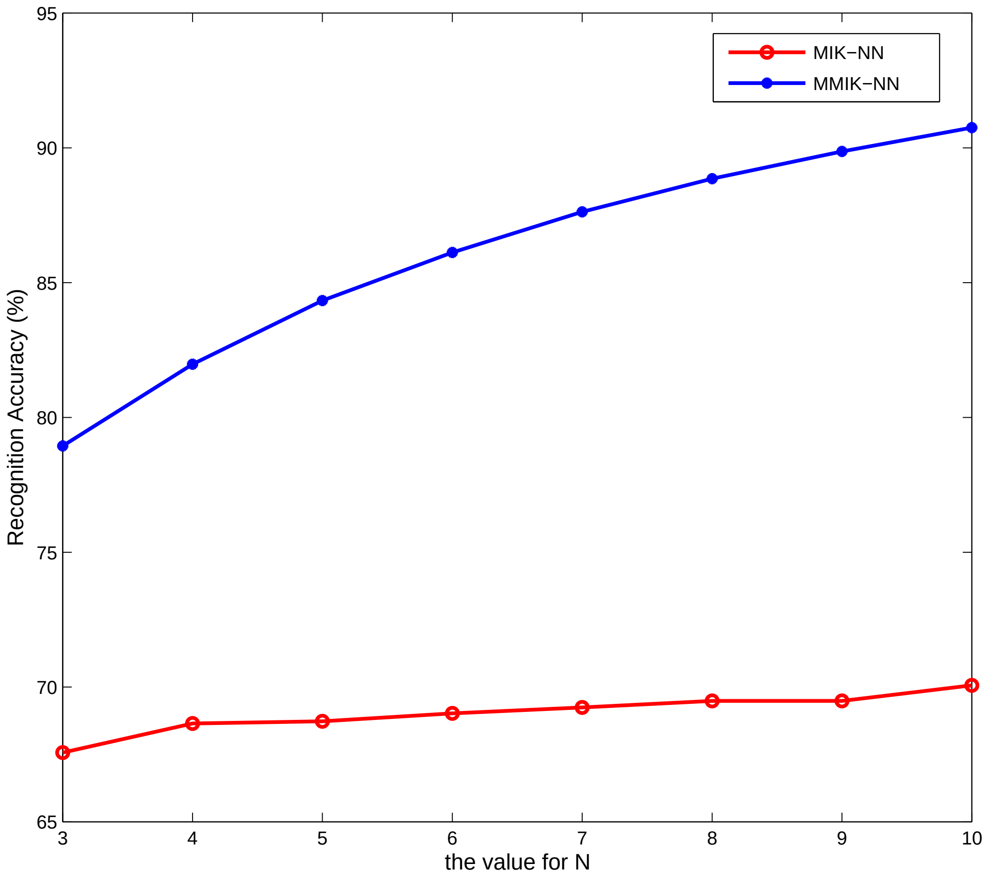

Figure 6.13: Comparison results of mean accuracy for MIK-NN and MMIK-NN.

| | Markus | Jorma | Jing | Mats | Antti | Magazine |
|---|---|---|---|---|---|---|
| # of the subsets | 100 | 30 | 11 | 7 | 32 | 20 |
| # of tries | 50 | 30 | 20 | 20 | 30 | 30 |

Table 6.8: The number of subsets and repetitions for each person.

K-means+Multiple K-NN also attempts to pick up more diverse images in the image sequences. During the experiment, we again permutated the feature matrices and divided each feature matrix into few matrices of the same size. Different subjects have different numbers of images in the database, thus the number of sub-matrices varies for each person. See Table 6.8 for the settings of the experiments. K-means further classified each sub-matrix into $k$ clusters. The nearest neighbors from the centers of the $k$ clusters were selected and used for classification by means of 1NN, 3NN and 5NN. For more convincing results, we performed the experiments several times, see Table 6.8 for the number of repetitions.

| k clusters | Markus | Jorma | Jing | Mats | Antti | Magazine | Mean |
|---|---|---|---|---|---|---|---|
| 3 | 92.69 | 96.89 | 100.00 | 8.57 | 91.83 | 94.74 | 92.01 |
| 4 | 91.48 | 94.00 | 100.00 | 9.29 | 94.55 | 94.74 | 91.27 |
| 5 | 90.24 | 95.33 | 100.00 | 10.00 | 97.25 | 94.74 | 90.68 |
| 6 | 91.94 | 97.22 | 100.00 | 10.71 | 98.21 | 94.74 | 92.20 |
| 7 | 93.071 | 97.67 | 100.00 | 6.43 | 98.91 | 94.74 | 93.14 |
| 8 | 94.16 | 99.11 | 100.00 | 4.29 | 99.15 | 94.74 | 94.06 |
| 9 | 95.48 | 99.67 | 100.00 | 7.86 | 99.64 | 94.74 | 95.21 |

Table 6.9: Mean accuracy of K-means + Multiple K-NN.

Table 6.9 outputs the mean accuracy of each subset after several continuous tries and also the mean accuracy of the whole set for different values of $k$. For the subsets Markus and Jorma, when $k$ ranges from 3 to 9 clusters, the recognition accuracy first decreases and later increases. It might be the case that the poor result comes from bad choice of $k$. For example, the intrinsic property of data has been determined that it can properly be partitioned into $k$ clusters whereas our method forcefully divided it into $k+1$ or $k+2$ clusters. For the Jing subset, the recognition accuracy is 100% for different values of $k$ and the explanation has been given for Jing's gender information which helped to achieve the perfect result. The recognition accuracy for the Mats subset is still really low due to its small image set. In addition, the accuracy for Mats is surprisingly low when the number of clusters becomes 7, 8 and 9. This might be caused by the inability of k-means to function well when the number of images to be clustered is only slightly larger than the number of clusters. Based on the overall results, K-means + Multiple K-NN might not be a good choice for multiple-image face recognition. However, on further research on utilizing the diversity of images, one could employ the idea of mode-seeking process, for instance the mean shift method [55].

# CHAPTER 7

# Summary and conclusions

The objective of this thesis is to find improved solutions to the online face recognition problem by using different kinds of machine learning methods and image processing techniques. We have presented methods for two schemes of online face recognition: online single-image face recognition and online multiple-image face recognition.

Online single-image face recognition has been implemented with three machine learning approaches: PCA-based, 2DPCA-based and SVM-based methods. On one hand, both the PCA-based method and the 2DPCA-based method are unsupervised learning approaches for feature extraction. While PCA is used to extract the features of 1D image vectors from a low subspace by means of the MPEG-7 face descriptor, 2DPCA is proposed to extract the features of an 2D image matrix computationally more efficiently. On the other hand, SVM as a classical supervised learning approach is incorporated with two popular multi-class classification techniques, one-against-one and one-against-all, to accomplish the face recognition task.

The illumination variations which pose a significant challenge to face recognition were also considered during the work. The focus of our research was turned to illumination normalization, and solutions were given in terms of image processing techniques (AHE and RGIC) as well as invariant feature extraction algorithms (Eigenphases and LBP). The recognition was implemented in the same fashion with PCA and SVM.

Overall, some novel methods have been proposed to tackle the multiple-image face recognition task. Multiple-image k-nearest neighbors (MIK-NN) uses multiple k-nearest neighbor classifiers to classify the image sequences and applies majority rule to determine the final identity for each image sequence. Modified multiple-image k-nearest neighbors (MMIK-NN) extends the idea of MIK-NN to obtain more variability to the image sequence for higher recognition accuracy. Another way to gain diversity of images is the combination of k-means and multiple k-nearest neighbors but it did not work as we expected in our experiments.

Furthermore, our experiment results show that LBP+$\chi^2$ based method achieves good recognition accuracy with low computation complexity when variations of illumination and pose were introduced in the facial images. It also elucidates that multiple-image face recognition methods could improve the recognition rate, and modified multiple-image k-nearest neighbors outperforms the other algorithms in multiple-image face recognition.

## 7.1 Future research

To sum up, reliable online face recognition poses a great challenge to researchers and still offers substantial potential for further research. One approach in further work could be the integration of human interaction. It requires the user to give feedback whenever the identity of a person is recognized by the online face recognition system. The feedback could be an answer whether the person was correctly recognized and further the selection of the correct identity based on a candidate list provided by the system. Human interaction undoubtedly would help to substantially improve the recognition result. Another issue that we can take into account in further research is the recognition of unknown faces, which would allow that the images of the unknown face could be automatically stored in the training set for the purpose of subsequent recognition of the unknown face.

# BIBLIOGRAPHY

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

[2] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[3] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces. Technical Report CMU-RI-TR-01-02, Robotics Institute, Pittsburgh, PA, 2001.

[4] F. S. Samaria and A.C. Harter. Parameterisation of a Stochastic Model for Human Face Identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, 1994.

[5] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone. Face Recognition Vendor Test 2002: Evaluation Report. Technical report, NISTIR 6965, Gaithersburg Maryland, March 2003.

[6] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge. pages 947–954, 2005.

[7] J. Fan, N. Dimitrova, and V. Philomin. Online Face Recognition System For Videos Based on Modified Probabilistic Neural Networks. In *Proceedings of IEEE International Conference on Image Processing 2004*, pages III: 2019–2022, 2004.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[9] O. Bousquet, U. von Luxburg, and G. Rätsch, editors. *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*. Springer, 2004.

[10] Ronald T. Azuma. A Survey of Augmented Reality. *Presence*, 6:355–385, 1997.

[11] A. Ajanki, M. Billinghurst, M. Kandemir, S. Kaski, M. Koskela, J. Laaksonen, K. Puolamäki, and T. Tossavainen. Ubiquitous Contextual Information Access with Proactive Information Retrieval and Augmentation. In *Proceedings of International Workshop in Ubiquitous Augmented Reality (IWUVR 2010)*, to appear.

[12] W. W. Bledsoe. The Model Method in Facial Recognition. Technical report, Panoramic research Inc., Palo Alto, CA, 1964.

[13] M. A. Turk and A.P. Pentland. Face Recognition Using Eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[14] R. Chellappa K. Etemad. Discriminant Analysis for Recognition of Human Face Images. *Journal of the Optical Society of America*, 14:1724–1733, 1997.

[15] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997.

[16] G. Guo, S.Z. Li, and K. Chan. Face Recognition by Support Vector Machines. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 196–201, 2000.

[17] A. J. O'Toole, D. A. Roark, and H. Abdi. Recognizing Moving Faces: A Psychological and Neural Synthesis. *Trends in Cognitive Sciences*, 6(6):261–266, 2002.

[18] *Information Technology - Multimedia Content Description Interface - Part 3: Visual*, first edition, 2002. International Standard.

[19] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang. Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[20] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[21] R. Basri and D.W. Jacobs. Lambertian Reflectance and Linear Subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[22] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. H. Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive Histogram Equalization and its Variations. *Computer Vision, Graphics and Image Processing*, 39(3):355–368, 1987.

[23] S. Shan, W. Gao, B. Cao, and D. Zhao. Illumination Normalization for Robust Face Recognition Against Varying Lighting Conditions. In *Proceedings of IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 157–164, 2003.

[24] T. Ahonen, A. Hadid, and M. Pietikäinen. Face Recognition with Local Binary Patterns. In *Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science 3021, Springer.*, 2004.

[25] M. Savvides, B. Kumar, and P. K. Khosla. Eigenphases vs. Eigenfaces. *In Proceedings of the 17th International Conference on Pattern Recognition 2004*, 3, 2004.

[26] J. Ruiz del Solar and J. Quinteros. Illumination Compensation and Normalization in Eigenspace-based Face Recognition: A comparative study of different pre-processing approaches. *Pattern Recognition Letters*, 29(14):1966–1979, 2008.

[27] X. Y. Tan and B. Triggs. Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions. In *Analysis and Modelling of Faces and Gestures*, volume 4778, pages 168–182, 2007.

[28] *Improving Identification Performance by Integrating Evidence from Sequences*, volume 1, 1999.

[29] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face Recognition From Long-Term Observations. In *Proceedings IEEE European Conference on Computer Vision*, pages 851–868, 2002.

[30] S. Z. Li and A. K. Jain. *Handbook of Face Recognition.* Springer, 2004.

[31] G. Hinton and T. J. Sejnowski, editors. *Unsupervised Learning: Foundations of Neural Computation.* MIT Press, 1999.

[32] T. Cover and P. Hart. Nearest Neighbor Pattern Classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[33] I.T. Jolliffe. *Principal Component Analysis.* Springer Verlag, 1986.

[34] Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

[35] C.-W. Hsu and C.-J. Lin. A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[36] Guodong Guo and S.Z. Li. Content-based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks*, 14(1):209 – 215, 2003.

[37] K. Delac, M. Grgic, and M. S. Bartlett, editors. *Recent Advances in Face Recognition.* IN-TECH, 2008.

[38] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice Hall, 2008.

[39] R. Forchheimer and T. Kronander. Image Coding - from Waveforms to Animation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12), 1989.

[40] A. V. Oppenheim and J. S. Lim. The Importance of Phase in Signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

[41] M. Hayes, J. Lim, and A. V. Oppenheim. Signal Reconstruction from Phase or Magnitude. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):672–680, 1980.

[42] T. Ojala, M. Pietikäinen, and D. Harwood. A Comparative Study of Texture Measures with Classification based on Featured Distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[43] G. Zhao and M. Pietikäinen. Dynamic Texture Recognition using Local Binary Patterns with an Application to Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

[44] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[45] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:2037–2041, 2006.

[46] J. Macqueen. Some Methods for Classification and Analysis of Multivariate Observations. *In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley*, pages 281–297, 1967.

[47] S. Lloyd. Least Squares Quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[48] T. Kanungo, D. M. Mount, N. S. Netanyahu, D. C. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[49] Mathworks. Matlab: the language of technical computing. Website, 1994. http://www.mathworks.com/products/matlab/: Last Accessed: 07 May 2010.

[50] J. Laaksonen, M. Koskela, and E. Oja. Picsom-self-organizing Image Retrieval with MPEG-7 Content Descriptors. *IEEE Transactions on Neural Networks*, 13(4):841–853, 2002.

[51] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`: Last Accessed: 05 May 2010.

[52] C.-C. Chang and C.-J. Lin. *A Practical Guide to Support Vector Classification*, 2010.

[53] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.

[54] R. Bhagavatula and M. Savvides. Eigen and Fisher-Fourier Spectra for Shift Invariant Pose-Tolerant Face Recognition. In *Proceedings of International Conference on Advances in Pattern Recognition*, pages 351–359. Springer, 2005.

[55] Y. Z. Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790 –799, 1995.