THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Department of Computing

iJADE Face Recognizer - A Multi-Agent Based Pose and Scale Invariant Human Face Recognition System

Tony Ao Ieong Wai Heng

A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Philosophy

June 2006

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________(Signed)

_____Tony Ao Ieong Wai Heng_______(Name of student)

Abstract

Human face recognition is a challenging pattern recognition problem. Problems with current face recognition systems include the difficulty of recognizing faces presented at various angles, at various scales, with varying facial expressions, under uneven illumination, and in cluttered scenes. This research proposes an automated face recognition system that attempts to deal with these difficulties at each stage of the face recognition processes: face detection, feature extraction, and face identification. In face detection, we deal with the issue of detecting and localizing faces in a cluttered scene. We begin by identifying possible skin regions using a Gaussian mixture model of human skin tones on the HSV color space. We then use Neural Networks to classify and detect faces within these skin regions at a variety of image scales and use a committee network to detect face with different poses. In the next step, feature extraction, we use Gabor features derived from Gabor wavelet representation to extract the most salient, distinctive, and invariant facial features. We adopt the Gabor feature extraction technique because these features are robust to change in illumination and facial expression. To identify faces presented at a variety of angles, we adopt template-based Gabor Eigenface features and feature-based Gabor Labeled Elastic Graph face features. The third and final step, face identification attempts to determine the identity of a query face by comparing it with a face database. At this stage, the search operation is conducted using agent technology and agent technology allows the system to operate at various scales and on various face models without any deterioration in the system performance. The proposed approach was tested on a variety of images and face databases and compared with other face recognition models to show the benefits and improvement.

This research was implemented on iJADE Face Recognizer, a multi-agent based pose and scale invariant human face recognition system. iJADE (intelligent Java Agent Development Environment) is an intelligent agent-based platform for supporting the implementation of artificial intelligence (AI) functionalities. Each of the three major processes in the proposed face recognition system has been designed in a separate module and each module is implemented using intelligent agents. Agent technology provides a collaborative platform that can be used to build sophisticated applications and allows different agents to interact with each other. The performance of the iJADE Face Recognizer is superior to that of other approaches because the agent platform operates in an asynchronous runtime environment which enables parallel processing and under a distributed architecture supports a number of varied operating scales. iJADE Face Recognizer has numerous potential applications. For example, as an automatic human face surveillance system, it can automatically analyze scenes and extract human faces in complex environments and can efficiently and effectively perform invariant human face feature extraction, identification and recognition.

Publications

Papers Published/Accepted

1 Tony W.H. Ao Ieong and Raymond S.T. Lee, "iJADE Face Recognizer - A Multi-Agent Based Pose and Scale Invariant Human Face Recognition System", *Knowledge-based Intelligent Information and Engineering Systems (LNAI 3214)***,** pp. 594-601, Springer-Verlag Berlin Heidelberg, 2004.

2 Tony W.H. Ao Ieong, Toby H.W. Lam, Alex C.M. Lee and Raymond S.T. Lee, "iJADE Tourist Guide – A Mobile Location-Awareness Agent-Based System for Tourist Guiding", accepted to be appeared in the *Knowledge-based Intelligent Information and Engineering Systems, Lecture Notes in AI as part of the LNCS/LNAI series*, Springer-Verlag, 2005.

3 Toby H.W. Lam, Tony W. H. Ao Ieong and Raymond S.T. Lee, "Silhouette Spatio-temporal Spectrum (SStS) for Gait-based Human Recognition", *Proceedings of the 3rd International Conference on Advances in Pattern Recognition (ICAPR 2005),* pp. 309-315, 2005.

4 King Hong Cheung, Jane You, James Liu and Tony W. H. Ao Ieong, "Appearance-Based Face Recognition Using Aggregated 2D Gabor Features", *Knowledge-based Intelligent Information and Engineering Systems (LNAI 3214)***,** pp. 572-579, Springer-Verlag Berlin Heidelberg, 2004.

Papers Submitted

5 Tony Ao Ieong, Toby Lam, Wing Lin and Alex Lee, "iJADE Tourist Guide – An intelligent Ontology-based Location-Aware Agent-Based Tourist Guide System", submitted to *IEEE Transactions on Knowledge and Data Engineering (TKDE).*

6 Eddie Chan, Tony Ao Ieong and Raymond Lee, "The Design and Implementation

of a Multi-Agent Based Intelligent Context –Aware News Reporting System – iJADE News Reporter", submitted to *IEEE Transactions on Knowledge and Data Engineering (Special Issue on Knowledge and Data Engineering in the Semantic Web Era).*

7 Florence Tang, Tony Ao Ieong and Raymond Lee, "iJADE News Personalizer: Adaptive Agent-based News Retrieval & Personalization System", submitted to *Web Intelligent to Web Intelligent and Agent Systems: An International Journal (WIAS).*

8 Burly Tan, Tony Ao Ieong and Toby Lam, "iJADE SCM System – The Design and Implementation of an Intelligent Agent-based Supply Chain Management", submitted to *IEEE Systems, Man and Cybernetics (SMC) Part A.*

9 Toby Lam, Tony Ao Ieong and Raymond Lee, "Ontia iJADE: An Intelligent Ontology-based Agent Framework for Semantic Web Service", submitted to *IEEE Transactions on Knowledge and Data Engineering (TKDE).*

Acknowledgement

I wish to take this opportunity to express my sincere gratitude to the following bodies:

1. The Department of Computing of the Hong Kong Polytechnic University for providing necessary resources and facilities for my research and studies;
2. My supervisor Dr. Raymond Lee for his encouragement and guidance to me throughout my studies;
3. Mr. Martin Kyle for his kind and professional proofreading of this thesis; and
4. My family and friends, in particular, Erica and Toby, for their care, love and support to me through the years of my studies.

I am certain that without the generous and tremendous support, encouragement and assistance from the above-mentioned bodies I would not be able to complete my studies in the Degree of Master of Philosophy.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Humans identify each other by their faces. Humans can do this quickly and easily and this skill is just as well demonstrated in recognizing people in portraits and photographs. This skill is quite robust against large changes in viewing angle, lighting condition, or pose, and notwithstanding occlusions, changes in facial expression and hairstyle, or the effects of aging and allows humans to quickly identify people from a multitude of person entries stored in the brain.

The human face is a natural choice as a biometric element for use in identification, being both familiar and accessible and socially acceptable and now that advances in the computational power of computers allow the application of ever more complex algorithms. The face is becoming proportionally more viable as a widespread method of personal identification. There are numerous applications now in government use and on the market, in criminal identification, security systems and credit card verification. Just one example is FaceOK [FaceOK], which provides an online face authentication solution and real-time face surveillance system.

In general, there are two types of face comparison scenarios: identification/recognition and verification/authentication [Gong 2000, Takacs & Wechsler 1997]. In Identification/recognition, an image of an unknown individual is collected and the individual is identified after a search of a large set of images. The size of the gallery is typically about 50-100 and in many applications may involve thousands of images. Airport surveillance systems might for various reasons use identification/recognition to identify individual passengers. In verification/authentication, a query face image is compared against a template face image in order to verify a claim of identity. Systems that compare an individual against a mug shot or security badge photo would use for verification/authentication.

There are limitations to contemporary face recognition or surveillance systems. They can recognize faces only under constrained conditions and in simple environments, where the face view is frontal, there is little variance in illumination, pose, facial expression and without occlusions. In this research we propose an automated human face recognition system on color image to deal with this problematic variation. The proposed system integrates intelligent agent technologies, to deliver an accurate, efficient and robust agent-based face recognition system that can deal with variation in uncontrolled environments.

1.1. Motivations

Contemporary face recognition systems can recognize faces mainly in a constrained environment and very few systems can operate on cluttered scenes. The proposed face recognition system deals with this in two ways. Firstly, the system exploits the color information in color images. This approach allows processing to be focused only on facial data, speeding up the process and reducing the false detection rate. Secondly, the system uses face poses information to improve recognition accuracy.

The system performance of the proposed face recognition system is further improved by being integrated into an intelligent multi-agent-based platform which supports the implementation of artificial intelligence (AI) functionalities. We use this agent technology because it provides an asynchronous runtime environment and distributed architecture that allows the automatic delegation of tasks to autonomous, highly mobile agents, creating an intelligent and automatic system that can perform recognition tasks robustly and efficiently. The multi agent technology also increases the efficiency and scalability of the proposed system and the system can operate in multi-scales and models without substantial deterioration of the system performance. The proposed face recognition system may serve as an automatic human face surveillance system. Using the face recognition models and intelligent agent technologies proposed in this work, it can provide an efficient and automatic scene analysis and figure-ground segmentation in complex environments and a reliable human face feature extraction, identification and recognition system.

1.2. Objectives

Given a color image of a cluttered scene, the system should locate all the human faces in the image and recognize and match each identity against a face database. The human faces in the image may appear different from their images in the database. These variations may be in terms of poses, scales, facial expressions, evenness of illumination and makeup.

A multi-agent based pose and scale invariant human face recognition system – iJADE (intelligent Java Agent-based Development Environment) Face Recognizer has been developed that can operate robustly and quickly under poor, uncontrolled, and changing conditions. It can identify human faces from a color image of a cluttered scene. The recognition rate is improved by using appearance and feature-based face recognition methods. The efficiency and scalability of the system are greatly increased by using mobile agent technology.

1.3. Organization of this Thesis

This thesis consists of six chapters, organized as follows:

- Chapter 1 introduces the work.
- Chapter 2 introduces face recognition, including background to face recognition, the relevant literature, and the description of the processes and techniques of a face recognition system.
- Chapter 3 introduces agent technology. This chapter provides the background to agent-based systems and discusses the benefit of using agent technology. As the iJADE model is adapted in this research, this chapter also presents details of the iJADE model and its practical applications.
- Chapter 4 proposes the iJADE Face Recognizer. This chapter describes the system architecture of the proposed system, the details of the algorithm involved in each module and the system flow.
- Chapter 5 presents the experimental results of the proposed model. The experiment setup and results of each module are described in detail, and compare the recognition rate of the proposed model with other contemporary face recognition methods.
- Chapter 6 concludes this thesis, which summarizes its contributions, discusses limitations of the proposed models and suggests directions for further work.

Chapter 2. Literature Review - Face Recognition

This chapter covers the basic knowledge and literature review on face recognition. It is organized in 6 major sections. Section 2.1 figures out the difficulties encountered in face recognition. Section 2.2 reviews the literature on contemporary face recognition models. Section 2.3 introduces the face recognition system, while Sections 2.4, 2.5 and 2.6 respectively describe the techniques and models involved in the face detection, feature extraction, and face identification processes.

2.1. Challenges in Face Recognition

The main challenge in vision-based face recognition is the presence of a high degree of variability in human face images. While there can be small inter-class variations due to the similarity of individual appearances, there can be potentially very large intra-class variations due to head pose, lighting, facial expression, hair style, and aging. Before We start to design the algorithm to tackle the face recognition problem, We should investigate what kinds of difficulties are faced within the process. There are two types of variation. A face can change its appearance due to either intrinsic or extrinsic factors [Gong 2000, Yang 2002].

Intrinsic variation takes place independently of any observer and is due purely to the physical nature of the face. It includes identity, facial expression and disguises. Each person has his/her own face and we can identify by it. We express our feelings with our faces. Disguises refer to glasses, hair style, or makeup that cause variations on face (Figure 1). Besides, aging is also a factor.

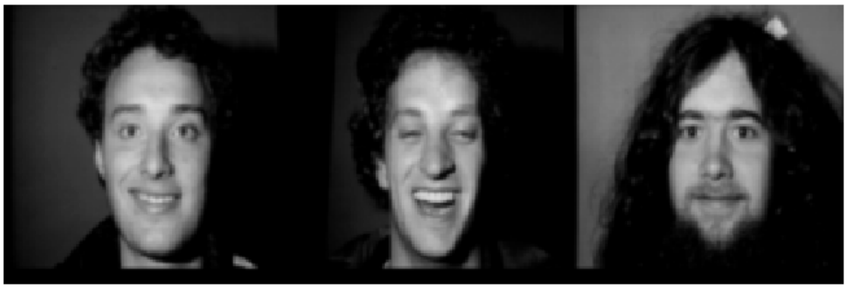


Figure 1: Faces with different expressions and with a disguise. [Samaria & Harter 1994]

Extrinsic variation arises when the face is observed via the interaction of light with the face and the observer. Common factors are view geometry, illumination and occlusion. Different viewing angles generate different poses and different scales of face, such as frontal, left and right profile faces and these being the most significant sources of variations. Illumination variance is due to the condition and direction of lighting, and in particular to self-shadowing from other facial features (Figure 2). Other objects present in the scene can also cause occlusion of the face so that part of a facial feature is lost. In fact, such type of extrinsic variation is most difficult to deal with.

Figure 2: Face with different pose and illumination. [Samaria & Harter 1994]

So the tasks of faces perception have to be performed consistently and robustly under all sorts of changes in external conditions characterized by the extrinsic sources of variations.

2.2. Face Recognition Models

Current vision-based recognition techniques can be mainly categorized into two groups based on the kind of face representation used [Brunelli & Paggio 1993]:

- Template-based representation uses holistic texture features, stores the whole face pattern in an array and compares them using a suitable metric such as the Euclidean distance [Belhumeur 1997].
- Feature-based representation uses geometrical features of the face, for example by extracting the relative position and attributes of distinctive features of the eyes, mouth, eye brow, and nose [Jeng 1998].

Experimental results show that Template-based methods generally perform better in recognition tasks than those methods based on templates. This is because it is difficult to robustly extract geometrical features, especially in face images of low resolution and of poor quality [Brunelli & Paggio 1993]. However, template-based recognition techniques have their own limitations in recognizing human faces in images with wide variations in head pose and in illumination. The following subsections describe the contemporary face recognition models that use these two approaches.

2.2.1. Template-based Approach

One of the most widely accepted algorithms in template-based approach is the Eigenface Method [Belhumeur 1997]. This method is based on Principal Components Analysis (PCA) technique. One of its main advantages is the dimensionality reduction scheme that enables recognition to be performed rapidly.

Fisherface is another well-known template-based approach. It is based on Linear Discrimination Analysis (LDA) [Belhumeur 1997]. LDA a is classical statistical technique using the projection which maximizes the ratio of scatter among the data of different classes to the scatter within the data of the same class. Features obtained by LDA are useful for pattern classification since they make the data of same class closer to each other and the data of different classes further away from each other. A LDA mixture model proposed by Kim [Kim 2003] suggests that all classes are partitioned into several clusters and obtain a transformation matrix for each cluster. This can improve classification performance as the inadequacy of one transformation matrix over the whole data in the LDA model is overcome.

Experiments show, however, that the eigenface and fisherface methods are not robust in dealing with variations in lighting conditions. To overcome these problems, wavelets decomposition [Chui 1992] is adopted to break images into approximations and details of different levels of scales and work on several approximations images of each face. The Gabor wavelet filter was selected by Li and Liu [Li & Liu 2002] and Liu and Wechsler [Liu & Wechsler 2002] for the enhancement of the eigenface and fisherface respectively. The Gabor filter provides robustness against varying brightness and contrast of images. Furthermore, the filters can model the receptive fields of a simple cell in primary visual cortex. Wavelet decomposition has been widely used in various applications such as palmprint recognition [Kong 2003], object recognition [Zhang 2002], Chinese and English language character recognition [Tao 2001].

Barnsley and Jacquin proposed the use of a fractal image code in a template-based approach [Jacquin 1993]. This approach uses the concept of contractive Iterated Function System (IFS) in fractal object formation. Fractals are mathematical sets that exhibit self-similarity under different scales of magnification. Figure 3 shows an example of the construction of a fractal object, formally known as the Sierpinski Triangle. This fractal code is represented by three contractive and iterative affine transformations of itself. On continued application of the transformations to each successive resultant image, an approximation to the attractor is reached as shown in Figure 3(d), which remains approximately the same with further iterations. The attractor of a fractal is invariant to further applications of the code [Crownover 1995].

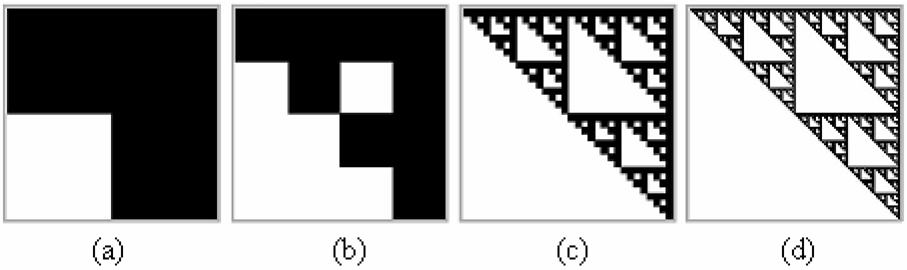


Figure 3: The iterative process that generates the Sierpinski Triangle fractal. (a) The first iteration using the Sieprinski Triangle transformations with a black input image, (b) second iteration, (c) fifth iteration, (d) tenth iteration (this is an approximation of the attractor). [Crownover 1995]

Fractal Image Code (FIC) was originally used in image compression but is now also proposed for use in face recognition [Hossien 2001, Tan 1999, Chandran & Kar 2002, Hossein & Chadran 2001, Tan & Yan 2001, Tan & Yan 2002, Kouzani & Sammut 2000]. Fractal image code depends on self-similarity and on applying the reference set of Partitioned Iterated Function System (PIFS), encoding an image into its fractal approximation then searching for self-similar sub-regions of the image. In particular, a larger domain block maps to a smaller domain block maps to a smaller range block through a geometric and affine transformation.

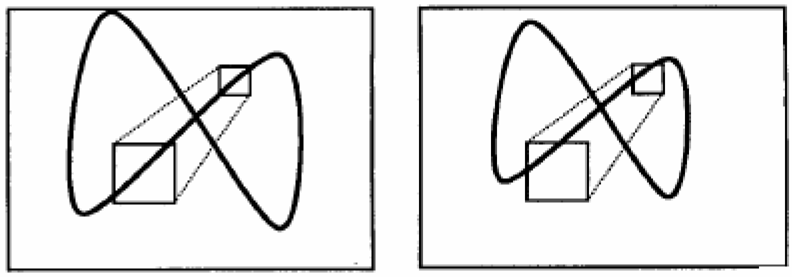


Figure 4: Illustrations of domain and region block mapping. [Tan 1999]

As shown in Figure 4, the left-hand illustration depicts an arbitrary image with a domain (big box) to range (small box) block transformation. The right-hand illustration shows that same image after certain rotation, scaling and shifting operations, with the same domain to range block transformation. In both cases, that same transformation captures the self-similarities in the image. The mapping coordinates of contractive domain to range block transformation in PIFS forms the fractal image code. In face recognition, we generate all the fractal image code of each face in a gallery first. The difference between the original unknown face image and the same image after applying the fractal image code transformation of the matching face image in the gallery to the unknown face image is then determined using the Euclidean distance measure. This results in a new distance measure that refers to the fractal neighbor distance (FND) [Tan & Yan 2001, Tan & Yan 2002]. Recognition of an unknown face is achieved by selecting the entry from the gallery of fractal codes that minimizes the FND. This method is invariant to some degrees of translation, rotation, scaling and differences in illumination without extra effort, unlike other template-based methods [Hossein & Chadran 2001, Kouzani & Sammut 2000].

2.2.2. Feature-based Approach

Feature-based approach extracts landmark features from the major components of the human face such as the eye, eyebrow, mouth, nose, and ear. A Face Bunch Graph (FBG) models a face as an elastic graph and extracts the facial landmarks from each node in the graph [Wiskott 1997]. Elastic graph matching is performed on the target face image and each node should locate to its corresponding facial landmark inside the image even if the viewpoint changes during matching. The extracted feature from each node in the graph is based on a Gabor wavelet transform, and it is called a “jet”. The phase of complex Gabor wavelet coefficients in a jet could help to achieve a more accurate location of the nodes and disambiguate patterns which might be similar in their coefficient magnitudes.

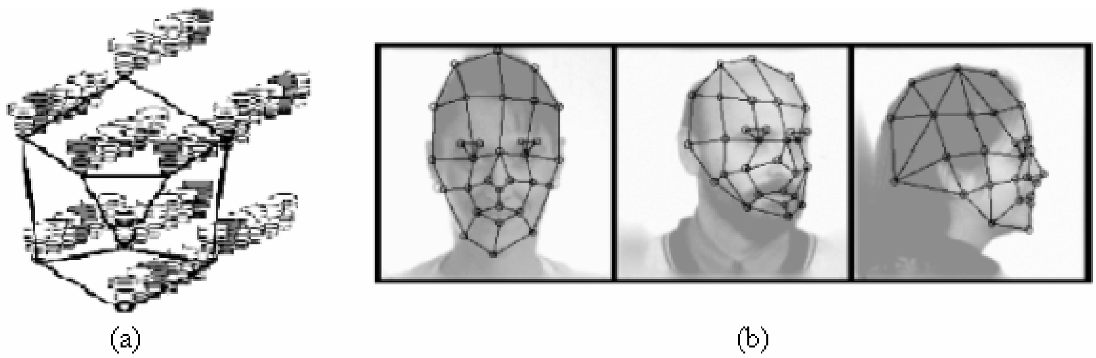


Figure 5: Illustrations of (a) Face Bunch Graph whose nodes are associated with bunch of jets, (b) Object adapted grids for different pose. [Wiskott 1997]

Figure 5(a) shows each node in a FBG which is associated with a bunch of jets, and each set of jets in all nodes corresponds to the whole facial landmark feature of a face. Figure 5(b) shows the target locations of each node after the elastic graph is matched to a face image with different poses. By matching the elastic face graph to a known face and extracting a set of jets, face matching can be done by maximizing a graph similarity between an image and the jet in the FBG having an identical pose. Another similar approach from Lee [Lee 2003] applies an Active Contour Model (ACM) to face detection and feature extraction of facial landmarks using a snake method [Davison 2000] and using an Elastic Graph Dynamic Link Model (EGDLM) to perform invariant human face identification. The idea is to use a snake, which is a continuous curve, to deform itself to trace the contour of the face and then extract the features in facial landmarks using a Morlet wavelet filter. This is illustrated in Figure 6. Face matching can be done by finding the object having the minimum difference.

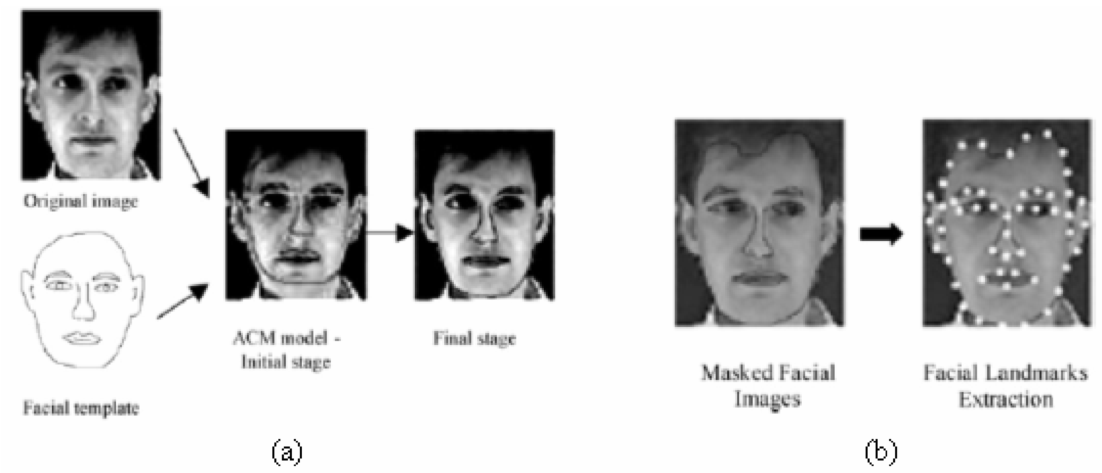


Figure 6: (a) Facial contour extraction using ACM, (b) facial feature extraction from landmarks (white points). [Lee 2003]

Increases in available computational power now allow the use of computationally intensive 3D face modeling methods. Blanz and Vetter present a model of 3D faces morphable into images for recognition [Blanz & Vetter 2003]. The model can overcome variations in pose, ranging from frontal to profile views, and a wide range of illuminations, including cast shadows and specular reflections. Promising results are achieved in their proposed algorithm, 95.0% and 95.9% correct in large variations in pose and illumination respectively.

2.3. Major Processes in Face Recognition System

Automated face recognition involves several important processes: face detection for locating human faces, face tracking for following moving subjects, face modeling for representing human faces, feature extraction for taking the salient feature from face, and face identification for comparing represented faces and identifying a query subject. These processes are illustrated in Figure 7.

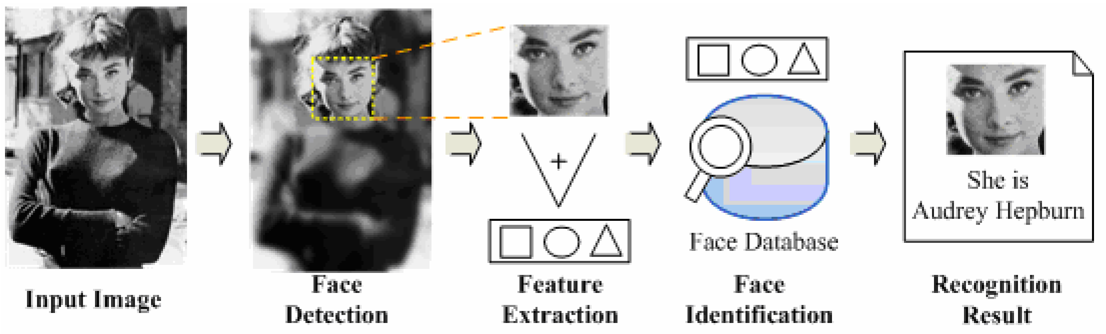


Figure 7: Illustration of main processes in face recognition system

2.4. Face Detection Process

The first step, face detection, is a crucial step in the recognition process. Face detection is required so that one can focus on the processing of the facial data only [Gong 2000]. Faces have to be detected from a scene, whether the scene is simple or complex. Firstly, illumination or color intensity normalization should carry out to reduce the bias from lighting variance in order to provide a better detection rate at later steps. Secondly, in order to find candidate faces within the entire image, face localization is carried out by locating facial features. Skin tone color, the eye, eyebrow, mouth, nose and ear are all facial features that can be used to detect a face. Thirdly, to help eliminate false positive candidate faces, we conduct eye pair or eye-mouth triangle location [Hsu 2002].

Another approach is to match a face template by scanning with multi-resolutions of the image. If a non-face template is provided, this can be viewed as a two-class (face vs. non-faces) classification problem [Liu 2003]. A good face detection scheme can greatly reduce processing expenses in the later stage. Several factors make the detection problem difficult. Even if the viewing geometry is fairly well controlled (e.g. the images are all face-frontal), the presence of facial hair, make up, etc., will obscure or alter the appearance of facial features which is essential for many detection schemes. The presence of noise, occlusions, and variations in scale and orientation makes this problem even harder. Humans handle these problems by extracting extra information from the entire image such as the location of the body, but for machines, this merely creates a body recognition problem.

One possible strategy for determining whether the image contains any faces is to start with low-level cues, such as edges, color, and/or motion, and to combine them by means of structural models. This will allow the location of potential face targets. Rein-Lein suggested that color information is used to simplify face localization in complex environments [Hsu 2002]. The proposed algorithm works on the presence of varying lighting conditions and complex backgrounds. After minimizing variations in static color images using lighting compensation techniques and a nonlinear color transformation, it uses skin color detection to detect faces. To make the skin color luma-independent, it adopted the YCbCr color space in color transformation, as this color space is perceptually uniform. The next step is to detect the eyes and mouth in different components in YCbCr color space. Figure 8 illustrates the construction of a mouth map in mouth detection.

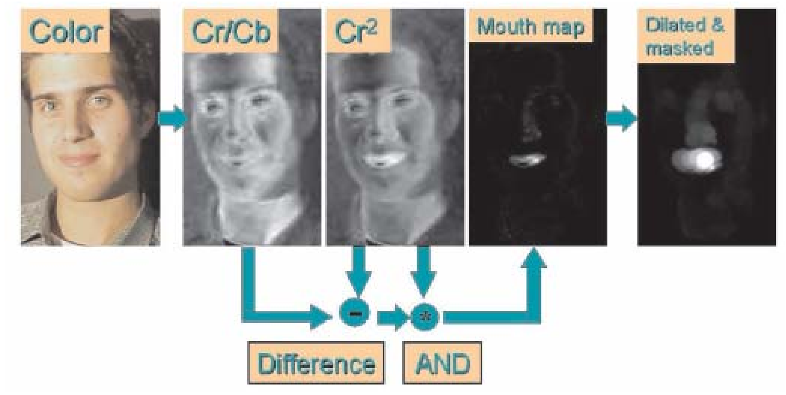


Figure 8: Construction of mouth map [Hsu 2002]

Finally, it forms an eye-mouth triangle and verifies with the skin regions and then utilizes the Hough transform to extract the best-fitting ellipse to represent the face. Figure 9 illustrates the detection results and also works in complex scene with multiple faces.

Figure 9: Face detection result on half-profile faces (some with facial hair) [Hsu 2002]

Gabor wavelet network is used in tracking facial features in face detection [Feris & Cesar 2000, Feris 2001]. It combines Gabor wavelet filter and a radial basis feedforward network to perform detection and a discrete face template is represented as a linear combination of continuous 2D odd-Gabor wavelet functions. Efficient real time face tracking in wavelet subspace by orthogonally projecting the video sequence frames into wavelet subspace and detects the wavelet based face template. It is robust to facial expressions and affine deformations. The number of wavelets in the representation can be adjusted according to the available computational resources.

The aim of face detection is to locate all of the faces regions in the image and extract them for recognition. An effective perceptual system must exhibit the ability to direct information processing to the most relevant aspects of the perceptual input in the basis of a coarse analysis of that input. In order to analyze and recognize human faces in realistically unconstrained environments, focused visual attention can be crucial to the correctness and effectiveness of the desired interpretation. Color, as a low-level cue, provides particularly useful and complementary pre-attentive and knowledge-based visual cues for focusing attention on human faces [Yang 2002].

2.4.1. Color Cue in Complex Scene

In recent years, an increasing body of research has addressed the specific problem of automatic face detection based on skin color [Hsu 2002, Vang & Yuan 2001, Birchfield 1998]. RGB signals reside in a 3D color space and each RGB pixel is then a point in this space. The pixels of a face image form a distribution in this color space which can be modeled by estimating a probability density function. Intensity is distributed across the RGB values so that a face's distribution varies with scene brightness. There is a research on comparing performance of a skin chrominance model that is invariant to scene brightness [Terrillon 2000]. We picked normalized RGB and HSV color spaces to do the analysis.

2.4.2. Normalized RGB Color Model

The normalized RGB model is considered to be capable of characterizing human faces with less variance in color. Colors of each pixel are expressed by the combination of RGB components, and the brightness value I = R+G+B. Since the color information is very sensitive to the brightness value of the pixel, each color component value can be normalized with the brightness value I (R+G+B) as follows:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \tag{2.1}$$

where r + g + b = 1. Indeed, the normalized color values can be expressed only with r and g.

About 7 million skin color pixels were collected from the Internet and movies. These consist of 250 images of human skin from Asians only. We extract the r and g components from the pixels in a normalized RGB model and the pixels are allocated in 100 x 100 bins and the distribution is plotted as shown in Figure 10. Figure 10(a) shows the skin color distribution among the r and g components, while 10(b), 10(c) and 10(d) shows the top down view, r component profile view and g profile view respectively. We can observe that the skin color falls in a small cluster in the normalized RGB color model and forms a bell shape, which can be estimated by a Gaussian model.

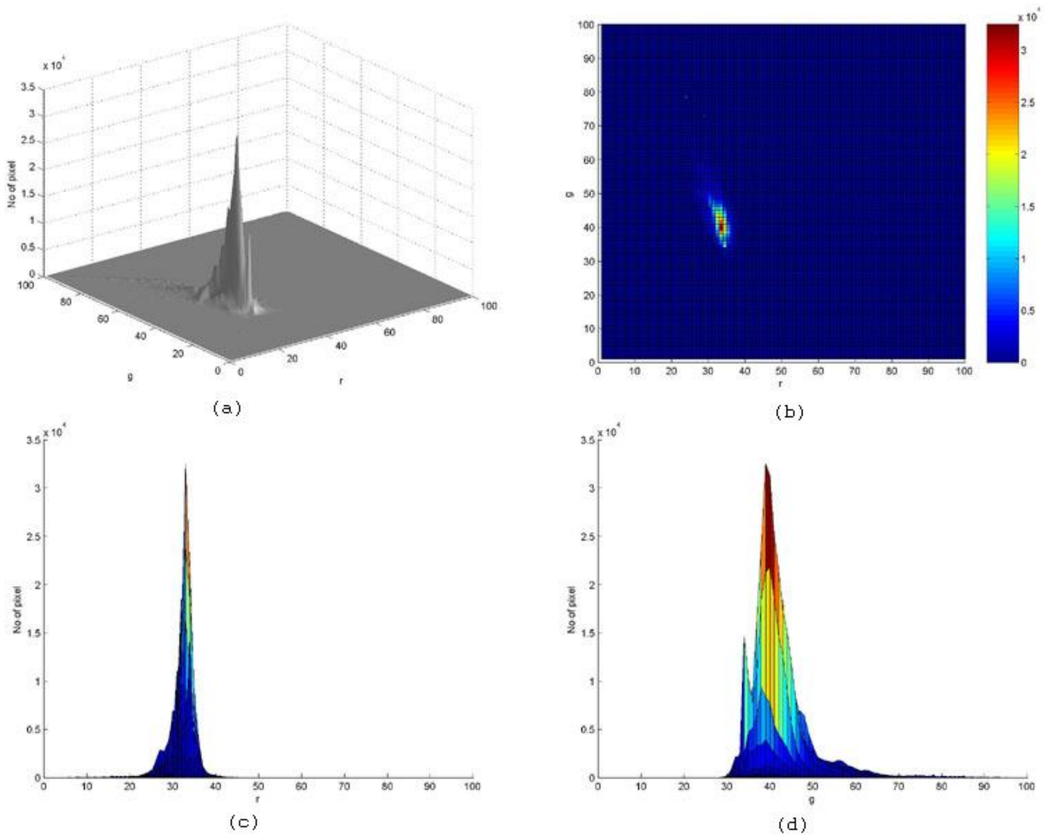


Figure 10: Distribution of skin color in r g components of normalized RGB color model

2.4.3. HSV Color Model

HSV consists of Hue (H), Saturation (S) and Value (V). Hue corresponds to the intuitive notion of “color” while saturation is the vividness or purity of that color and Value defines the brightness of that color as shown in Figure 11. HSV separates the brightness component and so it is invariant to the illumination effect and gives a better density function on skin color.

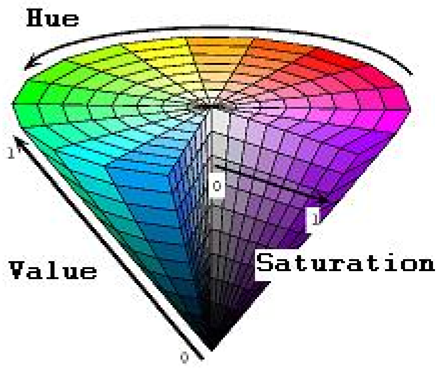


Figure 11: HSV color model

It is hard to estimate the probability density function (pdf) of skin color in normal RGB color space as it is sensitive to illumination, so non-linear transforms to a color model which separates the illumination component and is useful for analysis. HSV color model is suitable for use in estimating the skin color density distribution, and Gaussian distribution can characterize the properties of skin color. We pick out the HS components to plot the distribution in Figure 12. Although the skin color falls into a small cluster in HS color space, it is suggested that a single Gaussian distribution is not sufficient to model human skin color [Yang & Ahuja 2001, Caetano 2003].

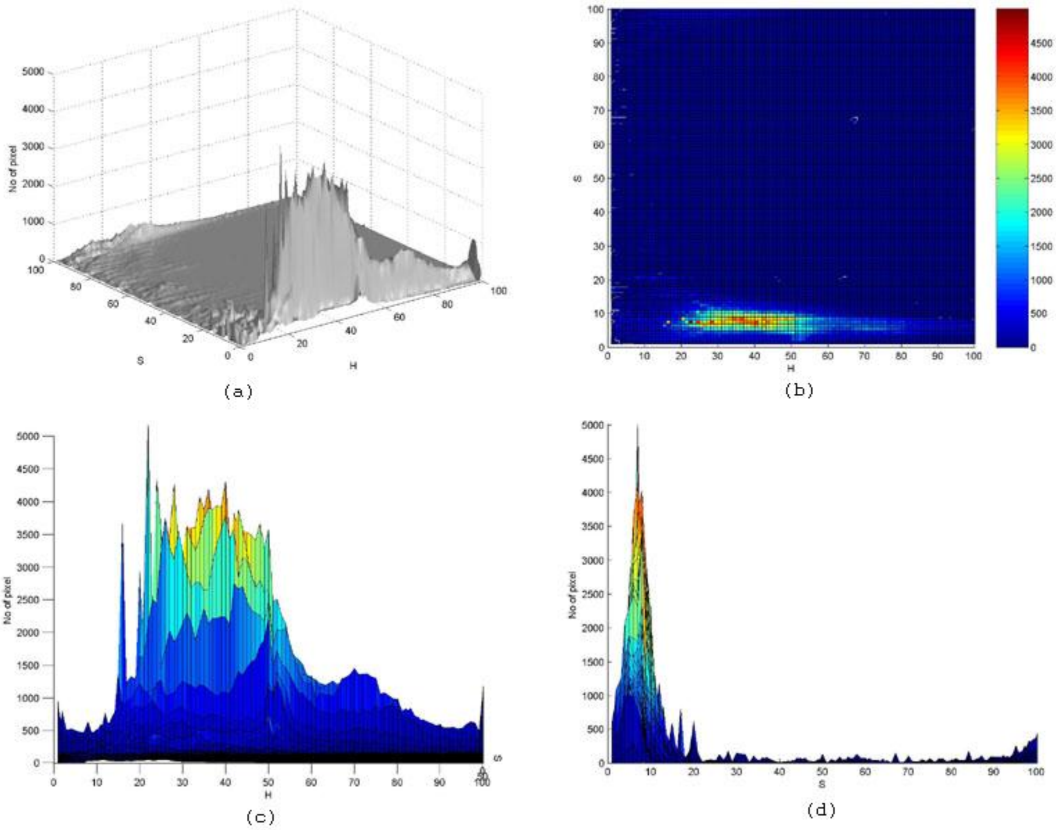


Figure 12: Distribution of skin color in HS components of HSV color model

2.4.4. Multivariate Gaussian Model

It has been suggested that skin color can be modeled by a single Gaussian pdf, even for samples coming from different ethnic groups [Gong 2000, Yang & Ahuja 2001, Caetano & Barone 2001]. Considering independence between the samples, the model can be obtained from the maximum-likelihood criterion, which has an analytical solution for a single Gaussian pdf and results in the following estimation:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_x \tag{2.2}$$

$$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{\mu})(x_k - \hat{\mu})^T \tag{2.3}$$

where n is the number of samples, x_k is the vector representing the k*th* sample, $\hat{\mu}$ is the mean vector and $\hat{\Sigma}$ the covariance matrix of the estimated Gaussian pdf. The resulting probability density that models the data is then

$$p(x) = \frac{1}{(2\pi)^{d/2}\left|\hat{\Sigma}\right|^{1/2}} \times \exp\left[-\frac{1}{2}(x-\hat{\mu})\hat{\Sigma}^{-1}(x-\hat{\mu})^T\right] \qquad (2.4)$$

where d is the dimension. The estimated single Gaussian model of r g color is shown in Figure 13 and that of HS color model is shown in Figure 14. The single Gaussian model provides better estimation in r g than does HS color model. This is because the distribution in HS is scattered, leading to over estimation that occurs in HS color model. Gaussian mixture model is adopted in order to provide a better estimation.

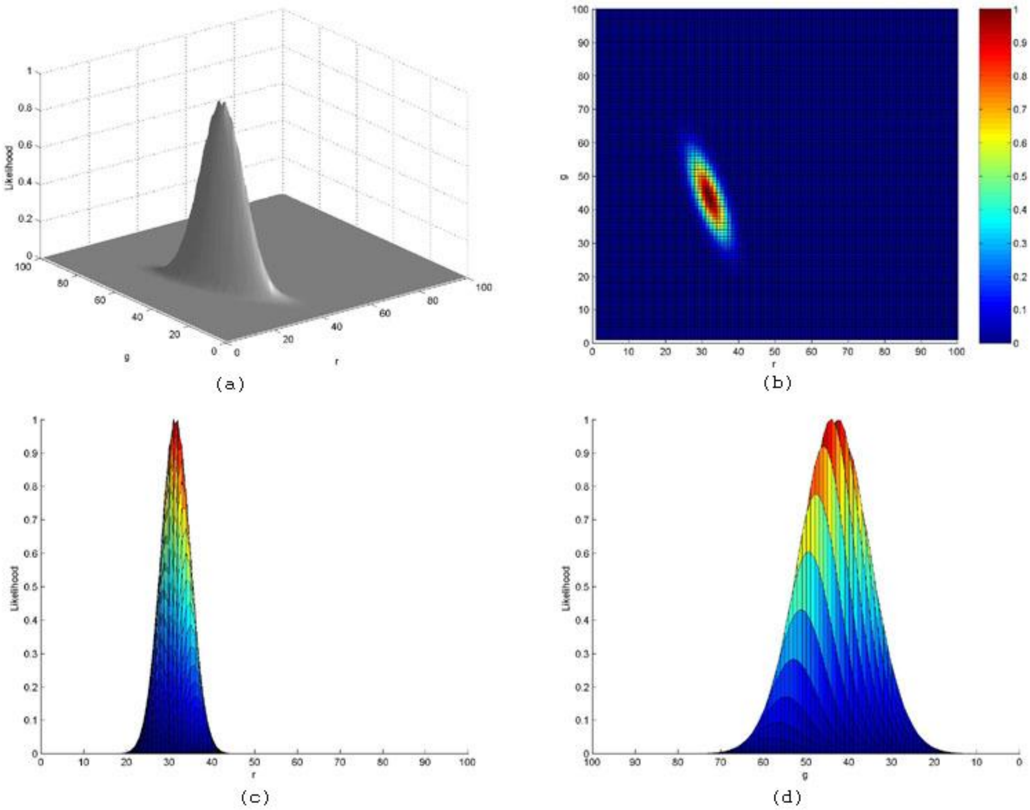


Figure 13: Estimated single Gaussian model in r g components in normalized RGB model

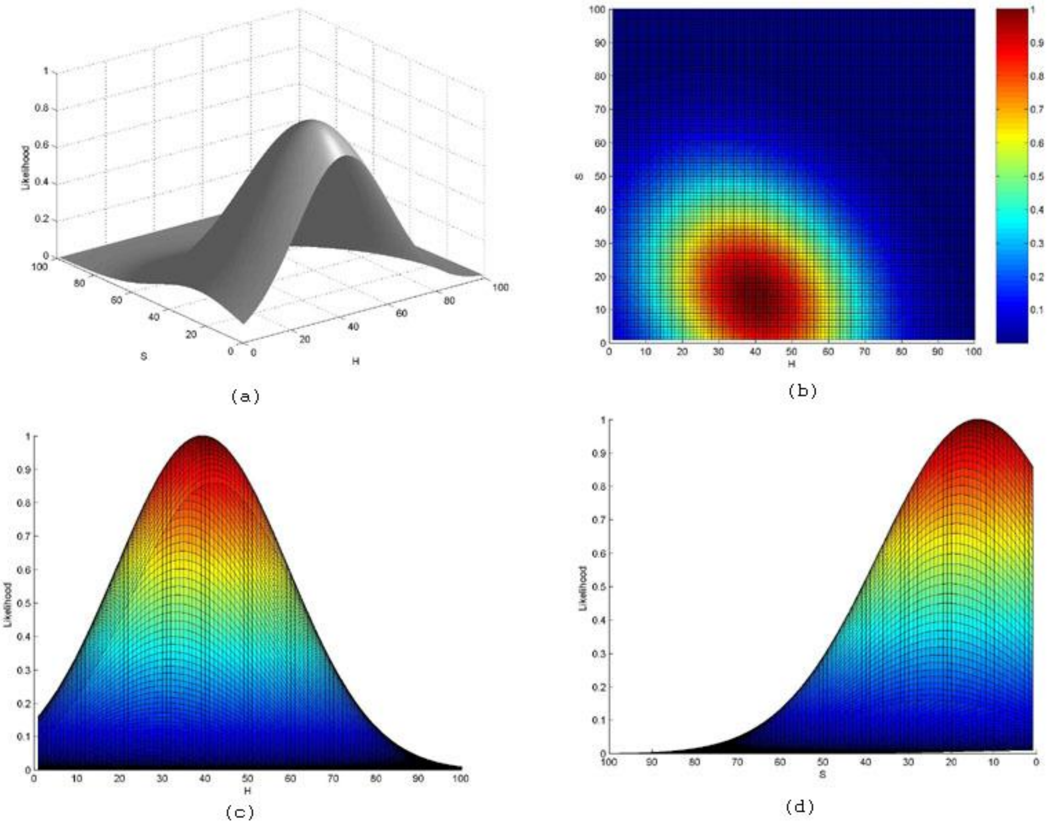


Figure 14: Estimated single Gaussian model in HS components in HSV color model

2.4.4.1. Gaussian Mixture Model

The motivation for using a Gaussian mixture is that, in spite of the fact that the whole data is well-clustered and one Gaussian can provide good performance, there are in fact two different races and it is reasonable to say that an optimal result can be obtained if we addresse one Gaussian to each race. A Gaussian mixture model [Gong 2000] is defined as:

$$p(x) = \sum p(x \mid k) P(k) \tag{2.5}$$

where $p(x \mid k)$, $k = 1 \ldots K$, are K Gaussian density functions. The parameters in such a model are the means, μ_k, covariance matrices, Σ_k, and mixing parameter, $P(k)$. These can be estimated from a data set using an Expectation-Maximizations (EM) algorithm

[Gong 2000].

2.4.4.2. Expectation-Maximization

Given a data set $X = \{x_1, \dots x_m, \dots x_M\}$, EM [Gong 2000] aims to find parameter values that maximize likelihood or, equivalently, minimize the negative log-likelihood function:

$$\varepsilon = -\sum_{m=1}^{M} \ln p(x_m) = -\sum_{m=1}^{M} \ln\left(\sum_{k=1}^{K} p(x_m \mid k)P(k)\right) \tag{2.6}$$

The EM algorithm is an iterative method for minimizing ε by monotonically reducing its value. The model parameters need to be initialized before applying EM. A simple initialization method is to assign the means to a randomly chosen subset of data points. An often effective method is the use of a clustering algorithm such as K-means to divide the data into disjoint subsets. A Gaussian component can then be assigned to each subset. Whereas K-means performs a ‘hard’ clustering of the data into disjoint subsets, a Gaussian mixture can be thought of as performing ‘soft’ clustering in which each data point belongs to a greater or lesser extent and to each of the Gaussian components or ‘clusters’.

Let the initial parameter values be labeled as *old* values. EM then performs an iterative approximation in order to try to find parameter values that maximize the likelihood by using the following updates rules:

(1) Evaluate the posterior probability for every mixture component *k*

$$p(k \mid x) = \frac{p(x \mid k)P(k)}{p(x)} \tag{2.7}$$

where $p(x \mid k)$ is the probability that data point x could be generated by the *k*th mixture component.

(2) Update the parameters to their *new* values μ_k^{new}, Σ_k^{new} and $P^{new}(k)$ where

$$\mu_k^{new} = \frac{\sum_{m=1}^{M} P^{old}(k \mid x_m) x_m}{\sum_{m=1}^{M} P^{old}(k \mid x_m)} \tag{2.8}$$

$$\Sigma_k^{new} = \frac{\sum_{m=1}^{M} P^{old}(k \mid x_m) \left[x_m - \mu_k^{new}\right]\left[x_m - \mu_k^{new}\right]^T}{\sum_{m=1}^{M} P^{old}(k \mid x_m)} \tag{2.9}$$

$$P^{new}(k) = \frac{1}{M} \sum_{m=1}^{M} P^{old}(k \mid x_m) \tag{2.10}$$

(3) Repeat steps (1) and (2) for a pre-determined number of iterations or until suitable convergence.

We used Netlab [Netlab] to generate the mixture model, and Figure 15 shows the estimated Gaussian mixture model with 2 Gaussians in HS color space. It provides better estimation than a single Gaussian model, and one Gaussian with small variance and the other with very large variance, so it cannot be observed directly.

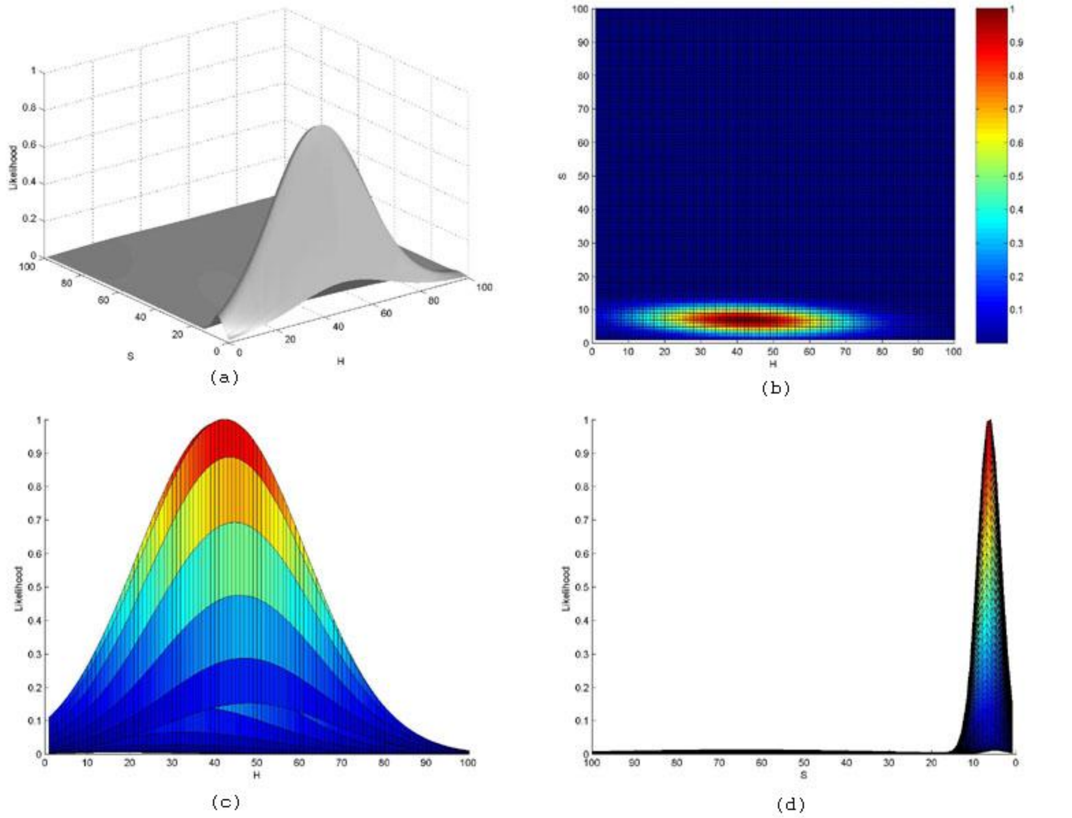


Figure 15: Estimated Gaussian mixture model with 2 Gaussians in HS components in HSV color model

2.4.5. Neural Networks for Face Detection

Neural Networks (NNs) are widely adopted in the field of pattern recognition due to their learning capability and their parallel nature. A neural network face recognition system (FADER – Face Detection and Recognition) [Aitkenhead & McDonald 2003] was developed. It fully utilizes NN to carry out the whole face recognition process including face detection, substructure detection (feature extraction) and facial recognition. Three different NN models are applied in each step, and a novel adaptation of the Hebbian connections strength adjustment model gave them better results than ordinary models. A neural networks committee machine is proposed for human face recognition. This machine uses multi-features of human faces using a NN committee machine [Zhao 2004], which consists of several independent neural networks trained using different image blocks of the original images in different feature domains. The final classification results of the committee machine represent a combined response of the individual networks. A new neural network model with the Constrained Generative Model (CGM) [Feraud 2001] was proposed to detect faces in images with complex backgrounds, and the model can detect side view faces and decrease the number of false alarms by means of a conditional mixture of networks. A neural networks based convolutional face finder [Garcia & Delakis 2004] was suggested to robustly detect highly variable face patterns, rotated up to ±20 degrees in image plane and turned up to ±60 degrees in complex real world images. The detection procedure acts like a pipeline of simple convolution and sub-sampling modules that treat the raw input image as a whole. The finder provides very high detection rates with a particularly low level of false positives without requiring the use of multiple networks for handling difficult cases. Here we study the mechanism of neural networks in details and determine how neural networks can be applied as classifiers in face detection.

A neural network can be defined as a model of reasoning based on the human brain. The brain consists of a densely interconnected set of nerve cells, or basic information-processing units, call neurons [Negnevitsky 2002]. Each neuron has a very simple structure, it consists of a cell body, soma, a number of fibres called dendrites, and a single long fibre called the axon. While dendrites branches into a network around the soma, the axon stretches out to other neurons. Figure 16 is a schematic of a biological neural network.

Signals are propagated from one neuron to another by complex electrochemical reactions. Chemical substances released from synapses cause a change in the electrical potential of the cell body. When the potential reaches its threshold, an electrical pulse, an action potential, is sent down through the axon. The pulse spreads out and eventually reaches synapses, causing them to increase or decrease their potential. In response to this stimulation pattern, neurons demonstrate long-term changes in the strength of their connections. Neurons also can form new connections with other neurons. Even entire collections of neurons may sometimes migrate from one place to another. These mechanisms form the basis for learning in the brain.

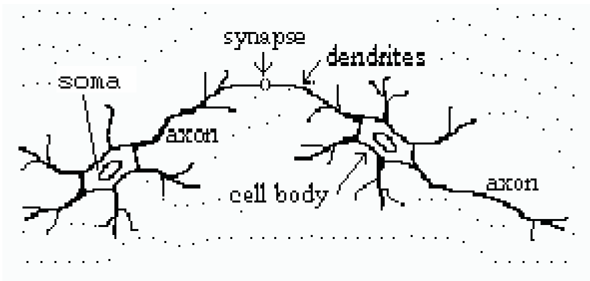


Figure 16: Biological neural network

2.4.5.1. Artificial Neural Network

Artificial neural networks (ANNs) [Haykin 1999] are biologically inspired and consist of a number of very simple and highly interconnected processors, also called neurons, which are analogous to the biological neurons in the brain. The neurons are connected by weighted links passing signals from one neuron to another. Each neuron receives a number of input signals through its connections and yet never produces more than a single output signal. The output signal is transmitted through the neuron’s connection and then splits into a number of branches that transmit the same signal. The signal is not divided among these branches. The outgoing branches terminate at the incoming connections of other neurons in the network. Figure 17 represents connections of a typical ANN.

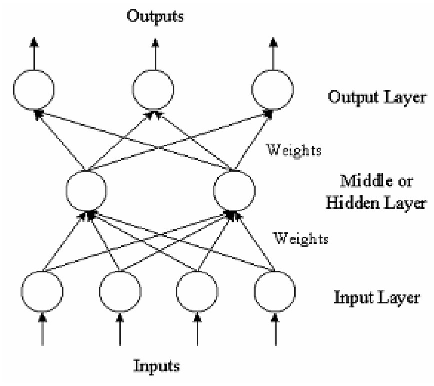


Figure 17: Architecture of a typical artificial neural network

2.4.5.2. The Perceptron

A perceptron [Haykin 1999] provides the procedure for training a simple ANN. It models a single neuron with adjustable synapse weights and a hard limiter as shown in Figure 18.

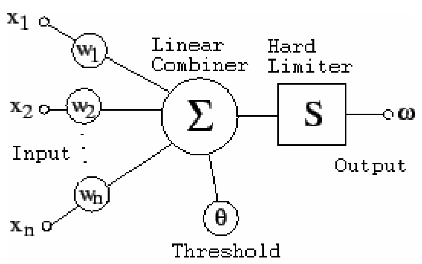


Figure 18: Single-layer perceptron

It consists of a linear combiner followed by a hard limiter. The weighted sum of the inputs is applied to the hard limiter, which by making a comparison with a threshold produces an output equal to +1 if its input is positive and equal to -1 if its input is negative. The aim of the perceptron is to classify inputs, or in other words externally applies stimuli $x_1, x_2, \ldots, x_n$, into one of two classes, say A_1 and A_2. Thus, in the case of an elementary perceptron, the n-dimensional space is divided by a hyperplane into two decision regions. The hyperplane is defined by the linearly separable function:

$$\sum_{i=1}^{n} x_i w_i - \theta = 0 \tag{2.11}$$

The perceptron learns its classification tasks by making small adjustments in the weights to reduce the difference between the actual and desired outputs of the perceptron. This is known as Hebb learning. The initial weights are randomly assigned, usually in the range [-0.5,0.5], and then updated to obtain the output consistent with the training examples. For a perceptron, the process of weight updating is particularly simple. If at iteration *p*, the actual output is Y(*p*) and the desired output is $Y_d(p)$, then the error is given by

$$e(p) = Y_d(p) - Y(p) \qquad \text{where } p = 1,2,3,\ldots \tag{2.12}$$

Iteration *p* here refers to the *p*th training example presented to the perceptron. If the error, *e(p)* is positive, we need to increase perceptron output *Y(p)*, but if it is negative, we need to decrease *Y(p)*. Taking into account that each perceptron input contributes $x_i(p)$ x $w_i(p)$ to the total input $w_i(p)$ tends to increase perceptron output *Y(p)*, whereas if $x_i(p)$ is negative, an increase in $w_i(p)$ tends to decrease *Y(p)*. Thus the following perceptron learning rule can be established:

$$w_i(p+1) = w_i(p) + \alpha \times x_i(p) \times e(p) \tag{2.13}$$

where α is the learning rate, a positive constant less than unity.

2.4.5.3. Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) [Haykin 1999] is a feedforward neural network with one or more hidden layers. Typically, the network consists of an input layer of source neurons, at least one middle or hidden layer of computation neurons, and an output layer of computational neurons. The input signals are propagated in a forward direction on a layer-by-layer basis as shown in Figure 17.

Neurons in the hidden layer detect the features. The weights of the neurons represent the features hidden in the input patterns. These features are then used by the output layer in determining the output pattern. With one hidden layer, it can represent any continuous function of the input signals and with two hidden layers even discontinuous functions can be represented. These are illustrated in Figure 19 below [Bishop 1995].

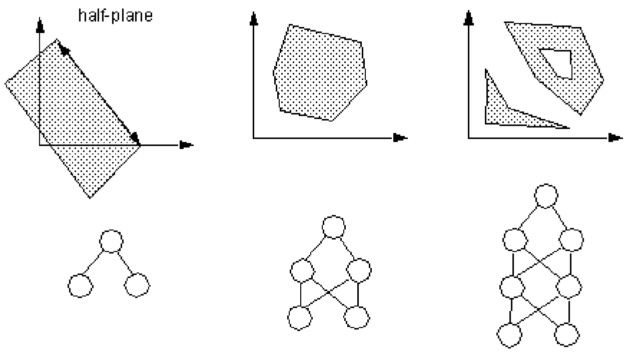


Figure 19: Classification in NN with different hidden layers, left with no hidden layer, center with one and right with two hidden layers.

Back-propagation is the learning algorithm in MLP. The learning algorithm has two phases. A training input pattern is presented to the network input layer. The network then propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated. A neuron determines its output in a manner similar to perceptron. Firstly, it computes the net weighted input as before:

$$X = \sum_{i=1}^{n} x_i w_i - \theta \qquad (2.14)$$

where n is the number of inputs, and θ is the threshold applied to the neuron. Secondly, this input value is value is passed through the activation function. However, unlike a perceptron, neurons in the back-propagation network use a sigmoid activation function:

$$Y^{sigmoid} = \frac{1}{1+e^{-X}} \qquad (2.15)$$

The derivative of this function is easy to compute. It also guarantees that the neuron output is bounded between 0 and 1.

The steps for the back-propagation training algorithm for MLP are as follows:

Step 1: Initialization

Set all the weights and threshold levels of the network to random uniformly distributed inside a small range.

Step 2: Activation

Activate the back-propagation neural network by applying inputs $x_1(p),x_2(p),\ldots,x_n(p)$ and desired outputs $y_{d,1}(p),y_{d,2}(p),\ldots,y_{d,n}(p)$.

(a) Calculates the actual outputs of the neurons in the hidden layer:

$$y_j(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j\right] \tag{2.16}$$

where n is the number of inputs of neuron j in the hidden layer, and sigmoid is the sigmoid activation function.

(b) Calculates the actual output of the neurons in the output layer:

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}(p) - \theta_k\right] \tag{2.17}$$

where m is the number of inputs of neuron k in the output layer.

Step 3: Weight training

Updates the weight in the back-propagation network propagating backward errors associated with output neurons.

(a) Calculates the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \tag{2.18}$$

where

$$e_k(p) = y_{d,k}(p) - y_k(p) \tag{2.19}$$

Calculates the weight corrections:

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p) \tag{2.20}$$

Updates the weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \tag{2.21}$$

(b) Calculates the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) \times w_{jk}(p) \tag{2.22}$$

Calculates the weight corrections:

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p) \tag{2.23}$$

Updates the weight at the hidden neurons:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \tag{2.24}$$

Step 4: Iteration

Increases iteration p by one, goes back to step 2 and repeat the process until the selected error criterion is satisfied.

Face Classifier

In this study, we use MLP as a template-based face classifier. It can be trained using back-propagation on a training set consisting of similar number of faces and non-face images. Training the network for the face detection task is challenging because of the difficulty in characterizing typical non-face images which represent the rest of the world except face images. It is easy to get a representative sample of images which contain faces, but much harder to get a representative sample of those which do not. We avoid the problem of using a huge training set for non-faces by selectively adding images to the training set as training progresses. This “bootstrap” method reduces the size of the training set needed [Rowley 1998, Sung & Paggio 1994].

Face template

First of all, we use the MIT CBCL face database [Sung & Paggio 1994] of about 5000 frontal face and non-face samples (Figure 20) to construct a distribution-based generic face template with all its permissible pattern variations in a high dimension image windows vector space (e.g. 19 x 19 pixels in size). The facial region in the sample contains eyebrows, eyes, nose and month. Such a region provides distinct facial features and yet offers a relative invariance to hair style. The region is then reduced to a coarser (e.g. 19 x 19) feature vector. The advantages of adopting lower resolution facial features are reduced computational costs and increased tolerance [Lin & Kung 1997].

Figure 20: Some face and non-face samples. [Samaria & Harter 1994]

For a more robust representation, there is a number of image preprocessing methods which can be used to usefully reduce the variability of the image vectors. Normalizing the overall brightness and variance of the images partially removes variations largely due to the intensity of illumination and the imaging system response. A simple lighting correction model can help to reduce effects due to directional illumination. Histogram equalization can also be used [Gonzalez & Woods 2002]. Figure 21 shows the image preprocessing results. In order to produce more variations in the sample of face templates, we artificially generate some virtual examples of face patterns from rotating, mirroring, translating or scaling of the real face templates in a database. For non-face samples, we collect randomly to build the initial face/non-face decision boundary that can be refined later during the training in MLP.

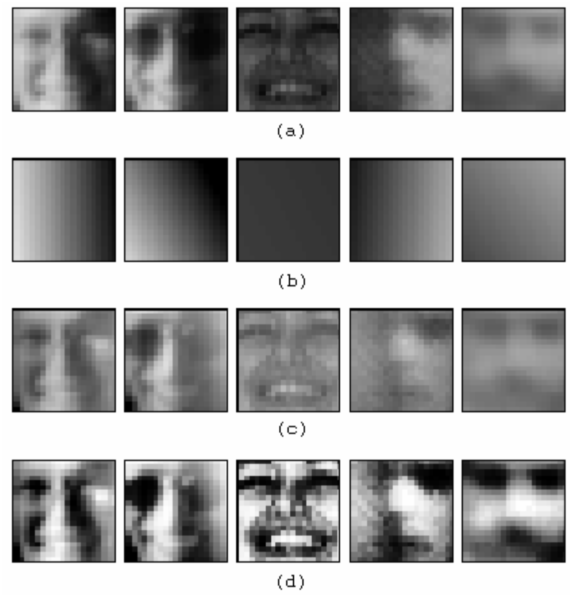


Figure 21: The steps in preprocessing of face templates. (a) Original face templates. (b) Best fit linear function. (c) Lighting corrected model (linear function subtracted). (d) Histogram equalized templates. [Gonzalez & Woods 2002].

Face Classifier Training

After obtaining the normalized face training samples, we can train a decision procedure on a sequence of “face” and “non-face” examples to empirically discover a set of operating parameters and thresholds that separates “face” patterns from “non-face” patterns in MLP face classifier.

The class of non-face images encountered while performing face detection, even only within attended probable face regions, is extremely broad. Accurate modeling of this distribution by performing density estimation is computationally infeasible given limited data. However, a decision surface can successfully discriminate faces from a non-faces distribution. In fact, only those near-face images that lie close to the face space and are therefore easily confused with face images need to be considered. Such confusable near-face images can be selected from a more extensive non-face data set using an iterative training method in which images incorrectly classified as faces are included in the training set for future training iterations as shown in Figure 22. This method is called "bootstrapping" [Rowley 1998, Sung & Paggio 1994].

Initially, the MLP is trained by back-propagation on equal numbers of face and non-face images, but the resulting classifier will perform quite poorly due to bad estimation among the training samples. In order to increase the classifier accuracy, further training is needed using more carefully selected non-face images that lie close to the desired decision boundary. The near-faces can be obtained by using the patterns erroneously detected as face in non-face images samples with multi-resolution scanning. This false detection is known as a false positive. These near-face samples are then used to retrain the MLP and repeated until a sufficiently low number of false positives are detected.

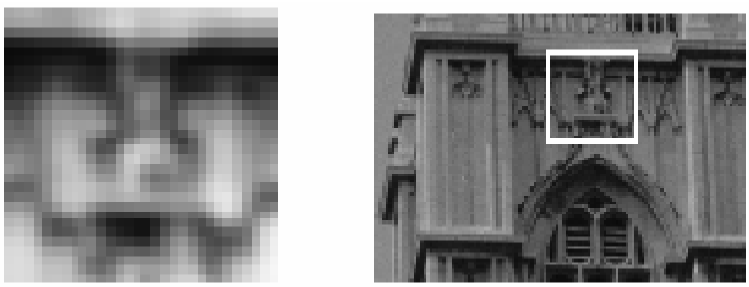


Figure 22: An example of a naturally occurring "non-face" pattern that resembles a face. Left: the pattern viewed in isolation. Right: The pattern viewed in the context of its environment. [Sung & Paggio 1994]

2.4.5.4. Radial Basis Function Network

As the shortcomings of the traditional multilayer perceptrons (MLP) neural network are the long training time of the back-propagation training algorithm and the arbitrariness in choosing the number of hidden layer neurons, RBFNN is more promising as it offers a faster and simpler two stages training with each stage relatively independent of the other [Haykin 1999]. An RBFNN classifier, as shown in Figure 23, is defined by ($\vec{x} \in R^n$):

$$s(\vec{x}_p) = w_0 + \sum_{i=1}^{M} w_i \Phi_i \left(\left\| \vec{x}_p - \vec{c}_i \right\| \right) \tag{2.25}$$

where x is the input vector, c_i is the centre of each RBF unit. One of the most common activation functions for RBF units is the Gaussian function, which is defined by:

$$\Phi(\vec{x}) = \exp\left\{ -\sum_{j=1}^{n} \frac{(x_{pj} - c_{ij})^2}{2\sigma_{ij}^2} \right\} \tag{2.26}$$

where σ is the variance across each dimension. Hence, the determination of the nonlinear map *s(x)* has been reduced to the problem of solving the following set of linear equations for the coefficients w_i:

$$\begin{bmatrix} s(\vec{x}_1) \\ s(\vec{x}_2) \\ \vdots \\ s(\vec{x}_N) \end{bmatrix} = \begin{bmatrix} 1 & a_{11} & \cdots & a_{1M} \\ 1 & a_{21} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{N1} & \cdots & a_{NM} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix} \tag{2.27}$$

where $a_{ij} = \Phi_j \left(\left\| \vec{x}_i - \vec{c}_j \right\| \right)$. We rewrite the above equation to matrix form $s=Aw$. By using the normal equation with the total-squared error between the actual output and the target output, we obtain the weight matrix by:

$$w = \left(A^T A\right)^{-1} A^T d \tag{2.28}$$

where d is the target output matrix. In face classification, we assign 1 to face samples and -1 to the non-face samples in the target output, and the input is classified as a face if its network output is greater than 0 during evaluation. In this study, the MIT CBCL face database [Sung & Paggio 1994] is used for the training set in the classifier. The first stage of the RBFNN training is to find the centre of each RBF unit using a clustering method, k-means is an example. The second stage is to find the set of weight matrices in the network. Thus the training algorithm is fast and simple as compared with the back propagation in MLP.

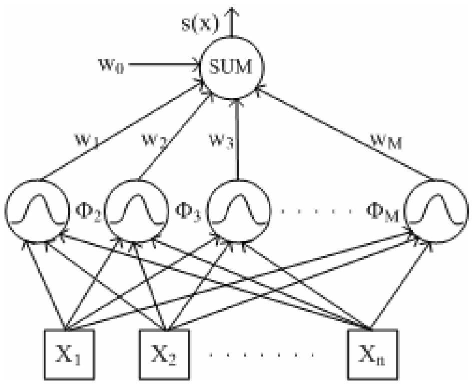


Figure 23: Radial Basis Function Neural Network (RBFNN)

2.5. Feature Extraction Process

After we have located the face candidate(s), we use the data from face detection such as the eye-mouth locations and face boundary for feature extraction, to normalize the face candidate for modeling and representation. As stated in section 2.2 before, face representation has two groups, template and feature.

Template-based approaches focus on the face boundary. They crop out the face portion to perform the principle component analysis. Principle component analysis lowers the dimension of the original data that represents the whole pattern. Eigenface [Belhumeur 1997] is a typical example. Although it is simple and fast, it suffers from the usual shortcomings of straightforward correlation based-approaches, such as sensitivity to face orientation, size, variable lighting conditions, and noise. The reason for this vulnerability of direct matching methods lies in the classification in high dimensionality of data. As a consequence, they lack robustness.

Feature-based approaches extract landmark features from the components of a face such as eye, eyebrow, mouth, nose and ear. The features include relative geometrical information and textural information. Feature-based approach utilizes the geometric information that is generated by face detection. Generic 3D face model [Blanz & Vetter 2003] and elastic face graph [Wiskott 1997] are common face geometry models. They match the facial features in their models with the features in the facial candidates from face detection. This is called “Masking”. Then feature extraction performs on each facial feature’s nodes in their models. The most common masking technique is the Active Contour Model (ACM) in which the snake is a typical example [Davison 2000]. Snake models trace the contour of the target object, so that facial components that are crucial for recognition are fitted to the individual’s facial geometry. Snake models are more robust than template-based approaches. They are more robust to different face poses as the face model is deformable to fit the input face image. But the defects of this approach are sensitive to scale and occlusion and computationally expensive.

The aim of feature extraction is to extract the most salient features from the target face such that it can be distinguished from others under different conditions such as pose, illumination, scale and facial expression. Gabor filter, Gabor filter bank, Gabor transform and Gabor wavelet are widely applied to image processing, computer vision and pattern recognition. This function can provide accurate time-frequency location, as well as robustness against varying brightness and contrast of images. Furthermore, the filters can model the receptive fields of a simple cell in the primary visual cortex [Kong 2003]. So in this study, we use a Gabor wavelet to extract facial features.

2.5.1. Gabor Feature Analysis

The Gabor wavelet [Kosko 1992], which captures the properties of spatial localization, orientation selectivity, spatial frequency selectivity and quadrature phase relationship seems to be a good approximation to the filter response profiles encountered experimentally in cortical neurons. Gabor wavelets have been found to be particularly suitable for image decomposition and representation when the goal is the derivation of local and discriminating features. Most recently, Donato et al. has experimentally shown that the Gabor filter is effective for classifying facial actions [Liu & Wechsler 2001].

2.5.2. Gabor Wavelets

Gabor wavelets are used for image analysis because of their biological relevance and computational properties. The Gabor wavelets, whose kernels are similar to the 2-D receptive field profiles of the mammalian cortical simple cells, exhibit strong characteristics of spatial locality and orientation selectivity and are optimally localized in the space and frequency domains.

The Gabor wavelets (kernels, filters) can be defined as follows [Lyons 1999]:

$$\Psi(k,x) = \frac{k^2}{\sigma^2}\exp\left(-\frac{k^2x^2}{2\sigma^2}\right)\left[\exp(ik.x) - \exp\left(\frac{\sigma^2}{2}\right)\right] \tag{2.29}$$

The multiplicative factor k^2 ensures that filters tuned to different spatial frequency bands have approximately equal energies. The term $\exp(-\sigma^2/2)$ is subtracted to render the filters insensitive to the overall level of illumination. The Gabor wavelet representation allows the description of spatial frequency structure in the image while preserving information about spatial relations. The complex amplitude of the transforms is used as features to test for the presence of spatial structure and restricted to a band of orientations and spatial frequencies within the Gaussian envelope. The amplitude information degrades gracefully with shifts in the image location at which it is sampled over the spatial scale of the envelope. Figure 24 shows the structure of Gabor wavelets.

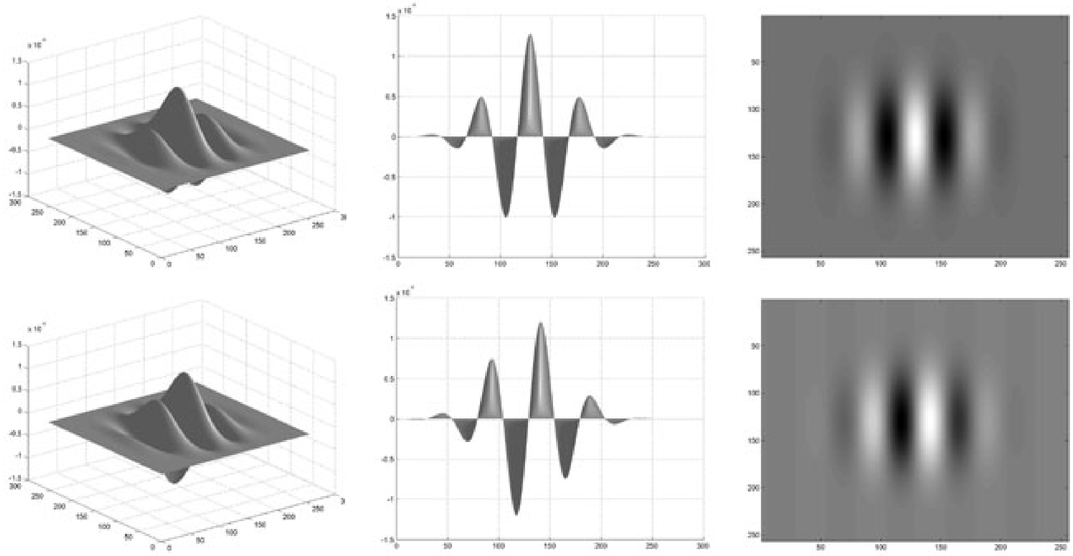


Figure 24: Gabor wavelets. Upper: Surface, profile and top-down view of real part. Lower: Imaginary (complex) part.

In most cases one would use Gabor wavelets at five different scales, e.g. $k_i=\pi/2^i$ with i from 1 to 5, and six orientations, e.g. from 0 to 150 degrees in 30 degree steps).

The kernels exhibit strong characteristics of spatial locality and orientation selectivity, making them suitable for image feature extraction when one's goal is to derive local and discriminating features.

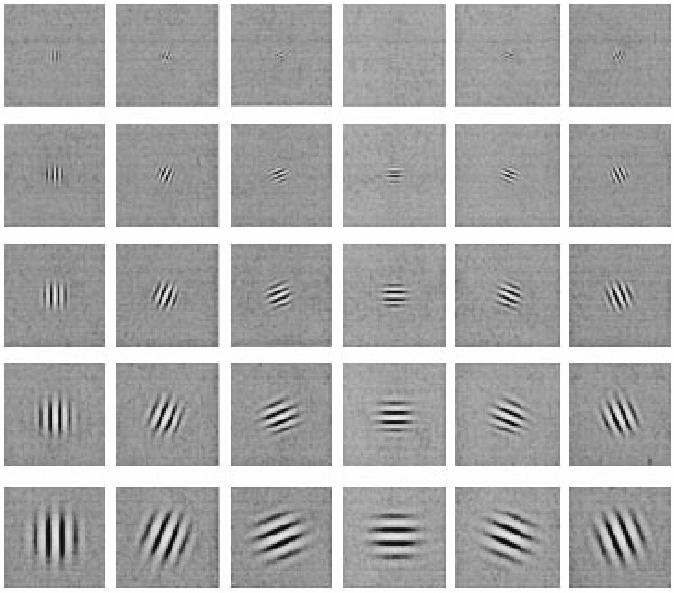


Figure 25: A set of Gabor wavelets with 5 scales and 6 orientations. [Liu & Wechsler 2001]

2.5.3. Gabor Feature Representation

The Gabor wavelet representation of an image is a convolution of the image with a family of Gabor kernels defined as follows:

$$O_{k,x} = I \otimes \psi(k, x) \tag{2.30}$$

The convolution outputs (both the real part and the magnitude) of a sample image exhibit strong characteristics of spatial locality, scale and orientation selectivity corresponding to those displayed by the Gabor wavelets. Such characteristics produce salient local features, such as eyes, nose and month that are suitable for visual event recognition. Gabor feature vector derived from Gabor wavelet representation of faces is obtained in set of convolution outputs with 5 scales and 6 orientations (total 30).

2.6. Face Identification Process

The last step, face identification, is the easiest step [Gong 2000]. It mainly concerns with the similarities between the features of an input face and those of other faces in a gallery. Distance or energy differences are the common metric for evaluating the similarity of two feature vectors. Verification concentrates on the extent of similarity between the input object and the claimed object. It will interpret as ‘they are the same’ if the difference is lower than a preset threshold. Recognition verifies all the objects in a gallery. It finds the objects with minimal differences from the objects that have passed the preset threshold. Euclidean distance is a common mathematical metric for determining vector difference. Although the differences between objects can be calculated easily, the computation will scale up as the number of sampled images in a gallery increases.

The aim of identification is to establish the identity of a query face from a set of labeled faces in database by their features. This measures the similarity of the features between the query face and the faces database usually by means of the Euclidean distance. This method is given by [Gong 2000]:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \tag{2.31}$$

where x and y are in Euclidean n-space.

Normalized data will produce a better similarity measure, so Pearson’s correlation coefficient can be used without prior normalization [Gong 2000]:

$$h(x,y)=\frac{\sum_{i=1}^{n}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i-\mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i-\mu_y)^2}} \tag{2.32}$$

where μ_x and μ_y are the mean values of x and y respectively.

After obtaining the similarity measure, it can be determined whether they are the same by a threshold. By using another threshold, the lower boundary that is totally different, we can define the degree of similarity between the samples also.

In summary, three major steps in face recognition are detection, feature extraction and identification. Face detection plays an important role during the process as its intermediate and final results will be re-used in later steps, so a good detection method can improve the recognition accuracy as well as efficiency. Invariant features of human faces are essential for performing recognition accurately across different poses and scales in the process of feature extraction. The efficiency of the system could be greatly improved with the aid of agent technology in the face identification process. In next section, we will discuss how to facilitate agent technology to improve the efficiency in face recognition system.

Chapter 3. Agent Technology

Many different distributed applications and mobile computing systems now operate on the internet. Agent technology, with its ability to automatically delegate tasks, and its autonomous and highly mobile characteristics in the network environment, is starting to play an important role in distributed and mobile computing. In this study, we adopt iJADE, an intelligent Java Agent Development Environment [iJADE] to implement an agent-based face recognition system. Contemporary agent-based development environments such as IBM Aglet [Aglet] focus only on the mobility of agents with simple multi-agent communication schemes, but iJADE overcomes this deficiency by providing a layer called the "Conscious (Intelligent) Layer", which supports the implementation of different AI functionalities for developing intelligent multi-agent applications. We can utilize the AI functionalities and distributed agent framework to implement a highly scalable and efficient face recognition system.

Agent technology provides a collaborative platform to build sophisticated application with interactive services among different modules. This makes it particularly suitable for deploying the proposed system. An asynchronous runtime environment in an agent server enables parallel processing to increase system performance, while distributed architecture in an agent system supports operation on various scales without damaging the system performance. By integrating the proposed face recognition models with the intelligent agent technologies, we deliver an accurate, efficient and robust agent-based face recognition system that is invariant to different conditions under uncontrolled environments.

This chapter covers the background of agent technology and discusses the benefits of using agent technology. Section 3.1 describes the basic properties and models of software agents. Section 3.2 describes the benefits of employing agent technology. Finally, iJADE model is introduced and also its practical applications are presented in Section 3.3.

3.1. Software Agent

A software agent is a software object that has one or more of the following properties [Lange & Oshimam 1999]:

- Autonomous: an agent should respond to stimulation automatically and should not require constant human guidance.
- Proactive: an agent should have the ability to sense the environment and respond to stimulation appropriately.
- Mobile: an agent should be able to migrate itself from one host to another.
- Collaborative: an agent should be able to make conversation with other agents to achieve sociability.

There are two major types of software agent in an agent model, .i.e. Mobile Agents and Stationary Agents [Lee 2003]. A mobile agent is a program that can move around on agent servers through a network. It consists of program code and its state when moves from one place to another. Then it will perform the new execution cycle on the new server according to its embedded program and state. Stationary agent refers to static agents which can stay in the execution environment in which they are born. They have access only to the resources of the local host. If a static agent requires other resources from other hosts, it must talk with the mobile agent which will then obtain those resources. The stationary agent acts as the middle layer or access control layer between the other agents and the local resources, e.g. database, disk space, files, or camera.

3.2. Benefits of Employing Agent Technology

Current Internet applications are primarily based on the client-server model. All transactions are carried out by request/response interactions over the Internet. Network resources are consumed during the interactions and user may experience long delays if the network is heavily loaded or unstable. The mobile agent model provides a new approach to overcome this scenario. We can send out the agent with the instructions and information to the target server, and the agent will have multiple interactions with the server locally or other agents located at the server. Only the agent with results will be returned to the user. This greatly reduces the requirements upon network bandwidth and response times. Thus the mobile agent platform provides a highly scalable and distributed development environment.

The major reasons for using mobile agents are as follows [Lange & Oshimam 1999]:

1. They reduce the network load – Most of the traditional client/server applications consist of a lot of message transfers. Using mobile agents will allow a piece of program code to be sent to the remote host to converse there.
2. They overcome network latency – In a hard real-time system, a delayed message is not accepted. For example, action must be taken immediately when a manufacturing processes fails. The control system can send an agent to the corresponding site to do and design the appropriate action locally.
3. They encapsulate protocols – An appropriate protocol should be made for the data exchange processes. It will be more difficult if the message content is

sensitive and needs to be encoded for security reason. Therefore, sending an agent to communication locally on the remote host can easily solve this problem.

4. They execute asynchronously and autonomously – Agent can be sent to the remote and stand by hosts so that it can work asynchronously with the creation host and automatically give the response to the stimulant from the remote host.
5. They adapt dynamically – Mobile agent can be sent to different hosts that are using different execution environments. The mobile agents still have the abilities to react autonomously.
6. They are naturally heterogeneous – Agent application programmers only need to take care about the execution environment because the agent platform already makes the mobile agent transport-layer-independent.
7. They are robust and fault-tolerant – Agent can have the abilities to react to the execution environment dynamically. Also agent can set a warming message to the control unit if it checks that there is something wrong in the remote host.

3.3. iJADE model

The aim of iJADE (intelligent Java Agent Development Environment) is to provide comprehensive 'intelligent' agent-based APIs and applications for future AI based applications [iJADE]. Figure 26 depicts the two levels of abstraction that are used in the iJADE system: (a) the iJADE system level – ACTS (Application, Conscious, Technology and Supporting layers) models and (b) the iJADE data level – DNA (Data, Neural and Application layers) models. The ACTS model consists of (1) Application Layer, (2) the Conscious (Intelligent) Layer, (3) the Technology Layer, and (4) the Supporting Layer. The DNA model is composed of (1) the Data Layer, (2) the Neural-Network Layer and (3) the Application Layer. iJADE DNA model provides a comprehensive data manipulation framework that is based on neural-network technology. The "Data Layer" corresponds to the raw data and input "stimuli" (such as the facial images captured from a Web camera and the product information in a cyber store) from the environment. The "Neural-Network Layer" provides the "clustering" of different types of neural networks for the purpose of organization, interpretation, analysis and forecasting operations that are based on the inputs from the Data Layer". The neural networks are used by the iJADE applications in the "Application Layer". Another innovative feature of the iJADE system is the ACTS model, which provides a comprehensive layered architecture for the implementation of intelligent agent systems.

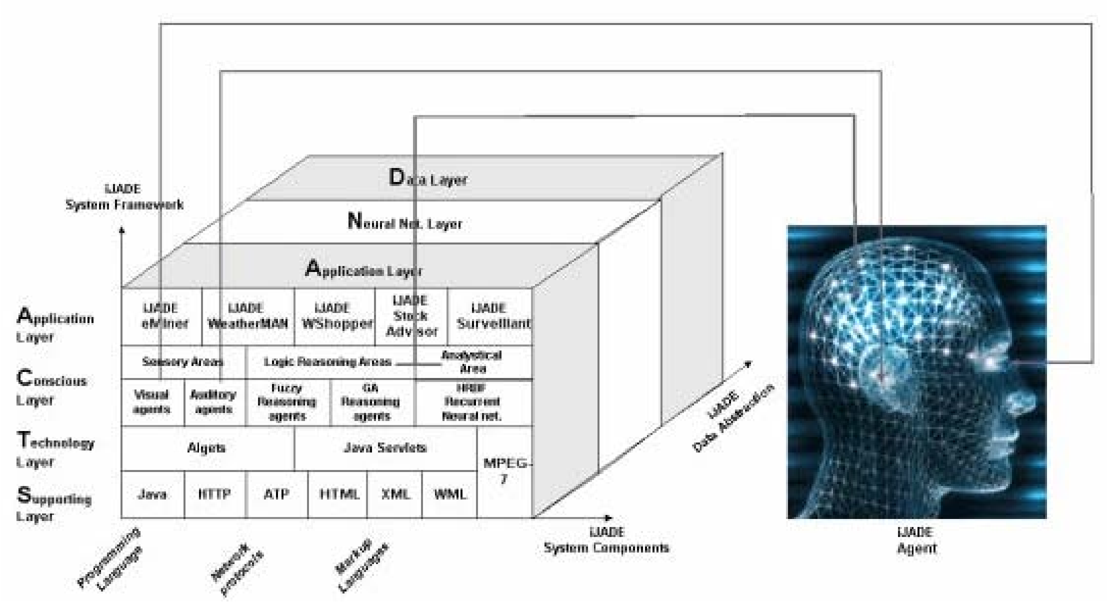


Figure 26: System architecture of the iJADE (v 2.0) model. [iJADE]

As iJADE provides a comprehensive intelligent agent-based platform for AI based application, various type of intelligent applications are developed on it. An intelligent multi-resolution composite neuro-oscillatory agent based surveillance system called iJADE Surveillant is proposed [Lee 2003]. This system integrates composite neuro-oscillatory wavelet-based scene segmentation module, active contour model with facial landmarks vectors feature extraction module, and elastic graph dynamic link based invariant human face identification module to provide an efficient and automatic scene analysis and figure-ground segmentation, and invariant human face extraction, identification and recognition system. Another practical application is iJADE Tourist Guide [Ao Ieong 2005]. iJADE Tourist Guide integrates GPS device and agent technology to provide location-awareness tourist information in mobile device for visitors. We overcome the limitation of computation power and network bandwidth in mobile device by using agent technology. This delivers a mobile computing solution for visitors who wish to conveniently browse for tourist information. In this research, we present a multi-agent based pose and scale invariant human face recognition system called iJADE Face Recognizer [Ao Ieong & Lee 2004]. iJADE provides an intelligent agent-based platform to support the implementation of various artificial intelligence (AI) functionalities, and we facilitate those AI features to implement an intelligent and automatic multi-agent based face recognition system. We can implement the three major processes in face recognition system into different module with intelligent agents easily. We deliver an accurate, efficient and robust agent-based face recognition system with invariant to different conditions on human faces under uncontrolled environment by integrating the proposed face recognition models with iJADE. The potential application of iJADE Face Recognizer is acting as an automatic human face surveillance system. It can efficiently and automatically analyze scene, segment out faces under complex environment and then carry out the invariant human face feature extraction and identification to identity all the faces from a color image.

Chapter 4. iJADE Face Recognizer

In this research, we present iJADE Face Recognizer [Ao Ieong & Lee 2004], a multi-agent based pose and scale invariant human face recognition system. We make use of those AI functionalities and mobility features of agent in iJADE to implement an intelligent and automatic multi agent based face recognition system. In iJADE Face Recognizer, we divide it into 3 major modules: 1) Face Detection Module, 2) Feature Extraction Module, and 3) Face Identification Module. In the following subsections, we will present the system architecture in section 4.1, and then describe the models and algorithms of face detection, feature extraction and face identification modules in sections 4.2, 4.3 and 4.4 respectively. In the last section 4.5, we will present the system flow of the proposed system.

4.1. iJADE Face Recognizer: An Overview

In this research, we introduce "iJADE Face Recognizer", a truly intelligent multi-agent based automatic human face recognition system. iJADE provides an intelligent agent-based platform to support the implementation of various artificial intelligence (AI) functionalities, and we make use of those AI features to implement an intelligent and automatic multi-agent based face recognition system. We can implement the three major processes in face recognition system into different module with intelligent agents easily. The system is based on the integration of the following 3 modules. 1) Automatic multi-scale face detector, it segments out all of the faces from the complex color input image with pose and size information. 2) Automatic facial feature selector, it selects the suitable feature extraction method based on the pose information of detected face and process the facial feature extraction. 3) Invariant face identifier, it identifies the query face with corresponding facial feature from face database. Figure

27 gives the system overview of multi-agent interaction in the iJADE Face Recognizer.

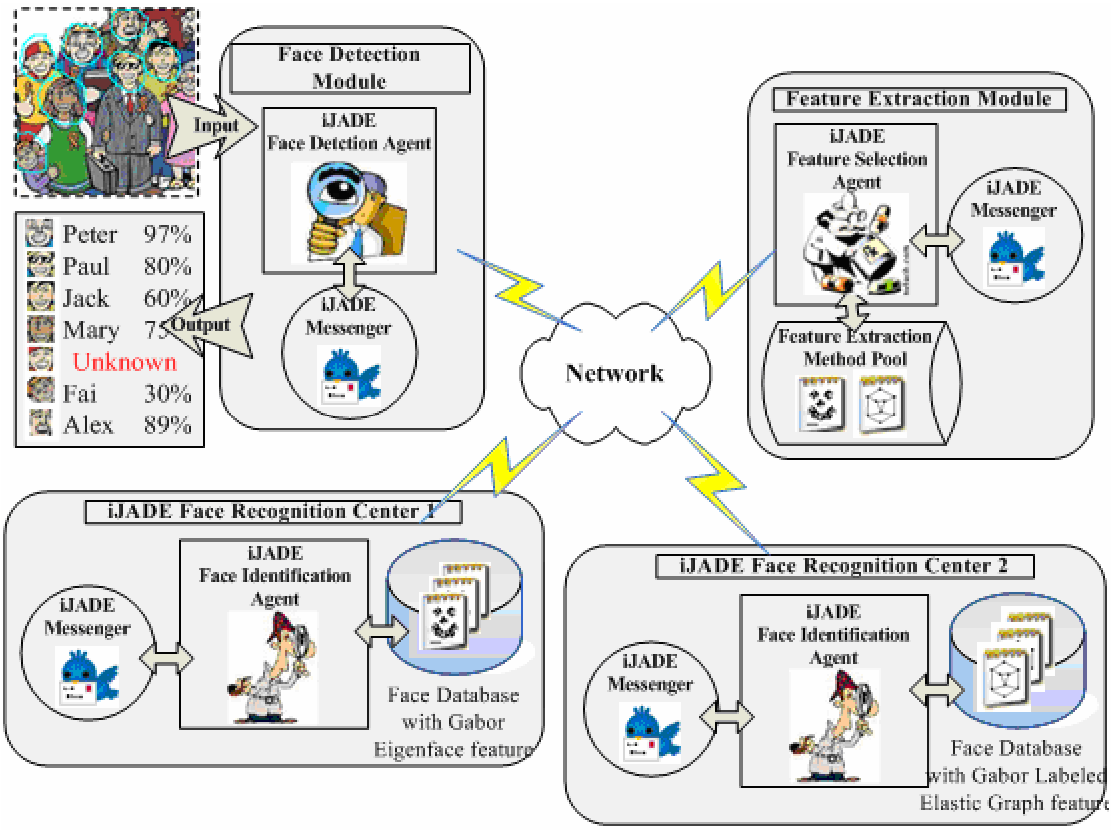


Figure 27: System overview of multi-agent interaction in the iJADE Face Recognizer

In the system of iJADE Face Recognizer, 4 types of iJADE agents are involved:

1. iJADE Messenger Agent – a mobile iJADE agent that acts as a messenger. It carries the result from one module to another one.
2. iJADE Face Detection Agent – a stationary iJADE agent that is situated at the client machine to act as a user interface to acquire input and to show output to user. Through the skin color model to approximate all possible face regions and the Mixture-of-Expert face classifier to detect in these regions in multi-scale progressively, all the face candidates with pose and size information are extracted and these results are sent to the next feature extraction modules by iJADE

Messenger Agent. Recognition output will show to user by this agent and this agent acts as the face registration interface as well.

3. iJADE Feature Selection Agent – a stationary iJADE agent situates at a iJADE server. The main duty of this agent is to select the suitable feature extraction method from the pose and size information of the extracted face candidate and then to perform the corresponding feature extraction process. It will then route the result to the server with suitable facial feature stored in the face database by iJADE Messenger. For example, if the pose of a face is frontal, it will select Gabor PCA method to extract the feature and then to pass these result to the iJADE Face Recognition Centre 1 to perform the identification as the centre 1 contains the face database with Gabor PCA feature.
4. iJADE Face Identification Agent – a stationary agent locates at the iJADE Face Recognition Centre Server. It performs the facial pattern matching of the facial features against the face database and face registration also. The result will send back to iJADE Face Detection Agent directly by iJADE Messenger.

The scalability and efficiency of the system can be increased by adding more iJADE Face Recognition Centre as the face identity in face database increases. Another duty of the iJADE Feature Selection Agent is acting as a dispatcher to balance the loading between each center, so the performance and efficiency can be improved.

4.2. Face Detection Module

The main purpose of the Face Detection Module is to locate all of the human faces in a color image with clustered scene. In this section, we describe the 2 major components in the Face Detection Module: Skin color filter and Face classifier. The skin color filter evaluates the possible skin regions in image and segments out these region-of-interests to next component to verify. The face classifier scans through these regions progressively and in multi-scale manner by sub-sampling, so that it can detect face in different scale. The overview of the face detection module is illustrated in Figure 28, and the back dots in the detection result in Figure 28 represent the centers of the windows classified as face.

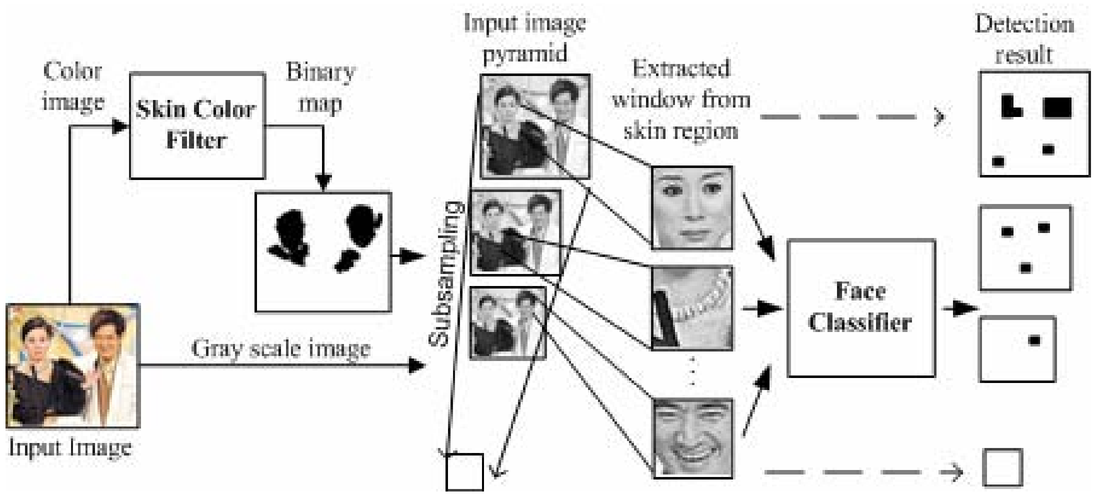


Figure 28: Overview of the Face Detection Module

4.2.1. Skin Color Filter

The aim of skin color filter is to segment out the possible skin regions from color image, and we can focus the face detection task within these skin regions in order to speed up the detection process and to reduce the false alarm rate. The skin color filter uses a Gaussian mixture model of skin color to evaluate the skin color likelihood of each pixel in color image. We obtained the Gaussian mixture model of skin color on HSV color space, since HSV separates the brightness component. It is invariant to the illumination effect and it would give better density function on skin color. Threshold is applied on the skin likelihood gray scale image to get the skin region binary image of the possible skin region. Morphological operation is applied on the binary image to reduce noise and obtain more complete skin regions, and the process of skin color filter is illustrated in Figure 29.

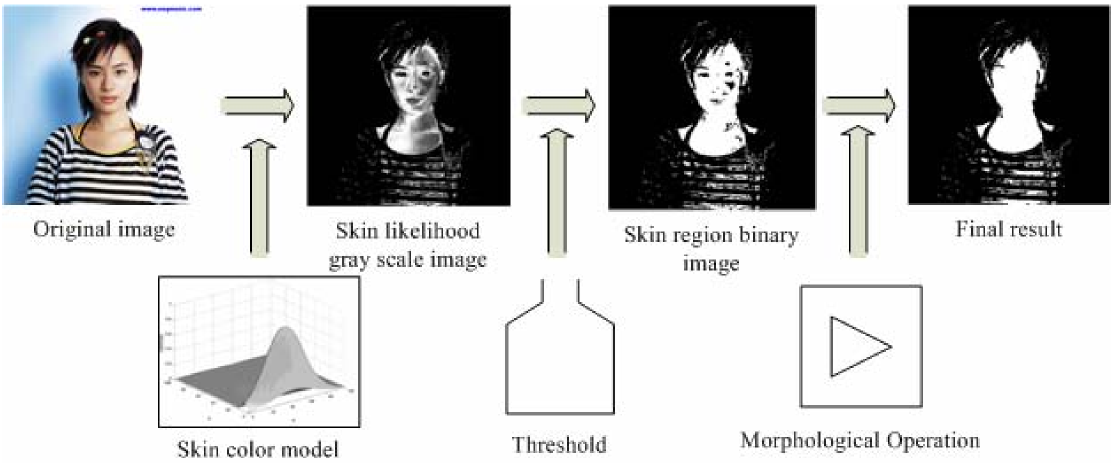


Figure 29: Skin color filter process

Figure 30 shows an example with threshold of 0.2, all of the skin regions are extracted by the skin color filter. We can focus the face detection task on these skin regions in order to speed up the detection process and to reduce the false detection rate as well.

Figure 30: Result of the skin color filter on a color image. Black pixels correspond to skin pixel

4.2.2. Committee of Face Classifier Network – Face Pose Classifier

Since we would handle face recognition across different poses, a committee of face classifier network is proposed in order to classify face in different poses [Feraud 2001]. The motivation of using a committee of face classifier is that one network is difficult to build up the decision boundary for different poses in face. A committee network can reduce the complexity and increase the efficient of each network, and it can simplify the training process by training each network individually [Lin & Kung 1997].

Each network detects its own subset of face in different pose. Only one network will give detected signal ideally if the examined window contains face as their subset of faces are disjointed. The committee network gives out not only the detected face candidates but also their corresponding pose according to the detected network. The information on detected pose is useful for the next steps of feature extraction and face identification since we can choose different recognition models for different poses to improve the accuracy in recognition.

Mixture of Expert Network

Mixture of Expert Network can be considered as the problem of learning a mapping in which the form of the mapping is different for different regions of the input space exists. Although a single homogeneous network could be applied to this problem, we might expect that the task would be made easier if we assigned different 'expert' networks to tackle each of the different regions. We use an extra 'gating' network, which also sees the input vector, to decide which of the experts should be used to determine the output [Bishop 1995]. This is known as Mixture-of-Experts (ME) network (Figure 31).

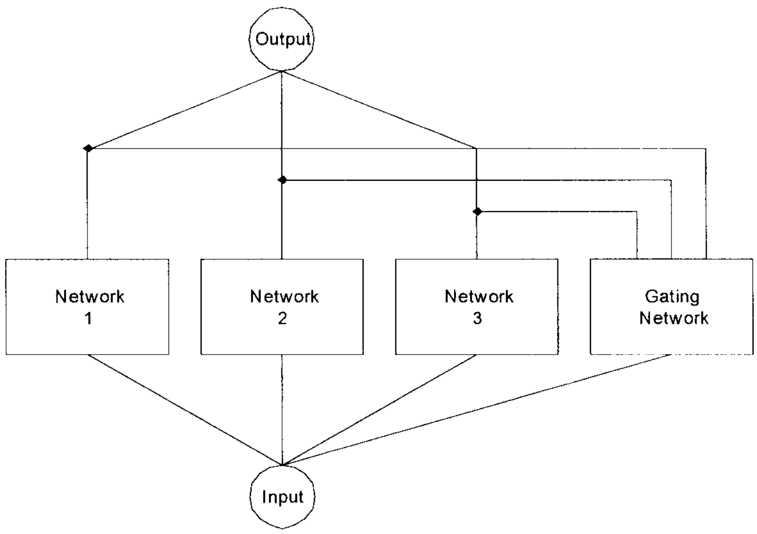


Figure 31: Architecture of a mixture-of-experts network

Training in ME network

Training in ME network and the gating network are carried out together. The goal of the training procedure is to have the gating network to learn an approximate decomposition of the input space into different regions, with one of the expert networks responsible for generating the outputs for input vectors falling within each region [Bishop 1995].

The key of the training procedure is in the definition of the error function, which is similar in the context of problem of modeling conditional distributions. The error function is given by the negative logarithm of the likelihood with respect to a probability distribution given by a mixture of M Gaussians of the form

$$E = -\sum_{n} \ln\left\{ \sum_{i=1}^{M} \alpha_i(x^n)\phi_i(t^n \mid x^n) \right\} \tag{4.1}$$

where $\phi_i(t^n \mid x^n)$ are Gaussian function given by

$$\phi_i(t \mid x) = \frac{1}{(2\pi)^{c/2}} \exp\left\{ -\frac{\left\| t - \mu_i(x) \right\|^2}{2} \right\} \tag{4.2}$$

These Gaussian functions have means $\mu_i(x)$ which are functions of the input vector x, and are taken to have unit covariance matrices. There is one expert representing the corresponding mean $\mu_i(x)$ where x is the input vector. The mixing coefficients $\alpha_i(x)$ are determined by the output γ_i of the gating network through a softmax activation function:

$$\alpha_i = \frac{\exp(\gamma_i)}{\sum_{j=1}^{M} \exp(\gamma_i)} \tag{4.3}$$

The ME network is trained by minimizing the error function simultaneously with respect to the weights in all of the expert networks and in the gating network. When the trained network is used to make predictions for new inputs, the input vector is presented to the gating network and the largest output is used to select one of the expert networks. The input vector is then presented to this expert network whose output $\mu_i(x)$ represents the prediction of the complete system for this input. It is also shown that the use of an error function based on a mixture of Gaussians leads to an automatic soft clustering of the target vectors into groups associated with the Gaussians components. In the context of the ME architecture, the ME network therefore leads to an automatic decomposition of the problem into distinct sub-tasks, each of which is effectively assigned to one of the network modules.

Mixture-of-experts face classifier

We adopted Mixture of expert network as the face classifier by using the conditional ensemble model of Feraud [Feraud 2001]. The ME network is trained on the face example as the conditional mixture and on the non-face example as the ensemble in which the target of the gate network is the mean output as shown in Figure 32.

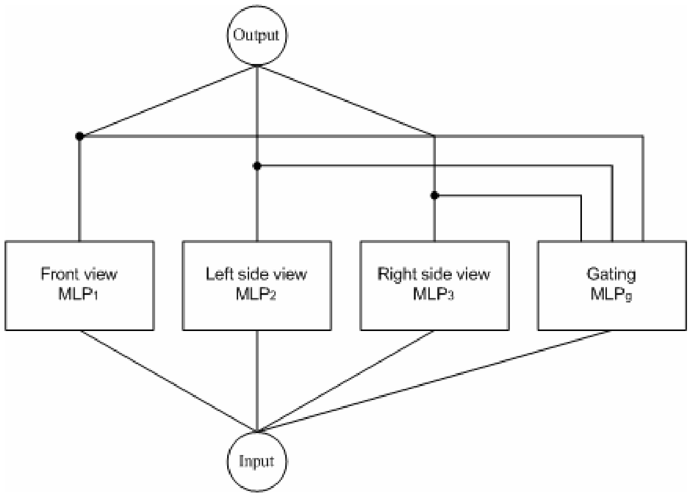


Figure 32: Architecture of Mixture-of-experts face classifier

For an example of ME network face classifier, 3 expert networks are used to classify different face poses, and so 5 sets are defined:

- f_f is the front view face set.
- f_l is the left side view face set.
- f_r is the right side view face set, with $f_f \cap f_l = \phi$, $f_f \cap f_r = \phi$, $f_r \cap f_l = \phi$
- $F = f_f \cup f_l \cup f_r$ is the face set.
- N is the nonface set, with $F \cap N = \phi$.

The goal is to evaluate $P(x \in F \mid x)$. Each expert network computes respectively:

- $P(x \in f_f \mid x \in f_f \cup N, x)$ *(MLP1(x))*
- $P(x \in f_l \mid x \in f_l \cup N, x)$ *(MLP2(x))*
- $P(x \in f_r \mid x \in f_r \cup N, x)$ *(MLP3(x))*

Each expert network detects its own subset of face, and only one network will give detected signal ideally if the examined window contains face as their subset of faces are disjointed. So the ME network gives out not only the detected face candidates but also their corresponding pose and size according to the detected expert network. This information is useful for the next steps of feature extraction and face identification, since we can choose different recognition models for different poses.

4.2.3. Description of Face Detection Module

The Face Detection Module is used for classifying a subwindow *x*, of size 19 x 19 pixels extracted from an image, as a face or as non-face. This module consists of 2 components (Figure 33):

- Skin Color Filter – It models by the Gaussian mixture skin model which evaluates the possible skin regions in image, and segment out the region-of-interests (ROIs) to next component to verify.
- Face Pose Classifier – It scans through the ROI progressively and in multi-scale manner by sub-sampling, so that it can detect face in different pose and scale.

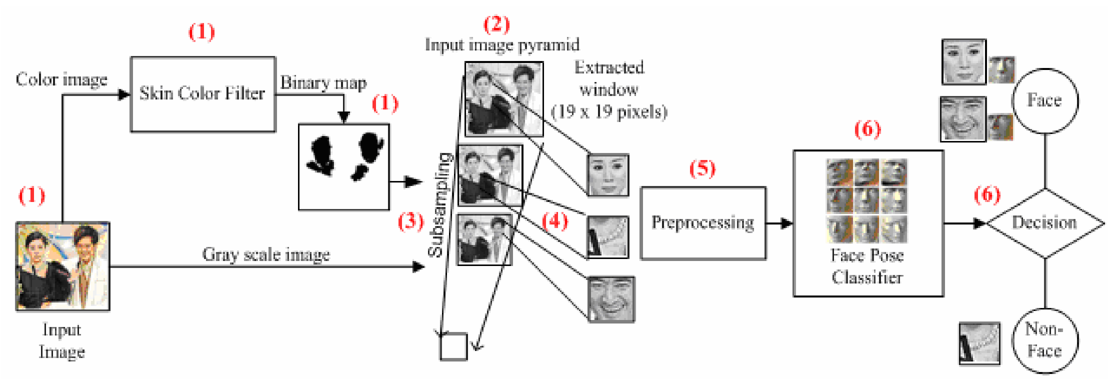


Figure 33: Overview of the face post classifier in face detection module

The steps in face detection module are as follows:

1) The color input image is processed by the Skin Color Filter. A gray scale image of skin probability in each pixel is generated by the Gaussian mixture skin model, and then the gray scale image is applied at morphological erosion and a threshold (e.g. 0.3). A binary image of possible skin region is obtained.
2) To detect faces anywhere in the input, sub-window (19 x 19 pixels) is extracted at every location in the image.
3) To detect faces larger than the window size, the input image is repeatedly reduced in size by sub-sampling with a certain factor (e.g. 1.2), and the window extraction is applied at each size.
4) The sub-windows contain a small number of skin pixels and are considered as background. The others, corresponding approximately to certain percent (e.g. 40%) of the total number of sub-windows, are evaluated by the following component: Face Pose Classifier.
5) The extracted window is pre-processed with lighting correction and histogram equalization.
6) The pre-processed window is then passed through the Face Pose Classifier that decides whether the window contains a face and the pose of that face.

4.2.4. Merging of Detections

In this section, we focus on how to merge the detection result from the previous section in order to eliminate the duplicated detections in different scales of the image and reduce the number of false positive detection results. Before we describe the merging algorithm, the variance property of the face classifier among the position and scale of face should be analyzed. In Figure 34, most human faces are detected at multiple nearby positions or scales. If a particular location is correctly identified as a face, then the other detected locations that overlap it among different scales are likely to be face also. Based on some testing results, we note that a correct face location usually overlaps among 3 scales of the image in sub-sampling factor of 0.8. Therefore we can derive the heuristic regarding overlap detections and preserve the location with 3 consecutive scales and eliminate the locations without this condition. Besides, for the multiple nearby detections in a correct face location with particular scale, we observe from some testing result that these multiple nearby detections usually occur within a 3x3 window around the face location. Thus, another heuristic regarding nearby detections, we combine the detection result within a 3x3 window.

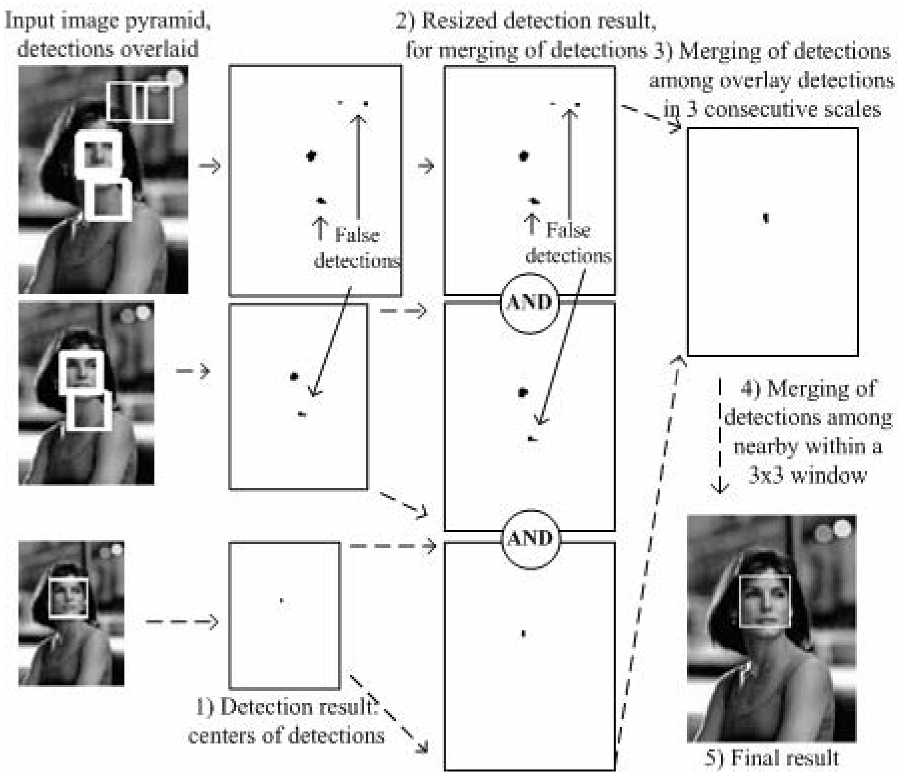


Figure 34: The framework for merging of detections

The implementation of the merging of detections can be based on these two heuristics. The first heuristic is overlap detections among 3 consecutive scales which are considered as correct detection, while false detections often occur with less consistency that could reduce many false detections. The second heuristic is nearby detections with a 3x3 window can be combined and this could eliminate the duplicated detections around the correct face region. The merging of detections is illustrated in Figure 34.

4.3. Feature Extraction Module

The aim of Feature Extraction Module is to extract the most salient features from the target face such that it can distinguish from others under different conditions such as pose, illumination, scale and facial expression. Gabor filter, Gabor filter bank, Gabor transform and Gabor wavelet are widely applied to image processing, computer vision and pattern recognition [Kong 2003]. This function can provide accurate time-frequency location and it also provides robustness against varying brightness and contrast of images. Furthermore, the filters can model the receptive fields of a simple cell in the primary visual cortex. So in this research, we has taken Gabor wavelet in feature extraction of face.

4.3.1. Facial Feature Representation

There are two kinds of facial feature representation approaches: Template and Feature based. Template-based uses holistic texture features and its advantage is that it can keep the geometrical information of the face pattern and also gives better recognition result. One of the most widely accepted algorithms in template-based approach is the Eigenface method. The Eigenface method is based on linearly projection of the image space to a low dimensional feature space. However, it uses Principle Components Analysis (PCA) for dimension reduction and yields projection directions that maximize the total scatter across all classes, i.e. across all images of all faces. In choosing the projection which maximizes total scatter, PCA retains unwanted variations due to lighting and facial expression. Thus, while PCA projections are optimal for reconstruction from a low dimensional basis, they may not be optimal from a discrimination standpoint [Belhumeur 1997]. By combining the advantage of Gabor feature which is robust to illumination and facial expression, PCA on Gabor facial feature can provide a good measurement on face recognition.

We can reconstruct the face pattern by adding up all the Gabor filter convolution output to a single image and by applying PCA on this image, low dimension Gabor Eigenface facial feature vector can be obtained. This feature can be compared by using suitable metric such as Euclidean distance to evaluate the similarity between samples. However, this method can handle the faces with nearly the same pose only. For the comparison between frontal and profile faces, severe error would produce. Therefore other method is required in order to overcome this constraint.

Feature-based approach seems to give a solution on the pose problem, as it is based on extract the feature on the facial landmarks without concerning the geometrical information of face [Wiskott 1997]. Elastic graph matching provides the model to locate all the fiducial points on facial landmarks in face image and then extract the Gabor coefficients of each node on the elastic graph. Feature vector can be generated by concatenating all the coefficients together [Lyons 1999].

As a result, we can choose Gabor Eigenface method for frontal-to-frontal face recognition. For frontal to profile and profile to profile face recognition, we use Gabor Labeled Elastic Graph method in order to overcome the pose difference.

4.3.2. Gabor Eigenface Feature

Gabor Eigenface feature is composed of Eigenface and Gabor features. The purpose of the Eigenface approach is to reduce the dimensions of face vectors. This is achieved by using the Principal Component Analysis [Haykin 1999] to find a set of new basis vectors (the eigenfaces). To construct the Eigenfaces, all the training vectors are put in a matrix X as the columns of X. The mean vector of the training face vectors is M. Then the covariance matrix E is transformed to a diagonal matrix A.

$$E = (X - M)(X - M)', \quad A = P'EP \tag{4.4, 4.5}$$

The columns of the matrix P are the eigen vectors of E and they are the Eigenfaces. Each face is projected onto the Eigenfaces and will then be represented by the coefficients. As the transformation *P'EP* diagonalizes the covariance matrix, the face space components of the face vectors are uncorrelated. As the number of eigenfaces is much smaller than the number of pixels of an image, the dimensions of face vectors are thus reduced. The dimensions can be reduced further by discarding those eigenfaces corresponding to smaller eigen values, as the coefficients computed at these eigenfaces have small variance across the whole training set and therefore are not important for discriminating different classes. However, experiments show that the Eigenface method is not robust in dealing with variation in lighting conditions. To overcome this problem, we resort to Gabor wavelets to break images into approximations and details of different levels of scales and work on several approximations images of each face.

In the step of face detection, the segmented face region is localized and then morphed with a fixed size for feature extraction in Gabor Eigenface method. This can increase the accuracy of the recognition rate as the localization and scale related issues are eliminated.

4.3.3. Gabor Labeled Elastic Graph Feature

In Gabor labeled elastic graph feature, each face where represented by a graph, each node labeled with a set of complex Gabor wavelet coefficients, is called a jet [Lades 1993]. Only the magnitudes of the coefficients are used for matching and recognition. When recognizing a face of a new image, each graph in the model gallery is matched to the image separately and the best match indicated the recognized person. Rotation in depth is compensated by elastic deformation of the graphs. Elastic graph matching aimed to locate the nodes in the graph to specific facial landmarks, is called fiducial points. The matching can use the phase of the complex Gabor wavelet coefficients to achieve a more accurate location of the nodes and to disambiguate the patterns which would be similar in their coefficient magnitudes [Wiskott 1997]. In this study, we exploit the pose information in face detection to improve the performance and efficient of this matching process. As we get the pose of a face, we can locate the approximated fiducial points from some predefined templates, and the template can act as the initial position of the nodes in matching process. This could enhance the accuracy in locating fiducial point and reduce computational effort significantly in the elastic graph matching process. The whole Gabor labeled elastic graph feature extraction process is illustrated in Figure 35.

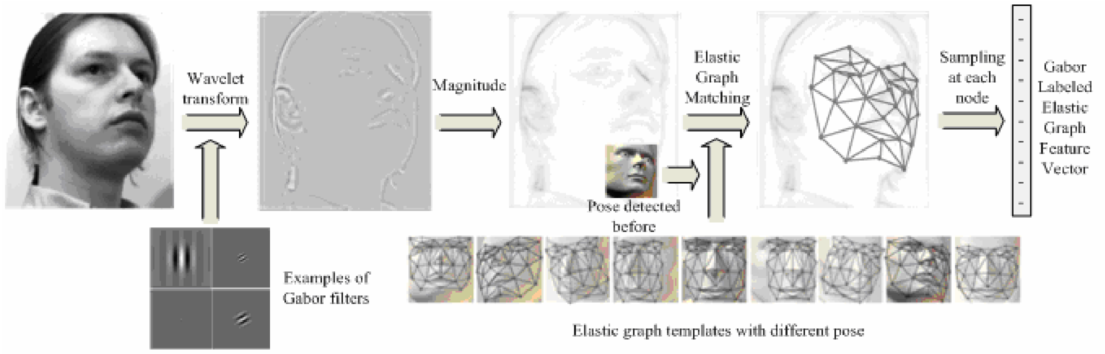


Figure 35: Gabor Labeled Elastic Graph feature extraction process

4.4. Face Identification Module

The aim of face identification module is to find out the identity of the query face from a set of labeled faces in database with their features. It measures the similarity of the features between the query face and the faces database usually by means of the Euclidean distance, and this method is given by [Gong 2000]:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \tag{4.6}$$

where *x* and *y* are in Euclidean *n*-space.

After obtained the similarity measures, we can determine whether they are the same by a threshold. By using another threshold, which is the lower boundary that they are totally different, we can define the degree of similarity between the samples within the threshold and lower boundary. The resulting decision of matched identity is the sample with minimum distance.

Agent technology provides asynchronous runtime environment and distributed architecture and it has the automatic delegation of task, autonomous and highly mobile characteristics. In iJADE Face Recognizer, it can perform the recognition task robustly and efficiently by using agent technology. We can utilize the multi agent technology to increase the efficiency and scalability of iJADE Face Recognizer, so the system can operate in various scales and models without deteriorating the system performance in the face identification process. We integrate the proposed face recognition models with the intelligent agent technology and thus we can deliver an accurate, efficient and robust agent-based face recognition system with invariant to different conditions on human faces under uncontrolled environment.

4.5. System Flow of iJADE Face Recognizer

In this section, we will describe the system flow and the collaboration of different agents in iJADE Face Recognizer. Figure 36 illustrates the overview of the system flow of iJADE Face Recognizer.

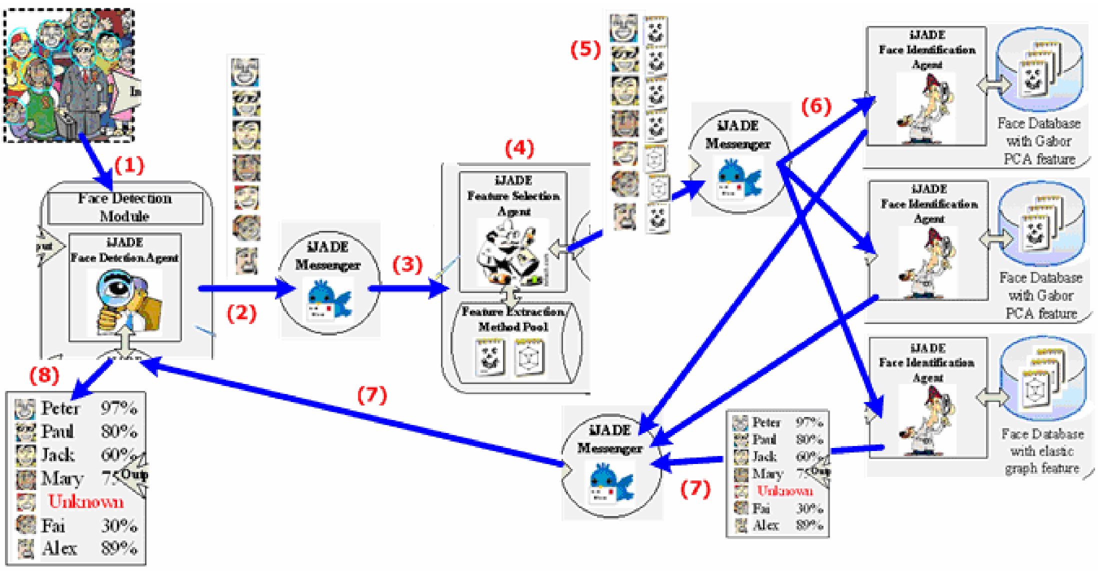


Figure 36: System flow of iJADE Face Recognizer

The abbreviations of the agents in iJADE Face Recognizer are as follows:

- FDA – Face Detection Agent

- MA – Messenger Agent
- FSA – Face Selection Agent
- FIA – Face Identification Agent

The system flow of iJADE Face Recognizer is as follows (Figure 36):

1) FDA receives the color image input from user, then goes through the face detection process to locate all the possible face patterns and find out its corresponding pose and size.
2) FDA will dispatch MA the detected face pattern with pose and size information to FSA.
3) FSA will analyze the face pattern with its information and carry out the suitable feature extraction method. The feature extraction method and its criteria can be obtained from the Feature Extraction Method Pool.
4) After the feature extraction process, FSA will clone different MAs and assign the face pattern with different extracted feature to each of them. The MA will then dispatch to different Face Recognition Center according to the feature of face pattern assigned.
5) All the MAs will transfer to corresponding Face Recognition Center in parallel manner in order to increase the efficiency and batch of the job will split into parts if replicated centres are available.
6) FIA will search the matched identity from its face database according to the feature of each face pattern and measure the degree of similarity.
7) MA is dispatched back to the FDA with recognition result.
8) FDA displays the recognition result to the user.

Chapter 5. Experimental Results and Discussion

In this section, 5 sets of experiments are carried out to evaluate the performance of each component in the proposed system. Since we could not find any system to compare with the proposed system directly, we analyze each component of the system separately. In section 5.1, we will analyze the performance of the skill color filters in normalized RBG and HSV color spaces with single and double Gaussian models. In section 5.2, we will present the improvement of using skin color filter in face detection module. In section 5.3, we will study the improvement of bootstrap method in training of MLP face classifier. In section 5.4, a comparison between MLP and RBFNN based face classifiers is given. Finally, a comparison of the face recognition rate of the proposed system with Eigenface method is presented in the section 5.5.

5.1. Skin Filter Performance Test

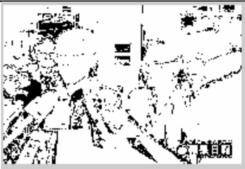
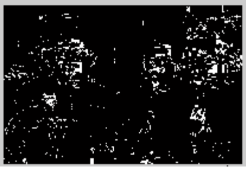
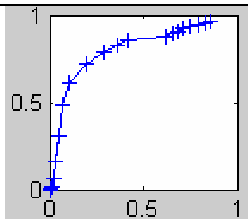
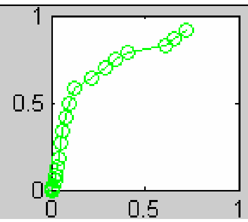
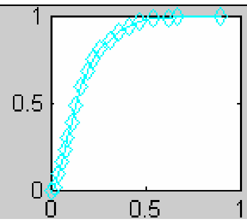
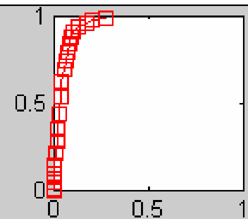
In this section, we carry out the skin filter performance test by analyzing the performance of skin color filter in single Gaussian and Gaussian mixture models, and in normalized RGB and HSV color spaces. Firstly, we obtain the skin color filters from training samples, which are about 7 million skin color pixels collected from 250 Internet's and movies' color images. Four skin color filters are evaluated as follows:

- Single Gaussian of rg components in normalized RGB color space (Model 1)
- Double Gaussian in Gaussian mixture model of rg components in normalized RGB color space (Model 2)
- Single Gaussian of HS components in HSV color space (Model 3)
- Double Gaussian in Gaussian mixture model of HS components in HSV color space (Model 4)

Secondly, we collect another testing sample set from the Internet and some movies, which contains 50 color images, to evaluate the performance of each filter. We first segmented the skin region from the original image by human as shown in Figure 37. The segmented result is acting as the ground truth to evaluate the performance of the skin color filters. Thirdly, we calculate the true positive and false positive rates in the skin color filters to generate the ROC curve in order to compare the performance across each filter. Two examples are described in details to explain the differences among the filters first and then conclusion is made from the ROC curve among different filters.

Figure 37: The original image (left) and the segmented skin region result (right) of an example in the experiment of skin color filter performance

Table 1: Output of the first example in the performance test of the skin color filters

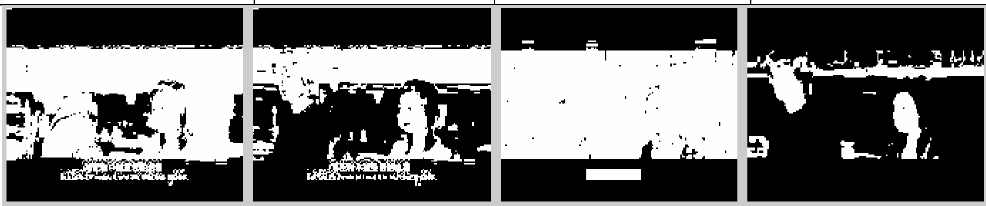
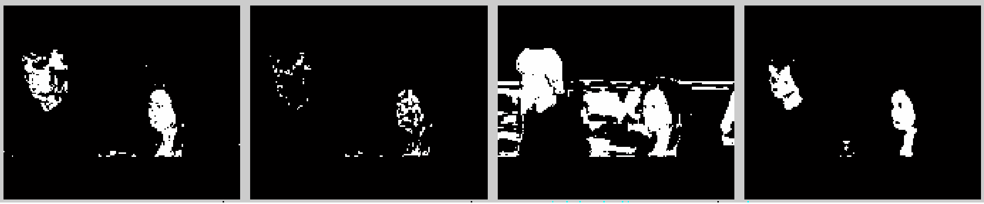
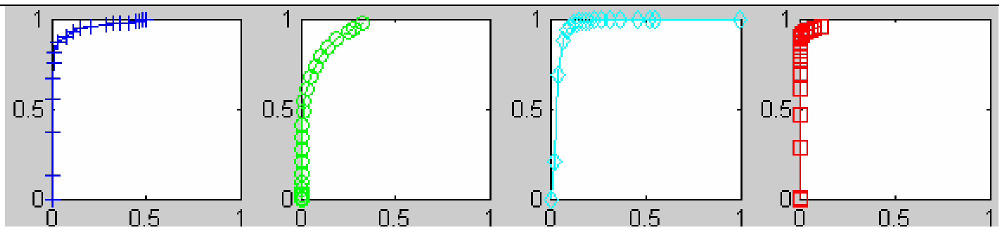
Th.	Single Gaussian in normalized rg color space (Model 1)	Double Gaussian in normalized rg color space (Model 2)	Single Gaussian in HS color space (Model 3)	Double Gaussian in HS color space (Model 4)
0.1				
0.3				
0.5				
0.7				
ROC				

The left hand side image in Figure 37 is collected from a newspaper website. This image contains 5 faces and many hands. The segmented skin region result of this sample image is presented at the right hand side, and this region result treats as the ground truth to evaluate the performance of each skin color filter. In Table 1, “Th.” refers to the threshold value, ROC refers to ROC curve, white pixel represents skin pixel in the image, y axis represents the true positive rate, and x axis represents the false positive rate. This table presents the results of each filter under the thresholds 0.1, 0.3, 0.5 and 0.7, and their corresponding ROC curves present on the last row of the table. We observe from the segmentation results that model 4 contains the smallest amount of false positive skin region in the thresholds 0.1 and 0.3 among all the models. The ROC curves us that model 4 has the highest performance as model 4 gives high true positive rate with low false positive rate than other models.

We did the same skin filter performance test on the left hand side image collected from a movie in Figure 38. This image contains 2 faces with complex background. Table 2 presents the results and the ROC curves of the skin color filters on this image. Model 4 gives slightly better performance than the others models and this model gives consistent results among different thresholds. However, other models present dynamic changes in true positive and false skin regions among different thresholds and this scenario is presented in the previous example also. Therefore, it is much easier to decide the threshold of model 4 for all situations and 0.3 is an example for the best threshold.

Figure 38: The original image (left) and the segmented skin region result (right) of the second example in the experiment of skin color filter performance

Table 2: Output of the second example in the performance test of the skin color filters

Th.	**Single Gaussian in normalized rg color space (Model 1)**	**Double Gaussian in normalized rg color space (Model 2)**	**Single Gaussian in HS color space (Model 3)**	**Double Gaussian in HS color space (Model 4)**
0.1				
0.3				
0.5				
0.7				
ROC				

In Figure 39, the overall performance of the four skin color filter models are presented by their ROC curves. This graph is generated from the test set with 50 color images from Internet and movies. It is clear to note that double Gaussian in HS color space skin color filter (Model 4) outperforms the others and has high true positive rate corresponded with low false positive rate. At the true positive rate of 0.9, model 4 gives 0.05 on false positive rate of skin color only but model 1, 2 and 3 give 0.6, 0.6, and 0.3 respectively. Moreover, as described in the last two skin filter performance tests before, model 4 is consistent under all situations, which means we can choose the threshold easily. From these experimental results, we can conclude that the skin color filter with Gaussian mixture model on HSV color space outperforms the models in single and double Gaussian on normalized RGB and single Gaussian in HSV color space. It is because HSV color model separates the brightness component and so it is invariant to the illumination effect and gives better density function on skin color and Gaussian mixture model provides better estimation of the skin color density function on that color space. Besides, it is suggested that double Gaussian model is good enough for estimation of skin color distribution [Caetano 2003] and the performance of higher order Gaussian models is very similar which is illustrated from the experimental results on the performance of multiple Gaussian mixture skin color filter in Figure 40. From the point of view on efficiency, double Gaussian model is suitable to adopt in skin color filter in face detection.

The limitation for the skin color filter is that if there is other color in the image which is very close to skin color such as background color and clothing color of the subjects, these regions will also be treated as the skin regions. The shortcoming of this limitation is causing wastage of computation on these non-face regions and this may increase the false face detection rate as the detection region is increased.

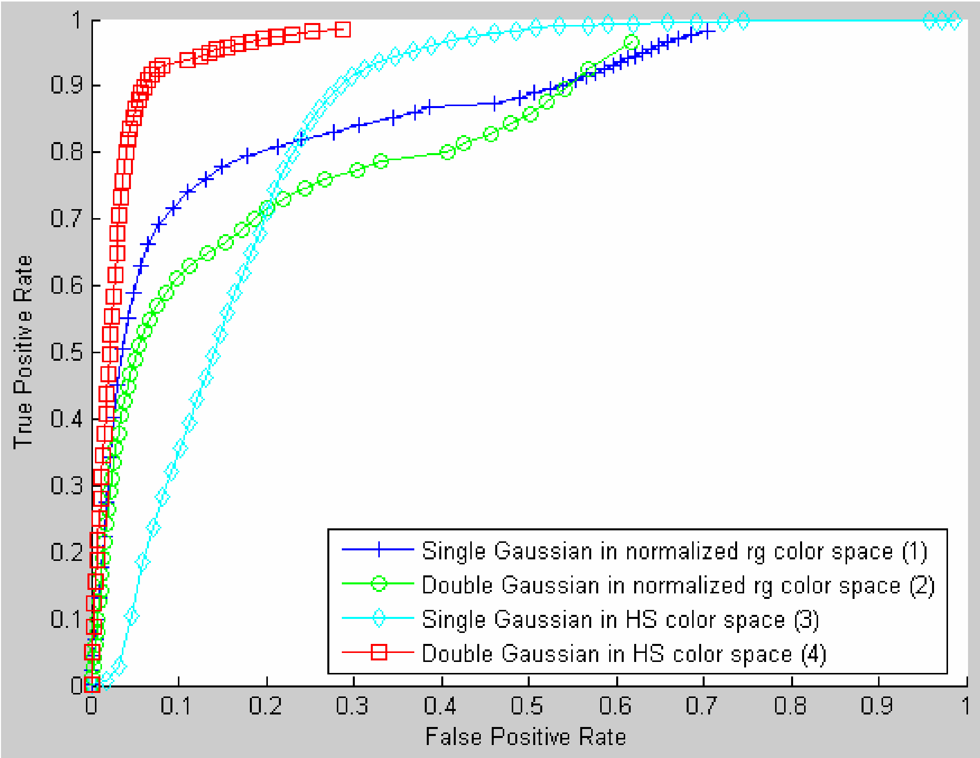


Figure 39: The ROC curves of the four skin color filter models

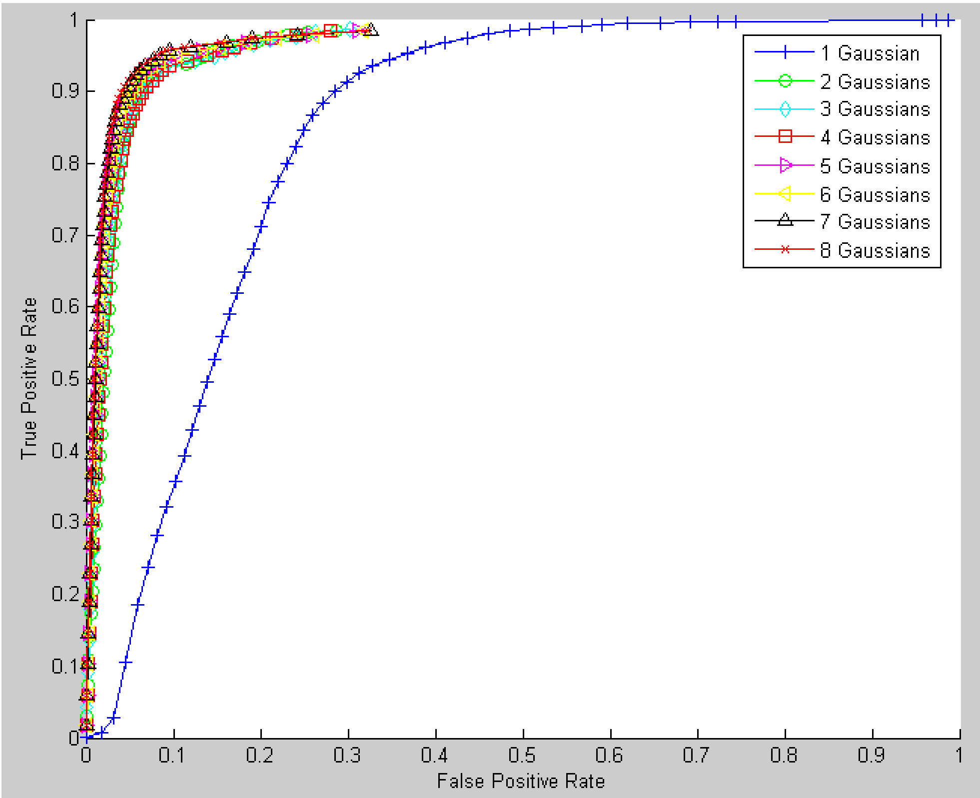


Figure 40: Performance of multiple Gaussian mixture skin color filters in HS components of HSV color space

5.2. Test of Skin Color Filter in Face Detection Module

We tested on the processing time and accuracy in detection after applying the skin color filter and merging of detections. This experiment is carried out in Matlab environment and running on a Pentium III 500MHz PC with 256MB memory. We first analyze the processing time and accuracy on the proposed model. We tested the algorithm on 40 still color images. The images are chosen from Video CD or Internet, and consist of both indoor scenes and outdoor scenes with clustered background. There are 93 faces in the test image set. The image size varies from 320x240 pixels to 450x278. We first trained up the system with the MIT CBCL face database (2429 face and 4548 non-face gray scale image samples with size 19x19 pixels) and test with this image set and the results with different methods and samples are presented in Table 3.

Table 3: The performance results on the RBFNN face classifier with different methods

Method	# of windows processed	Average processing time per image (in second)	Detection rate (# of detected faces)	False alarm rate (# of false alarm)
A	2975578	551	83.9% (78)	8.05e-4 (2395)
B	752121	143	79.6% (74)	3.66e-4 (275)
C	752121	143.9	78.5% (73)	1.25e-4 (94)
D	24560	189	40% (2)	0 (0)

Method A: Exhaustive search with the RBFNN face classifier only.

Method B: Search in skin region by skin color filter with the RBFNN face classifier.

Method C: The proposed model - Search in skin region by skin color filter with the RBFNN face classifier and combine the detection result with the heuristic merging method.

Method D: One typical example (Figure 30) in the test set that is tested with the proposed method.

Table 3 shows the performance of the RBFNN face classifier with different methods. In Table 3, the number of windows processed and the processing time in method B and C is greatly reduced to about one-fourth (# of windows processed in method C/# of windows processed in method A) of method A. The reason is that we use the skin color filter to narrow down the search region on the possible skin regions only. Moreover, the false alarm rate in method C is greatly reduced to 15.5% (false alarm rate of method A/false alarm rate of method Cx100%) of the rate in method A. It is because the heuristic merging method on 3 consecutive scales and 3x3 nearby window can reduce most of the uncertain false alarm and eliminate the duplicated detections. However, the detection rate has dropped in method B and C and this is because some of the faces with unpredicted skin color are omitted by the skin color filter. In method D, a test example for demonstration, the detection rate is low as there are many small faces and some occluded faces that could not be detected by the proposed system. No false alarm exists in this example even though the background is complex and this is mainly due to the benefit of the heuristic merging method. Another benefit by the skin filter in this example is that the processing time is 10 times faster than the method using exhaustive search.

5.3. Test of Bootstrap in MLP Face Classifier

We tested the performance of MLP Face Classifier before and after Bootstrap training in face detection process. Firstly, we use the MIT CBCL face database [Sung & Paggio 1994] of about 5000 frontal face and non-face samples to construct a distribution-based generic face template with all its permissible pattern variations in a high dimension image windows vector space (e.g. 19 x 19 pixels in size). The facial region in the sample contains eyebrows, eyes, nose and month. Such a region provides distinct facial features and yet offers a relative invariance to hair style. The region is then reduced to coarser (e.g. 19 x 19) feature vector. Advantages for adopting lower resolution facial features are reducing computational cost and increasing tolerance. In order to produce more variations in the sample of face templates, we can artificially generate some virtual examples of face patterns from rotating, mirroring, translating or scaling of the real face templates in database. For non-face samples, we can collect randomly to build the initial face/non-face decision boundary that can be refined later during the training in MLP.

After we obtain the normalized face training samples, we can train the MLP face classifier on a sequence of “face” and “non-face” examples to empirically discover a set of operating parameters and thresholds that separates “face” patterns from “non-faces” patterns. The class of non-face images encounters while performing face detection, even only within attended probable face regions, is extremely broad. Accurate modeling of this distribution by performing density estimation is computationally infeasible given limited data. However, a decision surface can successfully discriminate faces from non-faces distribution. In fact, only those near-face images that lie close to the face space and are therefore easily confused with face images need to be considered. Such confusable near-face images can be selected from a more extensive non-face data set using an iterative training method in which images incorrectly classified as faces are included in the training set for future training iterations. This method is called “bootstrapping” [Rowley 1998, Sung & Paggio 1994]. The evaluation of the bootstrap method is based on the MIT CBCL test set and the improvement is shown in Figure 41. We can conclude from the plot that bootstrap method can reduce false positive detection rate but is not applicable for improving true positive rate. The true positive detection rate can only be improved by training with more face samples.

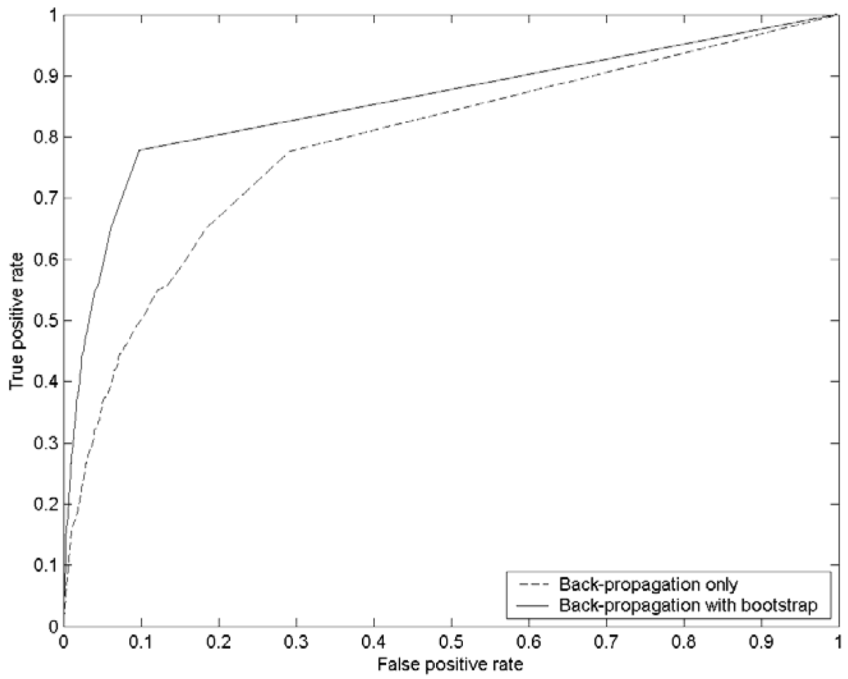


Figure 41: ROC curves for MLP classification rate by back-propagation with and without bootstrap

5.4. Comparison of MLP and RBFNN Based Face Classifiers

We presented the comparison between conventional MLP and RBFNN based face classifiers on training time, detection rate, false alarm rate and processing time in this session. All of the experiments are carried out in Matlab environment and running on a Pentium III 500MHz PC with 256MB memory. Comparison between RBFNN and conventional MLP face classifier is done on the CMU test set [CMU]. The CMU test set consists of 50 gray-scale images containing 149 faces and most of them are frontal faces. The numbers of hidden neurons used in the RBFNN face classifier and the MLP face classifier are both 30 and both of them are trained with the MIT CBCL face database with 2429 face and 4548 non-face samples.

Table 4: Results on the CMU test set

Method	Training time (in second)	Average processing time per image (in second)	Detection rate (# of detected faces)	False alarm rate (# of false alarm)
MLP	1208	574.9	63.1% (94)	1.14e-3 (1292)
RBFNN	221	836.3	71.1% (106)	1.58e-4 (179)

Table 4 presents the performance on MLP and RBFNN face classifiers in terms of training time, average processing time, detection rate and false alarm rate. Table 4 shows that RBFNN face classifier is well performed in training time, detection rate and false alarm rate. RBFNN has a simple and fast 2 stages training, so its training time is much faster than the MLP back propagation. In terms of detection rate and false alarm rate, the RBFNN face classifier provides a more generalization property under the same condition of training samples and the same number of hidden neurons, higher detection rate (8% more than MLP) and lower false alarm rate (13.9% of the false alarm rate of MLP) than MLP face classifier as a result.

5.5. Face Recognition Rate

In this section, we tested the face recognition rate of Eigenface method with the proposed Gabor Eigenface and Gabor Labled Elastic Graph method. The face database used in evaluation is the ORL face database [Samaria & Harter 1994] and we simply use Euclidian measurement as the classifier. The database contains 400 face images acquitted of 40 individuals (10 images per individual) over many years with various lighting conditions, facial expressions and poses. On the experiments of comparison between Eigenface and Gabor Eigenface, they are conducted by using the different training faces per person. During these experiments the dimensions of the face vectors are increased gradually in an attempt to find the dimension that gives best classification performance.

Table 5 presents the best recognition accuracy of Eigenface, Gabor Eigenface, Gabor Labeled Elastic Graph and iJADE Face Recognizer. The performance of Gabor Eigenface is better than Eigenface. It should be pointed out that PCA used all components (at most M-1, where M is the total number of training samples) for achieving the maximal recognition accuracy when there are 5 to 6 samples per person for training. Gabor Eigenface performs 100% accurately when dimension is 70 upwards in 6 training samples. A fixed dimension of 31 is sampled from the magnitude of the Gabor wavelet coefficient in Gabor Labeled Elastic Graph method, since there are 31 nodes in the elastic graph template. Dynamic method is adopted according to the pose of the test face in iJADE Face Recognizer.

Table 5: Comparison of the Top Recognition Accuracy (%) on various models

# of training samples / class	**Eigenface**	**Gabor Eigenface**	**Gabor Labeled Elastic Graph**	**iJADE Face Recognizer**
3	87.86% (40)	92.86% (90)	95%	96.42%
4	94.17% (30)	97.08% (60)	95.83%	97.92%
5	97.5% (160)	98% (110)	97%	99.5%
6	99.38% (150)	100% (70)	100%	100%

The values in parentheses denote the dimension of feature vectors for the best recognition accuracy in Eigenface models.

We figure out from that the proposed methods are better than the ordinary

Eigenface method. It is because Gabor wavelet representation which adopted in the proposed methods provides robustness against varying brightness and facial expression, while PCA retains unwanted variations due to lighting and facial expression. Moreover, iJADE Face Recognizer method combines the benefit of template-based Gabor Eigenface and feature-based Gabor Labeled Elastic Graph methods and compensates the weakness of these two methods so gives better recognition accuracy as a result.

Chapter 6. Conclusion and Future Works

This chapter concludes the thesis and is organized in four sections. Section 6.1 summarizes the research work. Section 6.2 summarizes the major contributions of the thesis. Section 6.3 points out the limitations of the proposed system. Finally, section 6.4 suggests the possible further works on the proposed system.

6.1. Summary of Research

In this research, we presented iJADE Face Recognizer, a multi-agent based pose and scale invariant human face recognition system with invariant to different conditions on human faces under uncontrolled environment. The target is to identify all of the faces from a color image with complex scene. A brief analysis on the difficulties in face recognition is presented, which includes intrinsic and extrinsic factors. The whole face recognition process involves three major processes: 1) face detection, 2) feature extraction and 3) face identification.

In face detection process, we proposed a skin color filter to speed up the detection process and a committee network to detect faces with different poses. In order to narrow down the searching region in image, skin color filter is developed which is based on Gaussian model in HSV color space. It produces the possible face regions to analyze in next module, a face classifier. Face samples are preprocessed by lighting correction and histogram equalization and these preprocessing can compensate for certain sources of image variation and make classification easier. Training in the classifier is difficult as non-face samples are hard to collect which represent the rest of the world except face. Therefore, Bootstrap model is introduced to find near face class during training in order to refine the face decision boundary in the classifier. High missing and false positive rate are observed in MLP face classifier, so RBFNN based face classifier is adopted to improve the efficiency and the pose information can be obtained by using a committee network of face classifiers. The major advantage of neural network model is that the efficiency of the system can be enhanced by learning from examples and the more samples processed, the more accurate result is obtained.

In feature extraction process, Gabor feature vector derived from Gabor wavelet representation of faces is adopted. Based on the pose of the face, different feature extraction method can be applied in order to produce an invariant facial feature for recognition. Template-based Gabor Eigenface feature extraction performs on frontal face while feature-based Gabor Labeled Elastic Graph feature extraction for other profile faces are adopted which eliminate the pose influence in recognition.

In face identification process, we adopt multi agent technology and the scalability and efficiency of the system can be greatly increased. Fault tolerance and fast parallel searching could be introduced if replicated face databases are available.

6.2. Summary of Contributions

We presented iJADE Face Recognizer in this research and many models are proposed in order to tackle the difficulties and speed up the process in each of the major processes on face recognition. The major contributions of this research are summarized as follows:

- Skin color filter: In face detection, color information from color image provides the visual cues for focusing attention on people faces. We overcome the difficulty of illumination effect on skin color by applying the nonlinear transformation to the HSV color space. HSV separates the brightness component and so it is invariant to the illumination effect and gives better density function on skin color. We adopted Guassian mixture model to estimate the skin color distribution in HS color space and we obtain the skin color likelihood of each pixel in color image from this Gaussian mixture skin color filter. The skin color filter locates all of the possible skin regions for detection and this greatly reduces the computation time and false alarm rate in the face detection process.

- Face pose classifier: A committee of face classifier network is proposed in order to classify face with different poses. The motivation of using committee of face classifier is that one network is difficult to build up the decision boundary for different poses. So multiple or a committee of network is used and it can reduce the complexity, increase the efficiency of each network and simplify the training process by training each network individually. The information on detected pose is very useful for the next steps of feature extraction and face identification, since we can choose different recognition models for different poses in order to improve the accuracy in recognition. The heuristic merging method in detection can greatly reduce the false alarm rate and eliminate the duplicated detections.
- Gabor facial feature: Gabor wavelet filter, which captures the properties of spatial localization, orientation selectivity, spatial frequency selectivity, and quadrature phase relationship and provides accurate time-frequency location, and robustness against varying brightness, contrast of image and facial expression on human face. Gabor Eigenface and Gabor Labeled Elastic Graph features are proposed to compensate the deficiency of template-based and feature-based facial features and overcome the difficulty of recognition among different poses. We exploit the pose information in face detection to select the suitable feature extraction method

and improve the accuracy in Gabor labeled elastic graph feature and better recognition rate is obtained as a result.

- Intelligent agent-based module: iJADE provides an intelligent agent-based platform to support the implementation of various AI functionalities. We facilitate those AI features to implement an intelligent and automatic multi-agent based face recognition system. Based on the three major processes in face recognition system, we implement the automatic multi-scale face detector as face detection module, automatic facial feature selector as feature extraction module, and invariant face identifier as face identification module in iJADE Face Recognizer.
- Scalable agent system: Agent technology, with its automatic delegation of task, autonomous and highly mobile characteristics, provides a collaborative platform to build sophisticated application with interactive services among different agents and is particularly suitable to deploy the proposed system on it. Asynchronous runtime environment in agent server enables parallel processing to increase system performance while distributed architecture in agent system supports operation in various scales without deteriorating the system performance as well.

6.3. Limitations

Low detection rate and high false alarm rate are the major problems in face detection module. It is because the dimension of face template is very large and the computational costs and training samples must increase exponentially with the dimension problem. The number of hidden unit in neural network based face classifier is another difficult problem.

Locating the nodes in the graph to specific facial landmarks in elastic graph matching method is an unsolved problem at this moment. The proposed elastic graph templates for different poses can provide the approximate position of each node and these templates for different poses can reduce the processing time on the matching process. However, the matching process for locating the exact position of facial landmarks still requires intensively computation.

6.4. Further Works

Further study can be focused on the invariant facial feature extraction model and contemporary approach such as elastic graph matching requires intensively computation and the accuracy is not good enough to work on every situation. The face manifold of Gabor PCA feature can be one of the possible solutions. Face recognition under occlusion is also another interesting topic that is worth to focus on.

Reference

[Aglet] IBM Aglets, http://aglets.sourceforge.net/

[Aitkenhead & McDonald 2003] Aitkenhead, M.J. and McDonald, A.J.S., "A neural network face recognition system", Engineering Application of Artificial Intelligence 16(3) 167-176, 2003.

[Ao Ieong & Lee 2004] Ao Ieong, Tony W.H. and Lee, Raymond S.T., "iJADE Face Recognizer - A Multi-Agent Based Pose and Scale Invariant Human Face Recognition System", KES 2004.

[Ao Ieong 2005] Ao Ieong, Tony W.H., Lam, Toby H.W., Lee, Alex C.M. and Lee, Raymond S.T., "iJADE Tourist Guide – A Mobile Location-Awareness Agent-Based System for Tourist Guiding", KES 2005.

[Belhumeur 1997] Belhumeur, Peter N., Hespanha, Joao P., Krigman, David J., "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", IEEE Trans. Pattern Analysis and Machine Intelligent 19(7) 711-720, 1997.

[Benjamin] Benjamin Caricature, http://benjamin.oportot.com/

[Birchfield 1998] Birchfield, Stan, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", Proc. of IEEE conf. on Computer Vision and Pattern Recognition, pp. 232-237, Jun 1998.

[Bishop 1995] Bishop, Christopher M., "Neural Network for Pattern Recognition", Clarendon Press, Oxford, 1995.

[Blanz & Vetter 2003] Blanz, Volker and Vetter, Tomas, "Face recognition based on fitting a 3D morphable model", IEEE Trans. Pattern Analysis and Machine Intelligent 25(9) 1063-1074, 2003.

[Brunelli & Paggio 1993] Brunelli, Roberto and Paggio, Tomaso, "Face recognition: features versus templates", IEEE Trans. Pattern Analysis and Machine Intelligent

15(10) 1042-1052, 1993.

[Caetano & Barone 2001] Caetano, T. S. and Barone, D.A.C., “A probabilistic model for the human skin color”, Proc. of IEEE international Conf. on Image Analysis and Processing, Palermo, Italy, pp. 279-283, Sep 2001.

[Caetano 2003] Caetano, Tiberio S., Olabarriaga, Silvia D., Barone, Dante A.C., “Do mixture models in chromaticity space improve skin detection?”, Pattern Recognition 36 3019-3021, 2003.

[Chandran & Kar 2002] Chandran, Sharat and Kar, S., “Retrieving faces by the PIFS fractal code”, Proc. IEEE Workshop on Application of Computer Vision, pp. 8-12, 2002.

[Chui 1992] Chui, C. K., “An introduction to wavelets”, Boston Academic Press, 1992.

[CMU] CMU face test set, URL: http://vasc.ri.cmu.edu/idb/html/face/index.html

[Crownover 1995] Crownover, Richard M., “Introduction to Fractals and Chaos”, Boston Jones and Bartlett, 1995.

[Davison 2000] Davison, N.E., Eviatar, H., Somorjai, R.L., “Snakes simplified”, Pattern Recognition 33 1651-1664, 2000.

[FaceOK] FaceOK, http://www.titanium-tech.com

[Feraud 2001] Feraud, Raphael, Bernier, Olivier J., Viallet, Jean-Emmanuel, Collobert, Michel, “A Fast and Accurate Face Detector Based on Neural Networks”, IEEE Trans. Pattern Analysis and Machine Intelligent 23(1) 42-53, 2001.

[Feris & Cesar 2000] Feris, Rogerio S. and Cesar Junior, Roberto M., “Tracking facial features using Gabor wavelet Networks”, Proc. Int. IEEE Symposium on Computer Graphics and Image Processing, pp. 22-27, 2000.

[Feris 2001] Feris, Rogerio S., Cesar, Roberto M., Kruger Jr., Volker, “Efficient real-time face tracking in wavelet subspace”, Proc. of ICCV'2001 Int. Conf. on Computer Vision, Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, BC, 2001.

[Garcia & Delakis 2004] Garcia, Christophe and Delakis, Manolis, “Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection”, IEEE Trans. Pattern Analysis and Machine Intelligent 26(11) 1408-1423, 2004.

[Gong 2000] Gong, Shaogang, McKenna, Stephen J, Psarrou, Alexandra, “Dynamic Vision: From Images to Face Recognition”, Imperial College Press, 2000.

[Gonzalez & Woods 2002] Gonzalez, Rafael C. and Woods, Richard E., “Digital Image Processing”, 2nd Edition, Prentice Hall, 2002.

[Haykin 1999] Haykin, Simon, “Neural Networks: A Comprehensive Foundation”, Prentice Hall, 1999.

[Hossein & Chadran 2001] Hossein, Ebrahimpour-Komleh, Chadran, Vinod, Sridharan, Sridha, “Robustness to expression variations in fractal-based face recognition”, IEEE Int. Symposium on Signal Processing and its Applications, Kuala Lumpur, Malaysia, pp. 359-362, Aug, 2001.

[Hossien 2001] Hossien, Ebrahimpour-komleh, Chandran, Dr V, Sridharan, Prof S, “Face recognition using fractal codes”, Proc. Int. Conf IEEE on Image Processing, Thessaloniki, Greece, pp. 58-61, 2001.

[Hsu 2002] Hsu, Rein-Lein, Mohamed Abdel-Mottaleb, “Face Detection in Color Images”, IEEE Trans. Pattern Analysis and Machine Intelligent 24(5) 696-706, 2002.

[iJADE] iJADE, http://www.ijadk.com

[Illusion] Illusion images, http://rootless.netfirms.com/funpic/page_01.htm

[Jacquin 1993] Jacquin, Arnaud E., “Fractal image coding: a review”, Processing of the IEEE 81(10) 1451-1465, Oct. 1993.

[Jeng 1998] Jeng, Shi-Hong, Liao, Hong Yuan Mark, Chin, Chuan Han, Chern, Ming Yang, Liu, Yao Tsorng, "Facial feature detection using geometrical face model: An efficient approach", Pattern Recognition 31 273-282, 1998.

[Kim 2003] Kim, Hyun-chul, Kim, Daijin, Bang, Sung Yang, "Face recognition using LDA mixture model", Pattern Recognition Letter 24 2815-2821, 2003.

[Kong 2003] Kong, Wa Kin, Zhang, David, Li, Wenxin, "Palmprint feature extraction using 2-D Gabor filters", Pattern Recognition 36 2339-2347, 2003.

[Kosko 1992] Kosko, Bart, "Neural Networks for Signal Processing", Prentice Hall, 1992.

[Kouzani & Sammut 2000] Kouzani, A.Z. and Sammut, F. He, K., "Towards invariant face recognition", Information Science, vol. 123, pp. 72-101, 2000.

[Lades 1993] Lades, Martin, Vorbruggen, Jan C., Buhmann, Joachim, Lange, Jorg, "Distortion Invariant Object Recognition in the Dynamic Link Architecture", IEEE Trans. on Computers 42(4) 300-311, 1993.

[Lange & Oshimam 1999] Lange, Danny B. and Oshimam, Mitsuru, "Seven good reasons for mobile agents", Communications of the ACM, Vol. 42, No. 3, March 1999.

[Lee 2003] Lee, Raymond S.T., "iJADE surveillant - an intelligent multi-resolution composite neuro-oscillatory agent-based surveillance system", Pattern Recognition 36 1425-1444, 2003.

[Li & Liu 2002] Li, Bai and Liu, Yihui, "When eigenfaces are combined with wavelets", Knowledge-Based Systems 15 343-347, 2002.

[Lin & Kung 1997] Lin, Shang-Hung and Kung, Sun-Yuan, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network", IEEE

Trans. Pattern Analysis and Machine Intelligent 8(1) 114-132, 1997.

[Liu & Wechsler 2001] Liu, Chengjun and Wechsler, Harry, “Independent Component Analysis of Gabor features for Face Recognition”, IEEE Trans. Pattern Analysis and Machine Intelligent 14(4) 919-928, 2001.

[Liu & Wechsler 2002] Liu, Chengjun and Wechsler, Harry, “Gabor feature based classification using the enhanced fisher linear discrimination model for face recognition”, IEEE Trans. Pattern Analysis and Machine Intelligent 11(4) 467-476, 2002.

[Liu 2003] Liu, Chengjun, “A Bayesian discriminating features method for face detection”, IEEE Trans. Pattern Analysis and Machine Intelligent 25(6) 725-740, 2003.

[Lyons 1999] Lyons, Michael J., Budynek, Julien, Akamatsu, Shigeru, “Automatic Classification of Single Facial Images”, IEEE Trans. Pattern Analysis and Machine Intelligent 21(12) 1357-1362, 1999.

[Negnevitsky 2002] Negnevitsky, Michael, “Artifical Intelligent: A Guide to Intelligent System”, Addison Wesley, 2002.

[NetLab] NetLab, http://www.ncrg.aston.ac.uk/netlab/

[Rowley 1998] Rowley, Henry A., Baluja, Shumeet and Kanade, Takeo, “Neural Nework-Based Face Detection”, IEEE Trans. Pattern Analysis and Machine Intelligent 20(1) 23-38, 1998.

[Samaria & Harter 1994] Samaria, Ferdinando and Harter, Andy, “Parameterization of a Stochastic Model for Human Face Identification”, Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL, December 1994. ORL Face Database, URL: http://www.uk.research.att.com/facedatabase.html.

[Sung & Paggio 1994] Sung, Kah-Kay and Poggio, Tomaso, “Example-based

Learning for View-based Human Face Detection", MIT, C.B.C.L. paper No. 112, 1994.

[Takacs & Wechsler 1997] Takacs, Barnabas and Wechsler, Harry, "Detection of Faces and Facial Landmarks Using Iconic Filter Banks", Pattern Recognition 30(10) 1623-1636, 1997.

[Tan & Yan 2001] Tan, Teewoon and Yan, Hong, "Object recognition based on fractal neighbor distance", Signal Processing 81 2105-2129, 2001.

[Tan & Yan 2002] Teewoon Tan, Hong Yan, "The fractal neighbor distance measure", Pattern Recognition 35 1371-1387, 2002.

[Tan 1999] Tan, Teewoon and Yan, Hong, "Face recognition by fractal transformations", IEEE IMDSP 6 3537-3540, 1999.

[Tao 2001] Tao, Yu, Lam, Ernest C.M. and Tang, Yuan Y., "Feature extraction using wavelet and fractal", Pattern Recognition Letter 22 271-284, 2001.

[Terrillon 2000] Terrillon, Jean Christophe, Shirazi, Mahdad N., Fukamachi, Hideo and Akamatsu, Shigeru, "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images", Proc. of 4th IEEE international Conf. on Automatic Face and Gesture Recognition, pp. 54-61, 2000.

[Vang & Yuan 2001] Vang, Yanjiang and Yuan, Baozong, "A novel approach for human face detection from color images under complex background", Pattern Recognition 34 1983-1992, 2001.

[Wiskott 1997] Wiskott, Laurenz, Fellous, Jean-Marc, Krüger, Norbert, Malsburg, Christoph von der, "Face recognition by elastic bunch graph matching", IEEE Trans. Pattern Analysis and Machine Intelligent 7 775-779, 1997.

[Yang & Ahuja 2001] Yang, Ming-Hsuan and Ahuja, Narendra, "Face Detection and

Gesture Recognition for Human-Computer Interaction", Kluwer Academic Publishers, 2001.

[Yang 2002] Yang, Ming-Hsuan, Kriegman, David J., Ahuja, Narendra, "Detecting Faces in Images: A Survey", IEEE Trans. Pattern Analysis and Machine Intelligent 24(1) 34-58, 2002.

[Zhao 2004] Zhao, Zhong-Qiu, Huang, De-Shuang, and Sun, Bing-Yu, "", Pattern Recognition Letter 25 1351-1358, 2004.

[Zhang 2002] Zhang, P., Bui, T. D. and Suen, C. Y., "Recognition of similar objects using 2-D wavelet-fractal feature extraction", Proc. Int. Conf. on Pattern Recognition, vol. 2, 316-319, Quebec City, Aug. 2002.