Face Detection in Colour Images by ICA-SVM Architecture

Tsz Ying Lui

Department of Electronic Engineering
Queen Mary, University of London

A thesis submitted for the degree of
Master of Philosophy

Oct 2004

ABSTRACT

We describe a face detection algorithm based on support vector machine (SVM). The algorithm consists of two–steps. The first step is a skin detection model which serves as a platform to reduce the searching space for potential face candidates. The second step reduces the computational complexity of the SVM architecture by projecting the image signals into a face subspace, constructed under ICA framework, to reduce the dimensionality of the problem while preserving the unique facial features. Experiments were conducted using various real world data and results are reported.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Ebroul Izquierdo for his in-depth guidance. Exchanges of ideas and fruitful discussions with Ebroul were and are always a big source of inspiration and instrumental in the success of my work.

Special thanks go to Mike Lee for his friendly support and fruitful discussions on my works and to all other colleagues for their conservation and assistance during my studies.

This thesis is dedicated to my parents for their love and support.

Contents

Chapter 1

Introduction

Computer vision, in general, aims to duplicate human vision, and traditionally, has been used in performing routine and repetitive tasks, such as classification in massive assembly lines. Today, researchers are trying to build intelligent machines that have different functions. Building machines with the faculty of vision is probably one of the most challenging problems human beings strive to solve. In this aspect, the human face is one of the most fascinating of all objects: powerful, expressive, and highly variable. At the same time, it is a highly specialised part of the body and the most convincing proof of an individual's identity. Despite this fact, one can list several relevant applications, such as face recognition, computer human interaction, and crowd surveillance, where all of these applications require face detection as a pre-processing step to obtain the "face object". In other words, many of the techniques that are proposed for these applications assume that the location of the face is pre-identified and is available for the next step.

1.1 Motivation

Face detection is one of the tasks that human vision can do without much effort. However, for machines, such an effortless task becomes challenging as face appearance depends mainly on the viewing conditions, the geometrical sensors and the photometrical parameters. Although the computer vision community started to pay attention to face processing three decades ago, there is as yet no solution with performance comparable to humans both in precision and speed.

In this thesis, we are mainly interested in the face detection problem, such as how to find - based on visual information - all the occurrences of faces regardless of who the person is. High precision is now technically achieved by building systems that learn from a lot of data in order to minimise errors on test sets. In most cases, the increase in precision is achieved at the expense of a degradation in run-time performance. However, in major applications not only high precision is demanded, the computation resource also has to be constrained at a reasonable cost.

1.2 Challenges

A general definition of face detection can be stated as identification of all regions that contain a face, in a still image or image sequence, independent of any three dimensional transformation of the face and lighting conditions of the scene. This statement implies an enormous statistical variation among all possible face images under all viewing conditions, resulting to the query: what is the set of all possible face images under all viewing conditions?

Pose Variation

Slight changes in the face's position often leads to large changes in the face's appearance. This is mainly because of the 2D transformations such as translations and rotations in the image plane. As faces are considered as highly semi-rigid 3D objects, rigid perspective transformations from 3D objects to 2D image plane cause various distortions to their appearances.

Lighting Condition

The variability in faces due to differences in illumination is usually dramatic. This not only lead to changes in contrast, but also in the configuration of the shadow. Moses et al. [56] have observed that the variability in the images of a given face due to illumination changes is greater than that from person to person under the same lighting. Although indoor lighting conditions can be well controlled and hence face detectors achieve very high performance in such conditions, lighting conditions for outdoor scenes are impossible to control, resulting in strong variations in facial appearances.

Shape Variation

Human faces are subject to various shapes due to physiognomy, ethnicity, physiological behaviour, and facial expressions. The shapes can be classified as either global shape variations (height, elongation, etc.) or local shape variations (nose and mouth shape, distance between the eyes, etc.). Such variability further complicates the task.

To tackle these problems, many face detection methods proposed the use of the pose and intensity information to model and learn different representations of faces under varying conditions. The main drawback of these approaches is that in order to detect faces seen in a particular pose and lighting condition, the faces must have been seen previously under the same conditions. Clearly, this makes the decision very complex and the visual selection task quickly gets out of hand. Hence, a detection algorithm should deal with as many of these sources of variability as possible. This is roughly equivalent to using a larger training set containing many training examples generated from real examples by changing the global conditions. It is clear that a larger training set generally improves the performance of a face detection algorithm, but it also usually increases the processing time for training and classification. Therefore, reducing the variability and reducing the size of the training set and complexity of the decision boundary should be balanced in order to achieve reasonable performance.

1.3 Thesis Outline

A comprehensive review of the works regarding face detection covered in literatures is given in Chapter 2. In Chapter 3, algorithms for facial feature extraction are described. Five different skin models are reviewed and compared in different colour spaces together with experimental results. The purpose of skin colour detection is to condense the searching space for potential face candidates and hence to reduce the system latency. To further speed up the system, meaningful facial features are extracted from original high dimensional image pixels using PCA and ICA frameworks. Unlike other approaches, the differences between PCA and ICA models are highlighted from the statistical point of view while experimental results follow subsequently. Chapter 4 describes the learning of face models using the SVM model together with various testing results. Finally, Chapter 5 concludes the entire study and highlights the future research direction.

Chapter 2

Towards Face Detection

Early efforts in face detection have dated back as early as the beginning of the 1970s, when simple heuristic and anthropometric techniques were used under various assumptions such as plain background, frontal face, etc. As a result of the rigidness of these systems, any changes in image conditions would mean fine–tuning of the system configuration, if not a complete re–design. These problems inhibited the growth of research interest until the 1990s, when practical face recognition and video coding systems started to become a reality. Over the past decade, there has been a great deal of research interest spanning several important aspects of face detection, including more robust segmentation schemes, employment of statistics and neural networks for face detection and recognition, and advanced feature extractors.

Face detection techniques require a priori information of the face, and can be effectively organised into three broad categories distinguished by their different approaches to utilising face knowledge. The techniques in the first category make explicit use of face knowledge and follow the classical detection methodology in which low level features are derived prior to knowledge–based analysis. The apparent properties of the face such as skin colour and face geometry are exploited at different levels. The aim is to find the structural features that exist invariant to the pose, viewpoint, or lighting conditions, and then use these features to locate faces. Since features are the main ingredients, these techniques are termed as feature–based approach. The techniques in the second group encode human knowledge of what constitutes a typical face by using rule–based methods where the rules capture the relationships between facial features. These methods are designed mainly for face localisation and are known as knowledge–based methods. The last but not the least category addresses face detection as a general pattern recognition problem. Classifications of face group are directly performed by using image–based representations of faces with training algorithms without feature derivation and analysis. Unlike the feature–based and knowledge–based approaches, these relatively new techniques incorporate face knowledge implicitly into the system through mapping and training schemes.

2.1 Feature–Based Approach

The development of the feature–based approach can be divided into three areas: low–level analysis, feature analysis and active shape models. Given a typical face detection problem, the objective of low–level analysis is to segment the potential face region by using the visual features from the pixel properties such as grey–scale and colour. However, features generated from this analysis are always ambiguous. Therefore, the visual features are organised into a more global concept of face and facial features in feature analysis stage by using prior knowledge of face geometry. Through feature analysis, feature ambiguities are reduced and locations of the face and facial features are determined. The last area involves the use of active shape models. For instance, point distributed models (PDM) have been developed for the purpose of complex and non–rigid feature extraction such as eye pupil and lip tracking.

2.1.1 Low–Level Analysis

Edges

The application of edge representation in the face detection problem can be dated back as early as the work by Sakai et al. [75]. By analysing line drawings of the faces from photographs, the facial features are located. Craw et al. [15] later proposed a hierarchical framework based on Sakai et al.'s work to trace a human head outline. More recent examples of edge–based face detection and facial feature extraction techniques can be found in [16, 80].

Edge detection is the foremost step in deriving edge representation. So far, many different types of edge operators have been applied. Among them, the Sobel operator was the most common filter being used. In [23] the MarrHildreth edge operator was used. A variety of first and second derivatives (Laplacian) of Gaussians have also been used in other methods. For instance, steerable and multi–scale orientation filters are used in [20]. The steerable filtering consists of three sequential edge detection steps, which include detection of edges, determination of the filter orientation of any detected edges, and stepwise tracking of neighbouring edges using the orientation information. The algorithm has allowed an accurate extraction of several key points in the eye.

In an edge–detection–based approach to face detection, edges need to be labelled and matched to a face model in order to verify correct detections. In [23] Govindaraju proposed an algorithm that labels edges as the left side, hairline, or right side of a front view of a face. The labelled components are then combined to form possible face candidate locations and matched to a facial model based on a cost function using the golden ratio[1]:

$$\frac{height}{width} \equiv \frac{1+\sqrt{5}}{2} \tag{2.1}$$

By testing this algorithm on a set of 60 images with complex backgrounds containing 90 faces, the system achieves 76% detection rate with an average of two false alarms per image.

Grey Information

Besides edge representation, the grey information within a face is also frequently used since facial features such as eyebrows, pupils, and lips appear generally darker than their surrounding facial regions. This prior knowledge of human faces can be exploited to differentiate various facial parts. In [24] the input images are first enhanced by contrast–stretching and grey–scale morphological routines to improve the quality of local dark patches and then the local grey minima within segmented facial regions are searched for local facial features. The extraction of dark patches is achieved by simply thresholding. In contrast, Hoogenboom and Lew [29] proposed an algorithm by using local maxima to indicate the bright facial spots such as nose tips. The local maxima are defined as bright pixels surrounded by eight dark neighbours. The detection points are then aligned with feature templates for correlation measurements.

On the other hand, Yang and Huang [97] explored the grey–scale behaviour of faces in mosaic (pyramid) images by using the fact that the macroscopic features of the face will disappear when the resolution of a face image is reduced gradually. They have observed that the face region will become uniform at low resolution. Based on this observation Yang proposed a hierarchical face detection framework. Starting at low resolution images, face candidates are established by a set of rules that searches for uniform regions. The face candidates are then verified by the existence of prominent facial features using local minima at higher resolutions. Their technique is later incorporated into a system for rotation invariant face detection by Lv et al. [51].

Colour

Compared with grey information, colour is a more powerful means for discerning object appearances. From the research works by McKenna et al. [53] it was found that different human skin colour tends to form a tight cluster in colour space even when faces of difference races are considered. This means colour composition of human skin differs little across individuals.

Among various colour spaces, the RGB colour space is one of the most widely used. Since the luminance change will cause large variation in skin appearance, normalised RGB colours are generally preferred for skin colour detection [27, 81, 83]. The normalised colours can be derived

[1] An aesthetically proportioned rectangle used by artists.

from the original RGB components as follows:

$$
\begin{aligned}
r &= \frac{R}{R+G+B} \\
g &= \frac{G}{R+G+B} \\
b &= \frac{B}{R+G+B}.
\end{aligned}
\tag{2.2}
$$

For a colour histogram based on r and g, it has been shown that the skin colour only occupies a small cluster in the histogram. Therefore, the likelihood of a pixel belonging to face region can be deduced by comparing its colour information with respect to the r and g values of the face cluster.

Besides rgb colour model, several other alternative models are also actively being used in the face detection research. For instance, the HSI colour representation has been shown to have advantages over other models in giving large variance among facial feature colour clusters. Hence this model is used to extract facial features such as lips, eyes, and eyebrows. It is also widely used in face segmentation schemes [68, 17].

On the other hand, it is found that the I–component in the YIQ colour space could enhance the skin region of Asians [16]. Other colour models applied to face detection include HSV [85], YCrCb [1, 76], YUV, CIE–xyz [11], L*a*b* [46], and L*u*v* [28].

A comparative study of several widely used colour spaces for face detection was presented by Terrilon et al. [87]. In their study, they compared normalised TSL (tint–saturation–luminance), *rg* and CIE–xy chrominance spaces, and CIE–DSH, HSV, YIQ, YES, CIE–L*u*v*, and CIE L*a*b* chrominance spaces. In each colour space, the skin colour distributions are modelled as either a single Gaussian or a Gaussian mixture density model. They extracted Hu's moments [30] as features and trained a multi–layer perceptron neural network to classify the face candidates. In general, they showed that skin colour in normalised chrominance spaces can be modelled with a single Gaussian and performs very well, while a mixture–model of Gaussians is needed for the unnormalised spaces. In their face detection test, the normalised TSL space provided the best results, but the general conclusion was that the most important criterion for face detection is the degree of overlap between skin and non–skin distributions in a given space, which is highly dependent on the number of skin and non–skin samples available for training.

Texture

Human faces have a distinct texture that can be used to separate them from other objects. Augusteijn and Skufca [2] presented a piece of work showing that human faces can be detected through the identification of face–like textures. The textures are computed using second–order statistical features on sub–images of 16×16 pixels. Three types of textures are computed including hair, skin and others. Then a cascade correlation neural network is used for supervised texture classification together with a self–organising map to form clusters of different texture classes. When a face candidate is presented to the system, votes of the occurrence of hair and skin textures are used to infer the presence of a face. However, they only reported the results of texture classification, not face detection.

Similar to [2], Dai and Nakano also applied the same texture model to face detection [16]. However, they incorporated colour information together with the face–texture model. They designed a scanning scheme for face detection in colour scene in which the orange–like parts including face areas are enhanced. They reported that their approach has the advantage in detecting faces that are not upright or have features such as beards and glasses. The detection rate is 100% for a test set of 30 images with 60 faces.

Geometric Measures

Besides edge, grey information, colour and texture, the geometric features of human faces have also been explored. As early as the work in Reisfeld and Yeshurun [71], a generalised symmetry operator that is based on edge pixel operation is introduced to local facial features by taking the fact that facial features are symmetrical in nature. Given an image, the symmetry magnitude is assigned at every pixel location based on the contribution of surrounding pixels. By using this symmetry magnitude map, the system achieves 95% success rate in detecting eyes and mouths in a database consisting of various backgrounds and orientations.

Later Lin and Lin [47] proposed a dual–mask operator that is similar to Reisfeld and Yeshurun [71] operator but with less complexity by exploiting the radial symmetry distribution on bright and dark facial parts. Different from Reisfeld and Lin, Tankus et al. [85] proposed a new attentional operator based on smooth convex and concave shapes by making use of the derivative of gradient orientation with respect to the y–direction which is termed Y–Phase. From their experimental results, it was shown that the Y–Phase has a strong response at the negative x–axis for concave and convex objects (paraboloids). Since facial features generally appear to be parabolic, their Y–Phase response will also give a strong response at the x–axis. They also proved that Y–Phase is invariant under a very large variety of illumination conditions and is insensitive to strong edges from non–convex objects and texture backgrounds.

2.1.2 Feature Analysis

Features generated from low–level analysis are likely to be ambiguous. For instance, segmenting facial regions using a skin colour model will always detect background objects of similar colour. To reduce this ambiguity, many face detection systems have employed the prior knowledge of face geometry to characterise and verify these features. One common approach is to perform sequential feature searching based on the relative positioning of individual facial features, and to enhance the confidence of a feature's existence if nearby features are detected. Starting with the determination of prominent facial features, the feature searching techniques try to detect other less prominent features by using anthropometric measurements of face geometry. For instance, the hypothesis of a pair of symmetric dark regions found in the face area may represent the eyes on the face, and subsequently increase the confidence of a face existence. Because of the distinct side–by–side appearance of human eyes, a pair of eyes is the most commonly applied reference feature among all the facial features. Other features include a main axis of the face, top of the head and mouth, etc.

Besides sequential feature searching, many methods that combine multiple facial features have also been proposed for face detection or localisation. Most of them initially utilise global features such as skin colour, size and shape to find the possible face candidates and then verify these candidates using local detailed features such as eyebrows, nose and hair. A typical approach begins with the detection of skin–like regions and then groups the skin–like pixels together using connected component analysis or clustering algorithms. If the shape of a connected region has an elliptic or oval shape, it becomes a face candidate for further verification using other local features.

A typical feature searching and multiple facial feature analysis example could be found in the work of Jeng et al. [37]. The system detects various facial features based on anthropometric measures, and predicts the presence of a face based on the existence of these features. The system first searches for possible locations of the eyes in binarised pre–processed images. Then it goes on to search for a nose, a mouth, and eyebrows at each possible eye position. Each facial feature has an associated evaluation function, which is used to determine the final most likely face candidate. Finally, each facial feature is weighted according to their facial importance with manually selected co–efficients as follows:

$$E_{face} = 0.5E_{eye} + 0.2E_{mouth} + 0.1E_{Reyebrow} + 0.1E_{Leyebrow} + 0.1E_{nose} \tag{2.3}$$

where E represents existence of particular facial feature. 86% detection rate was reported on a data set of 114 test images taken under controlled imaging conditions, but with subjects positioned at various directions with a cluttered background.

Wu et al. [96] presented a face detection method in colour images using fuzzy theory. In their system, two fuzzy models are used to describe the distributions of skin and hair colour in CIE XYZ colour space. The appearances of faces in images are represented by five head–shape models (one frontal and four side–views). Each shape model is a 2D pattern of $m \times n$ square cell assigned with two properties: the skin proportion and the hair proportion where the portions indicate the ratios of the skin/hair area within the cell to the area of the cell. For each input image, each pixel is classified as hair, face, hair/face, and hair/background based on the distribution models, thereby segmenting the image into skin–like and hair–like regions. The skin–like and hair–like regions are compared with the head–shape models, and are labelled as face candidates if they are similar. The eye–eyebrow and nose–mouth features are then extracted from each face candidate using horizontal edges for further verification.

A imilar pproach s roposed y obottka nd itas 83] or ace ocalisation sing hape and colour. In their system, skin–regions are first segmented in HSV colour space. Then connected component analysis is applied for region growing at a coarse resolution where the best fitted ellipse is computed for each connected component using geometric moments. Connected components that are well approximated by an ellipse are selected as face candidates. These candidates are verified by searching for facial features such as eyes and mouths based on the observation that they are darker than the rest of a face. Later, Terrillon et al. [86] proposed a similar method where a Gaussian skin colour model is first used to classify skin pixels and for each connected component, a set of 11 lowest–order geometric moments are computed using Fourier and radial Mellin transforms to characterise its shape. For detection, a neural network is trained with the extracted geometric moments. A detection rate of 85% was reported from their experiments based on a test set of 100 images.

Recently, Smeraldi et al. [82] proposed an algorithm based on eye movements. In the system, they constructed a template of the search targets (the eyes) by averaging Gabor responses from a retinal sampling grid centered on the eyes of the subjects in the training set. Six orientations and five different frequencies are employed for feature extraction. During the detection of the eyes, a saccadic search algorithm is applied. Initially, the algorithm randomly places the sampling grid in the image and subsequently moved it to the position where the Euclidian distance between the node of the sampling grid and the node in the search target is the smallest. The grid is moved around until the saccades become smaller than a threshold. If the search started at a blank area in the image where no target can be found, a new random position is tried. From a set of 800 frontal view face images, 84% correct detection of the eyes was reported.

Almost at the same time, Maio and Maltoni [52] proposed a two–stage face detection system using Hough transform. Using a gradient–type operator over local window (7 × 7 pixels), the input image is first converted to a directional image from which the Hough transform is applied to search for ellipses. The selected face candidates are then matched with a set of 12 binary templates for verification. Fig. 2.1 shows their proposed system. The proposed system functions in real time and a correct detection in 69 out of 70 test images was reported with no false alarms, where the test images consist of single faces of varying sizes with complex backgrounds.

2.1.3 Active Shape Models

Different from the techniques mentioned in the previous sections, active shape models depict the actual physical appearance of object and hence give higher–level object features. The most well–known face detection system using active shape model is developed by Cootes et al. [34]. They proposed the use of a new generic flexible model, which they termed Point Distribution Models (PDM) to provide an efficient interpretation of the human face. The model is based on a set of labelled points that are only allowed to vary to certain shapes according to a training procedure.

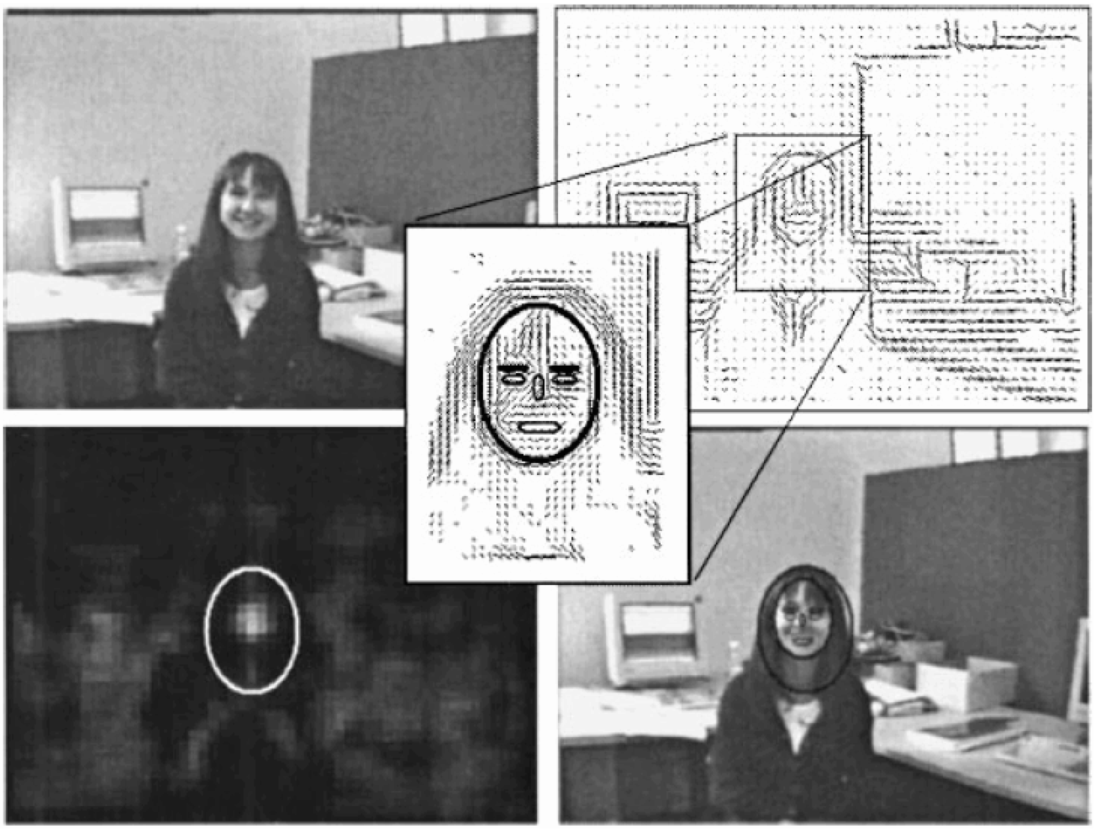

Figure 2.1: The face detection system re–produced from Maio and Maltoni [52].

PDM is a compact parameterised description of the shape based upon statistics in which the contour of the object shape is represented by a set of labelled points. Given a training set consists of objects of different sizes and poses, variations of these labelled points are parameterised via training. By using principal component analysis (PCA), variations of the features are then constructed as a linear flexible model. The model comprises the mean of all the features in the sets and the principal modes of variation for each point. Let $\mathbf{x}$ represents a point on the PDM, and $\bar{\mathbf{x}}$ is the mean feature in the training set for that point, then:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{v} \tag{2.4}$$

where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots \mathbf{p}_t]$ is the matrix of the t most significant variation vectors of the covariance of deviations, and $\mathbf{v}$ is the weight vector for each mode.

In [45], Lanitis et al. developed a face PDM as a flexible model. The model depicts the global appearance of a face that includes all the facial features such as eyebrows, nose, and eyes. They manually labelled 152 control points and trained these points with 160 training face images to obtain a face PDM. They showed that the model can approximate up to 95% of the face shapes in the training set by using only 16 principal weights. To fit a PDM to a face, the mean shape model is first placed near the face, and then each point is moved towards its corresponding boundary point by employing a local grey–scale search strategy. During the deformation, the shape is only allowed to change in a way which is consistent with the information modelled from the training set.

The advantage of face PDM is that only compact parameterised points are sufficient to describe a face shape. In their subsequent work, Lanitis et al. [44] have incorporated a genetic algorithm (GA) and multi–resolution approach to address the problem in multiple face candidates. They also showed that the global characteristic of the model allows all the features to be located simultaneously and thereby removes the need for feature searching. Furthermore, the occlusion of a particular feature does not pose a severe problem since other features in the model can still contribute to a global optimal solution.

2.2 Knowledge–Based Methods

In this approach, face detection methods are developed based on the rules derived from human faces. The rules describe the features of a face and their relationships by using the researcher's knowledge of human faces explicitly. For instance, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth with more darker red colour than other components of the face. These relationships can be represented by their relative distances and positions. Usually, different facial features are extracted first and then the face candidates are identified based on the coded rules. A verification process is usually applied to reduce false detection.

The main challenge with this approach is the difficulty in translating human knowledge into well–defined rules. If the rules are too restrictive, it might be too difficult for faces to pass all the rules for detection. If the rules are too general, they may give many false positives. In general, using heuristics about faces work well in detecting frontal faces in uncluttered scenes, but they are too difficult to be extended to detecting faces in different poses since it is challenging to enumerate all possible cases.

One of the early works on knowledge–based face detection is presented by ang and Huang [97]. Their hierarchical knowledge–based system consists of three levels of rules. By scanning a window over the input image, a set of rules are applied to identify the possible face candidates at each location. The rules at the highest level are general descriptions of what a face looks like while rules at lower levels rely on details of facial features. Fig. 2.2 depicts an example of the multi–resolution hierarchy of images created by averaging and sub–sampling. Examples of rules used in the lowest resolution (Level 1) include: "the central part of the face (the dark shaded parts in Fig. 2.3) has four cells with a basically uniform intensity", "the upper–round part of a face (the light shaded parts in Fig. 2.3) has a basically uniform intensity", and "the differences between the average grey values of the central part and the upper–round part is significant". Once the face candidates are searched at the lowest resolution, they are further

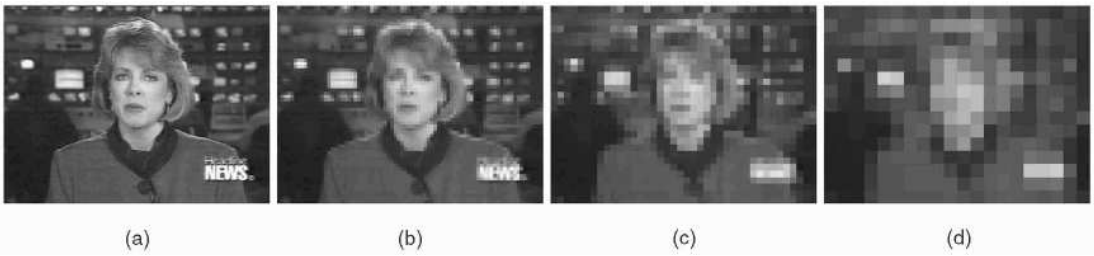

Figure 2.2: he multi–resolution hierarchy images created by averaging and sub–sampling, re–produced from ang and Huang [97]. (a) Original image. (b) n=4. (c) n=8. (d) n=16. From left to right, the figure shows the original image and lower resolution images in which the intensity of each pixel is replaced by the average intensity of $n \times n$ pixels.

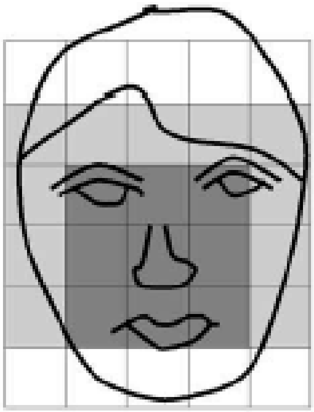

Figure 2.3: Facial region used in knowledge–based top–down method, re–produced from Yang and Huang [97].

processed at finer resolutions. At Level 2, local histogram equalisation is performed on the selected face candidates followed by edge detection. Candidates that passed Level 2 are finally examined at Level 3 with another set of rules in response to the facial features of eyes and mouth. By evaluating the system on a test set of 60 images, the system correctly located faces in 50 test images while there are 28 images in which false alarms appeared. Although the system did not result in high detection rate, the idea of using multi–resolution to reduce the required computations has been used in later face detection works [29].

Following Yang and Huang's work [97], [29] presented a similar rule–based face localisation method. First, facial features are detected with a projection method that locates the boundary of a face. Let $I(x, y)$ be the intensity value of an $m \times n$ image at position (x, y), the horizontal and vertical projections of the image are defined as:

$$HI(x) = \sum_{y=1}^{n} I(x, y)$$
$$VI(y) = \sum_{x=1}^{m} I(x, y). \tag{2.5}$$

Once the horizontal projection of the input image is obtained, the two local minima are determined by detecting abrupt changes in *HI*. The two local minima are assumed to correspond to the left and right side of the head. Similarly, the local minima are detected in the vertical projection which corresponds to the locations of mouth lips, nose tip and eyes. These detected facial features constitute a face candidate. Consequently, a set of rules detecting eyes, nose and mouth are applied to validate the candidate. The proposed method has been tested using a set of frontal faces extracted from the European ACTS M2VTS (MultiModal Verification for Teleservices and Security Applications) database. The database contains 37 different people and each image sequence contains only one face in a uniform background. The reported detection rate was 86.5%. However, it is difficult to detect a face in a complex background by using the horizontal and vertical projections and this method cannot detect multiple faces. Fig. 2.4 shows examples of this method.

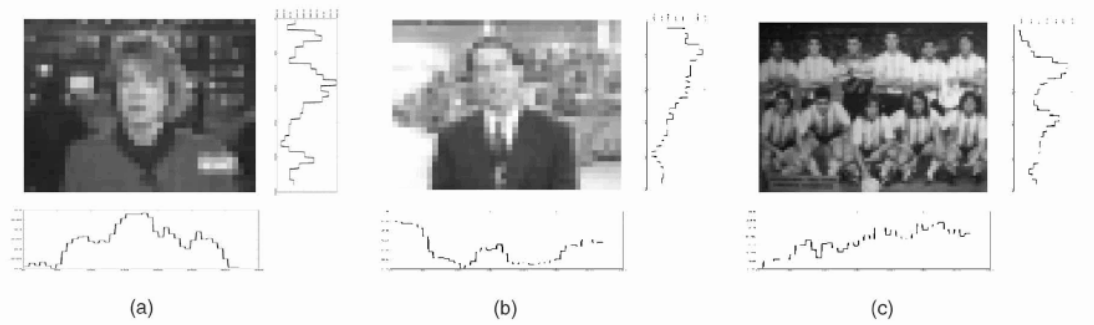

Figure 2.4: The horizontal and vertical projections of the input images, re–produced from [29]. It is possible to detect a single face by searching the peaks in horizontal and vertical projections as in (a). However, this method is difficult to detect a face in a complex background as shown in (b) and cannot detect multiple faces as shown in (c).

2.3 Image–Based Approach

It has been shown in the previous sections that the main problem in face detection by explicit modelling of facial features is the unpredictability of face appearance and complex environmental conditions. Although some of the recent researches have tackled the problem of unpredictability to some extent, most are still limited to quasi–frontal faces. The need for techniques to perform face detection in more hostile scenarios has inspired a new research area in which face detection is treated as a pattern recognition problem. By treating the problem of face detection as one of learning to recognise face pattern from examples, the specific application of face knowledge is avoided. This eliminates the potential modelling errors due to incomplete or inaccurate face knowledge. The fundamental approach of face pattern recognition is to classify examples into face and non–face prototype classes via a training process. The simplest image–based approach is template–matching [21, 40] but this approach does not perform as well as those using more complex techniques.

Most of the image–based approaches apply a window scanning technique for detecting faces. This scanning algorithm performs an exhaustive search of the input image for candidate face positions at all scales. Almost all the image–based systems have variations in the implementation of this algorithm. Depending on the methods and computational efficiency, the size of the window, the sub-sampling rate, and the step size vary across different systems.

In the following sections we divided the image–based approach into linear subspace methods, neural networks, and statistical approach. In each section, some representative methods are presented.

2.3.1 Linear Subspace Methods

Images of human faces lie in a subspace of the overall image space. To represent this subspace several methods have been explored in the literature including principal component analysis (PCA), linear discriminant analysis (LDA), and factor analysis (FA).

In the late 1980s, Sirovich and Kirby [81] developed a technique using PCA to efficiently represent human faces. The technique searches the principal components of the distribution of faces, which are expressed in terms of eigenvectors (commonly referred to as eigenfaces), for a given set of face images. Each individual face of the face set can then be approximated by a linear combination of a few largest eigenvectors using appropriate weights. Turk and Pentland [89] later extended this technique for face recognition by exploiting the distinct nature of the weights of eigenfaces in individual face representation. Since the re–constructed face is an approximation, a residual error is defined as a measure of “faceness”. They termed this residual error as “distance–from–face–space” (DFFS), which gives a good indication of face existence. The base procedures of their algorithm are as follows: Given a data set of n face images, Γ1, Γ2, · · · , Γn, the average face is defined by:

$$\Psi = \frac{1}{n}\sum_{i=1}^{n}\Gamma_i. \tag{2.6}$$

By subtracting the average face from each image, we obtain:

$$\Phi_i = \Gamma_i - \Psi \tag{2.7}$$

Let $\mathbf{D} = [\Phi_1\Phi_2\ldots\Phi_n]$ and $\mathbf{C} = \mathbf{D}\mathbf{D}^T$. The eigenvectors $\boldsymbol{u}_i$ of $\mathbf{C}$ are called the principal components (eigenfaces), which span the subspace called face space. An input image Φ can be projected onto this face space by

$$\mathbf{w}_k = \boldsymbol{u}_k^t \Phi, \quad k = 1, \ldots m, \tag{2.8}$$

where m is the number of principal components selected to span the face space. Since the principal components with small corresponding eigenvalues do not carry significant information in this representation, m is usually selected to be much smaller than n. The re–constructed image can be obtained by

$$\Phi_r = \sum_{k=1}^{m} \mathbf{w}_k \boldsymbol{u}_k. \tag{2.9}$$

The re–construction error $\varepsilon = \|\Phi - \Phi_r\|^2$ is the DFFS.

Pentland et al. [65] later proposed a facial feature detector using DFFS generated from so–called eigenfeatures (eigeneyes, eigennose, eigenmouth). The eigenfeatures are obtained from various facial feature templates sampled from a training set of 128 faces. Since features of different discrete views were used during the training phase, the detector has a better ability to account for features under different viewing angles. The performance of the eye locations was reported to be 94% with 6% false positive in a database of 7,562 frontal face images on a plain background. A slightly reduced, but still accurate performance for nose and mouth locations was also shown in [52]. The DFFS measure has also been used for facial feature detection in combination with Fisher's linear discriminant for face and facial feature detection [66].

Later, Moghaddam and Pentland [55] have further developed this technique within a probabilistic framework. As the orthogonal complement of face space is normally discarded when using PCA representation, Moghaddam and Pentland found that this leads to the assumption that the face space has a uniform density. Therefore they proposed a maximum likelihood detector which takes into account both the face space and its orthogonal complement to handle arbitrary densities. They reported a 95% detection rate on a set of 7,000 face images when detecting the left eye. A similar approach was also presented in [50] in which case PCA is applied for modelling both the face class and the pseudo–faces (non–faces, but face–like patterns) class, together with matching criteria based on a generalised likelihood ratio. In [54] Meng and Nguyen used PCA to model both faces and background clutter (an eigenface and an eigenclutter space).

Although PCA is an intuitive and appropriate way of constructing a subspace to represent an object class in many cases, it is not necessarily optimal for modelling the manifold of face images. This inspired another point of view that face space might be better represented by dividing it into sub–classes. Several methods have been proposed along this direction, most of which are based on mixtures of multi–dimensional Gaussians. This technique was first applied for face detection by Sung and Poggio [84]. Their method consists mainly of four steps:

1. The input sub–image is pre–processed by re–scaling it to 19×19 pixels, applying a mask

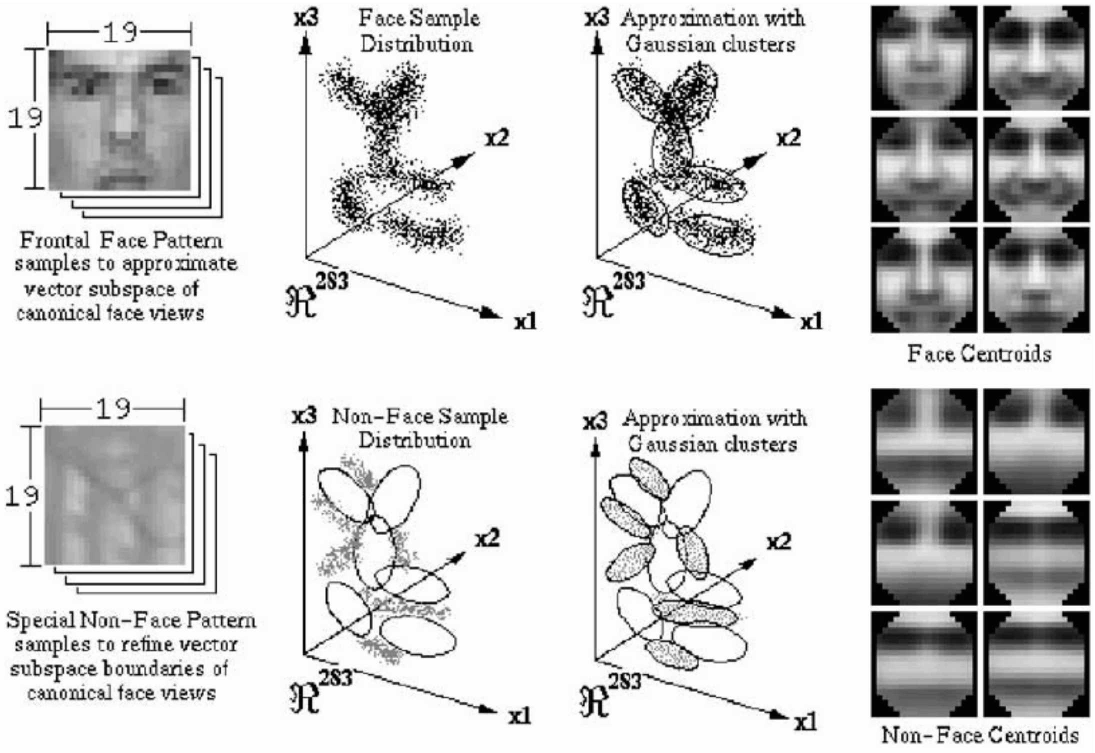
19
19
Frontal Face Pattern samples to approximate vector subspace of canonical face views
x3
Face Sample Distribution
x2
x1
$\Re^{283}$
x3
Approximation with Gaussian clusters
x2
x1
$\Re^{283}$
Face Centroids
19
19
Special Non-Face Pattern samples to refine vector subspace boundaries of canonical face views
x3
Non-Face Sample Distribution
x2
x1
$\Re^{283}$
x3
Approximation with Gaussian clusters
x2
x1
$\Re^{283}$
Non-Face Centroids

Figure 2.5: Face and non–face Gaussian clusters, re–produced from Sung and Poggio [84]. A set of Gaussians are used to estimate the density function for face and non–face patterns. The centers of these Gaussians are shown on the right.

for eliminating near–boundary pixels and subtracting a best–fit brightness plane from the unmasked window pixels, and finally applying histogram equalisation.

2. A distribution–based model of canonical face– and non–face patterns is constructed. By using an elliptical K–means clustering algorithm with an adaptively changing normalised Mahalanobis distance metric, 12 multi–dimensional Gaussian clusters with a global mean and a covariance matrix were constructed, of which six represent face, and six represent non–face pattern prototypes as shown in Fig. 2.5.

3. Two values are computed from each cluster. One is a Mahalanobis–like distance between the new image and the prototype centroid, defined within the subspace spanned by the 75 largest eigenvectors of the prototype cluster, while the other is the Euclidean distance from the new image to its projection in the subspace. Thus a 24–dimensional image measurement is computed. Fig. 2.6 shows the distance measures used in the system.

4. A multi–layer perceptron (MLP) is trained for face and non–face classification from the 24–dimensional image measurement vector. The MLP is not fully connected, but exploited some prior knowledge of the domain. The training set consists of 47,316 image, where 4,150 are examples of face patterns.

For face detection, a new image is first pre–processed through step 1 and 3, and then classified

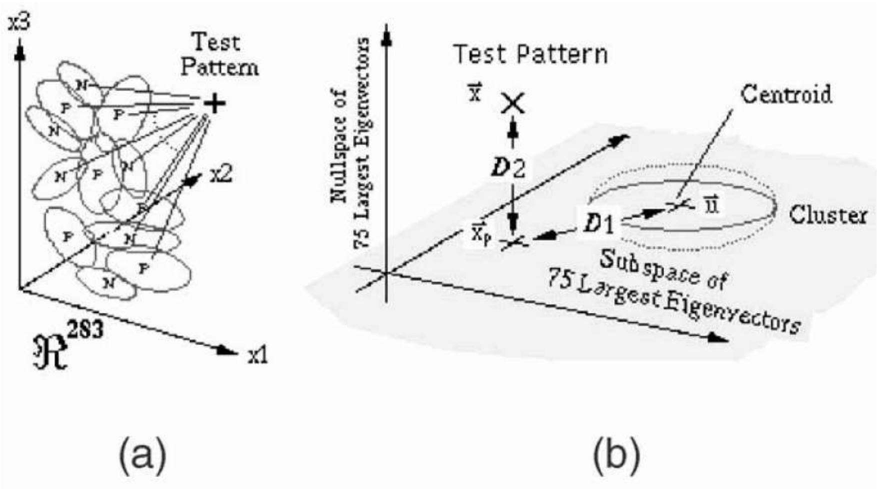

Figure 2.6: The distance measures used by Sung and Poggio, re–produced from their work [84]. Two distance metrics are computed between each input test pattern and the prototype clusters. (a) A set of 12 distances between the input pattern and the 12 cluster centroids are computed. (b) Each distance measurement between the input pattern and a cluster centroid is a 2D value. D_1 is the Mahalanobis distance between the input pattern's projection and the cluster centroid in a subspace spanned by the 75 largest eigenvectors. D_2 is the Euclidean distance between the input pattern and its projection in the subspace. Therefore, the distance measurement for each input pattern is a vector of 24 values.

using the MLP.

In [27], a similar approach to that of Sung and Poggio was explored by using grey–level features in combination with texture features. A more computationally efficient method was proposed in [32]. Another variation of this approach was presented in [25] where linear discriminant analysis is applied for feature selection before training the neural network classifier. A similar framework has also been proposed by Rajagopalan et al. [69]. They presented a new clustering algorithm using higher order statistics for modelling the face and non–face classes and reported good results.

In [97], the Fisher's Linear Discriminant (FLD) is used to project samples from the high dimensional space to a lower dimensional feature space. From their experiments, the FLD method outperforms the widely used eigenface method. One possible explanation is that FLD provides a better projection than PCA for pattern classification since it aims to find the most discriminant projection direction, and hence, the projected subspace may be superior to the eigenface projection. In their system, the training face and non–face samples are decomposed into several sub–classes using Kohonen's Self Organizing Map (SOM). Fig. 2.7 shows a prototype of each face class. Then the within–class and between–class scatter matrices are computed, thereby generating the optimal projection based on FLD. The density function for each sub–class is modelled as a Gaussian distribution whose parameters are estimated using maximum–likelihood. For face detection the input image is scanned with a rectangle window in which the class–dependent probability is computed. The maximum–likelihood decision rule is used to infer the presence

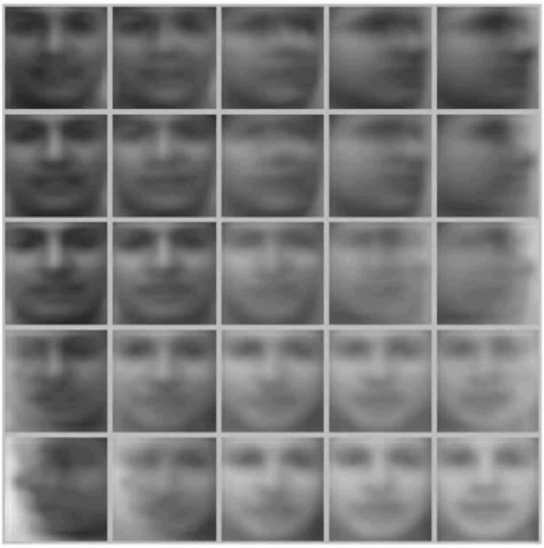

Figure 2.7: Prototype of each face class used in SOM, re–produced from Yang et al. [97].

of a face or not. 93.6% detection rate was reported on a set of 225 test images with 619 faces.

2.3.2 Neural Networks

Neural networks are popular techniques in the area of pattern recognition. Complex learning algorithms, auto–associative and compression networks, and networks evolved with genetic algorithms are all examples of the widespread use of neural networks. For face recognition this implies that neural approaches might be applied for all parts of the system, and this had indeed been shown in several papers [6, 38].

The first application of neural networks to face detection problems can be traced back as early as the work in [38] based on MLPs, where promising results were reported on fairly simple data sets. A more advanced neural approach that reported results on a large, difficult data set was by Rowley et al. [73]. They designed a retinally connected neural network that incorporates face knowledge implicitly. Fig. 2.8 depicts their system. The neural network is designed to look at windows of 20 × 20 pixels (thus 400 input units). There is one hidden layer with 26 units, where 4 units look at 10 × 10 pixel sub–regions, 16 look at 5 × 5 sub–regions, and 6 look at 20 × 5 pixels overlapping horizontal stripes. The input window is pre-processed through lighting correction (a best fit linear function is subtracted) and histogram equalisation. This pre-processing method was adopted from Sung and Poggio's system [84] mentioned earlier. One problem that arises with window scanning techniques is that of overlapping detections. Rowley et al. deal with this problem through two heuristics::

1. Thresholding: the number of detections is counted in a small neighbourhood surrounding

the current location, and a face is detected at this location if the number is higher than a pre–defined threshold.

2. Overlap elimination: when a region is classified as a face according to thresholding, then overlapping detections are likely to be false positives and thus are rejected.

For further improvements, they trained multiple neural networks and combined the output with an arbitration strategy (ANDing, ORing, voting, or a separate arbitration neural network). Later, this algorithm is applied in a person tracking system in [77] and for initial face detection in the head tracking system of La Cascia et al. [43].

Recently, Rowley et al. [74] improved this system by combining with a router neural network to detect faces at all angles in the image plane. In this system, a fully connected MLP with one hidden layer was constructed with 36 output units (one unit for each 10° angle) to decide the angle of the face. 79.6% detection rate was reported in two large data sets with a small number of false positives.

On the other hand, Feraud et al. [21] suggested a different neural approach, based on a constrained generative model (CGM). Their CGM is an auto–associative fully connected MLP with three layers of weights and 300 (15×20) input and output units (corresponding to the size of the image). The first hidden layer has 35 units, while the second hidden layer has 50 units. The idea behind this model is to force a non–linear PCA to be performed by modifying the projection of non–face examples to be close to the face examples. Classification is obtained by considering the re–construction error of the CGM (similar to PCA, explained in the previous section).

During the training phase, Feraud et al. used a training algorithm based on the bootstrap algorithm of Sung and Poggio [84] and also a similar pre–processing method. To further control the learning process, they used an additional cost function based on the minimum description

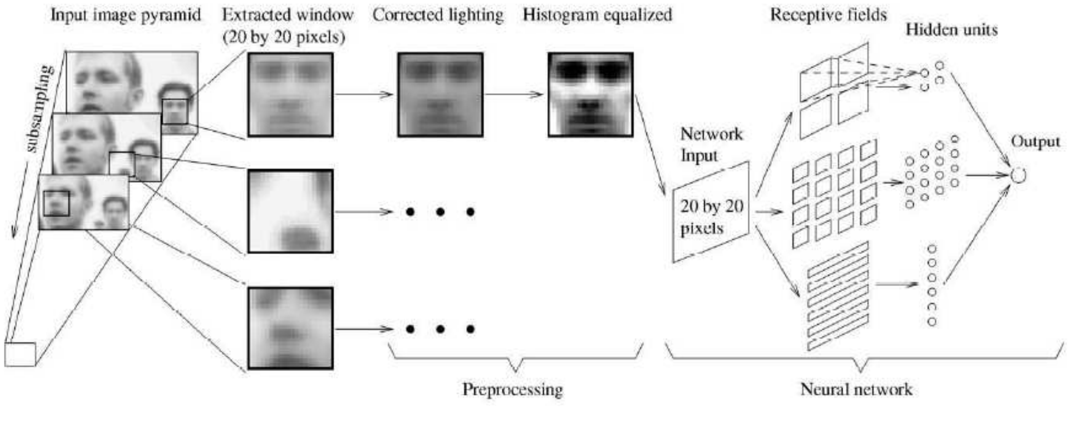

Figure 2.8: The system architecture of Rowley et al., re–produced from their work in [73].

length (MDL). More recently, they applied the system further to the problem of finding face images on the web in [22] by including colour information and multiple views.

In Lin et al. [48], a fully automatic face recognition system was proposed based on probabilistic decision–based neural networks (PDBNN), which is a classification neural network with a hierarchical modular structure. The network is similar to the DBNN [42], but it has an additional probabilistic constraint. The network consists of one subnet for each object class combined with a winner–take–all strategy. Thus, the network has only one subnet representing the face class. The learning rules are similar to DBNN, which means that the teacher only tells the correctness of the classification (as opposite to exact target values) by using a LUGS (locally unsupervised globally supervised) learning technique. With LUGS, each subnet is trained individually with an unsupervised training algorithm (K–mean and vector quantisation or the EM algorithm). The global training is performed to fine–tune decision boundaries by employing reinforced or anti–reinforced learning when a pattern in the training set is misclassified. The system used images from the MIT data set [72] but scaled them down to approximately 46 × 35, and a 12 × 12 window is used to scan through the images with 1 pixel search step.

2.3.3 Sparse Network of Winnows

In [72], Roth et al. proposed a new learning architecture called SNoW (sparse network of winnows). SNoW is a sparse network of linear functions that utilises the Winnow update rule [49]. It is specially tailored for learning in domains in which the potential number of features taking part in a decision is very large or may be unknown. In [72], SNoW is a neural network consisting of two linear threshold units (LTU) where one unit represents the face class while the other represents non–faces. The two LTUs operate on an input space of Boolean features. The system derives features from 20 × 20 input windows in the following way: for 1 × 1, 2 × 2, 4 × 4, and 10 × 10 subwindows, compute *position* × *intensity mean* × *intensity variance.* By discretising the mean and variance into a pre–defined number of classes, the Boolean feature is in a 135,424 dimensional feature space. The LTUs are separated from each other and are sparsely connected over the feature space. The system is trained with a simple learning rule that promotes and emotes eights n ases f isclassification. imilar o he reviously entioned ethods, Roth t l. 72 se he ootstrap ethod f ung nd oggio 84] or enerating raining samples nd re–process ll mages ith istogram qualisation.

2.3.4 Statistical Approach

Apart from sub–linear space and neural networks, there are also several statistical approaches to face detection, which includes systems based on information theory, support vector machines and Bayes' decision rule. In [14], Colmenarez and Huang proposed a system based on Kullback relative information (Kullback divergence), which is based on an earlier work of maximum likelihood face detection [13]. This divergence is a non–negative measure of the difference between two probability density functions P_{X^n} and M_{X^n} for a random process X^n:

$$H_{P\|M} = \sum P_{X^n} \ln \frac{P_{X^n}}{M_{X^n}} \tag{2.10}$$

During training a join–histogram is used to create probability functions for the classes of faces and non–faces for each pair of pixels in the training set. Since pixel values are highly dependent on neighbouring pixel values, X^n is treated as a first order Markov process and the pixel values in the grey–level images are quantised to four levels. Colmenarez and Huang use a large set of 11 × 11 images of faces and non–faces for training to produce a set of look–up tables with likelihood ratios. To further improve the system performance and reduce computational requirements, pairs of pixels that contribute poorly to the overall divergency are dropped from the look–up tables and not used in the face detection task. Later in [13], Colmenarez and Huang further improved on this technique by including error bootstrapping. More recently, this technique has been incorporated in a real–time face tracking system.

Support Vector Machines

In Osuna et al. [63], a system based on support vector machine (SVM) was proposed for face detection. The proposed system follows the same framework as the one developed by Sung and Poggio [72] as described before. The system is trained with a SVM with a 2^{nd}–degree polynomial as a kernel function with a decomposition algorithm which guarantees global optimality. Training is performed with the bootstrap learning algorithm, and the images are first extracted from an oval mask to eliminate pixels at the corners, and then pre–processed with lighting correction and histogram equalisation. Kumar and Poggio [41] recently incorporated Osuna et al.'s SVM algorithm in a system for real–time tracking and analysis of faces. They applied the SVM algorithm on segmented skin regions in the input images to avoid exhaustive scanning.

Bayes Classifiers

Besides SVMs, Schneiderman and Kanade [64, 65] described two face detectors based on Bayes' decision rule, which is presented as a likelihood ratio test in the following equation:

$$\frac{P(\mathit{image}|\mathit{object})}{P(\mathit{image}|\mathit{non\text{–}object})} > \frac{P(\mathit{non\text{–}object})}{P(\mathit{object})} \tag{2.11}$$

If the likelihood ratio (left side) is greater than the right side, then it is decided that an object (a face) is present at the current location. Using Bayes' decision rule is optimal if the representations for *P(image|object)* and *P(image|non–object)* are accurate. In the first proposed face detection system of Schneiderman and Kanade [78], the posterior probability function is derived based on a set of modifications and simplifications, which are listed as follows:

- The resolution of a face image is normalised to 64×64;
- Pixels of the face images are decomposed into 16×16 sub–regions and it is assumed that there is no statistical dependency among the sub–regions;
- The sub–regions are projected onto a 12–dimensional PCA subspace;
- The face subspace (constructed by PCA) of the entire face region is normalised to have zero mean and unit variance.

In the second proposed system [79], instead of representing the visual attributes of the image by local eigenvector co–efficients as in the first approach, they are represented by locally sampled wavelet transforms. A wavelet transform can capture information regarding visual attributes in space, frequency, and orientation and thus should be well suited for describing the characteristics of the human face. In their work, the wavelet transform is a three–level decomposition using a 5/3 linear phase filter–bank, which decomposes the image into 10 sub–bands. From these sub–bands, 17 visual attributes, of each consisting of 8 co–efficients, are extracted and treated as statistical independent random variables. The co–efficients are quantised to three levels and the visual attributes are represented using histograms. With this approach, a view–based detector is developed with a frontal view detector and a right profile detector (to detect left profile images, the right profile detector is applied to mirror reversed images). Between these two systems, the eigenvector system gives the best performance results, but this is due to the data set consisting of mostly frontal–view faces. In a separate experiment on a data set consisting mostly of profile–view faces, the wavelet detector outperformed the eigenvector detected (which of course had been modified to detect profile views also). Bayes' decision rule has also been applied for face detection.

Hidden Markov Models

A Hidden Markov Model (HMM) assumes that patterns can be characterised as a parametric random process, where the parameters of this process can be estimated in a precise, well–defined manner. In a pattern recognition problem using HMM, a number of hidden states need to be decided first to form the model. Then the model can learn the transitional probability between states from the training examples where each example is represented as a sequence of observations. The goal is to maximise the probability of observing the training data by adjusting the model parameters. Once the model has been trained, the output probability of an observation determines the class to which it belongs. Intuitively, a face pattern can be divided into several regions such as forehead, eyes, nose and mouth, which can be recognised if these regions are observed in appropriate orders. Therefore, a HMM–based method usually treats a face pattern as a sequence of observation vectors where each vector is a strip of pixels representing a meaningful facial region, as shown in Fig. 2.9(a). The boundaries between strips of pixels are represented by probabilistic transitions between states, as shown in Fig. 2.9(b),

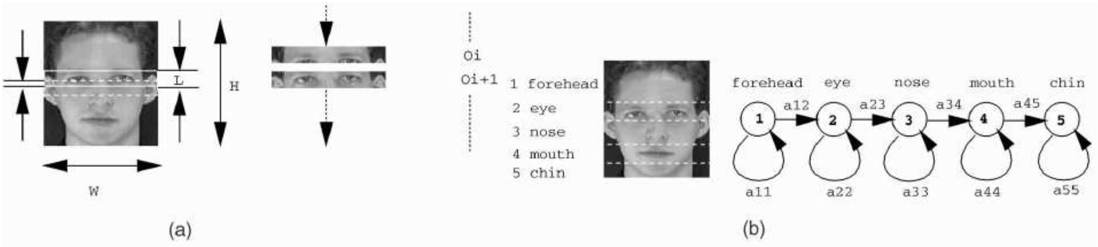

Figure 2.9: Face detection using HMM, re–produced from Samaria and Young [76].(a) Examples of observation vectors used to train an HMM model. Each face sample is divided into several facial regions. Observation vectors are constructed from a window of $W \times L$. By scanning the image vertically with P pixels of overlap, an observation sequence can be constructed. (b) An HMM with five states is trained with sequences of observation vectors.

and each region is modelled as a multi–variate Gaussian distribution. After the model has been trained, the output probability of an observation determines the class to which it belongs.

In [76], the HMM method was used for facial feature extraction and recognition. Since significant facial regions such as hair, forehead, eyes, nose, and mouth occur in the natural order from top to bottom, each of these regions is assigned to a state in a one–dimensional continuous HMM. Fig. 2.9(b) shows the five regions used in their system. During training, each image is segmented into five blocks from top to bottom as shown in Fig. 2.9(a). These blocks form the observation sequences for the image and the trained HMM determines the output class that the image belongs.

2.4 Face Descriptors in MPEG-7

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the Emmy Award winning standards known as MPEG-1, MPEG-2, and MPEG-4 standard. MPEG-7 is a standard for describing the multimedia content data that supports some degree of interpretation of the information's meaning. The aim of MPEG-7 is to provide standardised support on a range of applications across different medium [2].

The core technologies developed in MPEG-7 are the description of audio–visual data content in multimedia environments, which come in the form of MPEG-7 descriptors. The standard MPEG-7 visual description tools consist of basic structures and descriptors that cover the following basic visual features: colour, texture, shape, motion, localisation, and face recognition. Each category consists of elementary and sophisticated descriptors. Here, we will only describe the face recognition descriptor associated in the MPEG-7.

[2] More information about MPEG-7 can be found at the MPEG homepage (http://mpeg.tilab.com), the MPEG-7 Consortium website (http://www.mp7c.org), and the MPEG-7 Alliance website (http://www.mpeg-industry.com).

The face recognition descriptor defined in MPEG-7 can be used to retrieve face images that match a query face image. The descriptor represents the projection of a face vector onto a set of basis vectors which span the space of possible face vectors. The face recognition feature set is extracted from a normalised face image. This normalised face image contains 56 lines with 46 intensity values in each line. The centre of the two eyes in each face image is located on the 24th row and the 16th and 31st column for the right and left eye respectively. This normalised image is then used to extract the one dimensional face vector which consists of the luminance pixel values from the normalised face image arranged into a one dimensional vector using a raster scan starting at the top-left corner of the image and finishing at the bottom-right corner of the image. The face recognition feature set is then calculated by projecting the one dimensional face vector onto the space defined by a set of basis vectors.

2.5 Face Image Database

Although many face detection methods have been proposed, less attention has been paid to the development of an image database for face detection research. Table 2.1 summarised some common face image databases used in face detection.

2.6 Conclusion

In the previous sections, an extensive review of feature–based, knowledge–based and image–based algorithms for face detection has been presented, together with a brief presentation of some of the application areas. The following is a concise summary with conclusions representing the main issues in this chapter.

• Face detection is currently a very active research area and the technology has come a long way since the survey of Chellappa et al. [10]. The last couple of years have shown great ad-vances in algorithms dealing with complex environments such as low quality grey–scale images and cluttered backgrounds. Although some of the best algorithms are still too computation-ally expensive to be applicable for real–time processing, this is likely to change with coming improvements in computer hardware.

• Feature–based methods are applicable for real–time systems where colour and motion is available. Since an exhaustive multi–resolution window scanning is not always preferable, feature–based methods can provide visual cues to focus attention. In these situations, the most widely used technique is skin colour detection. Out of the feature–based approaches which perform on grey scale static images, Maio and Maltoni's [52] algorithm seems very promising, showing good detection results while still being computationally efficient.

Data Set	Description	Web location
AT&T Database	400 images of 40 subjects, 10 images per subject.	http://www.uk.research.att.com
CMU Database	130 images with 507 labelled faces. The images are grey in colour and of varying size and quality.	http://vasc.ri.cmu.edu/idb/html/face/
FERET Database	Collection of male and female faces. Each image contains a single person with certain expression in uncluttered background.	http://www.nist.gov/humanid/feret
Harvard Database	Cropped, masked face images under various illumination conditions.	ftp://ftp.hrl.harvard.edu/pub/faces/
M2VTS Database	A multi–modal database containing various image sequences.	http://poseidon.csd.auth.gr/M2VTS/index.html
MIT Database	Faces of 16 people, 27 images of each person under various lighting conditions, scale and head orientation.	ftp://whitechapel.media.mit.edu/pub/images/
MIT CBCL	Collection of 6,977 training images (2,429 faces and 4,548 non–faces) and 24,045 test images (472 faces and 23,573 non–faces)	http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html
Yale Database	Face images with glasses, different expressions, and under various lighting conditions.	http://cvc.yale.edu
UMIST Database	564 images of 20 subjects where each subject covers a range of poses from profile to frontal views.	http://images.ee.umist.ac.uk/danny/database.html
University of Bern Database	300 frontal face images of 30 people and 150 profile images.	ftp://iamftp.unibe.ch/pub/Images/FaceImages/

Table 2.1: Face Image Database

• Knowledge–based methods use explicit rules that describe the features of a human face and their relationships when prior knowledge of human faces is available. By using these rules, face candidates can be identified and this usually follows with a verification process to reduce false detection. The main challenge with this approach is the difficulty in translating human knowledge into well–defined rules without being too restrictive or too general. Experience from the literature suggested that this approach works well in detecting frontal faces in uncluttered scenes.

• Image–based approaches are the most robust techniques for processing grey–scale static images. Sung and Poggio [84] and Rowley et al. [73] set the standards for research on this topic, and the performances of their algorithms are still comparable to more recent image–based approaches. Since all these algorithms are based on multi–resolution window scanning to detect faces at all scales, this makes them computationally expensive. Multi–resolution window scanning can be avoided by combining the image–based approach with a feature–based method as a pre–processor with the purpose of guiding the search based on visual clues such as skin colour.

• It is not easy to evaluate and compare current algorithms. Since there are no standard evaluation procedures or agreements on the number of faces in the data set, it is hard to draw any conclusions. There is a need for an evaluation procedure for face detection algorithms similar to the FERET [66] test for face recognition.

- The human face is a dynamic object but with a standard configuration of facial features which can vary within a limited range. It is a difficult problem to detect such dynamic objects and considering the changes in faces over time (facial hair, glasses, wrinkles, skin colour, bruises) together with variations in pose, developing a robust face detection algorithm is still a hard problem to solve in computer vision systems.

Chapter 3

Facial Feature Extraction

The aim of this chapter is two–fold. First, to compare and evaluate different skin colour models and thereby to estimate the properties of human skin colours. In doing so, we can effectively detect skin regions in colour images and hence reduce the searching space for possible face candidates. The second objective is to develop a facial feature extraction method to capture the unique characteristics of different face representations. In particular, the principal component analysis (PCA) and independent component analysis (ICA) models are introduced and compared from a statistical point of view. The aim is to construct a subspace that models the distribution of human faces. Both skin information and subspace facial representation will then be used in the next chapter to develop an algorithm to detect faces in colour images.

3.1 Skin Colour Segmentation

The objective of skin modelling is to find a decision rule that could discriminate between skin and non–skin pixels. Although pixel–based skin detection has long history, only few papers had provided comparisons of different techniques being published. The main reason is that there is a lack of a standard skin pixel database for comparison and evaluation, and most authors use their own collected data sets. In this section, we describe several published skin modelling techniques, try to find out their characteristic features and compare their performances using the same set of data samples.

In general, these techniques can be categorised into non–parametric modelling and parametric modelling. The idea of non–parametric modelling methods is to estimate a skin colour distribution from training samples without deriving an explicit model or use some heuristic rules to discriminate skin pixel from non–skin pixel. The advantages of non–parametric methods are that they are fast in training and less dependent on the shape of distribution of the training sample. However, they need more storage space and are unable to generalise the training data. Explicit thresholding and skin probability maps are examples of this approach. On the other hand, the parametric modelling methods try to model the skin distribution explicitly. They are the counterpart of the non–parametric methods. These methods need more training time and are more dependent on the training sample. The advantages are that they need much less storage space since only the model parameters are needed for estimation and can be fine–tuned to a particular application. However, since they depend more on the shape of distribution, and the colour space used, the performance varies significantly from one colour space to the other. Single Gaussian modelling and mixtures of Gaussian are the most popular methods for the parametric approach.

3.1.1 Explicit Thresholding

The easiest way to build a skin classifier is to use explicit rules based on prior knowledge of human skin. The advantage of this method is its simplicity and speed. Following the work of Peer et al. [64], we classify a pixel as skin pixel in RGB colour space if it satisfies the following conditions:

$$\begin{aligned} &R > 95,\ G > 40,\ B > 20,\ |R - G| > 15,\ R > G, \\ &R > B,\ \mathit{and}\ \max\{R, G, B\} - \min\{R, G, B\} > 15 \end{aligned} \tag{3.1}$$

3.1.2 Bayes Skin Probability Map

For Bayes skin probability map (SPM), we compute the probability *P(skin|c)* of observing a skin pixel given a concrete colour value c based on a conditional probability *P(c|skin)* – a probability of observing colour c, knowing that it is a skin pixel. By using the Bayes rule, we can compute this probability as:

$$P(skin|c) = \frac{P(c|skin)P(skin)}{P(c|skin)P(skin) + P(c|\neg skin)P(\neg skin)}. \tag{3.2}$$

The *P(c|skin)* and *P(c|¬skin)* are directly computed from the sample population, whilst the prior probabilities *P(skin)* and *P(¬skin)* are estimated from the overall number of skin and non–skin samples in the training set. From the above equation, the ratio of *P(skin|c)* and *P(¬skin|c)* can be written as:

$$\frac{P(skin|c)}{P(\neg skin|c)} = \frac{P(c|skin)P(skin)}{P(c|\neg skin)P(\neg skin)}. \tag{3.3}$$

Comparing this ratio to a threshold Θ gives the skin/non–skin decision rule, which after some manipulations can be re–written as:

$$\frac{P(c|skin)}{P(c|\neg skin)} > \Theta. \tag{3.4}$$

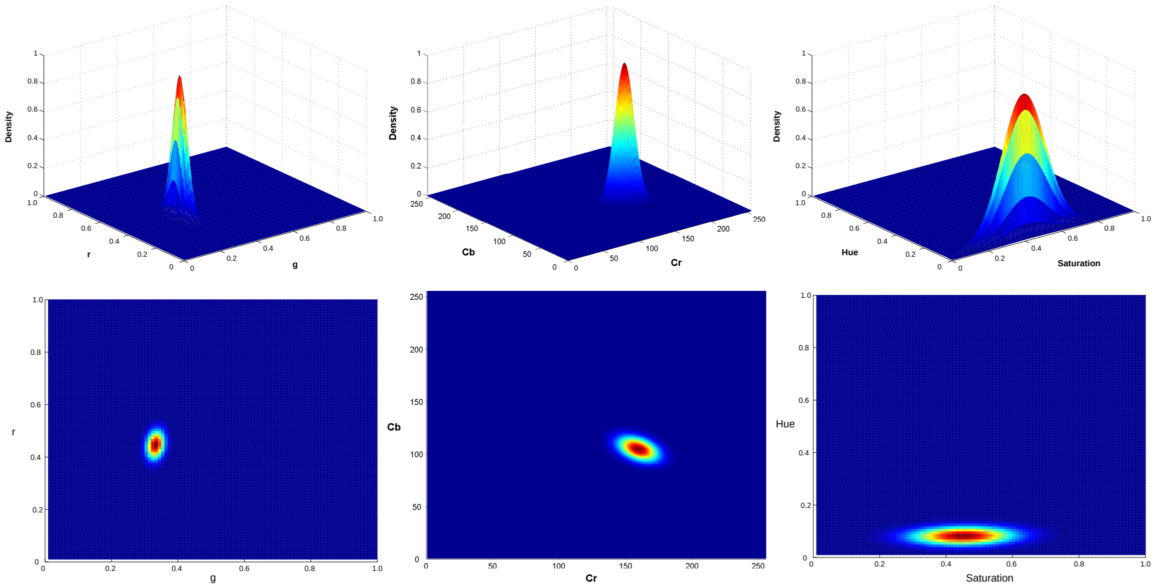

Figure 3.1: Estimated probability density of the skin colour using single Gaussian distribution in rgb, YCrCb and HSV colour space (from left to right).

3.1.3 Gaussian Modelling

For a single Gaussian, the skin colour distribution can be modelled by a Gaussian joint probability density function (pdf) which is defined as

$$P(\mathbf{c}|skin) = \frac{1}{\sqrt{2\pi}|\Sigma_s|^{1/2}} e^{-\frac{1}{2}(\mathbf{c}-\mu_s)^T \Sigma_s^{-1}(\mathbf{c}-\mu_s)}, \tag{3.5}$$

where $\mathbf{c}$ is a colour vector and μ_s and Σ_s are the mean vector and covariance matrix respectively. The probability $P(\mathbf{c}|skin)$ can be used directly as a measure of how likely a colour belongs to skin colour.

Fig. 3.1 depicts the estimated skin colour distribution modelled by single Gaussian in rgb, YCbCr and HSV colour spaces. rgb is a representation of normalised RGB which is defined in Eq.(2.2). As the sum of normalised components is known (r+g+b=1), the third component does not hold any significant information and can be ignored. The first insight of the result is that the skin colour distribution occupies a tiny space in rgb and YCbCr colour space. However, in HSV, the distribution has a big variance in Saturation values within a small range of Hue values. Since the Hue value represents the nature of colour while the Saturation value measures how strong the colour is, this suggests that human skin colour only occupies a tiny portion of the natural colour but because of the different lighting conditions of images, it may cause a wide variance in the whole colour space.

Although skin colour distribution can be modelled by single Gaussian, it may not be enough for more complex skin distributions. In this case, the Gaussian mixture model (GMM) can be used in which the pdf is given by:

$$P(\mathbf{c}|skin) = \sum_{i=1}^{k} \pi_i P_i(\mathbf{c}|skin) \tag{3.6}$$

where k is the number of mixture components, π_i is the mixing parameters and $\pi_i P_i(\mathbf{c}|skin)$ is each single Gaussian in the mixture model. The model parameters can be estimated by using the Expectation Maximization (EM) algorithm.

The EM algorithm consists of two steps: the expectation step (E–Step) and the maximization step (M–Step). Given n as the total number of feature points, $\mathbf{x}_i$ representing one of them, m being the number of clusters and d the dimensionality of the feature space, the E–Step is expressed as follows:

$$f(\mathbf{x}_i|\Phi) = \sum_{j=1}^{m} \pi_j f_j(\mathbf{x}_i|\theta_j) \tag{3.7}$$

$$f_j(\mathbf{x}_i|\theta_j) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_i-\mu_j)^T \Sigma_j^{-1}(\mathbf{x}_i-\mu_j)} \tag{3.8}$$

where $\theta_j = (\mu_j, \Sigma_j)$ is the set of parameters for the probability density function $f_j(\mathbf{x}_i|\theta_j)$ with μ_j, Σ_j and π_j as the mean, the covariance matrix and the mixing proportion of cluster j, respectively, subject to the conditions that $\pi_j > 0$ and $\sum_{j=1}^{m} \pi_j = 1$. $\Phi = (\pi_1, \ldots \pi_m, \theta_1, \ldots \theta_m)$ is the set of all model parameters, and $f(\mathbf{x}_i|\Phi)$ is the probability density function of the observed data point $\mathbf{x}_i$ given parameters Φ.

In the M–Step the algorithm iterates to re–estimate the model parameters that maximise the log–likelihood $\log f(\mathbf{X}|\Phi)$ using the updated equations:

$$E[z_{ij}] = p(z_{ij} = 1|\mathbf{X}, \Phi^{(t)}) = \frac{\pi_j^{(t)} p_j(\mathbf{x}_i|\Phi_j^{(t)})}{\sum_{s=1}^{m} p_s(\mathbf{x}_i|\Phi^{(t)})\pi_s^{(t)}} \tag{3.9}$$

$$\pi_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} E[z_{ij}] \tag{3.10}$$

$$\mu_j^{(t+1)} = \frac{1}{n\pi_j^{(t+1)}}\sum_{i=1}^{n} E[z_{ij}]\mathbf{x}_i \tag{3.11}$$

$$\Sigma_j^{(t+1)} = \frac{1}{n\pi_j^{(t+1)}}\sum_{i=1}^{n} E[z_{ij}](\mathbf{x}_i - \mu_j^{(t+1)})(\mathbf{x}_i - \mu_j^{(t+1)})^T \tag{3.12}$$

where $E[z_{ij}]$ is the expected value of the probability that data point i belongs to cluster j. At each iteration step, $\log f(\mathbf{X}|\Phi)$ is maximised until all the parameters are converged or a pre–defined step size is reached.

There are two ways to initialise the model parameters: the first is to initialise the means randomly and set the covariance matrix to identity matrix whilst the other is to use K–Means clustering to estimate the initial means for each component. Experiments suggest that both techniques generate similar results if good candidates are selected as starting points, but the second approach gives faster convergence of the data. Thus, the second approach is adopted here. One practical problem in using EM algorithm is the selection of model order. If a high model order is used, the model parameters estimated may over–fit the training data. On the other hand, if a small model order is used, the parameters may not be estimated correctly. In our experiments we used the principle of Minimum Description Length (MDL) as described in [91] to select the model order automatically. The MDL is a heuristic method in the sense that it does not minimise an error function between the estimated and the true model order but instead it defines various information criteria that only depend on the unknown model order. One of the most popular MDL criteria, the information criterion of Rissanen, is defined as:

$$\mathrm{MDL}(K) = -\ln[L(\mathbf{X}|\Phi)] + \frac{1}{2}M\ln(n). \tag{3.13}$$

The first term $-\ln[L(\mathbf{X}|\Phi)]$, the maximised mixture likelihood of $P(\mathbf{X}|\Phi)$, measures the system entropy which can be seen as a measure for the number of bits needed to encode the observations $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n]$ with respect to the model parameters Φ

$$P(\mathbf{X}|\Phi) = \prod_{i=1}^{n} f(\mathbf{x}_i|\Phi). \tag{3.14}$$

The second term $\frac{1}{2}M\ln(n)$ measures the additional number of bits needed to encode the model parameters and serves as a penalty for models that are too complex. M describes the number of free parameters and is given by $M = 2dK + (K-1)$ where K is the selected model order (number of clusters) and d is the dimension of the feature space.

Thus the optimal number of model order is determined by the following iterative procedure: first, the maximum likelihood of the Gaussian mixtures is computed using K–Means [1] for model initialisation and EM for parameter estimation with model order 2 to 4. Then the value of MDL is calculated using Eq.(3.13) for each model order and the model parameters are selected for the minimum value of MDL.

By using the MDL principle, the best model order selected in GMM for both rgb and YCbCr colour space is 2. However, in YCbCr colour space, we found that weighting of the second mixture is so insignificant that it can be almost neglected, in which way it means a single Gaussian model can well represent the skin distribution in YCbCr colour space. On the other hand, the best model order selected in HSV colour space is 3. As from Fig. 3.2, we can see that the middle mixture captures the most skin distribution while two other less weighted mixtures model the rest of the distribution.

To evaluate the performance of different skin models, two millions skin pixels and one million non–skin pixels are collected to train the models. For model testing, 560,000 skin pixels and 1,215,000 non–skin pixels are used and the results are measured in terms of true positive (TP) and false positive (FP), which are summarised in Table 3.1. To our surprise, the non–

[1] The K in K–Means is the number of clusters, so it is same as K in the document.

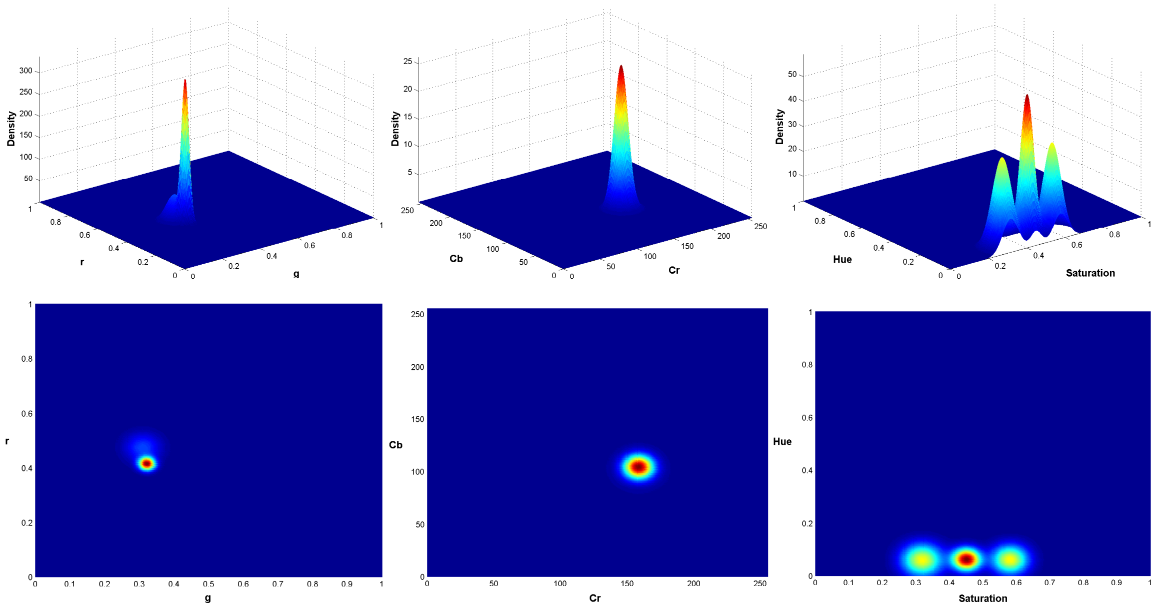

Figure 3.2: Estimated probability density of the skin colour using Gaussian mixture models in rgb, YCrCb and HSV colour space (from left to right).

parametric methods perform slightly better than parametric methods. This is possibly because of the presence of outliers, usually due to the variations of different lighting conditions, in the training set which give noises during the parameter estimation. Overall, none of the models outperformed compared with the rests. We also found that the skin models in HSV colour space give a higher false positive rate than the others. From Fig. 3.1 and Fig. 3.2, we have pointed out that the skin distribution in HSV occupies a much bigger area portion than the other colour spaces, and hence it has a bigger overlapping area with the non–skin colour. The conclusion from the experimental results is that the performance of skin modelling mainly depends on the distribution of skin samples in each corresponding colour space, and hence the tightness of the cluster and the overlapping with non–skin samples are the main criteria. The complexity of the skin model is insignificant in terms of detection accuracy.

Method	True Positive (TP)	False Positive (FP)
Empirically Thresholding in RGB	91.6%	15.55%
Bayes SPM in RGB	87.83%	9.6%
Single Gaussian in rgb	90.22%	24.91%
Single Gaussian in CbCr	91.16%	19.75%
Single Gaussian in HSV	87.65%	30.5%
Gaussian Mixture in rgb	90.12%	22.84%
Gaussian Mixture in CbCr	89.06%	23.33%
Gaussian Mixture in HSV	84.23%	24.11%

Table 3.1: Performance of different skin detection models

3.2 Face Patterns in Subspace Modelling

In this section, we describe two subspace models – principal component analysis (PCA) and independent component analysis (ICA) – to capture the distribution of face patterns.

3.2.1 Principal Component Analysis

Given a random vector $\mathbf{x}$ with d elements from a N sample of data, PCA is a statistical technique that frequently reduces the number of dimension from d to t where $t \leqslant d$ in the feature space of $\mathbf{x}$ under which the retained variance is a maximum.

Consider a linear model as follows with:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{W}\mathbf{b} \tag{3.15}$$

where $\mathbf{x}$ and $\mathbf{b}$ are $d \times 1$ column vectors, $\mathbf{W}$ is a $d \times d$ matrix, and $\bar{\mathbf{x}}$ is the mean vector, then the n samples of observed data set can be written as:

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{W}\mathbf{b}_i, \quad i = 1, 2, \ldots, n \tag{3.16}$$

This model is, however, unidentifiable in the sense that there is an infinite number of equally good solutions since the matrix $\mathbf{W}$ and the vector b can both be transformed by inserting any non–singular matrix $\mathbf{M}$ and its inverse $\mathbf{M}^{-1}$ on the right–hand side as:

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{W}\mathbf{M}\mathbf{M}^{-1}\mathbf{b} \\ \Rightarrow \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{W}'\mathbf{b}' \end{aligned} \tag{3.17}$$

To make the model well determined, the following constraints must be made:

1. Pairwise uncorrelated: $E\{\mathbf{b}_i\mathbf{b}_j\} = 0,\ i < j,$ for $i, j \in [1, 2, \ldots, d]$
2. Normalisation: $\mathbf{w}_i^T\mathbf{w}_i = 1$, for $i = 1, 2, \ldots, d$
3. Orthogonality: $\mathbf{w}_i^T\mathbf{w}_j = 0$ for $i, j \in [1, 2, \ldots, d]$ and $i \neq j$
4. The elements of $\mathbf{W}$ are real and positive
5. The entries of $\mathbf{W}$ are sorted in order

These constraints imply that $\mathbf{W}$ is a square symmetry matrix and we can now re–write Eq.(3.16) to:

$$\mathbf{b}_i = \mathbf{W}^T(\mathbf{x}_i - \bar{\mathbf{x}}) \tag{3.18}$$

Therefore, we are looking for a weight vector that maximise the criteria represented by the elements of the diagonal matrix:

$$\begin{aligned}
\mathbf{D} &= E\{\mathbf{b}_i\mathbf{b}_i^T\} \\
&= E\{\mathbf{W}^T(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{W}\} \\
&= \mathbf{W}^T E\{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\}\mathbf{W} \\
&= \mathbf{W}^T\mathbf{C}_\mathbf{x}\mathbf{W} \\
\Rightarrow \mathbf{W}\mathbf{D} &= \mathbf{C}_\mathbf{x}\mathbf{W} \\
\Rightarrow \sigma_i^2\mathbf{w}_i &= \mathbf{C}_\mathbf{x}\mathbf{w}_i, \ \ i = 1, 2, \ldots, d \qquad (3.19)
\end{aligned}$$

where $\mathbf{C}_\mathbf{x}$ is the covariance matrix. This is the well–known eigenproblem when the normalisation constraint is imposed. The solution is given in terms of eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d$ of the matrix $\mathbf{C}_\mathbf{x}$ ordered according to their corresponding eigenvalues σ_i^2.

By choosing the first t orthogonal vectors $\mathbf{w}_i$, for $i = 1, 2, \ldots, t$, the vector $\mathbf{b}_i = \mathbf{W}^T(\mathbf{x}_i - \bar{\mathbf{x}})$, where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t)$, is thus a t–dimensional reduced representation of the observed vector $\mathbf{x}_i$. The projection onto this principal subspace minimises the squared re–construction error $\sum \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$. The optimal linear re–construction of $\mathbf{x}_i$ is given by $\hat{\mathbf{x}}_i = \bar{\mathbf{x}} + \sum_{i=1}^t \mathbf{w}_i\mathbf{b}_i$.

3.2.2 Independent Component Analysis

Independent component analysis is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multi–variate data in which the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed to be non–Gaussian and mutually independent, and hence they are called as the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA. ICA is superficially related to principal component analysis and factor analysis. It is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail. In recent years, there has been great research interest in ICA spanning different application areas. The world leading research group on ICA is the department of computer science in University of Helsinki [2].

In fact, PCA can be derived as a special case of ICA using Gaussian source models. As was shown from the previous section, PCA chooses an orthogonal matrix which allows optimal linear re–construction of the input in the sense of minimising the mean square error, and the re–constructed data are uncorrelated. However, independence is a stronger concept, in the sense

[2] More ICA related information and the research work of University of Helsinki can be found at http://www.cs.helsinki.fi/u/ahyvarin/whatisica.shtml.

of statistics, than uncorrelatedness. Since ICA is based on this stronger concept, consequently, it is expected to be better than PCA. In the following sections, we will prove that independence is a stronger property than uncorrelatedness before going into details of ICA model.

3.2.3 Statistical Independence and Uncorrelatedness

Given a random vector $\mathbf{x} \in \mathbb{R}^n$ from a χ distributed data set, the correlation r_{ij} between the ith and jth component of $\mathbf{x}$ is contained in the second moment [61] as:

$$\begin{aligned} r_{ij} &= E\{x_i x_j\} \\ &= \int_{-\infty}^{\infty} x_i x_j p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_i x_j p_{x_i x_j}(x_i, x_j) dx_i dx_j \end{aligned} \qquad (3.20)$$

$p_{\mathbf{x}}(\mathbf{x})$ is the probability density. Hence, the correlation matrix is $\mathbf{R_x} = E\{\mathbf{x}\mathbf{x}^T\}$. Since the covariance matrix is $\mathbf{C_x} = E\{(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\}$ and by substituting this back to the equation, we can be easily see that $\mathbf{R_x} = \mathbf{C_x} + \bar{\mathbf{x}}\bar{\mathbf{x}}^T$.

Two random variables $\mathbf{x}$ and $\mathbf{y}$ are said to be uncorrelated if their cross–variance or covariance matrix is 0:

$$\mathbf{C_{xy}} = E\{(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{y}-\bar{\mathbf{y}})^T\} = 0. \qquad (3.21)$$

If $\mathbf{x}$ and $\mathbf{y}$ are uncorrelated,

$$\begin{aligned} \mathbf{R_{xy}} &= \mathbf{C_{xy}} + \bar{\mathbf{x}}\bar{\mathbf{y}}^T, \\ \mathbf{C_{xy}} &= 0, \\ E\{\mathbf{x}\mathbf{y}^T\} &= R_{\mathbf{xy}} = \bar{\mathbf{x}}\bar{\mathbf{y}}^T = E\{\mathbf{x}\}E\{\mathbf{y}\}^T. \end{aligned} \qquad (3.22)$$

However, when the two random variables are taken from the same data sample, this uncorrelatedness condition cannot be satisfied due to the high correlation between each component and itself. In this situation, the condition now is that different components of x are mutually uncorrelated, which means the covariance matrix is:

$$\mathbf{C_x} = E\{(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\} = \mathbf{D} \qquad (3.23)$$

$\mathbf{D}$ is a diagonal matrix for which the diagonal elements are variances $\sigma_{x_i}^2 = E\{(x_i - \bar{x}_i)^2\}$ and with zero elements off the diagonal. Each pair–wise components of $\mathbf{x}$ is uncorrelated in the way that:

$$c_{x_i x_j} = E\{(x_i - \bar{x}_i)(x_i - \bar{x}_j)^T\} = 0, \text{ for } i \neq j. \qquad (3.24)$$

On the other hand, two random variables x and y are said to be independent if and only if

$$p_{x,y}(\mathrm{x,y}) = p_x(\mathrm{x})p_y(\mathrm{y}). \tag{3.25}$$

If we let $f(x)$ and $g(y)$ be two integrable functions, then the two independent random variables x and y will hold the following property:

$$\begin{aligned} E\{f(x)g(y)\} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)g(y)p_{x,y}(x,y)dxdy \\ &= \int_{-\infty}^{\infty} f(x)p_x(x)dx \int_{-\infty}^{\infty} g(y)p_y(y)dy \\ &= E\{f(x)\}E\{g(y)\} \end{aligned} \tag{3.26}$$

And it can be easily seen that, if we let $f(x) = x_i$ and $g(y) = x_j$ for $i \neq j$, this implies that

$$E\{x_i x_j\} = E\{x_i\}E\{x_j\} \tag{3.27}$$

Eqs.(3.25) and (3.27) imply that variables x_i and x_j are uncorrelated. Hence, we can see that if the components of $\mathbf{x}$ is mutually independent they are also mutually uncorrelated.

In data analysis, the distributions of all data sets could be categorised as either a Gaussian (normal distribution) or as non–Gaussian. According to the sign of their fourth order cumulant statistics, also known as kurtosis, the non–Gaussian distributions can be further divided into sub–Gaussian distribution (*platykurtic*) and super–Gaussian distribution (*leptokurtic*). Kurtosis is the fourth–order cumulant of a data set, which is defined as:

$$\begin{aligned} kurt(x) = \kappa_4 &= E\{x^4\} - 3[E\{x^2\}]^2 \\ &= \mu_4 - 3\mu_2^2 \end{aligned} \tag{3.28}$$

and the normalised kurtosis is defined as:

$$\bar{\kappa} = \frac{\mu_4}{\mu_2^2} - 3 \tag{3.29}$$

where μ_i stands for the i–th moment of the data set. A distribution which has zero kurtosis is called *mesokurtic*, or a Gaussian distribution. If the kurtosis is less than zero, it is said to be a sub–Gaussian distribution. If the kurtosis is greater than zero, it is said to be a super–Gaussian distribution.

Gaussian Distribution

Given an n–dimensional random vector $\mathbf{x}$ from a Gaussian distribution, its probability density function (PDF) is defined as:

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C_x}|^{n/2}} \exp(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{C}_{\mathbf{x}}^{-1}(\mathbf{x}-\bar{\mathbf{x}})). \tag{3.30}$$

If we assume that the components of $\mathbf{x}$ are mutually uncorrelated, Eq.(3.23) implies that $\mathbf{C_x} = diag(\sigma_{x_1}^2, \sigma_{x_2}^2, \ldots, \sigma_{x_n}^2)$, so:

$$\begin{aligned}
& \mathbf{C}_{\mathbf{x}}^{-1} = diag(\frac{1}{\sigma_{x_1}^2}, \frac{1}{\sigma_{x_2}^2}, \ldots, \frac{1}{\sigma_{x_n}^2}) \\
\Rightarrow \quad & (\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{C}_{\mathbf{x}}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) \\
= \quad & \frac{(x_1-\bar{x}_1)^2}{\sigma_{x_1}^2} + \frac{(x_2-\bar{x}_2)^2}{\sigma_{x_2}^2} + \ldots + \frac{(x_n-\bar{x}_n)^2}{\sigma_{x_n}^2} \\
\Rightarrow \quad & p_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{n} p_{x_i}(x_i). \qquad (3.31)
\end{aligned}$$

Hence we can see that uncorrelated Gaussian distribution also implies independence.

Non–Gaussian Distribution

Unlike Gaussian distributions, there is not such standard form for non–Gaussian distributions and it may not be possible to prove directly whether uncorrelatedness implies independence for non–Gaussian distributions. But we may assume that uncorrelated non–Gaussian distributions also lead to independence and if we could find any one example that contradict this assumption, we would disprove this hypothesis.

A widely used super–Gaussian distribution is the Laplace or the so–called double exponential distribution, which is defined as:

$$p_x(x) = \frac{1}{2b}e^{-\frac{|x-\mu|}{b}} \tag{3.32}$$

where b is a real number and μ is the mean of $p_x(x)$. The mean, variance and normalised kurtosis of the Laplace distribution are:

$$\begin{aligned}
\bar{x} &= \mu \\
\sigma_x^2 &= 2b^2 \\
\bar{\kappa}_x &= 3. \qquad (3.33)
\end{aligned}$$

Given an n–dimensional random vector $\mathbf{x}$ from the Laplace distribution with elements x_i, for

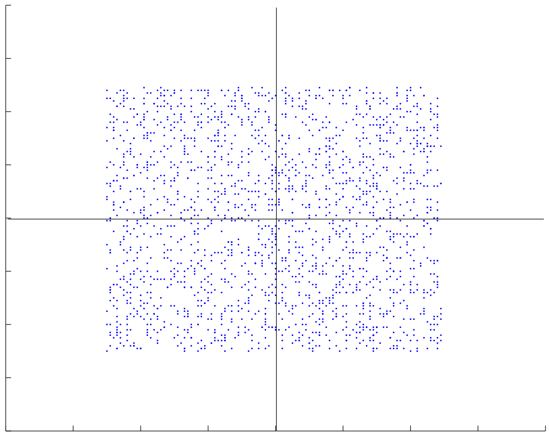

Figure 3.3: Uniform distribution.

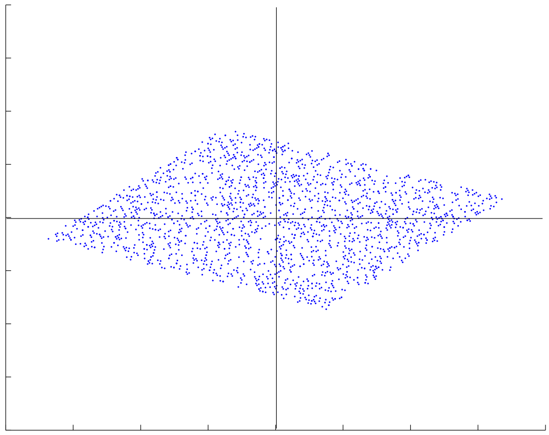

Figure 3.4: Rotated uniform distribution.

$i = 1, 2, \ldots, n$, its probability density function (PDF) is

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{2\|\mathbf{b}\|} e^{-\frac{\|\mathbf{x}-\bar{\mathbf{x}}\|}{\|\mathbf{b}\|}}. \tag{3.34}$$

If the elements of vector $\mathbf{x}$ are mutually uncorrelated, then again apply Eq.(3.23) we will see that uncorrelatedness does not imply independence for Laplace distribution, since

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x}) &= \frac{1}{2\sqrt{(b_1^2 + b_2^2 + \ldots + b_n^2)^2}} \exp(-\frac{\sqrt{(x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2 + \ldots + (x_n - \bar{x}_n)^2}}{\sqrt{b_1^2 + b_2^2 + \ldots + b_n^2}}) \\ &\neq \frac{1}{2^n b_1 b_2 \ldots b_n} \exp(-(\frac{|x_1 - \bar{x}_1|}{b_1} + \frac{|x_2 - \bar{x}_2|}{b_2} + \ldots + \frac{|x_n - \bar{x}_n|}{b_n})) \\ &= p_{x_1}(x_1) p_{x_2}(x_2) \ldots p_{x_n}(x_n). \end{aligned} \tag{3.35}$$

Now let us consider an often used sub–Gaussian distribution. Fig. 3.3 shows a set of random data generated from Gaussian distribution and are whitened beforehand, which means $\bar{\mathbf{x}} = 0$ and $\mathbf{C_x} = \mathbf{I}$. Now consider a linear transformation of the data, as depicted in Fig. 3.4, and let the rotation matrix as $\mathbf{R}$, thus $\mathbf{x}' = \mathbf{Rx}$. Following Eq. 3.23, the covariance matrix of the transformed data **x**' is:

$$\begin{aligned}
\mathbf{C}_{\mathbf{x}'} &= E\{(\mathbf{x}' - \bar{\mathbf{x}})(\mathbf{x}' - \bar{\mathbf{x}})^T\} \\
&= E\{(\mathbf{x}')(\mathbf{x}')^T\} \\
&= \mathbf{R}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{R}^T \\
&= \mathbf{RIR^T} \\
&= \mathbf{I} \qquad (3.36)
\end{aligned}$$

The new data is uncorrelated as well. From Fig. 3.4 we can easily see that the value of x_2 is at the extreme values of x_1. Hence, the new data is uncorrelated but it is not independent.

To conclude, we have shown that if the elements of a random vector **x** are mutually independent, it implies that the elements are mutually pair–wise uncorrelated as well. However, it is not always true the other way round; i.e.: if the elements of a random vector **x** are uncorrelated, they may not necessarily be statistically independent. Therefore, independence is a stronger property than uncorrelatedness in statistics.

3.2.4 ICA Model

Assuming that we observe n linear mixtures $x_1, \ldots, x_n$ of n independent components, we can define ICA by using a statistical "latent variables" model as follows:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \cdots + a_{jn}s_n, \quad \text{for all } j. \qquad (3.37)$$

where x_j is a observed random mixture variable and s_k is each independent component. Usually it is assumed that both the mixture variables and the independent components have zero mean. If this is not true, then the observed variables x_i can always be centered by subtracting the sample mean, which makes the model zero-mean. If we denote **x** and **s** as the random vectors whose elements are $x_i, \ldots, x_n$ and $s_1, \ldots, s_n$ respectively and likewise by **A** the matrix with elements a_{ij}, by using this vector–matrix notation, the above mixing model can be written as:

$$\mathbf{x} = \mathbf{As} = \sum_{i=1}^{n} \mathbf{a}_i s_i. \qquad (3.38)$$

The statistical model in Eq.(3.38) is known as the independent component analysis, or ICA model. The ICA model is a generative model that describes how the observed data are generated by a process of mixing the components s_i which are assumed to be statistically independent as defined in the previous section. These independent components are latent variables in a way that they cannot be directly observed. Both the mixing matrix $\mathbf{A}$ and the latent variables $\mathbf{s}$ must be estimated from the observed data $\mathbf{x}$. Once the mixing matrix $\mathbf{A}$ is estimated, then we can compute its inverse, $\mathbf{W}$, and obtain the independent components simply by:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{3.39}$$

3.3 ICA and PCA

Following our discussions in the previous sections, we can see that PCA can be derived as a special case of ICA. PCA is based on the assumption of Gaussian source model while ICA is based on non–Gaussian distribution. When the sources are Gaussian, the likelihood of the data depends only on the first– and second–order statistics. For images, Oppenheim and Lim [62] have shown that second–order statistics capture the amplitude spectrum of images while the higher–order statistics capture the phase spectrum. It was also illustrated in [80] that the phase spectrum contains the structural information in images which derives human perception but not the power spectrum. Since PCA is only sensitive to the power spectrum, it might not be particularly well–suited for representing natural images.

PCA may nevertheless be carried out on non–Gaussian distributions but there will be usually dependencies remaining in the results. In addition, Bell and Sejnowski [3] have empirically observed that many natural signals are better described as linear combinations of sources with long tailed distributions, which are known as "super–Gaussian" sources.

To sum up, ICA has the following potential advantages over PCA:

1. It is based on a stronger statistical concept than PCA;
2. It provides a better probabilistic model of the data that can better identify when the data concentrate in n–dimensional space;
3. It uniquely identifies the mixing matrix $\mathbf{W}$;
4. It can find a non–orthogonal basis which may re–construct the data better than PCA in the presence of noise;
5. It is sensitive to higher–order statistics other than just the second–order statistics and, hence, it potentially captures the structural information of the source.

3.4 Performing ICA on Face Images

Let $\mathbf{X}$ be a data matrix with n_r rows and n_c columns where each column represents a face image. We may think of each column of $\mathbf{X}$ as independent trials of a random experiment. In this sense we think of the ith row of $\mathbf{X}$ as the specific value (pixels) taken by a random variable $\mathbf{X}_i$ (a face image) across n_c independent trials. Therefore, pixels are random variables and images are trials. The goal is to find a good set of basis images to represent a database of faces. Fig. 3.5 shows the idea of this ICA architecture for face images.

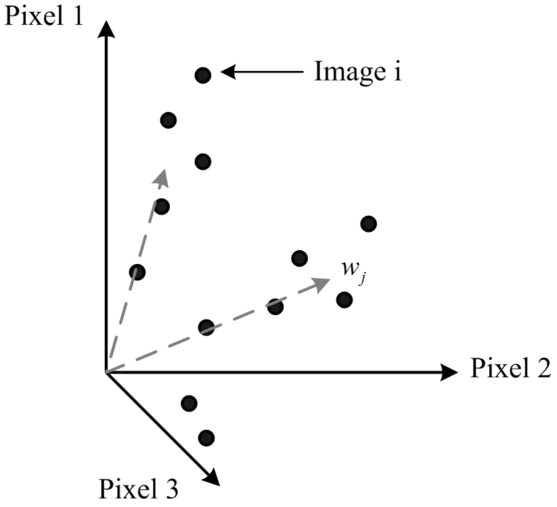

Figure 3.5: The architecture of ICA representation for face images.

Due to the high dimensional space of the original data, we first apply PCA to reduce the dimensionality. It should be noted that the use of PCA in the input did not throw away the high–order statistical relationships; the relationships still exist in the data but were not separated. Let $\mathbf{P}_m$ denote the matrix containing the first m PC axes in its column. We perform ICA on $\mathbf{P}_m^T$ to produce m independent source images in the rows of $\mathbf{U}$ as follows:

$$\begin{aligned} \mathbf{W}\mathbf{P}_m^T &= \mathbf{U} && (3.40)\\ \mathbf{P}_m^T &= \mathbf{W}^{-1}\mathbf{U} && (3.41) \end{aligned}$$

The PC representation of the set of zero–mean images in $\mathbf{X}$ based on $\mathbf{P}_m$ is defined as $\mathbf{R}_m = \mathbf{X}\mathbf{P}_m$. A minimum squared error approximation of $\mathbf{X}$ based on $\mathbf{P}_m$ is obtained by $\hat{\mathbf{X}} = \mathbf{R}_m\mathbf{P}_m^T$. By replacing $\mathbf{P}_m^T$ with equation Eq.(3.41) we obtain:

$$\hat{\mathbf{X}} = \mathbf{R}_m\mathbf{W}^{-1}\mathbf{U}. \qquad (3.42)$$

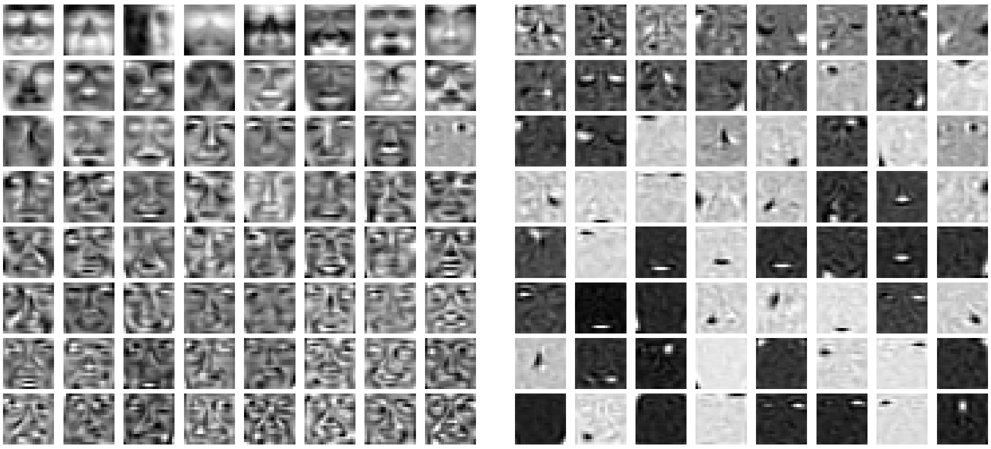

Figure 3.6: Examples of face representations extracted using PCA and ICA models. The first 64 PCA–based eigenfaces are shown at the left–hand side ordered according to their eigenvalues. The right–hand side gives the corresponding 64 ICA–based face representations.

where the rows of $\mathbf{R}_m\mathbf{W}^{-1}$ contains the co–efficients for the linear combination of statistically independent source $\mathbf{U}$ that comprised $\hat{\mathbf{X}}$. Therefore, the IC representation of the face images based on the set of m statistically independent feature images, $\mathbf{U}$, is given by:

$$\begin{aligned} \mathbf{B} &= \mathbf{R}_t\mathbf{W}^{-1} \\ &= \mathbf{X}_t\mathbf{P}_m\mathbf{W}^{-1}, \end{aligned} \tag{3.43}$$

where $\mathbf{X}_t$ is a test image. The source images estimated by the rows of $\mathbf{U}$ are then used as basis images to represent faces. Fig. 3.6 shows some examples of ICA–based face representation constructed from 2,429 face examples from the training set of the MIT CBCL Face Data [3]. Fig 3.7 depicts examples of re–constructed images with strong noises using PCA and ICA projections. The result shows that ICA re–construct noisy images better than PCA, one possible explanation is that the ICA facial subspace captures more facial structural information than PCA subspace and hence gives better performance.

3.5 Conclusion

In this chapters, different skin models have been presented together with the evaluation results and discussions of their advantages and disadvantages. Experimental results suggest that the performance of different skin models mainly depends on the distribution of skin samples in each corresponding colour space. Therefore, the conclusion is that the criteria of selecting an

[3] http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.

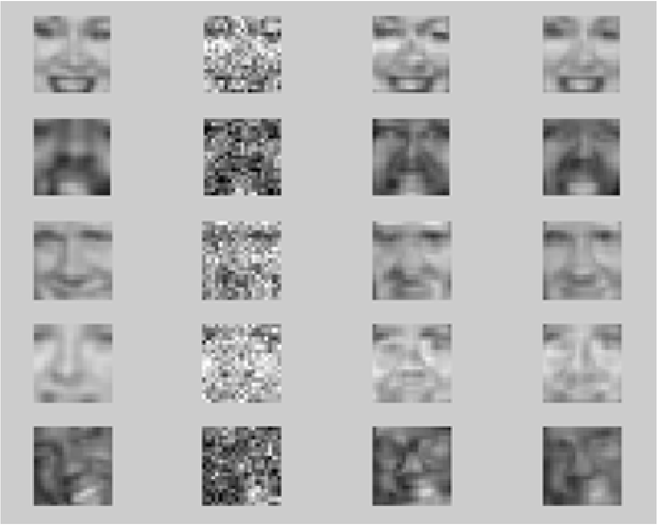

Figure 3.7: Re–construction of images under strong noises using PCA and ICA projections. From left to right, the images are the original images, images with strong noise, re–constructed images using PCA projection and images using ICA projection.

appropriate skin model depends only on the tightness of the skin cluster and its overlapping with non–skin samples.

We have also presented an account of feature extraction for face detection within the PCA and ICA framework. The PCA and ICA models are compared from a statistical point of view and we showed that ICA is based on a stronger statistical concept than PCA and consequently it provides a better probabilistic model. From the discussion of ICA and PCA, we have developed the underlying framework for facial feature extraction which is used in the next chapter.

Chapter 4

Face Detection with SVM

In this chapter, we present the algorithm that learns structural facial pattern through the use of support vector machine (SVM) algorithm and utilises colour information to detect faces in colour images. The advantage of using SVM is that it is a maximal margin classifier and, consequently, this gives low expected probability of generalisation errors. Due to its gerenalisation property, it naturally fits the task of face detection.

4.1 Support Vector Machine

4.1.1 Structural Risk Minimisation

A SVM is a maximum margin classification tool based on structural risk minimisation principle. The goal of a SVM is to produce a model which predicts target class value of data instances in the testing set.

Given a set of labelled pairs of training instances $(\mathbf{x}_i, y_i)$, $i = 1, \dots m$ where $\mathbf{x}_i \in \mathbb{R}^d$, $y \in 1, -1$ and m is the number of samples, we seek to learn the mapping of $\mathbf{x}_i \mapsto y_i$. This is actually done by estimating a function of $\mathbf{x} \mapsto f(\mathbf{x}, \alpha)$, where α are the adjustable parameters of the function. However, if no restriction is placed on the estimation of the class function, a function that does well to the training samples need not generalise well to unseen examples. To see this, it should be noted that for each function f and any test set $(\bar{\mathbf{x}}_1, \bar{y}_1), \dots, (\bar{\mathbf{x}}_{\bar{m}}, \bar{y}_{\bar{m}}) \in \mathbb{R}^d \times \{-1, 1\}$, satisfying $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_{\bar{m}}\} \cap \{\mathbf{x}_1, \dots, \mathbf{x}_m\} = \{\}$, there exists another function f^* such that $f^*(\mathbf{x}_i) = f(\mathbf{x}_i)$ for all $i = 1, \dots, m$, and $f^*(\bar{\mathbf{x}}_i) \neq f(\bar{\mathbf{x}}_i)$ for all $i = 1, \dots, \bar{m}$. Since only training samples are available to us, we have no means to choose which of the two functions is preferable. Hence, a practical way is to minimise the training error (or empirical risk) by:

$$R_{empir}(\alpha) = \frac{1}{2m}\sum_{i=1}^{m} \mid f(\mathbf{x}_i,\alpha) - y_i \mid \tag{4.1}$$

The expectation of the test error is therefore:

$$R(\alpha) = \int \frac{1}{2} \mid f(\mathbf{x},\alpha) - y_i \mid dP(\mathbf{x},y) \tag{4.2}$$

where $P(\mathbf{x}, y)$ is the probability distribution of the training data assuming that the data are generated independently.

The Vapnik–Chervonenkis (VC) theory [90] shows that it is imperative to restrict the class of functions that f is chosen from to one, which has the capacity that is suitable for the amount of available training data. This provides bounds on the test error and minimisation of these bounds lead to the principle of structural risk minimisation. The best–known capacity concept of VC theory is the VC dimension which is defined as the largest number of points that can be separated in all possible ways by using functions of the given class. If we choose some η such that $0 < \eta < 1$, by using VC theory, which is the bound for all functions of the class, with probability of $1 - \eta$, so that:

$$R(\alpha) \leq R_{empir}(\alpha) + \sqrt{\frac{h(\log\frac{2m}{h} + 1) - \log(\frac{\eta}{4})}{m}} \tag{4.3}$$

where h is a non–negative integer known as the Vapnik Chervonenkis (VC) dimension. It is a measure of the notation of capacity mentioned above. The right–hand side of Eq.(4.3) gives the risk bound of the model and the second term on the right–hand side is called the "VC confidence".

Although it is usually impossible to compute the left–hand side of Eq.(4.3), we can still easily compute the right–hand side of the equation if we know the value of h. Therefore, given several different functions $f(\mathbf{x}, \alpha)$ and choosing a fixed, sufficiently small η, the function that minimises the right–hand side is the one that gives the lowest upper bound of the actual risk.

4.1.2 Construction of Hyperplane Classifier

In this section we will look at how the machine can learn the hyperplane that separate the training data. Again let us label the training data $(\mathbf{x}_i, y_i)$, $i = 1, \dots m$, $y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbb{R}^n$. Consider the class of hyperplanes as:

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0, \quad \mathbf{w} \in \mathbb{R}^n,\ b \in R, \tag{4.4}$$

and the decision function is:

$$f(\mathbf{x}) = \operatorname{sgn}((\mathbf{w} \cdot \mathbf{x}) + b). \tag{4.5}$$

To construct f from the empirical data, we have to find a learning algorithm that meets two conditions: first, among all the separating hyperplane there exists a unique one yielding the maximum margin of separation between classes:

$$\max_{\mathbf{w},b} \ \min\{\|\mathbf{x} - \mathbf{x}_i\| : \mathbf{x} \in \mathbb{R}^n, \ (\mathbf{w} \cdot \mathbf{x} + b = 0), \ i = 1, \ldots, m\}; \tag{4.6}$$

and second, the capacity decreases with increasing margin. Let $d_+(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) point, the margin will be defined as $d_+ + d_-$. In a linear separation case, the SVM simply looks for a hyperplane that maximise the separating margin. This can be formulated as follows:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad \text{for } y_i = +1 \tag{4.7}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad \text{for } y_i = -1. \tag{4.8}$$

Eqs.(4.7) and (4.8) can be combined into one set of inequalities:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \ \ \forall i. \tag{4.9}$$

Points that satisfy Eq.(4.7) will lie on the hyperplane $H_1 : \mathbf{x}_i \cdot \mathbf{w} + b = 1$ with normal $\mathbf{w}$ and perpendicular distance from the origin $|1-b|/\|\mathbf{w}\|$. Similarly, points that satisfy Eq.(4.8) will lie on the hyperplane $H_2 : \mathbf{x}_i \cdot \mathbf{w} + b = -1$ with the same norm $\mathbf{w}$ and perpendicular distance from the origin $|-1-b|/\|\mathbf{w}\|$. Hence $d_+ = d_- = 1/\|\mathbf{w}\|$ and the margin is simply $d_+ + d_- = 2/\|\mathbf{w}\|$. It can be seen that H_1 and H_2 are parallel since they have the same normal and that no training points fall between them. Hence we are looking for a pair of hyperplanes that maximises the margin by minimising the $\|\mathbf{w}\|^2$ subject to constraints in Eq.(4.9). Putting them together to construct the optimal hyperplane will solve the following optimisation problem:

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } \ y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{4.10}$$

We may solve Eq.(4.10) by its Lagrangial dual:

$$\max_{\alpha \geq 0}(\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)) \tag{4.11}$$

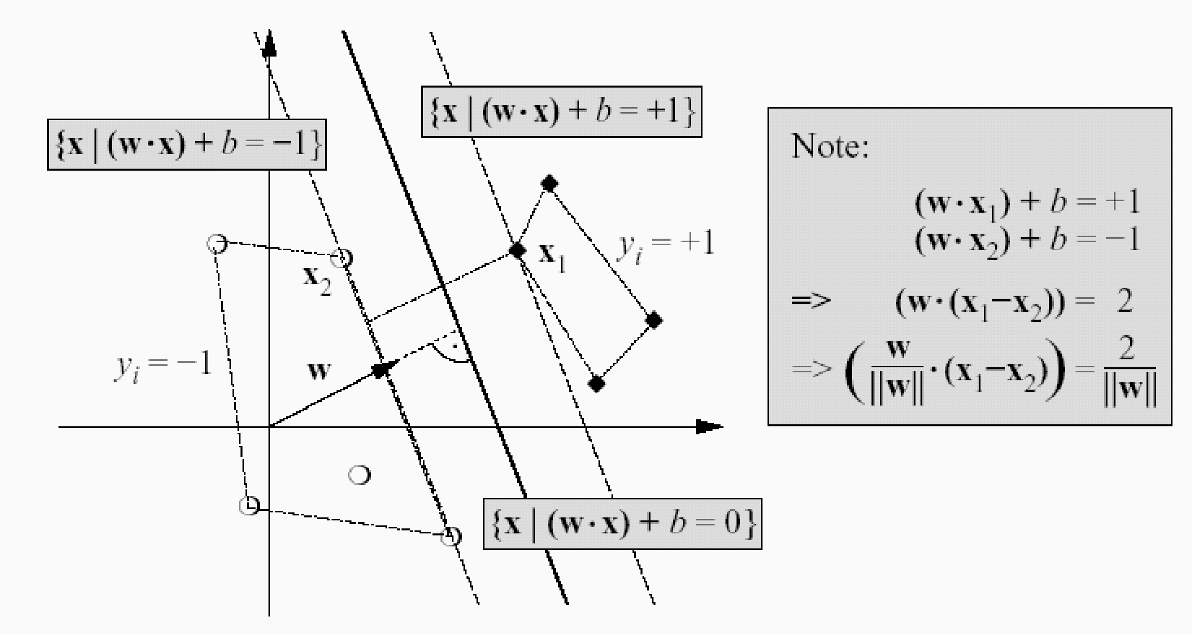

Figure 4.1: An example of binary classification problem. The optimal hyperplane is orthogonal to the shortest line connecting the convex hulls of the two classes, and intersect it half–way between the two classes. Re–produced from Chen et al. [11].

where

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i \cdot ((\mathbf{x}_i \cdot \mathbf{w}) + b) - 1). \tag{4.12}$$

The Lagrangian L has to be minimised with respect to the primal variables $\mathbf{w}$ and b and simultaneously maximised with respect to the dual variables α_i, subject to the constraints $\alpha_i \geq 0$. This is a convex quadratic programming problem. To simplify this dual problem, as $L(\mathbf{w}, b, \alpha)$ is convex when α is fixed, for any given α,

$$\begin{aligned} \frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) &= 0 \\ \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) &= 0 \end{aligned} \tag{4.13}$$

which leads to the conditions:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{4.14}$$

and

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{4.15}$$

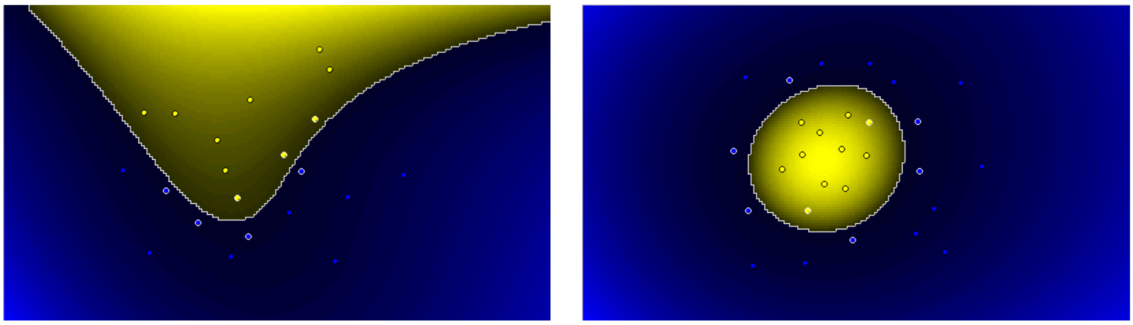

Figure 4.2: An example of binary classification problem using polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d$. Circles and crosses are two classes of training samples, the lines give the decision boundary. Support vectors found are marked by extra circles. From top left to bottom right, the parameter values are $r = 1$ and $d = 1, 2, 4, 6$.

Substituting these constraints into Eq.(4.11), gives:

$$\max_{\alpha} = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4.16}$$

$$\text{subject to} \begin{cases} \alpha_i \geq 0, \quad i = 1, \ldots, m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

Following the above discussion, the hyperplane decision function can be written as:

$$f(\mathbf{x}) = \operatorname{sgn}(\sum_{i=1}^{m} y_i \alpha_i \cdot (\mathbf{x} \cdot \mathbf{x}_i) + b). \tag{4.17}$$

Thus, the solution vector $\mathbf{w}$ is an expansion in terms of a subset of training samples, whose α_i is non–zero, called as support vectors. These support vectors lie on the separating margin while all the remaining training set are irrelevant. In other words, the hyperplane is completely determined by the support vectors only, it does not depend on other training samples.

4.1.3 Kernel

From the previous sections, we will realise that the data appearing in the training problem is in the forms of dot product, $\mathbf{x}_i \cdot \mathbf{x}_j$. To generalised the above methods to the case where the decision function is not a linear function of data, [5] suggested that we can transform the data into some dot product space $\mathcal{H}$, which need not be identical to $\mathbb{R}^d$. In other words, we are looking for a mapping:

$$\Phi : \mathbb{R}^d \rightarrow \mathcal{H} \tag{4.18}$$

where $\mathcal{H}$ is called a feature space. Then the training algorithm would only depend on the data through the dot products in $\mathcal{H}$, in the form of $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. It should be noted that the feature space $\mathcal{H}$ could possibly be an infinite dimensional space which would make it difficult to work with Φ explicitly. However, if there is a kernel function K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, then we would only need to use the kernel in the training algorithm without explicitly knowing what Φ is. Replacing $\mathbf{x}_i \cdot \mathbf{x}_j$ by $K(\mathbf{x}_i, \mathbf{x}_j)$ in previous equations, we obtain:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{m} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) + b \\ &= \sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \qquad (4.19) \end{aligned}$$

Some examples of kernel are:

$$\begin{aligned} \text{linear} &: K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \\ \text{polynomial} &: K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d, \;\; \gamma > 0 \\ \text{Gaussian function} &: K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2) \\ \text{radial basis function (RBF)} &: K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \;\; \gamma > 0 \\ \text{sigmoid} &: K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r) \qquad (4.20) \end{aligned}$$

where γ, r, and d are all kernel parameters.

To summarise, there are three benefits transforming the data into $\mathcal{H}$ via the kernel K: 1) we could define a similarity measure in the dot product in $\mathcal{H}$; 2) we could deal with the patterns geometrically by using linear algebra and analytic geometry; and 3) we could design a large variety of learning algorithm by choosing different mappings Φ.

4.1.4 C-Soft Margin Support Vector Classifier

The previous sections describe the construction of linear and non–linear SVM. In practice, separating hyperplane may not exist. For instance, a high noise level (presence of outliers) may cause a large overlap of the classes from which a separating hyperplane may not be constructed. This is particular true for image signals where noises are always presented. One practical solution for this problem is to introduce slack variables

$$\xi_i \geq 0, \quad i = 1, \ldots, m \qquad (4.21)$$

into Eq.(4.10) in order to relax the constraints to

$$y_i \cdot ((\mathbf{x}_i \cdot \mathbf{w}) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m. \qquad (4.22)$$

By controlling both the classifier capacity via $||\mathbf{w}||$ and the sum of the slacks $\sum_{i=1}^{m} \xi_i$, a generalised classifier can then be found. It has also been shown that the latter also provides an upper bound on the number of training errors.

Let $\xi = (\xi_1, \ldots, \xi_m)$, a soft margin classifier known as C-SVC can be found by minimising the objective function

$$\begin{aligned} &\min \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y_i \cdot ((\mathbf{x}_i \cdot \mathbf{w}) + b) \geq 1 - \xi_i \ \text{ and } \ \xi_i \geq 0, \ \text{ for } \ i = 1, \ldots, m \end{aligned} \tag{4.23}$$

under the constraints $C > 0$. The parameter C is to be chosen by the user, a larger C corresponding to assigning a higher penalty to errors. In practice, this trade–off parameter is usually fine–tuned using cross validation method during the learning phase. By putting the kernels back and re–writing the equation in Lagrange multipliers, again this becomes:

$$\max_{\alpha} = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4.24}$$

$$\text{subject to} \quad \begin{cases} 0 \leq \alpha_i \leq C, \ \ i = 1, \ldots, m \\ \sum_{i=1}^{m}\alpha_i y_i = 0 \end{cases}$$

The only difference from the separable case described before is that now we have the upper bound C on the Lagrange multipliers α_i where the influence of the individual pattern is limited and could be outliers.

4.1.5 Limitations of SVM

The main drawback of using SVM is that it depends too much on the kernel function for separating data. There, however, is no measure to choose the most appropriate kernel. The only solution is by trial and error in order to obtain a suitable kernel from experimental results. On the other hand, a huge number of support vectors is needed to maximise the generalisation ability. This will inevitably increase the computational burden. In addition, the run time will grow almost linearly with the increase in dimensionality of the feature vector. To tackle this problem, the most commonly adapted solution is to reduce the dimensionality of the feature vector by constructing a subspace of the original feature space.

4.2 Face Detection in Colour Images

In this section, we developed a two–step face detection algorithm based on support vector machines (SVM). The first step builds a skin detection model which serves as a platform to reduce the searching space for potential face candidates. The second step extracts representative facial features by projecting the image signals into a face subspace constructed under ICA framework. A set of experiments are conducted and the proposed face detector is evaluated in terms of precision and false alarm.

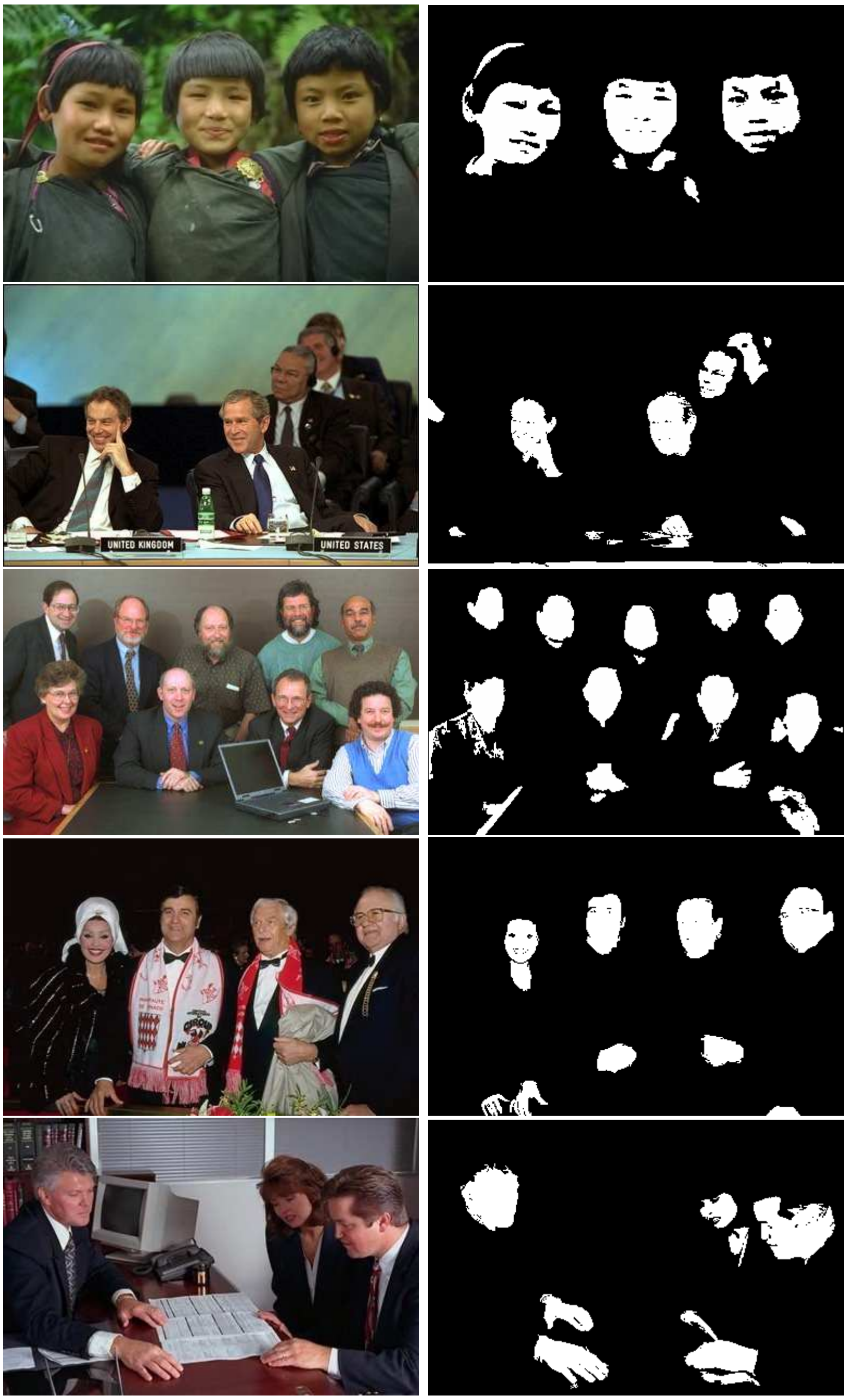

Figure 4.3: Examples of skin detection using single Gaussian model in YCbCr colour space.

4.2.1 Human Skin Colour Model

Although different people have different skin colour, studies have showed that these differences in colours tend to form a tight cluster in certain colour spaces [44]. To build a skin colour

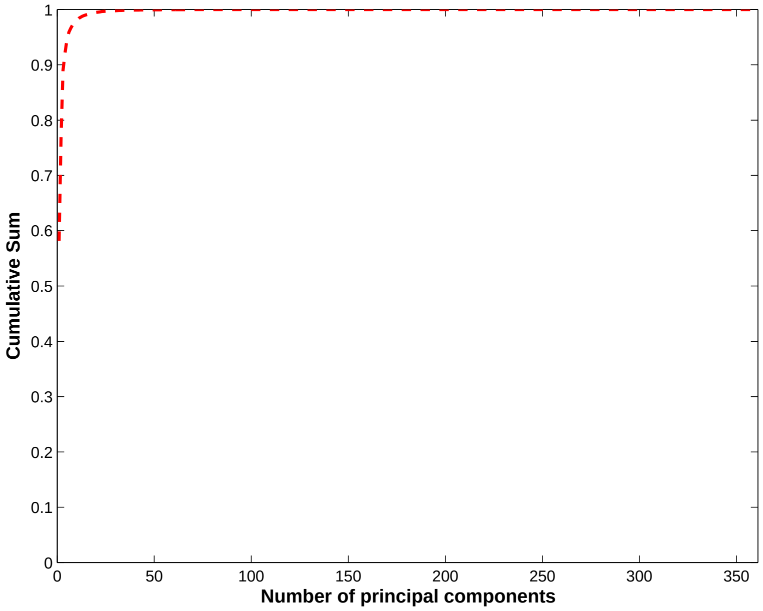

Figure 4.4: Cumulative covariance contained in the eigenvectors.

model, we use the single Gaussian skin colour model in YCbCr colour space as described in Chapter 3. A pixel is identified to a skin colour if its corresponding probability is greater than a threshold. Fig. 4.3 shows some examples of segmented regions from skin detection.

4.2.2 Learning Face Patterns using SVM Architecture

We use a total of 6,977 images (2,429 faces and 4,548 non–faces) collected from MIT CBCL Face Data [1] to train the SVM classifier. The images are in 19×19 resolution and are gathered into one matrix where each column represents one image vector. In order to reduce the dimensionality of the feature space, we extract the first 64 eigenvectors from the original matrix. Fig. 4.4 shows the cumulative covariance captured by all the 361 eigenvectors. The first 64 eigenvectors contain 99.97% of the total covariance from the data. Then the ICA kernel is built from this 64 eigenvector matrix using the algorithm described in Chapter 3. From Table 4.1, we can see that the polynomial kernel at degree 3 with parameter C=10 gives the best performance during training. In Table 4.1, the detection rate (DR) and the false alarm rate (FAR) are defined as:

$$\begin{aligned} DR &= \frac{TP}{TP+FN} \\ FAR &= \frac{FP}{TP+FP} \end{aligned} \tag{4.25}$$

where true positive (TP) is the number of faces that are correctly detected, false positive (FP) is the number of detected samples that do not correspond to faces, and false negative (FN) is

[1] The MIT CBCL Face Data is available at http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.

the number of faces that are not detected.

Kernel	DR	FAR
linear	81.71%	9.14%
polynomial (degree=3)	94.68%	2.58%
Gaussian	69.33%	8.66%
RBF kernel	68.97%	8.77%

Table 4.1: Performance of different kernels during training phase.

In total, the SVM classifier provides 360 support vectors. To evaluate the performance of the classifier, we use the test set from MIT CBCL Face Data which consists of 24,045 test images (472 faces and 23,573 non–faces). Our ICA-based system is compared to the intensity-based system as described in Osuna et al. [63] using the MIT CBCL data set. Fig. 4.5 shows the ROC (Receiver Operator Characteristic) curves for the two systems in both training and testing set.

An ROC curve is a graphical representation of the trade–off between the true positive and false positive rates for every possible cut–off. By tradition, the false positive rate is plotted on the X axis and the true positive rate on the Y axis. This could also be described with 1–specificity on the X axis and sensitivity on the Y axis. The ROC curve can be used for diagnostic test. A good diagnostic test is one that has small false positive and high true positive rates across a reasonable range of cut–off values while a bad diagnostic test is one where the false positive rate goes up linearly with the true positive rate.

If the ROC curve climbs rapidly towards upper left–hand corner of the graph, this means that the true positive rate is high and the false positive rate is low. One way to quantify how quickly the ROC curve rises to the upper left–hand corner is to measure the area under the curve. The larger the area, the better the diagnostic test. An ideal test will have an area of 1.0 which means it achieves both 100% sensitivity and 100% specificity. On the other hand, if the area is 0.5, then the test has effectively 50% sensitivity and 50% specificity which is no better than flipping a coin.

Table 4.2 shows the accuracy of experiment and ROC statistics for our work and Osuna et al.'s [63]. It clearly shows that our ICA-based approach is better than the intensity-based approach in both detection rate and the number of false alarm. The dimensionality of the feature space in our approach is only 64, which is significantly smaller than the work in Osuna et al. as their feature space is as large as 283. We can see that our ICA-SVM approach can effectively reduce the dimensionality of the problem and still give promising results.

4.2.3 Evaluation

We also tested our detection method using real–world colour photos collected from Corel image collection and Internet. The whole testing set contains 67 images (176 faces) with different

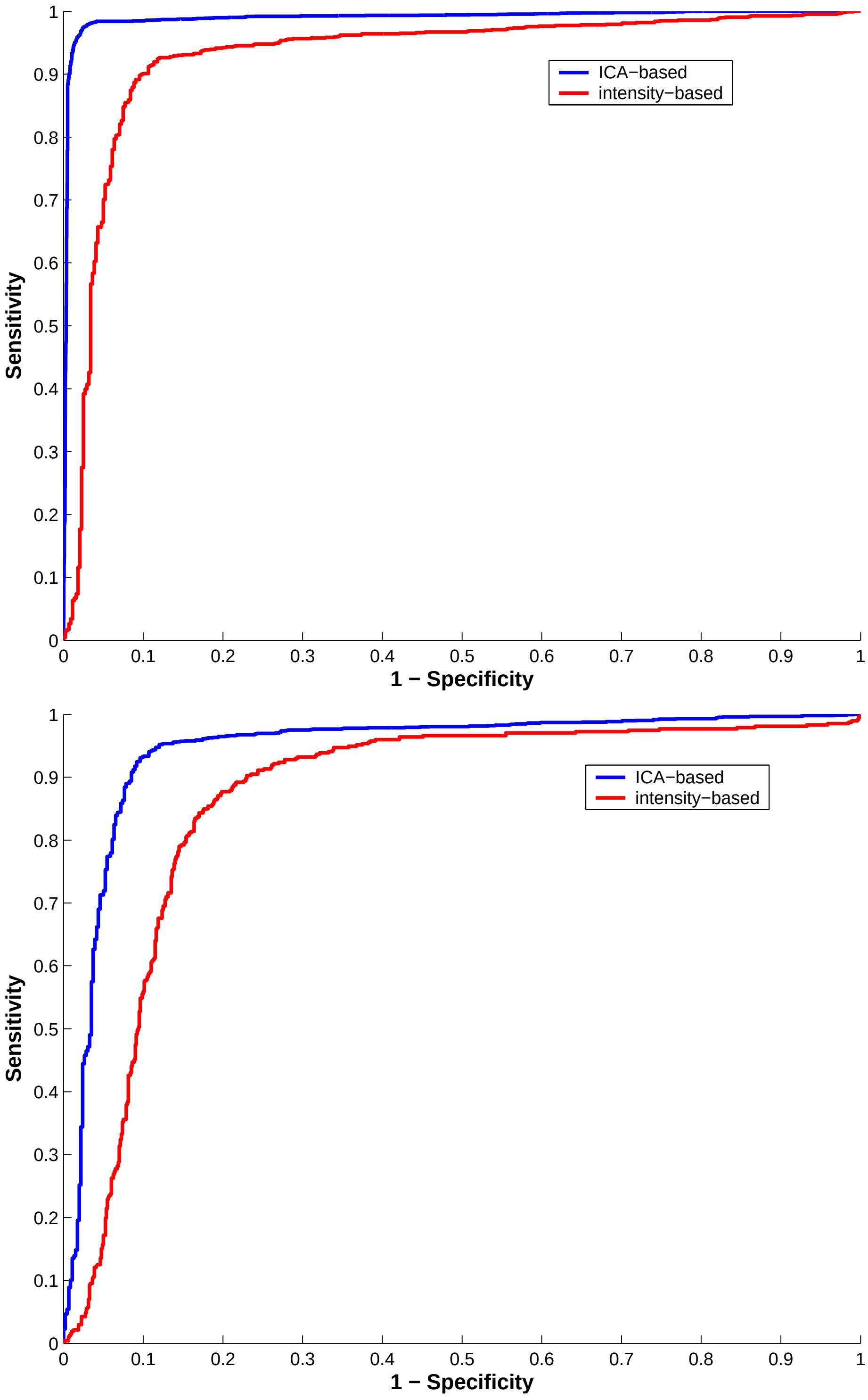

Figure 4.5: ROC curve for the CBCL face data (top:training set, bottom: testing set).

sizes and resolutions. Since it is difficult to collect a representative set of non–face examples, we use the previous trained SVM classifier and the bootstrap method [83] to include more

Training Set	DR	FAR	ROC Area	Standard Error
ICA-based	94.68%	2.58%	0.9902	0.0013
Intensity-based	89.87%	5.66%	0.9391	0.0086

Testing Set	DR	FAR	ROC Area	Standard Error
ICA-based	89.79%	4.75%	0.9444	0.0034
Intensity-based	84.24%	6.11%	0.8666	0.013

Table 4.2: ROC statistics for the CBCL data set.

non–face examples and retrain the initial classifier. The skin detection method is first applied to each input test image to retain only skin–like regions. Then the input image is re–sampled at different scales. A window of 19 × 19 [2] pixels size scans through the image and the pixels are extracted for face detection if more than 50% of the window pixels are counted as skin pixels. Since the decisions are made at the same location for several times, it is possible that detections will overlap so we use the voting techniques in order to reduce false alarms. This technique arbitrates between two (or more) overlapping detections by maintaining only the detections with the highest SVM score. The main drawback is that this heuristic sometimes will remove overlapping alarms corresponding to slightly overlapping faces. Furthermore, voting only removes a false alarm if it is close to another alarm with a higher SVM score, this heuristic cannot therefore deal with isolated false alarms.

The detection rate for our method is 88.6% and the number of false positive is 27. Fig. 4.6 shows some results of our methods. Our method detects frontal faces and faces with shadows. However occluded and rotated faces cannot be detected effectively due to lack of such examples in the training sets. Fig 4.7 also shows the result of face detection under strong lighting changes. The result suggests that our system fails to detect faces in a dark lighting condition where the skin pixels cannot be detected effectively. However, our system works well for images exposed to certain amount of light. In this situation, although the skin detection step gives more outliers, the ICA-SVM classification step can still detect human faces effectively.

[2] This gives image patches at 361 dimension, which is then projected to the facial subspace for feature extraction.

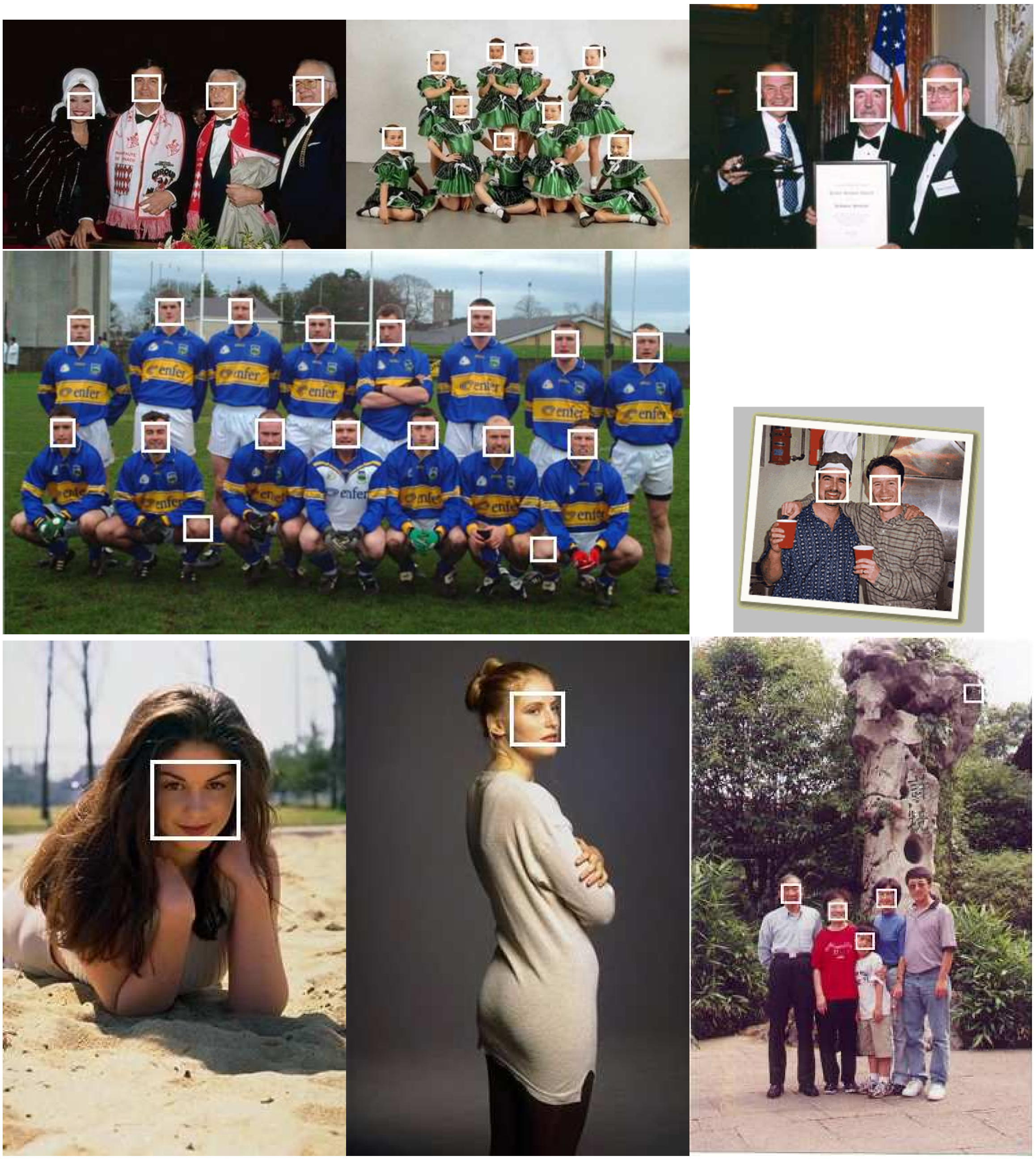

Figure 4.6: Results of face detection using the SVM–ICA architecture.

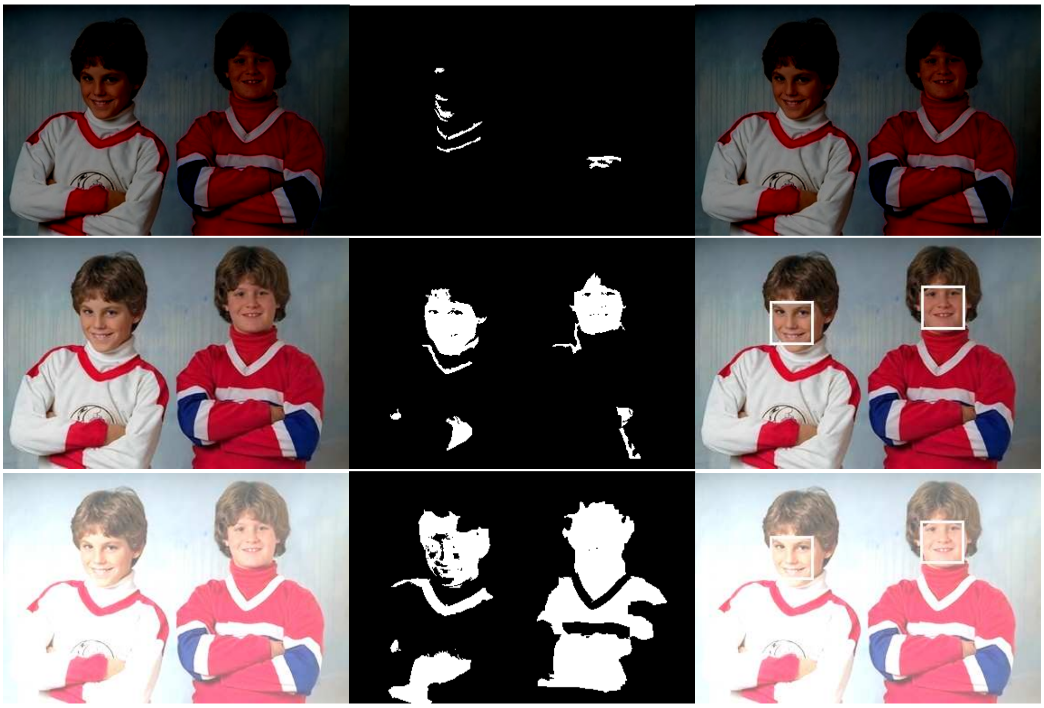

Figure 4.7: Results of face detection under different lighting conditions. From top to bottom, images are exposed to different lighting conditions: dark lighting, normal lighting and excessive lighting.

Chapter 5

Conclusion and Future Work

In this thesis, various aspects of research on intelligent human computer interaction are discussed in the context of computer vision and machine learning. In this chapter, we summarise our results of this work and sketch the future research directions.

5.1 Conclusion

In Chapter 3, we described different methods to detect human skin regions from still colour images and subsequently compared and evaluated different skin models which was followed by a discussion on their advantages and disadvantages. Experimental results suggest that the performance of different skin models mainly depends on the distribution of skin samples in each corresponding colour space. The tightness of the skin cluster and its overlapping with non–skin samples are the main criteria in selecting appropriate skin model.

We have also presented an account of feature extraction for face detection within the PCA and ICA framework in Chapter 3. We have compared these models from a statistical point of view and showed that ICA is based on a stronger statistical concept than PCA and, consequently, it provides a better probabilistic model. In addition, ICA is sensitive to higher–order statistics other than just the second–order statistics and therefore it potentially captures the structural information of the source better than PCA. From the discussion of ICA and PCA, we have developed the underlying framework for facial feature extraction which is used in the SVM learning architecture in Chapter 4.

In Chapter 4, we have described the algorithms of different SVM classifiers and the underlying ideas. Experimental results are reported using the MIT CBCL Face Data and real world photos in the task of face detection. The performance is evaluated using ROC curve, and the result suggested that our ICA-SVM approach can significantly reduce the dimensionality of the feature space and hence speed up the computation time. This approach also captures better facial features than features based on intensity only and gives promising performance.

5.2 Future Work

In Chapter 3, the ICA framework has been used for feature extraction. It is of great interest to see how kernel–ICA can be used in the same way for facial feature extraction. Since kernel–ICA can provide non–linear projection, we can formulate the feature space in non–linear subspaces, which are naturally more suitable for highly non–rigid complex objects such as faces.

On the other hand, training a SVM for a large-scale problem is challenging because it is computationally intensive and the memory requirement grows with square of the number of training vectors. Although skin detection helps to reduce the search space for potential face candidates, it is interesting to develop methods other than skin detection to identify regions of interest.

Future work can also focus on learning the "face kernel". Since we know that the performance of SVM architecture highly depends on the kernel and the best kernel depends directly on the problem at hand. If we could introduce our prior knowledge of human face into the kernel, we can significantly improve the performance of the algorithm. Instead of tuning the parameters of a given kernel, a more promising way is to try to learn the kernel matrix directly from the data. However, some questions remain unanswered. For instance, it is not clear what the best criterion is to optimise the kernel, and also how to design the kernel in a computationally efficient way. Therefore, important developments can be expected in this domain.

Appendix A

List of Publications

1. Tsz Ying Lui and Ebroul Izquierdo, "Scalable Object-based Image Retrieval", IEEE International Conference on Image Processing, ICIP2003, 14-17 Sep, Barcelona, Spain.

2. Sorin Sav, Vasileios Mezaris, Tsz Ying Lui and et al., "Region and Object Segmentation Algorithms in the Qimera Segmentation Platform", 3rd International Workshop on Content-Based Multimedia Indexing", CBMI 2003, 22-24 Sep, IRISA, Rennes, France.

3. Tsz Ying Lui and Ebroul Izquierdo, "Automatic detection of human faces in natural scene images by use of skin colour and edge distribution", 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, April 21-23, 2004, Instituto Superior Técnico, Lisboa, Portugal.

References

[1] A. Albiol, L. Torres, C. A. Bouman, and E. Delp. A simple and efficient face detection algorithm for video database applications. *In Proc. of the 2000 Int'l Conf. on Image Processing*, volume 2, pages 239–242, Vancouver, Canada, Sep 2000.

[2] M. F. Augustejin and T. L. Skujca. Identification of human faces through texture-based feature recognition and neural network technology. *In Proc. IEEE Conf. Neural Networks,* volume 1, pages 392–398, 1993.

[3] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation, 7*(6):1129–1159, 1995.

[4] P. Borges, J. Mayer, E. Izquierdo, "Robust and Transparent Color Modulation for Text Data Hiding", *IEEE Transactions on Multimedia*, Volume 10, Issue 8, pages 1479-1489

[5] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *In 5th Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, Pennsylvania, USA, 1992.

[6] J. Calic, S. Sav, E. Izquierdo, S. Marlow, N. Murphy, N. O'Connor, "Temporal Video Segmentation For Real-Time Key Frame Extraction", *27th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002).* Orlando, FL, 13-17 May 2002, Volume 4, pages 3632-3635.

[7] J. Calic, E. Izquierdo, "A Multiresolution Technique for Video Indexing and Retrieval", *2002 International Conference on Image Processing (ICIP 2002)*, Rochester, NY, September 22-25, 2002, Volume 1, pages 952—955.

[8] J. Calic, E. Izquierdo, "Towards Real-Time Shot Detection in the MPEG Compressed Domain", *3rd Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2001)*, Tampere, Finland, 16-17 May 2001, pages 1-5.

[9] J. Calic, E. Izquierdo, "Temporal Segmentation of MPEG Video Streams", *EURASIP Journal on Applied Signal Processing*, Issue 6, pages 561—565, 2002/6/26.

[10] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: *A survey. Proc. IEEE*, 83(5):705–740, 1995.

[11] C. Chen and S. P. Chiang. Detection of human faces in color images. *In IEE Proc. Vision Image Signal Process.*, volume 144, pages 384–388, 1997.

[12] P. H. Chen, C. J. Lin, and B. Scholkopf. A tutorial on nu-support vector machines. *In Applied Stochastic Models in Business and Industry*, 2005.

[13] A. J. Colmenarez and T. S. Huang. Maximum likelihood face detection. *In IEEE Proc. of 2nd Int. Conf. on Aut. Face and Gesture Recognition*, pages 307–311, Vermont, USA, Oct 1996.

[14] A. J. Colmenarez and T. S. Huang. Face detection with information-based maximum discrimination. *In IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, volume 6, pages 782–787, 1997.

[15] I. Craw, H. Ellis, and J. R. Lishman. Automatic extraction of face-feature. *Pattern Recognition Lett.*, pages 183–187, 1987.

[16] Y. Dai and Y. Nakano. Face-texture model based on SGLD and its application in face detection in a color scene. *Pattern Recognition*, 29(6):1007–1017, 1996.

[17] Dorado, E. Izquierdo, "Semantic Libelling of Images Combining Color, Texture and Keywords", *IEEE Proceedings 10th International Conference on Image Processing (ICIP 2003)*. Barcelona, Catalonia, 14-18 September 2003, pages 9-12.

[18] Dorado, E. Izquierdo, "Semi-Automatic Image Annotation Using Frequent Keyword Mining", *IEEE Proceedings 7th International Conference on Information Visualisation (IV 2003)*. London, England, 16-18 July 2003, pages 532-535.

[19] Dorado, E. Izquierdo, "Fuzzy Color Signatures", *IEEE Proceedings International Conference on Image Processing*. Rochester, New York, 22-25 September 2002, Vol. 1.

[20] N. Duta and A. K. Jain. Learning the human face concept from black and white images. In *Proc. of Int. Conf. on Pattern Recognition*, volume 2, pages 1365–1367, 1998.

[21] R. Feraud, O. Bernier, and D. Collobert. A constrained generative model applied to face detection. Neural Process. Lett. 5, pages 73–81, 1997.

[22] R. Feraud, O. Bernier, J. E. Viallet, and M. Collobert. A fast and accurate face detector for indexation of face images. *In Proc. 4th IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 77–82, 2000.

[23] V. Govindaraju. Locating human faces in photographs. Int. *Journal of Computer Vision* 19.

[24] H. P. Graf, E. Cosatto, D. Gibson, E. Petajan, and M. Kocheisen. Multi-modal system for locating heads and faces. *In IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 277–282, Killington, Vermont, USA, Oct 1996.

[25] Q. Gu and S. Z. Li. Combining feature optimization into neural network based face detection. *In Proc. of the 15th Int'l Conf. on Pattern Recognition*, pages 2814–2817, 2000.

[26] M. Hanke, E. Izquierdo, R. März, "On Asymptotics in Case of Linear Index-2 Differential-Algebraic Equations", *SIAM Journal on Numerical Analysis 1998*, Volume 35, Issue 4, pages 1326-1346

[27] R. Herpers, M. Michaelis, K.-H. Lichtenauer, and G. Sommer. Edge and key point detection in facial regions. *In IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 212–217, Killington, Vermont, USA, Oct 1996.

[28] H. Hongo, M. Ohya, M. Yasumoto, Y. Niwa, and K. Yamamoto. Focus of attention for face and hand gesture recognition using multiple cameras. *In Proc. 4th IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 156–162, 2000.

[29] R. Hoogenboom and M. Lew. Face detection using local maxima. *In IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 334–339, Killington, Vermont, USA, Oct 1996.

[30] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8:179–187, 1962.

[31] E. Izquierdo, et al., "Advanced Content-Based Semantic Scene Analysis and Information Retrieval: The Schema Project", *4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2003)*, World Scientific Publishing, London, England, 9-11 April 2003, pages 519-528.

[32] E. Izquierdo, M. Ernst, “Motion/Disparity analysis for 3D-Video-Conference Applications”, *1995 International Workshop on Stereoscopy and 3-Dimensional Imaging (IWS3DI 1995).* Santorini, Greece, September 1995.

[33] E. Izquierdo, M. Ghanbari, "Key Components for an Advanced Segmentation System", *IEEE Transactions on Multimedia,* Volume 4, Issue 1, pages 97-113.

[34] E. Izquierdo, V. Guerra, "An Ill-Posed Operator for Secure Image Authentication", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 13, Issue 8, pages 842-852.

[35] E. Izquierdo, S. Kruse, "Image Analysis for 3D Modeling, Rendering, and Virtual View Generation", *Elsevier Journal Computer Vision and Image Understanding*, 1998, Volume 71, Issue 2, pages 231-253.

[36] E. Izquierdo, J-R. Ohm, "Image-based rendering and 3D modeling: a complete framework", *Signal Processing: Image Communication*, Volume 15, Issue 10, 2000, pages 817-858.

[37] S. H. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, and Y. T. Liu. Facial feature detection using geometrical face model: An efficient approach. *Pattern Recognition*, 31(3):273–282, 1998.

[38] P. Juell and R. Marsh. A hierarchical neural network for human face detection. *Pattern Recognition*, 29(5):781–787, 1996.

[39] S. Kay, E. Izquierdo, "Robust Content Based Image Watermarking", *3rd Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2001)*, Tampere, Finland, 16-17 May 2001, pages 53-56

[40] C. Kervrann, F. Davoine, P. Perez, R. Forchheimer, and C. Labit. Generalized likelihood ratio-based face detection and extraction of mouth features. *Pattern Recognition Lett.*, 18(9):889–912, 1998.

[41] V. Kumar and T. Poggio. Learning-based approach to real time tracking and analysis of faces. *In Proc. 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 96–101.

[42] S. Y. Kung and J. S. Taur. Decision-based neural networks with signal/image classification applications. *IEEE Trans. Neural Networks*, 6:170–181, 1995.

[43] M. LaCascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: an approach based on registration of textured-mapped 3D models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:322–336, 2000.

[44] A. Lanitis, A. Hill, T. Cootes, and C. Taylor. Locating facial features using genetics algorithms. *In Proc. of Int. Conf. on Digital Signal Processing* 1995, pages 520–525.

[45] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic tracking, coding and reconstruction of human faces using flexible appearance models. *IEEE Electron. Lett.* 1994, pages 1578– 1579.

[46] Y. Li, A. Goshtasby, and O. Garcia. Detecting and tracking human faces in videos. *In Proc. of the 15th Int'l Conf. on Pattern Recognition*, volume 1, pages 1807–1810, 2000.

[47] C. C. Lin and W. C. Lin. Extracting facial features by an inhibitory mechanism based on gradient distributions. *Pattern Recognition*, 29:2079–2101, 1996.

[48] S. H. Lin, S. Y. Kung, and L. J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. Neural Networks*, 8:114–132, 1997.

[49] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 62:285–318, 1988.

[50] Z. Liu and Y. Wang. Face detection and tracking in video using dynamic programming. In *Proc. of the 2000 Int. Conf. on Image Processing*, pages 53–56, Vancouver, Sep 2000.

[51] X. G. Lv, J. Zhou, and C. S. Zhang. A novel algorithm for rotated human face detection. *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 760–765, 2000.

[52] D. Maio and D. Maltoni. Real-time face location on grey-scale static images. *Pattern Recognition*, 33:1525–1539, 2000.

[53] S. McKenna, S. Gong, and J. J. Collins. Face tracking and pose representation. *In British Machine Vision Conf.*, pages 755–764, Edinburgh, Scotland, Sep 1996.

[54] L. Meng and T. Nguyen. Two subspace methods to discriminate faces and clutters. *In Proc. of the 2000 Int'l Conf. on Image Processing*, volume 2, pages 215–218, Vancouver, Canada, Sep 2000.

[55] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1), 1997.

[56] Y. Moses, Y. Adini, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *In Proc. of the European Conf. on Computer Vision*, pages 286–296, 1994.

[57] M. Mrak, N. Sprljan, E. Izquierdo, "Motion estimation in temporal subbands for quality scalable motion coding", *IET Electronics Letters*, Volume 41, Issue 19, pages 1050-1051.

[58] M. Mrak, C. K. Abhayaratne, E. Izquierdo, "On the influence of motion vector precision limiting in scalable video coding", *7th International Conference on Signal Processing (ICSP 2004)*. Beijing, China, 31 August - 4 September 2004, Volume 2, pages 1143-1146.

[59] E. Mrówka, A. Dorado, W. Pedrycz, E. Izquierdo, "Dimensionality Reduction for Content-Based Image Classification", *IEEE Proceedings 8th International Conference on Information Visualisation (IV 2004)*. London, England, 14-16 July 2004, pages 435-438.

[60] N. O'Connor, E. Izquierdo et al., "Region and Object Segmentation Algorithms in the Qimera Segmentation Platform", *3rd International Workshop on Content-Based Multimedia Indexing (CBMI 2003),* Rennes, France, 22-24 September 2003, pages 1-8.

[61] E. Oja, A. Hyvarinen, and H. Karhunen. Independent Components Analysis. *John Wlley& Sons, Inc.*, 2001.

[62] A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proc. IEEE*, 69:529–541, 1981.

[63] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. *In IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pages 130–136, San Juan, Puerto Rico, Jun 1997.

[64] P. Peer, J. Kovac, and F. Solina. Human skin colour clustering for face detection. *In EURO2003 - Int'l Conf. on Computer as a Tool*, Ljubljana, Slovenia, September 2003.

[65] A. Pentland, B. Moghaddam, and T. Strarner. View-based and modular eigenspaces for face recognition. *In IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pages 84–91, Seattle, WA, USA, Jun 1994.

[66] J. P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Comput.*, 16(5):1090–1104, 1998.

A. Pinheiro, E. Izquierdo, M. Ghanhari, "Shape Matching using a Curvature Based Polygonal Approximation in Scale-Space", *IEEE Proceedings International Conference on Image Processing (ICIP 2000)*. Vancouver, BC, 10-13 September 2000, Volume 2, pages 538-541.

[67] L. N. Piotrowski and F. W. Campbell. A demonstration of the visual importance and flexibility of spatial–frequency, amplitude, and phase. *Perception*, 11:337–346, 1982.

[68] B. Raducanu and M. Grana. Face localization based on the morphological multiscale fingerprint. *In Proc. of the 15th Int'l Conf. on Pattern Recognition*, volume 2, pages 2925–2928, 2000.

[69] A. N. Rajagopalan, K. S. Kumar, J. Karlekar, R. Manivasakan, M. M. Patil, U. B. Desai, P. G. Poonacha, and S. Chaudhuri. Finding faces in photographs. *In Proc. of 6th Int'l Conf. on Computer Vision,* pages 640–645, 1998.

[70] J. Ramos, N. Guil, J. González, E. Zapata, E. Izquierdo, "Logotype detection to support semantic-based video annotation", J*ournal Signal Processing: Image Communication,* Volume 22, Issue 7-8, pages 669-679.

[71] D. Reisfeld and Y. Yeshurun. Robust detection of facial features by generalised symmetry. *In Proc. of 11th Int. Conf. on Pattern Recognition*, pages 117–120, The Hague, The Netherlands, Apr 1992.

[72] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In Advances in Neural Information Processing Systems 12, NIPS, pages 862–868, Denver, Colorado, USA, Nov 1999.

[73] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:23–28, 1998.

[74] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. *In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 38–44, 1998.

[75] T. Sakai, M. Nagao, and T. Kanade. Computer analysis and classification of photographs of human faces. *In Proc. First USA-Japan Computer Conf.*, pages 2–7, 1972.

[76] F. S. Samaria and S. Young. HMM based architecture for face identification. *Image and Vision Computing*, 12:537–583, 1994.

[77] S. Satoh. Comparative evaluation of face sequence matching for content-based video access. *In Proc. of 4th IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 163–168, 2000.

[78] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 45–51, Jul 1998.

[79] H. Schneiderman and T. Kanade. A statistical model for 3D object detection applied to faces and cars. *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 746–751, Jul 2000.

[80] A. W. Senior. Face and feature finding for a face recognition system. *In Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication*, pages 154–159, Wash- ington D. C., USA, Mar 1999.

[81] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Opt. Soc. Amer.*, 4:519–524, 1987.

[82] F. Smeraldi, O. Carmona and J. Bigun. Saccadic search with Gabor features applied to eye detection and real-time head tracking. *Image Vision Comput.* 18(4):323–329, 2000.

[83] K. Sobottka and I. Pitas. Face localization and feature extraction based on shape and color information. *In Proc. IEEE Int'l Conf. on Image Processing*, pages 483–486, 1996.

[84] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[85] A. Tankus, H. Yeshurun, and N. Intrator. Face detection by direct convexity estimation. In *Proc. of the 1st Int. Conf. on Audio- and Video-based Biometric Person Authentication*, pages 43–50, Crans-Montana, Switzerland, 1997.

[86] J. C. Terrillon, M. Daivd, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and invariant moments. *Proc. 3rd Int'l Conf. Automatic Face and Gesture Recognition*, pages 112–117, 1998.

[87] J. C. Terrillon, M. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *In Proc. 4th IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 54–63, 2000.

[88] E. Tsomko, H-J. Kim, E. Izquierdo, "Linear Gaussian blur evolution for detection of blurry images", *IET Image Processing*, Volume 4, Issue 4, pages 302-312.

[89] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cog. *Neuroscience*. 3:71–86, 1991.

[90] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, 2000.

[91] M. Walter, A. Psarrou, and S. Gong. Data driven model acquisition using minimum description length. *In British Machine Vision Conf. (BMVC)*, pages 673–683, University of Manchester, Sep 2001.

[92] S. Wan, E. Izquierdo, "Rate-distortion optimized motion-compensated prediction for packet loss resilient video coding", *Image Processing, IEEE Transactions on*, Volume 16, Issue 5, pages 1327-1338

[93] Y. Wang, E. Izquierdo, "High-Capacity Data Hiding in MPEG-2 Compressed Video", *9th International Workshop on Systems, Signals and Image Processing (IWSSIP 2002)*, World Scientific, Manchester, England, 7-8 November 2002, pages 212-218

[94] J. G. Wang and E. Sung. Frontal view face detection and facial feature extraction using color and morphological operations. *Pattern Recognition Lett.* 20, pages 1053–1068, 1999.

[95] N. Ramzan, S. Wan, E. Izquierdo, "Joint Source-Channel Coding for Wavelet-Based Scalable Video Transmission Using an Adaptive Turbo Code", *EURASIP Journal on Image and Video Processing*, Volume 2007, pages 1-12.

[96] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(6):557–563, 1999.

[97] G. Yang and T. S. Huang. Human face detection in a complex background. *Pattern Recognition*, 27(1):53–63, 1994.

[98] M. H. Yang, N. Ahuja, and D. Kriegman. Mixtures of linear subspaces for face detection. *Proc. 4th Int'l Conf. Automatic Face and Gesture Recognition*, pages 70–76