Robust face detection in cluttered images

Zhu, Xiaoling

2005

Zhu, X. (2005). Robust face detection in cluttered images. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4086>

<https://doi.org/10.32657/10356/4086>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:40:18 SGT

Robust Face Detection in Cluttered Images

Zhu Xiaoling

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Master of Engineering

2005

TA
1650
Z63
2005

*To my family,
for their encouragement and love.*

Acknowledgment

First and foremost, I am most grateful to my supervisor, Associate Professor Wang Han for his professional guidances, comments and supports in the process of my study. It has been a great pleasure to have Professor Wang as my supervisor. Moreover, I would like to thank Associate Professor Eric Sung for his helpful instruction, and Professor Soh Yeng Chai for his kindly concern.

Also, I wish to express my sincere gratitude to all the people who have assisted me during the years of my postgraduate study at Nanyang Technological University: Sun Yeow Cheng for his precious instruction and concern; Wang Xiao, Wang Lei, Li Jiang, Liu Nan, Liang Nanying, Chen Cheng, and other researchers in IRL for their valuable discussion and sharing with me their ideas and information.

Last but not least, I would like to dedicated my thesis to my family, especially to my father, who have always provided support and understanding to me. Without their endless encouragement, I would not have made any correct decision in the turning point of my study and life.

Summary

In this thesis, we present a robust face detection algorithm in cluttered images. It is a system based on a novel feature extraction method that uses wavelet-based sub-kernel principal component analysis (SKPCA) method. Usually, facial features such as eyebrows, eyes, nose and mouth are distinguished characteristics of human face, and that their global structure distribution is a guarantee of the existence of a face. The proposed method firstly transfers gray value images into a Haar wavelet spatial frequency domain to extract visually plausible features of the shape and interior detail of faces. Following this step, the constructed wavelet coefficients are mapped onto a higher dimensional feature space using SKPCA. By this way the structure distribution of faces can be captured. Theoretically, SKPCA is an extension of kernel principal component analysis algorithm (KPCA). In the process of SKPCA, a simple and effective subset choosing method is used to solve the computational and storage problems of KPCA. Via wavelet-based SKPCA feature extraction method, representative features of face images that include both local facial feature detail and global structure characteristic are constructed. Finally, a relatively simple support vector machine (SVM) is adopted as classifier.

To test the accuracy and robust performance of the proposed face detection system, different testing data sets are constructed. In particular, 24045 testing data in MIT CBCL database are used as Set-1, 994 FERET data as Set-2, and 84 MIT-CMU testing images as Set-3. In addition, comparisons with different feature extrac-

tion methods are carried out and the experimental results show that wavelet-based SKPCA outperforms the others.

Table of Contents

Acknowledgements	i
Summary	ii
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives	3
1.3 Major Contribution of the Thesis	5
1.4 Organization of the Thesis	6
2 Literature Review on Face Detection	8
2.1 Introduction	8
2.2 Feature Extraction and Face Representation	10
2.2.1 Feature-based Representations	11

TABLE OF CONTENTS	v
2.2.1.1 Facial Features	11
2.2.1.2 Skin-tone Color	14
2.2.2 Template Representations	16
2.2.2.1 Predefined Templates	16
2.2.2.2 Deformable Templates	17
2.2.3 Example-based Representations	18
2.2.3.1 Value-Based	19
2.2.3.2 Distribution-Based	22
2.2.3.3 Component-Based	24
2.2.3.4 Hidden Markov Model	25
2.2.3.5 Trace Transform	26
2.3 Classifier	27
2.3.1 Matching	28
2.3.2 Decision Tree	28
2.3.3 Probabilistic Approach	29
2.3.4 Neural Networks	32
2.3.5 Support Vector Machine	34
2.3.6 AdaBoost	36
2.4 Performance Comparison Between Face Detection Methods	37
2.4.1 Performance Comparison Between Different Face Representations	38

TABLE OF CONTENTS

vi

2.4.2	Performance Comparison Between Different Classifiers	39
2.5	Conclusion	41
3	Haar Wavelet Coefficients Face Representation	42
3.1	Introduction	42
3.2	Related Works	43
3.3	Principle of Haar wavelet transformation	44
3.4	Face representation using Haar wavelet transformation	48
3.5	Conclusion	51
4	Sub-kernel Principle Component Analysis Facial Feature Extrac- tion Method	52
4.1	Introduction	52
4.2	Related works	53
4.2.1	Principle Component Analysis in Face Detection	53
4.2.2	Kernel-based Feature Extraction in Face Detection	54
4.3	Kernel	55
4.4	KPCA feature extraction	58
4.5	SKPCA Construction	60
4.6	Face Representation Using SKPCA	64
4.7	Conclusion	68
5	Example-based Face Detection System	69

<i>TABLE OF CONTENTS</i>	vii
5.1 Preprocessing	69
5.2 Classifier - Support Vector Machine	70
5.3 Example-based Face Detection System Training Strategy	72
5.3.1 Discussions	73
5.3.1.1 Reinforcement Learning	73
5.3.1.2 Multiple Detection Problem	74
5.4 Conclusion	75
6 Experimental Results	77
6.1 Training and testing data	78
6.2 Experimental results	79
6.2.1 Performance of Reinforcement Training	80
6.2.2 Performance comparison between different feature extraction methods	82
6.2.3 Experimental performance in testing data	84
6.3 Conclusion	93
7 Conclusion and Recommendations	94
7.1 Conclusion	94
7.2 Recommendations	95
Author's Publications	97

TABLE OF CONTENTS

viii

Bibliography	98
--------------	----

List of Tables

2.1	Performance comparison between different face representations. The amount of testing data is 120; it includes 20 face images and 100 nonface images.	39
2.2	Performance comparison between different Classifiers. The amount of testing data is 120; it includes 20 face images and 100 nonface images.	40
6.1	Performance comparison between different Classifiers. The original size of training data is 6977.	81
6.2	Comparisons of different feature extractor. See from the data, our proposed method presents a good detection rate, especially reduces the false negative rate. The total amount of face testing data is 472. .	84
6.3	Comparative face detection performance of the Edgar Osuna et al., Sung et al. and Wavelet-based sub-KPCA SVM method on testing data set-1, which contains 472 faces and 23573 nonfaces images. . . .	84
6.4	The average intensities within face region of the corresponding lighting scope testing. The boldface indicates wrong classification. . . .	93

List of Figures

1.1 Examples of biometric features. 2

1.2 Face processing systems can be applied to: (a) surveillance system
and (b) Human Computer Interaction system. 2

1.3 The training and testing strategies of the proposed robust face detec-
tion system. 4

2.1 Examples of faces in different conditions. 9

2.2 (A) Yow and Cipolla model a face as a plane with six facial features (a
pair of eyebrows, a pair of eyes, nose, and mouth). (B) They model 6
features as pairs of oriented edges. (C) The feature selection process
starts with interest points, followed by edge detection and linking,
and sufficed a standard boundary following algorithm. 13

2.3 Hsu’s face detection algorithm. 14

2.4 Govindaraju’s face model, and the feature are selected as two arcs
and two lines. 17

LIST OF FIGURES

xi

2.5

Comparison of 5 classes by Lu. (a) PCA-based subspace. Different classes's data are mixed together. (b) D-LDA-based subspace. (c) Proposed DF-LDA-based subspace. Obviously, DF-LDA can separate different classes much better than the other two.

20

2.6

The 37 significant coefficients chosen by Papageorgiou.

21

2.7

(a) Haar wavelet used by Papageorgiou et al.; Haar-like features (b) Proposed by Viola et al. (c) Proposed by Lienhart et al.

22

2.8

Face and nonface clusters used by Sung and Poggio [110].

23

2.9

The distance measures used by Sung and Poggio [110].

23

2.10

(a) Diagrammatic description of the operation of Mohan's system. (b) Geometric constraints that are placed on the different components.

25

2.11

(a) The fourteen components used on Bileschi's component based face detection system.(b) The three right images are the results images after greedy optimization (brow, nose, and mouth).

25

2.12

(a) Observation vectors. (b) Hidden states.

26

2.13

(a) Definition of the parameters of an image tracing line. (b) A 19×19 image and its Trace transform.

27

2.14

Extracting shapes from the masked Trace transform in [104].

27

2.15

(a) Yow's belief network. (b) Yow's reinforcement belief network. . .

30

2.16

A separating hyperplane, written in terms of a weight vector $\mathbf{w}$ and a threshold b . x_1 and x_2 are support vectors.

35

2.17

Schematic depiction of a the detection cascade in [118].

37

2.18

Component selection.

39

LIST OF FIGURES	xii
3.1 Haar basis for V^2 , where $j = 2, i = 0, 1, 2, 3$	45
3.2 Haar wavelets for W^1 , where $j = 1, i = 0, 1$	46
3.3 Two dimensional Haar wavelet transformation of face images.	49
3.4 1 level Haar wavelet representation of an image.	49
3.5 The dissimilar face images are mapped into similar Haar coefficients feature vectors.	50
3.6 (a) LH subimage raw values. The large values rounded by three ellipses concentrate in the position of left eye, right eye, and mouth in image respectively. (b) HL subimage raw values. The densely large values concentrate in the position of nose.	50
4.1 Toy example of a binary classification problem mapped into feature space. The true boundary can be assumed an ellipse in the input space (left space). The task of the learning process is to estimate this boundary based on empirical data consisting of training points in both classes (plus and circles, respectively). When mapped into feature space via the nonlinear map $\Phi_2(x) = (z_1, z_2, z_3) = ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2)$ (right panel), the ellipse boundary becomes a flat hyperplane bound- ary problem.	57
4.2 Support vectors are defined as the data that nearest to the decision boundary. Based on the theory of structural risk minimization [8], if we discard the other non-SVs data from training set, the decision boundary constructed by the remaining data will be the same as the one constructed by all training examples, [88]. Therefore, SVs can represent original data well in classification problem.	61
4.3 Enlarge the chosen subset using Euclidean distance.	62

LIST OF FIGURES

xiii

4.4

Example of grip-search. Based on the calculated detection rate, we can choose the best pair of parameters of SVM algorithm.

63

4.5

Algorithm for wavelet transformation and subset chosen in MATLAB environment.

65

4.6

Algorithm for SKPCA feature space construction in MATLAB environment.

66

4.7

Algorithm for test data’s SKPCA feature extraction in MATLAB environment.

67

4.8

The process of constructed sub-kernel PCA.

67

5.1

Histogram equalization is used on the left image. We can observe from the right one the contract of a face is enhanced after this processing. .

70

5.2

This is an example of the architecture of support vector machines. Here 7, 1, 4, 1 are the 4 support vectors, and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the corresponding weights. After calculating the weights, a decision function is computed to class the input vector x : $f(x) = \text{sgn}(\sum_{i=1}^4 \lambda_i k(x, x_i) + b)$.

71

5.3

Training and testing steps of face detection system. The bootstrap-ping training step will improve the system performance.

72

5.4

Retrain and testing stage. In the retrain process, the input image is any image that doesn’t contain faces. The false outputs will be used as input to train classifier. In the testing process, the output will determine whether there is a face in the image. If it has, then draw a frame around the face image.

73

5.5

Multiple detection of the same face in nearby locations.

74

5.6	(a) Multiple detections of two faces in two regions. (b) Mass centers of two different regions are the final positions of the detected faces. .	76
6.1	Examples from the database of faces used for training. The images are gray level of size 19×19	78
6.2	Examples from the database of nonfaces used for training. The images are gray level of 19×19	79
6.3	Examples of CBCL test database, set-1. This test set includes 472 face images and 23573 nonface images.	79
6.4	Face detection examples from set-2 without wearing glasses. A square indicates a face region successfully detected. The resolution of the images is 256×384	80
6.5	Face detection examples from set-2 with glasses.	80
6.6	Face detection examples from set-3.	81
6.7	A 19×19 mask to eliminate the near-cheeks pixels.	82
6.8	Detection of large faces in set-3. The large face could be detected is up to 210×210	85
6.9	Detection of small faces in set-3. The smallest size of faces can be detected is 11×11	86
6.10	Detection of old people's faces in set-3. See from the images, old people's faces usually have wrinkle, and fleck. The face features in these people are quite different than the younger ones who have a "clear" faces.	86

LIST OF FIGURES

6.11 Detection of occlude faces in set-3. In these images, faces features contain glasses, and mustache. Some of the faces are also occluded by water drops or by hands. 87

6.12 Detection of faces in set-3 in different expressions. 88

6.13 Detection of blurred faces in set-3. 88

6.14 Rotate faces in set-3. However, because it is a classifier trained on frontal faces, its capacity to detect rotated face is limited. 89

6.15 Scope testing on rotation angles in-plane and in-depth. A square indicates a face region successfully detected. (a) The largest angle of rotated face in-plane is around 18°. (b) The largest angle of rotated face in-depth is around 35°. 89

6.16 Detection of faces in set-3 in different light conditions. 90

6.17 Light source comes from frontal. The corresponding lighting intensity is showed in the above row. From left to right, light intensity is descend. When the light intensity becomes too low, false detection appears (see the fifth image). 90

6.18 Light source comes from right side. The corresponding lighting intensity is showed in the above row. From left to right, the left side of the face becomes darker and darker. 91

6.19 Light source comes from left side. The corresponding lighting intensity is showed in the above row. When the light intensity is too low, the detection system will fail. 91

6.20 Light source comes from top. The corresponding lighting intensity is showed in the above row. System can detect all faces in the first five images. 91

6.21	Light source comes from the bottom. The corresponding lighting intensity is showed in the above row. The detection system doesn't work well in this case.	92
7.1	Most of the analyzed patterns in this image (size 285×450) belongs to the background class. The ratio of nonface to face patterns is 114000:2.	96

Chapter 1

Introduction

1.1 Background and Motivation

Biometric technologies, which are automatic methods of recognizing a person based on a physiological or behavioral characteristic, are becoming the foundation of highly secure identification and personal verification solutions. The interest of biometric is brought about by the increase of the level of security breaches and transaction fraud. Especially, in a world that is surrounded by threatens of terroristic attacks, the need for highly secure identification and personal verification technologies is becoming absolutely necessary. Biometric-based technologies which can provide dependable guarantee on surveillance by using unique personal data characteristic is therefore being received more and more attentions in different countries. Bio-features such as face, fingerprints, hand geometry, handwriting, retinal, vein, and voice can be measured to verify the identification of a person, Figure 1.1. Among these features, face is the most convenient one. Unlike other identification processing, such as fingerprints, retinal, vein, etc., which have to ask the cooperation of people, biometric technologies based on face is a noncooperative process; it allows people to be identified even simply walking by a digital camera. Further, the instrument used to

Figure 1.1: Examples of biometric features.

collect face images is very simple: a digital camera can do this job.

In addition to surveillance system, face processing biometric technologies is also popular in daily life. Abandoning the traditional devices such as keyboards and mouse, more effective and friendly Human Computer Interaction (HCI) is being developed nowadays. People are expecting that one day the self-service machine can “recognize” them by “looking at” their face, or their computer will turn on when they sit in front of its camera “eye” and log in to the right account by itself. A more automatic world will be brought by the development of face processing technologies.

Figure 1.2: Face processing systems can be applied to: (a) surveillance system and (b) Human Computer Interaction system.

However, till now most of the proposed face processing algorithms assume that the locations of faces are known and identified beforehand. To build a fully automated system that can analyze face images, a robust face detection system is indispensable. As an active research topic, face detection has received much attention in the recent twenty years, but there still exist a lot of difficulties in this field. Generally speaking, the difficulties of face detection come from the different appearances of human faces which are caused by orientation, pose, illumination, occlusion, and low

images quality. Therefore, looking for similar face representations under different conditions or searching robust classification algorithms are the two major solutions for the difficulties of face detection problems [16]. In this thesis, we focus on the face representation aspect.

Human beings can effortlessly detect faces in different poses and lighting conditions so there must exist properties or features which are invariant over these variabilities. Obviously, even under different conditions, facial features such as eyebrows, eyes, nose and mouth are still the distinguished characteristics of human face, and meanwhile their global structure distribution can guarantee the existence of a face. Therefore, if any face representation method can obtain these two kinds of information, it would be a successful modeling of a face. Motivated by this, Haar wavelet representation method is used to extract facial features at the first step. It is a method that can emphasize face characteristics by local difference. Then sub-kernel principal component analysis (SKPCA) method is adopted to provide the face global structure information. SKPCA is an extension of kernel principal component analysis algorithm (KPCA). It can be used to solve the computational and storage problems of KPCA. Via the proposed wavelet-based SKPCA feature extraction method, the distinguished local characteristics of face and their global structure distribution can be combined to model the face.

1.2 Objectives

The project aims to design a robust face detection system in cluttered images. Some robust face representation features are needed to model faces in different conditions. Figure 1.3 shows the proposed face detection strategies that are able to extract robust features.

Preprocessing is the first important step of robust face detection system. The pur-

Figure 1.3: The training and testing strategies of the proposed robust face detection system.

pose in this step is to reduce the effect of illumination. Usually, images under different illumination will result in different contrast level images [22]. In poor lighting condition, for example too bright or too dark, low-contrast images will be produced. To enhance the contrast and emphasize facial features such as eyes, noses and mouth, relevant histogram processing technique can be used properly.

In feature extraction step, the combination between distinguished local characteristics of human faces and the global structure distribution of these characteristics will be employed. Apparently, wavelets, which allow an image to be described in terms of a coarse overall shape, plus details that range from broad to narrow, are elegant techniques for representing the levels of facial detail. Theoretically, there exist many wavelets, but most of them are not suitable for the proposed case. To satisfy the

particular training data in our face detection system, which are 19×19 training images, the simplest wavelet will be appropriate. As to the global structure aspect PCA is the most useful approach to represent the structure distribution of the same class by using its dominant eigenvectors. However, PCA is not a powerful technique to provide representative features that can satisfy the criterion which requires that features should have small variance between the same class and, meanwhile, have large variance between the different classes. Therefore, in order to provide more classification information to the extracted features, KPCA will be used. It is a method that first maps images into a higher feature space before solving the eigenvalue problem. However, mapping to a higher feature space will cause computational and storage problems. Therefore, a method, which can solve all these problems, and at the same time would provide a dominant global structure characteristics, are proposed from the extension of KPCA method. Finally, robust features, that are relatively similar for faces in different conditions, should be constructed via the local and the global feature extractions.

After the distinguished features of face images are provided, the classification algorithm that use to classify faces and nonfaces images need not to be too complicated. A relatively simple classifier can be competent for this purpose.

1.3 Major Contribution of the Thesis

The following are the main contributions of the thesis:

1. *Combine the distinguished local facial features and their global structure distribution.* Since facial features such as eyebrows, eyes, nose and mouth are prominent characteristics, many approaches, for example feature-based approach and template matching approach, have made use of this. However, the global structure of the facial features are also an important evidence for the

existence of faces. Therefore, in order to improve performance of representation features, the combination of the local and global characteristics of human faces are proposed.

2. *Use sub-kernel principle component analysis (SKPCA) to extract the global structure characteristics by using the dominant eigenvectors in sub-kernel space.* KPCA algorithm is a powerful method that can extract global structure information and at the same time it can provide good separable features for different classes. However, mapping data to higher kernel space will cause computational and storage problems. In the proposed method, a simple and effective subset choosing method is used to solve these problems of KPCA.
3. *Construct a robust face detection system for cluttered images.* In order to show the accuracy and robust performance of our proposed system, three broad testing sets are constructed. In addition, scope testings of rotation and lighting condition are also carried out. Experimental results show that the proposed method have 98.16% detection rate and robust performance in different conditions, for example different orientations, poses, illuminations, occlusions, and low images quality.

1.4 Organization of the Thesis

The organization of the thesis is as follows:

Chapter 2: **Literature Review on Face Detection** reviews the various face detection methods that exist now. The review is divided into two major parts. One is the *Feature Extraction and Face Representation*, which will go over different face modeling methods. The other one is the *Classifier*, which will illustrate various classification algorithm. In the final part, performance comparison between different

face representations and different classifiers will be presented.

Chapter 3: **Face Representation in Spatial Frequency Domain** will illustrate the principle of Haar wavelet transformation, and show the advantage by using wavelet representation.

Chapter 4: **Sub-kernel Principle Component Analysis Facial Feature Extraction Method** will show the principle of kernel and KPCA feature extraction method. In this chapter, we will mathematically illustrate the computational and storage problems of KPCA. After that, the construction process of SKPCA will be illustrated.

Chapter 5: **Example-based Face Detection System Training Strategy** will show the overall structure of proposed face detection system. Special training strategy will be presented in this chapter. The principle of the used classifier will also be illustrated in last part.

Chapter 6: **Experimental Results** presents the performance results of the proposed system. In this chapter, three different testing sets and compare results by using different feature extraction methods will be listed. Results of scope testings of rotation and lighting condition will also be presented.

Chapter 7: **Conclusion and Recommendations** draws conclusions on the theory, and performance of the design system. Recommendations for future work will also be stated.

Chapter 2

Literature Review on Face Detection

2.1 Introduction

Face detection in single image, which involves the knowledge in image processing, computer vision, machine intelligence and pattern recognition, has achieved great progress in the last two decades. The interest in face detection is caused by its necessity in face processing technologies, that is: it is the first step to build a fully automated system which can analyze face images. Now, the definition of face detection: “Given an arbitrary image, the goal of face detection is to identify and locate human faces in that image regardless of face position, scale, orientation, pose and illumination.”

The difficulties of face detection are evident: first of all, it is a huge data processing problem. Let us consider a thumbnail 19×19 (361 pixels) image pattern. The combination of its gray value is $256^{361} = 2^{2888}$, which is about 90 times more than the world population (2^{32}). Second, there exist many challenges in face detection

Figure 2.1: Examples of faces in different conditions.

technologies:

1. **In-plane or out-of-plane rotation.** Normally, facial appearance would not always be upright frontal ones. To be a noncooperative surveillance system or an effective and friendly HCI system, face detection algorithm must be robust to face rotation.
2. **Occlusion.** In the real cases, people faces will present different obstacles, for example beards, mustaches, and glasses, therefore, part of facial features such as mouth and eyes will be occluded. Moreover, faces may be sometimes occluded by other objects or people. A robust face detection system should endure such kinds of perturbation in face appearance.
3. **Expression.** Face appearance will be quite different under different expression, therefore, even the same face will be dissimilar. This gives difficulty to face detection algorithm.
4. **Lighting.** Faces under different lighting conditions will also give different appearance.

5. **Image quality.** The characteristics of camera will affect the quality of images, therefore they will affect the appearance of faces in those images by adding different levels of noises.

Basically, face detection includes three different operations: preprocessing, feature extraction and classification [16]. Preprocessing is the first and a very important operation in face detection. Usually, the effect of preprocessing will impact the final face detection rate. The purpose of preprocessing is to enhance image conditions. Noise deduction algorithms and lighting adjustment will be done in this step. Sometimes, even rotation adjustment can be applied too. Referring to feature extraction and classification, they are the two major operations in face detection. Their relationship is complementary: an ideal feature extractor would create a representation that make the job of classifier trivial; conversely, an omnipotent classifier would not need any complicated feature extractor. Therefore how to balance the relationship of extractor and classifier in a face detection system is forced upon us for practical reason. Till now numerous methods have been proposed for face detection. In this review, two aspects of these methods will be illustrated: face modeling methods and classifiers.

2.2 Feature Extraction and Face Representation

In this section, we review face detection methods that emphasize feature extraction and face representation aspect, which also can be called face modeling. The goal of face modeling is to characterize an object by features whose values are very similar for objects in the same category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input images. Based on the essential ideas of face modeling, different methods can be broadly attributed to the following three

categories: feature-based representations, template representations, and example-based representations.

2.2.1 Feature-based Representations

The motivation of feature-based representations is caused by the observation that humans can effortlessly detect faces and objects in different poses and lighting conditions. Therefore, there must exist properties or features which are invariant over these variabilities. Numerous methods have been proposed to first detect facial features such as eyebrows, eyes, nose, and mouth, and then based on the existence of them, to deduce the presence of a face. It has been shown that face detection based on feature-based representation is successful to some extent. Edge detectors are commonly used to extract these facial features. And a statistical model which is built to describe their relationships will be used to verify the existence of a face. One problem with the feature-based algorithm is that the image features can be severely corrupted due to illumination, noise, and occlusion. Furthermore, edge detectors would not work well in images where the intensity values change violently under external conditions, for example, shadow. Normally, eyebrows, eyes, nose, mouth, as well as skin color are the five most important characteristics of human face. In the following, we will separate them into two parts to review the development of feature-based face detection.

2.2.1.1 Facial Features

First of all, we would like to review feature-based representation using facial features such as eyebrows, eyes, nose, and mouth. In 1993, instead of using edges like Sirohey's localization face detection method [102], Chetverikov and Lerch presented a simple face detection approach [13]. They used two dark blobs and three light

blobs to represent facial features, such as eyes, cheekbones, and nose. The model uses streaks to represent the outlines of the faces, eyebrows, and lips. Two triangular configurations are utilized to encode the spatial relationship among the blobs. A low resolution Laplacian image is generated to facilitate blob detection. Next, the image is scanned to find specific triangular occurrences as face candidates.

Later, Leung et al. developed a probabilistic method to locate a face in a cluttered scene based on local feature detectors and random graph matching [44]. Concretely, five features (two eyes, two nostrils, and nose/lip junction) are used to describe a typical face. For any pair of facial features of the same type (e.g., left-eye, right-eye pair), their relative distance is computed, and over an ensemble of images the distances are modeled by a Gaussian distribution.

In [125], Yow and Cipolla presented the most typical feature-based face detection method which used a feature-based representation method that used a large amount of evidence from the visual image and their contextual evidence to detect faces. The first stage is to apply a second derivative Gaussian filter, elongated at an aspect ratio of three to one, to a raw image. Interest points, which are detected at the local maxima in the filter response, indicate the possible locations of facial features. The second stage examines the edges around these interest points and groups them into regions. The perceptual grouping of edges is based on their proximity and similarity in orientation and strength. Measurements of a region's characteristics, such as edge length, edge strength, and intensity variance, are computed and stored in a feature vector. The labeled features are further grouped based on model knowledge of where they should occur with respect to each other. Figure 2.2 shows their face model and the process to find interested features. One attractive aspect of this method is that it can detect faces at different orientations and poses. The overall detection rate on a test set of 110 images of faces with different scales, orientations, and viewpoints is 85 percent. However, the reported false detection rate is 28 percent and the

Figure 2.2: (A) Yow and Cipolla model a face as a plane with six facial features (a pair of eyebrows, a pair of eyes, nose, and mouth). (B) They model 6 features as pairs of oriented edges. (C) The feature selection process starts with interest points, followed by edge detection and linking, and sufficed a standard boundary following algorithm.

implementation is only effective for faces larger than 60×60 pixels.

After Yow and Cipolla's work, Han et al. developed a morphology-based technique to extract eye-analogue segments for face detection [28]. They argue that eyes and eyebrows are the most salient and stable features of human face and, thus, useful for detection.

Later, Amit et al. presented a method for shape detection and applied it to detect frontal-view faces in still intensity images [1]. Their modeling method focused on spatial arrangements of edge fragments extracted from a simple edge detector using intensity difference. A rich family of such spatial arrangements, invariant over a range of photometric and geometric transformations, is defined. From a set of 300 training face images, particular spatial arrangements of edges which are more common in faces than backgrounds are selected. The classifier they used is CART algorithm which will be illustrated later. Their experimental results on a set of 100 images from the Olivetti (now AT&T) data set report a false positive rate of 0.2 percent per 1,000 pixels and a false negative rate of 10 percent.

2.2.1.2 Skin-tone Color

Human skin color has been used and proven to be an effective feature in many applications, for example face detection, face tracking etc. Although different races have different skin color, several studies have shown that the major difference lies largely on their intensity rather than their chrominance [26,27]. Several color spaces have been utilized to label pixels as skin including RGB [86], normalized RGB [40,59,107], HSV (or HSI) [103], etc.

Figure 2.3: Hsu's face detection algorithm.

The early face detection method that were based on skin-tone color can be cast back to 1998, Cai et al. [11], Yang et al. [122], and Kim et al. [40] who had tried to use Gaussian density functions to model skin color. The parameters in their unimodal Gaussian distributions are estimated using EM algorithm. Later, Garcia et al. [21] proposed a novel scheme for human faces detection in color images under nonconstrained scene conditions. Providing quantized skin color regions, they applied color clustering and filtering using approximations of the YC_bC_r and HSV skin color subspaces on the original images. A merging stage is then iteratively performed on the set of homogeneous skin color regions in the color quantized image, in order to

provide a set of potential face areas. Constraints related to shape and size of faces are applied, and face intensity texture is analyzed by performing a wavelet packet decomposition on each face area candidate in order to detect human faces. Fuzzy theory is also used to detect faces in color images.

Wu et al. [119] designed two fuzzy models to describe the skin color and hair color, respectively. They used a perceptually uniform color space to describe the color information. Via this, the accuracy and stability will increase.

Recently, Hsu et al. [31] proposed a face detection algorithm for color images in the presence of varying lighting conditions as well as complex backgrounds. Figure 2.3 shows his face detection algorithm. It contains two major modules: 1) face localization for finding face candidates and 2) facial feature detection for verifying detected face candidates. In the face modeling aspect, their algorithm estimated and corrected the color bias based on a lighting compensation technique. The corrected red, green, and blue color components are then nonlinearly transformed into the YC_bC_r color space. The skin-tone pixels are detected using an elliptical skin model in the transformed space. The parametric ellipse corresponds to contours of constant Mahalanobis distance under the assumption of Gaussian distribution of skin tone color. The detected skin-tone pixels are iteratively segmented using local color variance into connected components which are then grouped into face candidates based on both spatial arrangement of these components and the similarity of their color. The facial feature detection module will reject face candidate regions that do not contain any facial features, for example eyes, mouth, and face boundary.

Skin color information is an effective tool to identify facial area. However, color alone is usually not sufficient to detect or track faces. Formally, modular systems have to combine color with other information, such as shape and motion [27, 103, 122].

2.2.2 Template Representations

In template representations [6], a standard face pattern (usually frontal) is manually predefined or parameterized by a function. The existence of a face is determined by the correlation values between the standard template and test images. This approach has the advantage of being simple to implement. However, it has been proven to be inadequate for face detection since it cannot effectively deal with variation in scale, pose, and shape. Later, multiresolution, multiscale, subtemplates, and deformable templates have subsequently been proposed to achieve scale and shape invariance.

2.2.2.1 Predefined Templates

Early attempt of predefined template method can be cast back to Sakai et al. [84]. They used several subtemplates for the eyes, nose, mouth, and face contour to model a face. Each subtemplate is defined in terms of line segments. Lines in the input image are extracted based on the greatest gradient change and then matched against the subtemplates.

In 1987, Craw et al. presented a localization method based on a shape template of a frontal-view face [14]. They used a Sobel filter to extract edges and grouped them to search for the template of a face based on several constraints.

After Craw, Govindaraju et al. [23–25] presented a face modeling method. A face model is built in terms of features defined by the edges. These features describe the curves of the left side, the hair-line, and the right side of a frontal face as in Figure 2.4. Their system reports a detection rate of approximately 70 percent based on a test set of 50 photographs. However, the detected faces should be frontal upright ones, and they should not be occluded by other objects. Moreover, the size of the face should be greater than an absolute minimum, and resolution of the image must

be at least 66 ppi.

Figure 2.4: Govindaraju's face model, and the feature are selected as two arcs and two lines.

Sinha [101] had extended the template matching to the pairwise ratios of brightness of a few regions in face. He encoded a face as a set of brighter-darker relationship between the average intensities of 11 regions. This idea of using intensity differences between local adjacent regions has been later extended to a wavelet-based representation [21, 54, 55].

2.2.2.2 Deformable Templates

Using deformable template method to model facial features is proposed by Yuille et al. in [126]. In this method, they use parameterized templates to describe the facial features. An energy function is defined to link edges, peaks, and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters. Although their proposed method has a good performance, the limitation of it is that the deformable templates should be initialized in the proximity of the object of interest.

In [42], snake [38, 39, 98, 120] was combined with templates to detect faces. First "snakelets" are used to find candidate edges. Candidate ovals (face-locations) are then found from these snakelets using a voting method. For each of these candidate face-locations, they used a method introduced in [41] to find detailed facial features

If a substantial number of the facial features are found successfully, and their positions satisfy ratio-tests for being standard, the procedure positively reports the existence of a face at that location.

Lanitis et al. described a face representation method with both shape and intensity information [43]. In training images the sampled contours such as the eye boundary, nose, cheek are manually labeled, and a vector of sample points is used to represent shape. A face-shape point distribution model can be used to locate faces in new images by using active shape model (ASM) search to estimate the face location and shape parameters. The face patch is then deformed to the average shape, and intensity parameters are extracted. The shape and intensity parameters can be used together for classification.

2.2.3 Example-based Representations

Compared with the previous two methods: feature-based representation and template representation, example-based representations have superior high accuracy in face detection aspect. It is a method that takes the advantage of the unceasing development of computing power and storage. Its “templates”, unlike templates methods, whose templates are predefined by experts, are learned from large amount examples in images. Features in example-based representation are quite different from those in feature-based representation and template representation. Usually, they would not express any concrete visualization for eyes, nose, mouth, skin color, or facial shape. They are somewhat abstract. For example, pixel value, statistical distribution, or trace transform coefficients are used as features to represent faces. Researchers have proven that example-based approaches can handle detecting of faces in cluttered scenes and have shown a reasonable degree of success when extended to handle nonfrontal views. The common problem of example-based methods is the computational burden. For examples, the amount of faces and nonfaces is

strikingly huge. In the following, we will separate the example-based representation methods into several parts to review their achievements.

2.2.3.1 Value-Based

Taking the advantage of learning from huge examples of face images, the feature vectors can be as simple as only using pixel values. Consider a 19×19 image pattern. Therefore, the pixel value feature vector is 361 dimension. Some methods [69, 82, 83, 109] that value the training strategies or classification algorithms would like to use this kind of features. However, gray value alone is not a good representation of faces. They are unstable to rotation, expression, occlusion and sensitive to the changing of lighting conditions. Moreover, since every pixel in the image has the same weight in this representation, features only using pixel values cannot emphasize the characteristics of faces. Therefore, transformations is done to enhance the representative abilities of pixel values.

The motivation of using principle component analysis (PCA) [4, 56, 72, 100, 111, 113, 114] is to design a transformation such that the feature vector can be represented by reducing the number of effective features, yet retaining most of the intrinsic information. An early example of PCA is used by Turk and Pentland [113, 114] who have applied PCA [100] to face recognition and detection. They used PCA on a training set of face images, and got the principle components of faces, and then they used these components to construct an “eigenface space”, into which all the training and testing data should be projected and at last, distance would be calculated between image region and the face region in the “eigenface space”. This approach grapples the most distinctive differences between face and nonface, and meanwhile yields dimension reduction. In 1997, Schölkopf extended the normal PCA method by using kernel, and proposed a new KPCA method [88, 90–92], which play the same role as PCA but in much higher dimension feature space.

Figure 2.5: Comparison of 5 classes by Lu. (a) PCA-based subspace. Different classes's data are mixed together. (b) D-LDA-based subspace. (c) Proposed DF-LDA-based subspace. Obviously, DF-LDA can separate different classes much better than the other two.

PCA is a relatively simple, fast algorithm and has been shown to work well in some particular environment. However, in the cases that the distribution structures of different classes are similar, PCA would not be an ideal method to extract effective features, Figure 2.5. Therefore, linear discriminant analysis (LDA) [4, 34, 53, 56, 111] is proposed as a substitute method to perform dimensionality reduction while preserving as much class discriminatory information as possible. In [53], Lu et al. proposed a new algorithm that deals with both of the shortcomings of LDA in an efficient and cost effective manner. Their results indicated that the proposed method was superior to those of traditional FR approaches, such as the eigenfaces, fisherfaces, and D-LDA methods, Figure 2.5. Recently, Jing et al. [34] had proposed an improved LDA approach to overcome the efficiency and principle components selection problems of LDA. The limitation of LDA is brought from the its definition. First of all, it can only produces at most $c - 1$ features projections, where c is the number of different classes. Second, it is ineffective for distributions that deviate significantly from unimodal Gaussian distribution, and third, it is ineffective when the discriminatory information is not contained in the mean but rather in the variance of the data.

The appeal of wavelets (multiresolution theory) [2, 55, 105, 106] is obvious: features that might go undetected at one resolution may be easy to spot at another. The early research of wavelets in object detection was done by Papageorgiou and Oren et al. [66, 70, 71]. First, they used the wavelets to encode the difference in average intensity between local regions along different orientations, in a multiscale framework. The strong response from a particular wavelet indicates the presence of an intensity difference, or boundary, at that location in the image while weak response from a wavelet indicates a uniform area. Second, they used overcomplete Haar basis to propagate constraints between neighboring regions and describe complex patterns. Via this method, they got a rich, overcomplete dictionary of features. From these features, they manually chose 37 significant coefficients in the proper configuration.

Figure 2.6: The 37 significant coefficients chosen by Papageorgiou.

Extending from Haar wavelet, Viola et al. [117, 118] introduced Haar-like features in their rapid object detection scheme. They used a set of features which look like Haar Basis functions. In order to compute these features very rapidly at many scales they used the integral image representation, which can be computed from an image by using a few operations per pixel, for images. Once computed, any one of these Haar-like features can be computed at any scale or location in constant time. Based on these features, they applied AdaBoost algorithm for classification problem. Later, Lienhart et al. [46] extended the Haar-like features to rotated ones.

Figure 2.7: (a) Haar wavelet used by Papageorgiou et al.; Haar-like features (b) Proposed by Viola et al. (c) Proposed by Lienhart et al.

2.2.3.2 Distribution-Based

An intuitive solution to the problem of modeling faces is to allow the detector to project a tested image onto a learned subspace and determine whether or not it is close to that subspace. Eigenface, wavelet methods adopt such a scheme, and Sung and Poggio [110] also developed an adaptation of this scheme to create a detection method using Gaussian clusters in $\mathcal{R}^n$. Using a set of Gaussians, they estimated density functions for face and nonface patterns that learned from a set of training data. Their systems consisted of two stages: distribution-based method and a multilayer perceptron classifier, which will be illustrated in the next section. They grouped the face and nonface patterns into six face and six nonface clusters using a modified k -means algorithm. Figure 2.8 shows the clusters used by them. Each cluster is represented as a multidimensional Gaussian function with a mean image and a covariance matrix. Figure 2.9 illustrates the distance measures in their method. Two distance metrics are computed between an input image pattern and the prototype clusters. The first distance component is the normalized Mahalanobis distance between the test pattern and the cluster centroid, measured within a lower-dimensional subspace spanned by the cluster's 75 largest eigenvectors.

$$\mathcal{D}_1(\vec{x}, \vec{\mu}) = \frac{1}{2}(75\ln 2\pi + \ln|\sum_{75}| + (\vec{x} - \vec{\mu})^T \sum_{75}^{-1}(\vec{x} - \vec{\mu})). \quad (2.1)$$

Figure 2.8: Face and nonface clusters used by Sung and Poggio [110].

The second distance component is the Euclidean distance between the test pattern and its projection into the 75-dimensional subspace.

$$\mathcal{D}_2(\vec{x}, \vec{\mu}) = |(\vec{x} - \vec{x}_p)|^2 = |(I - E_{75}E_{75}^T)(\vec{x} - \vec{\mu})|^2. \quad (2.2)$$

This distance component accounts for pattern differences that are not captured by the first distance component. The last step is to use a multilayer perceptron network to classify face window patterns from nonface patterns using twelve pairs of distances to each face and nonface cluster.

Figure 2.9: The distance measures used by Sung and Poggio [110].

In [60, 61], Moghaddam et al. also presented an unsupervised technique for visual

learning using the distribution idea. It is a method that is based on density estimation in high-dimensional spaces using an eigenspace decomposition. Compared with the classic PCA method, the proposed method shows better performance in face representation.

2.2.3.3 Component-Based

In contrast to model the global appearance of a face like [110], Schneiderman and Kanade [96, 97] proposed a statistical classifier, to estimate the joint probability of local appearance and position of face patterns (subregions of the face) at multiple resolutions. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes are much more distinctive than the pattern found around the cheeks.

In [62], Mohan developed the idea of local area, and presented a general example-based framework for detecting objects in static images by components. The system is structured with four distinct example-based detectors that are trained to separately find the four components of human body: the head, legs, left arm, and right arm, Figure 2.10 (a). Geometric constraints, Figure 2.10 (b), would be applied on the detected components. Although it is a method that used to detect human body, later, Bileschi [5] extended it to face detection. Figure 2.11 (a) shows the 14 components used in his component-based face detection system arranged in a geometrically salient and vaguely disturbing pattern. After biasing, the corresponding components would be presented as a white spot in the corresponding position in the result images, Figure 2.11 (b).

Figure 2.10: (a) Diagrammatic description of the operation of Mohan's system. (b) Geometric constraints that are placed on the different components.

Figure 2.11: (a) The fourteen components used on Bileschi's component based face detection system.(b) The three right images are the results images after greedy optimization (brow, nose, and mouth).

2.2.3.4 Hidden Markov Model

Hidden Markov model [63,64,80] is another face representation method. The underlying assumption of the HMM is that patterns can be characterized as a parametric random process and that the parameters of this process can be estimated in a precise, well-defined manner. Firstly, a face pattern can be divided into several regions such as the forehead, eyes, nose, mouth, and chin. A face pattern can then be recognized by a process in which these regions are observed in an appropriate order (from top to bottom and left to right). Instead of relying on accurate alignment as in template matching or appearance-based methods (where facial features such

as eyes and noses need to be aligned well with respect to a reference point), this approach aims to associate facial regions with the states of a continuous density Hidden Markov Model. HMM-based methods usually treat a face pattern as a sequence of observation vectors where each vector is a strip of pixels, as shown in Figure 2.12.

Figure 2.12: (a) Observation vectors. (b) Hidden states.

2.2.3.5 Trace Transform

Recently, Petrou et al. proposed a novel representation method which is called trace transform [37, 73, 104]. It is a generalization of the Radon transform. It consists of tracing an image with straight lines along which certain functionals of the image function are calculated. Different functionals that can be used may be invariant to different transformations of the image. With the Trace transform, the image is transformed into another “image” which is a 2D function depending on parameters (ϕ, p) that characterize each line, Figure 2.13 (b). Figure 2.13 (a) defines these parameters. Parameter t is defined along the line with its origin at the foot of the normal.

In [104], Srisuk and Petrou et al. applied Trace transform method in face aspect and introduced a new face representation method called shape Trace transform (STT). Their idea is to change the traditional pixel-to-pixel comparisons of face images to the shape Trace transform comparisons. Figure 2.14 shows examples from the masked Trace transform. In [104], they applied simple thresholding on the original

Figure 2.13: (a) Definition of the parameters of an image tracing line. (b) A 19×19 image and its Trace transform.

Figure 2.14: Extracting shapes from the masked Trace transform in [104].

Trace transform image, and produced a binarised version of the Trace transform. The shapes of the outlines of the extracted regions may be used for the authentication task. Figure 2.14 shows the process to get STT images. Although Srisuk and Petrou et al. only used this method in face authentication aspect, however, based on the representative shape information provided by STT for each face, this method can also be applied in face detection.

2.3 Classifier

The task of classifier is to use the feature vector provided by the feature extractor to assign the object to a category. In the development of face detection techniques, different kinds of classifiers have been used. The ideas behind these methods are

various, but all of them are trying to find algorithms that can minimize the classification error rate. Broadly, the classification problem can be attributed to the following aspects.

2.3.1 Matching

Matching is a classification method that used the idea to minimize the Euclidean distance of testing images and training images, or maximize the correlation values of testing images and training images. It is the simplest classification method. In face detection, a face is detected if the faceness measure is above a predefined threshold [85,101,112]. In [84], Sakai et al. used correlation value to detect candidate locations of faces. Then, they performed the matching with the other subtemplates at the candidate positions. It is an idea to focus on attention of region of interest. The idea is adopted by later researchers. Differing from Sakai who used correlation values, Turk and Pentland [113] minimized the Euclidean distance between the PCA coefficients of faces. Although the matching approach has the advantage of being simple to implement, it is quite sensitive to image rotation, illumination, and image scale changes.

2.3.2 Decision Tree

Decision tree (DT) learning [76–78] is one of the most widely used and practical methods for inductive classification. By using DT algorithm, each branch of the tree can grow deep enough to perfectly classify the training examples. While this is sometimes a reasonable strategy, in fact it can lead to difficulties when there is noise in the data, or when the number of training examples is too small to produce a representative sample of the true target function. Also, DT is quite susceptible to overfitting. Therefore, in 1987, Quinlan [77] proposed ID3 to avoid this problem.

DT algorithm was firstly applied to face detection by Huang et al. [32]. Their approach for face detection involved three main stages: location, cropping, and post processing. The C4.5 algorithm [79], which is an outgrowth of the original ID3 algorithm [77], was used in the second stage to detect face. Their algorithm did not require multiple templates and was proposed achieving 96% accuracy rate. But the testing data used to verify the system is FERET, which only includes the upper body of each person and has monotone background. Therefore, the result cannot show the robustness of their system. Amit et al. [1] also presented a face detection system based on CART algorithm (classification and regression trees). It was applied to grow a classification tree from the training images and a collection of false positives identified from the generic background images. Therefore, given a test image, regions of interest are identified from the spatial arrangements of edge fragments. Each region of interest is then classified as face or background using the learned CART tree. Their experimental results on a set of 100 images from the Olivetti (now AT&T) data set report a false positive rate of 0.2 percent per 1,000 pixels and a false negative rate of 10 percent.

2.3.3 Probabilistic Approach

Many face detection classification approaches can be understood as probabilistic framework. A feature vector can be viewed as a random variable x , and this random variable is characterized for faces and nonfaces by the class-conditional density functions $p(x | \textit{face})$ and $p(x | \textit{nonface})$. Therefore, Bayesian classification or maximum likelihood can be used to classify a candidate image location as face or nonface. One problem of probabilistic approach is the requirement of prior knowledge of the probability of face and nonface.

In the work of Yow et al. [123], they proposed a probabilistic framework to assign and propagate probabilistic among the facial features and face groups so that

they could achieve a high confidence rate for true positive faces. They made use of Bayesian network (or belief networks), Figure 2.15, which are directed acyclic graphs, to propagate evidence. Belief networks have nodes representing random variables and arcs signifying direct dependencies specified in terms of conditional probabilities. Each node can take either of 2 values, True or False, and has a conditional probability table describing the conditional probability of each value given each possible combination of the values of the parent nodes. A belief network is initialized for each face candidate and virtual nodes are created for each feature and component face groups that is found in the process. The evidence for each facial feature or face group i is related to its Mahalanobis distance, M_{ij} , and the admission threshold for the j th feature class, τ_j , by: $P_i = (1 - \frac{M_{ij}}{\tau_j})$. Each facial feature that is detected is assigned 4 probability values, P_{brow} , P_{eye} , P_{nose} , P_{mouth} , using the above equation. When a higher level group is formed, only the probability of the corresponding feature is propagated. Likewise, only these values are updated in the propagation process. As a result, only true positive faces are updated to a high confidence level. Face variations in scale, orientation and viewpoint are treated in a subsequent paper [124,125].

Figure 2.15: (a) Yow's belief network. (b) Yow's reinforcement belief network.

The classification problem of HMM face modeling also belongs to probability aspect. One can train HMM to learn the transitional probability between states from the examples where each example is represented as a sequence of observations. The goal of training an HMM is to maximize the probability of observing the training data by adjusting the parameters in an HMM model with the standard Viterbi segmentation method and Baum-Welch algorithms [80]. After the HMM has been trained, the output probability of an observation determines the class to which it belongs. Nefian et al., in [65], used maximum likelihood to train the embedded HMM face modeling. They proposed a re-estimation equations for the maximum likelihood training algorithm. Compared with the other methods for face detection and recognition, their method provided a much higher accuracy.

In [94–97], Schneiderman et al. described a statistical method for 3D object detection. They represented the statistics of both object appearance and the “non-object” appearance using a product of histograms. Each histogram represented the joint statistics of a subset of wavelet coefficients and their position on the object. Their approach used many such histograms to represent a wide variety of visual attributes. Using this method, they developed the first algorithm that can reliably detect human faces with out-of-plane rotation and the first algorithm that can reliably detect passenger cars over a wide range of viewpoints. In particular, a object region is extracted, if satisfy Equation 2.3.

$$\frac{\prod_{x,y \in \text{region}} \prod_{k=1}^{17} P_k(\text{pattern}_k(x,y)|\text{object})}{\prod_{x,y \in \text{region}} \prod_{k=1}^{17} P_k(\text{pattern}_k(x,y)|\text{non-object})} \geq \lambda \quad (2.3)$$

where λ is the likelihood ratio threshold, and equals to

$$\lambda = \frac{P(\text{non-object})}{P(\text{object})}. \quad (2.4)$$

However, the high dimensionality of images makes structure learning especially chal-

lenging. In [93], Schneiderman et al. described an algorithm that searches for the structure of a Bayesian network based classifier in this large space of possible structures. The algorithm seeks to optimize two cost functions: a localized error in the log-likelihood ratio function to restrict the structure and a global classification error to choose the final structure of the network. The final network structure is restricted such that the search can take advantage of pre-computed estimates and evaluations. Similar to Schneiderman et al.'s work, Liu [49] also proposed a novel Bayesian Discriminating Features (BDF) method for multiple frontal face detection. Their Bayes classifier was claimed to achieve 98.5 percent face detection accuracy with one false detection.

Based on the method developed by Schneiderman et al. [95], Verma et al. [116] presented a new probabilistic method for detecting and tracking multiple faces in a video sequence. The three novel contributions of their method were: 1) Accumulation of probabilities of detection over a sequence. 2) Prediction of the detection parameters which are position, scale, and pose. 3) The representation of pose is based on the combination of two detectors, one for frontal views and one for profiles. The probabilities of detection are propagated over time using a Condensation filter and factored sampling. Prediction is based on a zero order model for position, scale, and pose; update uses the probability maps produced by the detection routine. The proposed method can handle multiple faces, appearing/disappearing faces as well as changing scale and pose. The same work also had been done by Loutas et al. [51, 52] who had presented a joint probabilistic face detection and tracking algorithm combining likelihood estimation and a prior probability.

2.3.4 Neural Networks

Neural network has been proven to be quite useful for pattern recognition problems. As face detection can be treated as a two class pattern recognition problem, various

neural network architectures have been proposed in this area, [7, 17–19, 36, 47, 75]. The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one drawback is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance.

Rowley *et al.* presented the most significant multilayer neural network for face detection. in [82]. Their upright frontal face detection system operates in two stages: It firstly applies a set of neural network-based filters to an image and then uses an arbitrator to combine the outputs. The filters examine each location in the image at several scales, looking for locations that might contain a face. First, a preprocessing step, adapted from [110], is applied to a window of the image. Then the preprocessed window is passed through a neural network. The network has retinal connections to its input layer. There are three types of hidden units: four which look at 10×10 pixel subregions, 16 which look at 5×5 pixel subregions, and six which look at overlapping 20×5 pixel horizontal stripes of pixels. Each of these types is chosen to allow the hidden units to detect local features that might be important for face detection. In particular, the horizontal stripes allow the hidden units to detect such features as mouths or pairs of eyes, while the hidden units with square receptive fields might detect features such as individual eyes, the nose, or corners of the mouth. After getting the raw output from single networks, they went to the second stage of the NN systems: merging overlapping detections from a single network and arbitrating among multiple networks. A threshold is applied to the output of single network, and the position of positive ones is obtained, and the centroids of all above threshold regions are computed. All detections contributing to a centroid are collapsed down to a single point. Each centroid is then examined in order, starting from the ones which had the highest number of detections within the specified neighborhood. If any other centroid locations represent a face overlapping with the current centroid, they are removed from the output pyramid. All remaining cen-

troid locations constitute the final detection result. After this they apply multiple networks and arbitrate between their outputs to produce the final decision. Simple arbitration schemes such as logic operators (AND/OR) and voting are used. Later, Rowley et al. extended this method to detect rotated faces by firstly preprocessing the face window (rotated one) to a canonical orientation (upright frontal) [83]. However, the detection rate decreased seriously to 76.9%.

2.3.5 Support Vector Machine

Support vector machine was introduced by Vapnik in the late seventies [115]. This algorithm has excellent accuracy, which is the most remarkable. Unlike the other high performance classifiers, it does not include a priori knowledge about the problem, such as probability approaches. The study of support vector learning is very useful in two respects. First, it is quite satisfying from a theoretical point of view: SV learning bases on some beautifully simple ideas and provides a clear intuition of what learning from examples is about. Second, it can lead to high performances in practical applications. Therefore SV algorithm is considered as lying at the intersection of learning theory and practice. Real-world applications often mandate the use of more complex models and algorithms-such as neural networks-that are much harder to analyze theoretically. The SV algorithm achieves both. It constructs models that are complex enough: it contains a large class of neural nets, radial basis function (RBF) nets, and polynomial classifiers as special cases. Yet it is simply enough to be analyzed mathematically, because it can be shown to correspond to a linear method in a high-dimensional feature space nonlinearly related to input space. Moreover, even though we can think of it as a linear algorithm in a high-dimensional space, by the use of kernels, all necessary computations are performed directly in input space. This is the characteristic twist of SV methods-we are dealing with complex algorithms for nonlinear pattern recognition, but for the sake of

analysis and algorithmics, we can pretend that we are working with a simple linear algorithm. In [9, 88], Burges and Scholkopf gave good overview of the concepts of SVMs and different applications of using SVMs are even given out by Scholkopf. During last twenty years, numerous methods have been proposed to speed up SVM algorithm, [8, 10, 35, 67, 74], meanwhile many applications are proposed based on such algorithm [3, 15, 57]

Figure 2.16: A separating hyperplane, written in terms of a weight vector w and a threshold b . x_1 and x_2 are support vectors.

The first one to apply SVM to face detection is Osuna et al. in 1997 [68, 69], and followed up with work by Romdhani et al. [81]. The key to the model for a support vector machine is the choice of a hyperplane that separates the face set from the non-face set, Figure 2.16. In [69], a hyperplane is chosen, specifically the hyperplane which maximizes minimum distance on either side. A support vector set is, roughly speaking, a set of vectors (images) which are close to this hyperplane, and can therefore be used on their own to reacquire the hyperplane. In [81], an attempt to improve the performance of a similar system is made via reduced set vector machines. The methods in both papers use quadratic programming heavily, and exploit properties of their models' kernel functions. In terms of training sets, Osuna et al. exploit their model and the fact that most vectors will be ignored, or at least meaningless, in their quadratic programming formulation. Because the hardware requirements

for training a support vector machine in a natural way are prohibitive, training data must be chosen in a nontrivial manner. First, a set of optimality conditions, specifically the Kuhn-Tucker conditions, are considered. Only those vectors which are relevant to the training, i.e. support vectors, are used. Memory requirements are quadratic in the size of this working vector set, so obviously minimizing it is key. The proposed solution is to decompose the problem into smaller sub-problems, a standard solution to such a problem when doing so is possible. Romdhani et al. [81] work further on reducing this vector set in order to improve performance.

2.3.6 AdaBoost

AdaBoost algorithm is introduced by Freund and Schapire in [87]. Unlike previous algorithms [20, 48], it adjusts adaptively to the errors of the weak hypotheses returned by WeakLearn, which is denoted a generic weak learning algorithm. It is Viola who first applies AdaBoost in face detection problem [118]. The overall form of the detection process of his classifier is that of a degenerate decision tree, what we call a “cascade” (see Figure 2.17). A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. Stages in the cascade are constructed by training classifiers using AdaBoost and then adjusting the threshold to minimize false negatives. Note that the default AdaBoost threshold is designed to yield a low error rate on the training data.

Recently, Li et al. proposed a novel learning procedure, called FloatBoost, for learning a boosted classifier for achieving the minimum error rate in face detection [45]. FloatBoost learning uses a backtrack mechanism after each iteration of AdaBoost learning to minimize the error rate directly, rather than minimizing an exponen-

Figure 2.17: Schematic depiction of a the detection cascade in [118].

tial function of the margin as in the traditional AdaBoost algorithms. It requires fewer weak classifiers than AdaBoost yet achieves lower error rates in both training and testing. They claimed that the FloatBoost learning method, together with a proposed detector pyramid architecture, leads to the first real-time multiview face detection system reported.

2.4 Performance Comparison Between Face Detection Methods

Although numerous face detection algorithms have been developed, most of their experimental results are reported using different test sets, or under different controlled conditions. To get an equal comparison, tests of different face detection methods which use the same training and testing database are required. In particular, in this comparison, the training data includes 50 face images and 100 nonface images, and the testing data includes 20 face images and 100 nonface images. The size of these two data set is both 19×19 .

2.4.1 Performance Comparison Between Different Face Representations

First of all, the performance comparison between different face representations will be done. In this comparison, the system accuracy as well as the false positive, false negative numbers will be used to evaluate the performance of different face representation methods. As to the classification algorithm, the same 2 degree polynomial SVM classifier will be used. (except template representation and HMM face modeling methods.)

In template representation, the mean face of all the face training data will be used as the predefined face template, and the minimum squared Euclidean distance (MSED) is used to do classification problem rather than the SVM classifier as others. In value-based representation, four different kinds of methods are compared. Thereinto, one level Haar wavelet decomposition are performed, and 4 degree polynomial are chosen as the kernel in KPCA method. The components that are selected in component-based representation is shown in Figure 2.18. The face is detected if any three of these four component detectors claiming detection of the components. In trace transform, the Radon transform is used as the trace function, and $360/128 = 1.8125$ degrees to sample a full circle (refer to Figure 2.13 (a)). Table 2.1 is the result of this comparison. From the results it can conclude that KPCA feature extraction method gains the highest detection accuracy, and Haar wavelet representation method has the smallest false negative number. Template representation is the most inaccuracy method for face detection. This is the natural result, because compared with other methods, template is more sensitive to variations in scale, pose, and shape of faces. Due to the particularly small size of our training images, it is not suitable to use feature-based representation of face, because this kind of method requires the size of training and testing images to be larger than a particular value, which depends on the performance of a detection system. Therefore, based on the experimental results,

Table 2.1: Performance comparison between different face representations. The amount of testing data is 120; it includes 20 face images and 100 nonface images.

Representation Methods		Detection Rate	False Negative No.(Total 20)	False Positive No.(Total 100)
Template representation+MSED		89.17%	4	9
Value-based	Gray value	95%	4	2
	Haar wavelet	95.83%	3	2
	PCA	95%	4	2
	KPCA	96.67%	4	0
Distribution-Based		92.5%	4	5
Component-Based		90%	7	5
HMM+probability		94.2%	4	3
Trace transform		90%	4	8

the example-based representation will be employed. In particular, the combination of an extended KPCA and Haar wavelet representation methods is used to get the global structure and local details information of faces.

Figure 2.18: Component selection.

2.4.2 Performance Comparison Between Different Classifiers

To test the performance between different classifiers, the same face representation method will be used as inputs to different classifiers. Concretely, gray value of images will be used as the inputs of different classifiers. Detection accuracy, false negatives, and false positives will be used to evaluate the performance comparison.

Table 2.2: Performance comparison between different Classifiers. The amount of testing data is 120; it includes 20 face images and 100 nonface images.

Classifiers	Detection Rate	False Negative No.(Total 20)	False Positive No.(Total 100)
Matching (minimum squared Euclidean distance)	85.83%	2	15
Decision tree	86.7%	10	6
Probabilistic (Bayes classifier)	85.83%	17	0
Neural network	92.5%	4	5
SVM	95%	4	2
AdaBoost	96.67%	2	2

As to the matching aspect, the squared Euclidean distance (SED) between testing data and training data will be computed. The same category as a training data will be assigned to a testing data if the testing data's SED with this training data is the smallest one compared with all the other training data. In probabilistic approach, Bayes classifier are used as the representative method. As an example of neural network, a feed-forward backpropagation network will be used. Concretely, a two layers network are chosen. 50 neurons are used in the first layer and 1 in the second. The 10 weak learners used in the AdaBoost classifier belongs to radial-based function (RBF), while the kernel in SVM classifier belongs to 2 degree polynomial. From the comparison results, Table 2.2, it can be concluded that AdaBoost with RBF weak learner gets the highest accuracy. Among these 6 methods, Bayes classifier is the slowest one, because it has to compute the inverse of covariance matrix whenever a testing data inputs. Compared with others, 2 degree polynomial SVM classifier is a relatively fast and accurate one. In the proposed robust face detection system, it will be adopted as classifier.

2.5 Conclusion

This chapter presents the main techniques applied to detection of faces from feature extraction aspect and classification aspect. For each technique, the main principle is outlined, as well as its merits and demerits. Performance comparisons between different feature extraction methods and different classifiers are also presented. Based on the motivation of making use of the local distinguished characteristics of human faces and the their global structure distribution, we adopt example-based representation method using the combination of an extended KPCA and Haar wavelet representation. As to the classification aspect in the proposed detection system, a relatively simple support vector machine is adopted.

Chapter 3

Haar Wavelet Coefficients Face Representation

3.1 Introduction

Although face is a clear conception for human beings, it is not an easy task for a machine to learn this concept. First of all, human faces are diverse and subject to changes such as lighting, expression, and rotation, which give the task of face concept learning direct difficulties. Moreover, variable background also can seriously affect the face representation; for example, if the color of background is similar to skin color, then it will cause the contrast of face and background to be weakened. Therefore it would make the color-based face detection method useless.

Even though it is difficult to construct face representation for a machine, the basic idea of seeking face representation does exist; that is to find representation that can achieve high inter-class variability with low intra-class variability from these changeable face images. Obviously, the most easily obtained features, image pixel values, are not good representation of faces, because values of image pixels will change dra-

matically in different lighting intensity or between different human races, etc. In such a situation, even intra-class variability of these face images is very large. On the other hand, the widely used traditional feature extraction method, edge detection, is not suitable to be used in cluttered images neither, because the edges produced by cluttered background maybe more drastic than the facial edge. Therefore, the inter-class variability of image edges is very small. Wavelets, however, can capture visually plausible features of the facial features and interior structure of faces which are invariant to certain transformations. The result of wavelet transformation is a compact representation where dissimilar object images from the same class will map to similar feature vectors. In these feature vectors, zero values represent uniform face regions and those non-zero values denote facial features' boundaries. Therefore, we choose wavelet as one part of the face representation. More concretely, we adopted Haar wavelet, because it is the simplest wavelet which can transform fastest and still can give a good result in our face detection system.

3.2 Related Works

As briefly introduced in 2.2.3.1, wavelet is an attractive feature extraction method. It receives much attention and development in pattern recognition after Papageorgiou et al., who first applied an overcomplete Haar wavelet to objects detection. Thereafter, different kinds of wavelet have been introduced to solve the face detection problem. The most popular one is the wavelet packet analysis. Garcia et al. [21] used it and quantized skin color regions merging in face detection problem. Similarly, Gundimada et al. used the same wavelet to propose a rotation invariant wavelet features. Unlike previous methods, which used a relatively complicated wavelet, we aim to use the simplest one, Haar wavelet. Because compared with others, it is the fastest one.

3.3 Principle of Haar wavelet transformation

The principle of Haar wavelet transformation in a two dimensional image can be illustrated by decomposition in two vertical one dimensional images respectively. To get a sense for how wavelets work, we give a simple example. Suppose we are given a one-dimensional “image” with a resolution of four pixels, having values $[9\ 7\ 3\ 5]$. We can represent this image in the *Haar basis* by computing a wavelet transform. To do this, we first average the pixels together to get the new lower resolution image with pixel value $[8\ 4]$. Clearly, some information has been lost in this averaging process. To recover the original four pixel from the two averaged values, we need to store some *detail coefficients* $[1\ -1]$ ($9-8 = 1$, $8-7 = 1$, $3-4 = -1$, $4-5 = -1$), which capture the missing information. Thus we have decomposed the original image into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition:

Resolution	Averages	Detail coefficients
4	$[9\ 7\ 3\ 5]$	
2	$[8\ 4]$	$[1\ -1]$
1	$[6]$	$[2]$

Finally, we will define the wavelet transform of our original four-pixel image to be the single coefficient $[6]$ representing the *overall average* of the original image, followed by the *detail coefficients* $[2\ 1\ -1]$ in order of increasing resolution. Thus, for the one-dimensional Haar wavelet transform of original four-pixel image is given by $[6\ 2\ 1\ -1]$.

From the table, we can know that a one-pixel image is just a constant over the entire interval (We can assume the interval is $[0, 1)$). Let V^0 be the vector space of all these functions. A two-pixel image has two constant pieces over the intervals $[0, 1/2)$ and $[1/2, 1)$. Let V^1 be the vector space of all these functions. If we continue

in this manner, the space V^j will include all piecewise-constant functions defined on the interval $[0, 1)$ with constant pieces over each of 2^j equal subintervals. The basis functions for the spaces V^j are called *scaling function*. A simple basis for V^j is given by the set of scaled and translated function:

$$\phi_{j,i}(x) := \phi(2^j x - i), \quad i = 0, \dots, 2^j - 1, \quad (3.1)$$

where

$$\phi(x) := \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{otherwise.} \end{cases}$$

Figure 3.1 shows the four functions which form a basis for V_2 .

Figure 3.1: Haar basis for V^2 , where $j = 2, i = 0, 1, 2, 3$.

After defining the simple basis, the next step is to map the input image data on to this basis. Mathematically, it is an inner product problem that defines on the vector spaces V^j . The “standard” inner product

$$\langle f|g \rangle := \int_0^1 f(x)g(x)dx, \quad (3.2)$$

for two elements $f, g \in V^j$ will do quite well for running example. We can now define a new vector space W^j as the orthogonal complement of V^j in V^{j+1} . In other words, we will let W^j be the space of all functions in V^{j+1} that are orthogonal to all functions in V^j under the chosen inner product. A collection of linearly independent functions $\psi_i^j(x)$ spanning W^j are called wavelets. The wavelets corresponding to the box basis are known as the *Haar wavelets*, given by

$$\psi_{j,i}(x) := \psi(2^j x - i), \quad i = 0, \dots, 2^j - 1, \quad (3.3)$$

where

$$\psi(x) := \begin{cases} 1, & 0 \leq x < \frac{1}{2} \\ -1, & \frac{1}{2} \leq x < 1 \\ 0, & \text{otherwise.} \end{cases}$$

Figure 3.1 shows the two Haar wavelets spanning W_1 .

Figure 3.2: Haar wavelets for W^1 , where $j = 1, i = 0, 1$.

Let's now apply these more sophisticated ideas to the former example. We begin by expressing our original image $I(x)$ as a linear combination of the basis functions in V^2 :

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x) \quad (3.4)$$

Note that the coefficients $c_0^2, \dots, c_3^2$ are just the four original pixel value [9 7 3 5].

Then we rewrite the expression for $I(x)$ in terms of basis functions in V^1 and W^1 , using pairwise averaging and differential:

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (3.5)$$

These four coefficients are the lower resolution image's pixel values c_0^1, c_1^1 ([8 4]) plus their detail coefficients d_0^1, d_1^1 ([1 -1]).

Finally, we'll rewrite $I(x)$ as a sum of basis functions in V^0 , W^0 and W^1 :

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (3.6)$$

Coefficients $c_0^0, d_0^0, d_0^1, d_1^1$ are [6 2 1 -1]. These four coefficients are the Haar wavelet transform of the original image. The four functions which shown above constitute the Haar basis for V^2 . Instead of using the usual four scaling functions in V^2 , we can use $\phi_0^0(x), \psi_0^0(x), \psi_0^1(x), \psi_1^1(x)$ to represent the overall average, the broad detail, and the two types of finer detail possible in a function in V^2 .

In preparation for image decomposition, we need to generalize Haar wavelets to two dimensions. Thus the scaling function can be defined as

$$\phi(x, y) = \phi(x)\phi(y), \quad (3.7)$$

and the three wavelet functions are,

$$\psi^1(x, y) = \phi(x)\psi(y),$$

$$\psi^2(x, y) = \psi(x)\phi(y),$$

$$\psi^3(x, y) = \psi(x)\psi(y).$$

More details of wavelet transform can be attained in [55].

3.4 Face representation using Haar wavelet transformation

Regarding learning, an image's two dimensional Haar wavelet coefficients can be achieved through two steps by using horizontal and vertical one dimensional Haar wavelet transformation on this image consecutively. As can be seen from Figure 3.3, low frequency filter $\phi(m)$ and high frequency filter $\psi(m)$ will first be applied along the horizontal of the image. Therefore an original 19×19 image turns into two 19×10 subimages, where one is the low frequency response of $\phi(m)$, and one is the high frequency response of $\psi(m)$. Thereafter, these two outputs will be used as inputs of two vertical filters, that is $\phi(n)$ and $\psi(n)$. Finally, through these two steps, we get four 10×10 subimages from the original input image, which can be observed from Figure 3.3. To facilitate illustration, we can denote these four subimages as LL, LH, HL, HH. In particular, LL represents low-pass filtering in the horizontal direction and low-pass filtering in the vertical direction too. That is the lower resolution coefficients which capture the overall coarse shape information. LH, which denotes low-pass filtering in the horizontal direction and high-pass filtering in the vertical direction, can preserve the detail information in horizontal direction. HL, on the other hand, represents vertical detail information, and HH represents

diagonal features. Figure 3.4, shows the 1 level Haar wavelet transformation, which we adopted in our proposed method. Concretely, the size of training data does not satisfy the 2^j condition [22]. Therefore, based on some compensation, the total size of the four output images is 20×20 which is different from 19×19 . From Figure 3.5, it can be observed that the dissimilar face images are mapped into similar Haar coefficients feature vectors. This is a characteristic of good representation. Observed from these examples, the detailed information of eye sockets, eyebrows, mouth and tip of the nose are preserved quite well in the LH and HL subimages.

Figure 3.3: Two dimensional Haar wavelet transformation of face images..

Figure 3.4: 1 level Haar wavelet representation of an image.

To get a deep understanding of the detail preserving capability of Haar wavelet coefficients, the raw values of LH and HL subimages are shown. From Figure 3.6, the values inside ellipse is relatively larger than the others. From (a), three parts of

Figure 3.5: The dissimilar face images are mapped into similar Haar coefficients feature vectors.

this image contain densely large values, and their positions in image correspond to left eye, right eye, and mouth respectively. From (b), the large values concentrate in the position of nose. Regarding to the previous learning, large values of wavelet responses represent the boundary of facial features. Therefore, the details of faces are preserved by different subimages respectively.

0.2381	0.3175	0.6190	0.8333	0.1032	0.4286	0.4921	0.4762	0.3175	0.0952
0.1111	0.2143	0.1667	0.0159	0.0952	0.1667	0.6508	0.1746	0.1032	0.0952
0.1270	0.0397	0.0317	0.1587	0.0079	0	0.3413	0.1032	0.2540	0.0317
0.1429	0.2143	0.2540	0.0238	0.0079	0.0079	0.0794	0.1270	0.0952	0.2540
0.0873	0.0476	0.0476	0.0079	0.1508	0.0556	0.1270	0.0238	0.0079	0
0.0794	0.0556	0.0714	0	0.4206	0.2778	0.1032	0.0079	0.0159	0.0159
0.1111	0.0317	0.2857	0.3651	0.2381	0.2222	0.1905	0.4921	0.0635	0.1270
0.1111	0.1190	0.0556	0.4841	0.5079	0.5952	0.3492	0.0873	0.0317	0.1587
0.1032	0.0317	0.2381	0.4286	0.4841	0.3829	0.1587	0.0556	0.1349	0
0	0	0	0	0	0	0	0	0	0

(a)

0.2063	0.0952	0.0635	0.0476	0.0556	0.4444	0.0635	0.0794	0.0000	0
0.0635	0.5635	0.1508	0.2063	0.4762	0.2778	0.9476	0.0317	0.1032	0
0.1905	0.1508	0.0000	0.1429	0.5794	0.5556	0.2778	0.0397	0.0476	0
0.3175	0.1190	0.0317	0.0873	0.5159	0.3254	0.2063	0.0635	0.3175	0
0.0714	0.1111	0	0.3889	0.5635	0.5635	0.2698	0.0079	0.1190	0
0.1746	0.0556	0.0873	0.7143	0.5000	0.4841	0.4524	0.0556	0.1111	0
0.1587	0.1270	0.1746	0.1746	0.1429	0.2540	0.1111	0.1587	0.0635	0
0.2857	0.1190	0.1349	0.1349	0.0952	0.1508	0.0317	0.1349	0.0794	0
0.0556	0.0952	0.0159	0.0476	0.0397	0.0079	0.3651	0.1032	0.3413	0
0.1111	0.0635	0.3016	0.1111	0.0794	0.1111	0.1746	0.2540	0.3016	0

(b)

Figure 3.6: (a) LH subimage raw values. The large values rounded by three ellipses concentrate in the position of left eye, right eye, and mouth in image respectively. (b) HL subimage raw values. The densely large values concentrate in the position of nose.

3.5 Conclusion

This chapter presents the representation for facial detail and local shape information by using Haar wavelet. From the illustration, it can be shown that wavelet transform can preserve the distinction among gray value that is useful for most pixel-based classifiers. This characteristic is related to the intrinsic property of wavelet transforms that preserve high and low frequency features during the signal decomposition. However, most classification errors occur at the boundary between classes. Therefore, since wavelet decomposition is applied to each local pixel, a wavelet-based feature extraction might not well differentiate classes among neighborhood pixels in the spatial domain. In order to construct a representative face feature, further techniques should be done based on these extracted wavelet coefficients.

Chapter 4

Sub-kernel Principle Component Analysis Facial Feature Extraction Method

4.1 Introduction

Although Haar wavelet is an ideal feature extraction method that can provide overall coarse shape information plus details of facial features, it only applies on local pixels, and does not describe the face global structure distribution information. Therefore, a wavelet-based feature extraction method only may not be effective for the classification problem. In the proposed method, further technique, SKPCA, is required to be based on these extracted wavelet coefficients. SKPCA, an extension of KPCA, is a method that can provide more facial global information than wavelet. By carefully combining SKPCA and wavelet techniques, we propose a novel face representation that takes advantage of the distinguished characteristics of human face, like eyebrows, eyes, nose and mouth, and as well as the face global structure distribution.

4.2 Related works

4.2.1 Principle Component Analysis in Face Detection

Mathematically, the basic idea behind kernel principle component analysis (KPCA) and principle component analysis (PCA) is similar, and that, KPCA is a nonlinear extension of PCA. Therefore, to review the development of KPCA, we cannot desert the development of PCA. The earliest attentions at using PCA can be cast back to Turk and Pentland's work in 1991. In [113], they first introduced PCA algorithm for face representation, and proposed the eigenface concept. The attraction of this method is that the eigenface can reduce feature vectors dimension while still preserving the major information of training data. Therefore it is a quite good face representation which can solve the computational problem of face detection. Although Turk and Pentland's work was proposed for face recognition, their idea was soon popular in face detection field. Many works are proposed based on PCA or modified PCA, [12, 29, 33, 58, 99, 108]. For example, Hotta in [29] proposed a view invariant face detection method based on local PCA cell. In order to extract the general features of faces at each view and position, Gabor filters and local PCA are used. Local PCA cells specialized to each view and position are made by applying Gaussian to the outputs of local PCA of Gabor features. By applying the Gaussian, only the local PCA cells which are similar view to an input give large values. This decreases the undesirable influence of the local PCA cells of other views. As a result, only one classifier can treat multi-view faces well by integrating the outputs of local PCA cells. Although this method can detect multiple view of faces, however, like other methods, it only uses several classifiers trained on different view points of training data to accomplish this task. Therefore, the proposed face representation method only cannot solve the multipose face problem.

Recently, PCA was extended into two-dimension [121]. As opposed to PCA, 2DPCA

is based on 2D image matrices rather than 1D vectors so the image matrix does not need to be transformed into a vector prior to feature extraction. Compared with traditional 1DPCA, 2DPCA has many advantages. In the first place, since 2DPCA is based on the image matrix, it is simpler and more straightforward to be used for image feature extraction. Second, 2DPCA is better than PCA in terms of recognition accuracy in all experiments. Third, 2DPCA is computationally more efficient than PCA and it can improve the speed of image feature extraction significantly. However, it should be pointed out that 2DPCA-based image representation was not as efficient as PCA in terms of storage requirements, since 2DPCA requires more coefficients for image representation than PCA.

4.2.2 Kernel-based Feature Extraction in Face Detection

Although PCA is an effective face representation, it may not perform well in the aspect of classification. Therefore, Schölkopf proposed kernel-based extension of PCA feature extraction method, [88–90,92]. It is a method that will first nonlinear map data into a much higher feature space before doing principle components extraction. Usually, it is used in recognition aspect, [50,88], this is because kernel-based feature extraction is a computationally expensive method. The burden of its computational problem will increase with the increase of training data. Therefore, KPCA is not quite popular in the face detection aspect where the training data is much larger than that in recognition.

However, in our proposed method, we introduce a sub-kernel principle component analysis method (SKPCA) to reduce the computational problem of KPCA. In the process of SKPCA, a simple and effective subset choosing method is used to choose the representative training data to construct the SKPCA feature space. Prior to the illustrating of the construction of SKPCA feature space, let's first briefly go through the principle of kernel and KPCA.

4.3 Kernel

In this section, we will study kernels k that correspond to dot products in feature space $\mathcal{H}$ via a map Φ (Suppose the input domain is $\mathcal{X}$). Mathematically, we can write the above transformation using the following note:

$$\begin{aligned}\Phi : \quad \mathcal{X} &\rightarrow \mathcal{H} \\ x &\mapsto \mathbf{x} := \Phi(x),\end{aligned}\tag{4.1}$$

that is,

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle .\tag{4.2}$$

Concretely, let's see the essence of kernel construction. Suppose we are given patterns $x \in \mathcal{X}$ where most information is contained in the d th order products (so-called monomials) of entries $[x]_j$ of x ,

$$[x]_{j_1} \cdot [x]_{j_2} \cdot \dots \cdot [x]_{j_d},\tag{4.3}$$

where $j_1, \dots, j_d \in \{1, \dots, N\}$. Often, these monomials are referred to as *product features*. These features form the basis of many practical algorithms; indeed, there is a whole field of pattern recognition research studying *polynomial classifiers*, which is based on extracting product features first and then applying learning algorithms to these features. In other words, the patterns are preprocessed by mapping into the feature space $\mathcal{H}$ of all products of d entries. This has proven quite effective in visual pattern recognition tasks, for instance. To understand the rationale of this, note

that visual patterns are usually represented as vectors whose entries are the pixel intensities. Taking products of entries of these vectors then corresponds to taking products of pixel intensities, and is thus akin to taking logical “and” operations on the pixels. Roughly speaking, this corresponds to the intuition that, for instance, a handwritten “8” constitutes an eight if there is a top circle *and* a bottom circle. With just one of the two circles, it is not half an “8”, but rather a “0”.

Let us take a look at this feature map in the simple example of two-dimensional patterns, for which $\mathcal{X} = R^2$. In this case, we can collect all monomial feature extractors of degree 2 in the nonlinear map

$$\begin{aligned}\Phi : R^2 &\rightarrow \mathcal{H} = R^3 \\ [x]_1, [x]_2 &\mapsto ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2).\end{aligned}\tag{4.4}$$

Figure 4.1 gives an example of the above mapping. From this figure, we can see the advantage of product features: mapping the input data to feature space $\mathcal{H}$, a nonlinear separable problem is reduced to a linear superable problem. By doing this, it not only has advantages on the computational point of view, but also has advantages on the statistical point of view (For example, it is easy to construct classification algorithms).

We now describe how dot products in polynomial feature spaces $\mathcal{H}$ can be computed efficiently. We employ kernel representations

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle,\tag{4.5}$$

which allow us to compute the value of the dot product in $\mathcal{H}$ without having to explicitly compute the map Φ . Now let’s see why this can work out successfully.

Figure 4.1: Toy example of a binary classification problem mapped into feature space. The true boundary can be assumed an ellipse in the input space (left space). The task of the learning process is to estimate this boundary based on empirical data consisting of training points in both classes (plus and circles, respectively). When mapped into feature space via the nonlinear map $\Phi_2(x) = (z_1, z_2, z_3) = ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2)$ (right panel), the ellipse boundary becomes a flat hyperplane boundary problem.

We start by giving an example for $N = d = 2$, as considered above. For the map

$$\Phi : ([x]_1, [x]_2) \mapsto ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2), \quad (4.6)$$

dot products in $\mathcal{H}$ take the form

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle = [x]_1^2[x']_1^2 + [x]_2^2[x']_2^2 + 2[x]_1[x]_2[x']_1[x']_2 = \langle x, x' \rangle^2, \quad (4.7)$$

In other words, the desired kernel k is simply the square of the dot product in input space. The same works for arbitrary N, d . Therefore, defining $\Phi_d(x)$ whose entries are all possible d th degree products of the entries of x , we have:

$$k(x, x') = \langle \Phi_d(x), \Phi_d(x') \rangle = \langle x, x' \rangle^d. \quad (4.8)$$

In more realistic cases, e.g., if x represents an image with the entries being pixel values, polynomial kernels $\langle x, x' \rangle^d$ enable us to work in the space spanned by products of any d pixels values - provided that we are able to do our work solely in terms of dot products, without any explicit usage of a mapped pattern $\Phi_d(x)$. For more detail of kernels, please refer to [91]. Factually, based on this definition, k is a matrix that represents the similarity of each two input data. In particular, given m input data, k will be a $m \times m$ matrix,

$$\begin{aligned}
 k &= \begin{bmatrix} k_{x_1x_1} & k_{x_1x_2} & \cdots & k_{x_1x_m} \\ k_{x_2x_1} & k_{x_2x_2} & \cdots & k_{x_2x_m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{x_mx_1} & k_{x_mx_2} & \cdots & k_{x_mx_m} \end{bmatrix} \\
 &= \begin{bmatrix} \langle x_1, x_1 \rangle^d & \langle x_1, x_2 \rangle^d & \cdots & \langle x_1, x_m \rangle^d \\ \langle x_2, x_1 \rangle^d & \langle x_2, x_2 \rangle^d & \cdots & \langle x_2, x_m \rangle^d \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_m, x_1 \rangle^d & \langle x_m, x_2 \rangle^d & \cdots & \langle x_m, x_m \rangle^d \end{bmatrix}, \quad (4.9)
 \end{aligned}$$

in which the entries with larger value represent high likeness of each two input data and those with smaller value represent low likeness of each two input data.

4.4 KPCA feature extraction

Impressed by the merit of product features and kernel, we adopt this method to extract the ideal features of faces image. Here, we use the kernel PCA, which will extract face images' principal component in feature space $\mathcal{H}$.

The basis principle of kernel PCA is essentially like its linear counterpart PCA. Still,

we consider the problem of face detection. Firstly, we map the input face image data into a feature space:

$$\Phi : \mathcal{X} \rightarrow \mathcal{H}, x \rightarrow \Phi(x), \quad (4.10)$$

After this, we use the centered data (assume that we are dealing with centered data, $\sum_{i=1}^m \Phi(x_i) = 0$.) to calculate the covariance matrix who takes the form:

$$C = \frac{1}{m} \sum_{j=1}^m \Phi(x_j) \Phi(x_j)^T. \quad (4.11)$$

Then we have to find eigenvalues $\lambda \geq 0$ and eigenvectors $\mathbf{v} \in \mathcal{H}$ satisfying

$$\lambda \mathbf{v} = C \mathbf{v}. \quad (4.12)$$

Combining (4.11) and (4.12), we know that all solutions $\mathbf{v}$ with $\lambda \neq 0$ lie in the span of $\Phi(x_1), \dots, \Phi(x_m)$. So instead we can consider the set of equations

$$\lambda \langle \Phi(x_n), \mathbf{v} \rangle = \langle \Phi(x_n), C \mathbf{v} \rangle, \quad n = 1, \dots, m, \quad (4.13)$$

and assume that

$$\mathbf{v} = \sum_{i=1}^m \alpha_i \Phi(x_i). \quad (4.14)$$

Combining (4.13) and (4.14), we get

$$\lambda \sum_{i=1}^m \alpha_i (\Phi(x_n) \cdot \Phi(x_i)) = \frac{1}{m-1} \sum_{i=1}^m \alpha_i (\Phi(x_n) \cdot \sum_{j=1}^m \Phi(x_j) (\Phi(x_j) \cdot \Phi(x_i))), \quad n = 1, \dots, m.$$

Substituting the above function using kernel $k_{ij} := \langle \Phi(x_i) \cdot \Phi(x_j) \rangle$, we get

$$m\lambda\alpha = k\alpha. \quad (4.15)$$

Then we can get the eigenvalues λ and corresponding eigenvectors α . For the purpose of principle component extraction, we need to compute projections onto the eigenvectors $\mathbf{v}$ in $\mathcal{H}$ (From (4.14), we know that $\mathbf{v} = \sum_{i=1}^m \alpha_i \Phi(x_i)$). Let the test image x map to $\mathcal{H}$, then calculate

$$\langle \mathbf{v}, \Phi(x) \rangle = \sum_{i=1}^m \alpha_i^n k(x_i, x). \quad (4.16)$$

For more details about the algorithm of KPCA and how to get the centered data in $\mathcal{H}$, please refer to [88, 89].

4.5 SKPCA Construction

In section 4.3, we have discussed how dot product in polynomial feature space $\mathcal{H}$ can be computed efficiently by using kernel representation,

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \langle x_i, x_j \rangle^2. \quad (4.17)$$

Consequently, kernel PCA is computational comparable to a linear PCA on m observations with a $m \times m$ dot product matrix, because of computing the value of the dot product in $\mathcal{H}$ without having to explicitly compute the map Φ . However, in the case where we need to use a large number of observations m , k would not be easy to compute. Moreover, whenever import a testing data, its extracted wavelet coefficients should be always multiplied with k before principle components extrac-

Figure 4.2: Support vectors are defined as the data that nearest to the decision boundary. Based on the theory of structural risk minimization [8], if we discard the other non-SVs data from training set, the decision boundary constructed by the remaining data will be the same as the one constructed by all training examples, [88]. Therefore, SVs can represent original data well in classification problem.

tion, and evaluation times will be m (See equation $\langle \mathbf{v}, \Phi(x) \rangle = \sum_{i=1}^m \alpha_i^n k(x_i, x)$). This is an arduous job when classifier has to scan the whole image in order to find all faces contained in it. Therefore, we intend to engender a subset containing the representative observations of the original training data, and use this subset to estimate kernel k . Unlike Schölkopf [88], who also had computed k using a subset while still extracting principal components from all m observations, we will only use the chosen subset to construct the eigen space.

In our method, we construct a subset by using support vectors (SVs) which are chosen by support vector machine (SVM) algorithm, and enlarge it to some extend by Euclidean distance. From [8], support vectors are defined as the data that nearest to the decision boundary. Thus, based on the propose of structural risk minimization, only these support vectors need to be considered when classifying classes [8]. So if we discard the other non-SVs data from training set, the decision boundary constructed by the remaining data will be the same as the one constructed by all training examples (Figure 4.2). This shows that SVs can represent original data well in classification problem. One of the method to find SVs is support vector machine algorithm. Here, we adopt polynomial SVM. Experiment result showed that these SVs could do classification problem well (If we use these SVs to test on

Figure 4.3: Enlarge the chosen subset using Euclidean distance.

testing database, the detection rate is high.). However, every classifier has bias; the extracted SVs may not include all the genuine SVs. In order to reduce the classification error by only using extracted SVs to estimate kernel k , we will use Euclidean distance between SVs and training data, which has excluded those extracted SVs, to enlarge the subset amount in order to insure that the chosen subset can include as many genuine SVs as possible, Figure 4.3. The Euclidean distance used in our method has following format:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^m (x_{1i} - x_{2i})^2}. \quad (4.18)$$

Finally, gathering all these chosen data, we construct a subset, whose amount is l , where $l \ll m$. The kernel created by this subset will become

$$\hat{k}(x_i, x_j) = \langle \Phi(x)_i, \Phi(x)_j \rangle, \text{ where } i, j = 1, \dots, l. \quad (4.19)$$

And the principal components can be extracted by

$$l\hat{\lambda}\hat{\alpha} = \hat{k}\hat{\alpha}. \quad (4.20)$$

To evaluate the chosen subset and extracted principal components, we have com-

Figure 4.4: Example of grip-search. Based on the calculated detection rate, we can choose the best pair of parameters of SVM algorithm.

pared the sub-kernel PCA with kernel PCA in classification task. Experimental results show that the sub-kernel PCA can speed up feature extraction process with no loss in accuracy.

When considered the computational price of this SKPCA feature extraction method, the burden is quite different in training and testing process. In training process, which needs to create a suitable SVs subset to construct Kernel eigen space, the computational burden is quite heavy. First of all, in order to find the most representative SVs, we adopted grid-search method [30], which requires numerous calculations. Let us take Figure 4.4 as an example. Firstly, we should painstakingly calculate the detection rate using many different pairs of parameters, for example, d and c in the polynomial SVM algorithm; and then carefully decide which pair of parameters can produce the best result, which means the corresponding SVs extracted through this pair of parameters are the most representative subset of original training data. After that, in order to enlarge the chosen SVs, Euclidean distance is used. It is known that the computational burden of Euclidean method is quite heavy [16].

However, after the representative subset has been chosen. To construct SKPCA features in testing process becomes an easy task, because the computational problem in training will not impact the testing process. In testing, we only make use of the

constructed sub-kernel space to extract the representative SKPCA features of testing data. It does not require subset-chosen techniques like those in the training process. Moreover, the dimension of this sub-kernel eigen space is much smaller than the one constructed using original training data. Therefore, in testing process, the detection time is greatly shortened.

4.6 Face Representation Using SKPCA

After illustrating the principle of sub-kernel PCA, we present how to extract face feature using this algorithm. Figure 4.5 shows the first part of SKPCA algorithm in MATLAB environment. After reading in a 19×19 image, one level Haar Wavelet transformation will be applied on it, and get 4 10×10 subimages. In our method, we make use of all these four outputs, and normalize their coefficients values to $[0, 1]$.

Till now, we finish the first step of our proposed wavelet-based SKPCA method - the detail extraction of facial characteristics. After this, SKPCA will be applied on these extracted Haar wavelet coefficients. First of all, we use polynomial SVM to find the support vectors of all the training Haar wavelet coefficients, line 8 in Figure 4.5 shows the idea in MATLAB program. Following this step, we use trial and error method to choose the appropriate Euclidean distance between SVs and the Haar wavelet coefficients training data. In particular, the training data, which has the distance with SVs between $[1.8, 6]$, will be chosen as an entries of the chosen subset. lines 9-21 in Figure 4.5 shows this idea in MATLAB program. Eventually, the amount of chosen Haar wavelet coefficient images, namely the chosen subset, is 837.

After these representative data is ready, the SKPCA feature space can be constructed by them. Figure 4.6 is the standard code of feature space's construction in MATLAB. Line 6-7 in Figure 4.6 show the centralize of training data. It is a

```

%READ TRAINING DATA IMAGES;
... ..
1    image=double(imread(fileName))/255;

%ONE LEVEL HAAR WAVELET TRANSFORMATION
2    [Haar_Coef,S_image] = wavedec2(image,1,'haar');

%NORMALIZE HAAR COEFFICIENTS TO [0 1]
3    e = 10^(-10);
4    fmin = min(Haar_Coef);
5    fmax = max(Haar_Coef);
6    Haar_Coef = (Haar_Coef-fmin+e)/(fmax-fmin+e);

%SUBSET:
%STEP 1: USE SVM CLASSIFIER TO CHOOSE SUBSET
7    Bias=1;
8    [AlphaY, SVs, Bias, Parameters, nSV, nLabel] = PolySVC(Haar_Coef, Train_Label, 12, Bias);

%STEP 2: USE EUCLIDEAN DISTANCE TO ENLARGE THE SVs SUBSET
... ..
9    sub_f = zeros;
10   dis = Haar_Coef - SVs;
11   Norm_Dis = norm(dis);

%THE EUCLIDEAN DISTANCE BETWEEN Haar_Coef AND SVs SHOULD BETWEEN
%[1.8, 6] (based on the trial and error), sub_f STORES THE FINALE SUBSET
12   IF Norm_Dis <= 6
13       D = sub_f(:,k)-wout_svsl(:,j);
14       Norm_D = norm(D);
15       IF Norm_D >= 1.8
16           sub_f = [sub_f,wout_svsl(:,j)];
17           k = k+1;
18       END
19   ELSE
20       sub_hist_im = [sub_hist_im,wout_svsl(:,j)];
21   END

```

Figure 4.5: Algorithm for wavelet transformation and subset chosen in MATLAB environment.


```

%CONSTRUCT THE SKPCA FEATURE SPACE
1  [eigvect,eigval, oneM, K, Kt, nx, degree]=kernel_pca(sub_f,2);

    %FUNCTION: kernel_pca
2  FUNCTION [eigvect,eigval, oneM, K, Kt, nx, degree]=kernel_pca(x,degree)
3      tol=1e-10; %tolerance on zeroness of eigenvalue
4      nx=size(x,1);
5      K =(x*x'+1).^degree;
6      oneM=ones(nx,nx)./nx;
7      Kt=K-oneM*K-K*oneM+oneM*K*oneM;
8      [eigvect,eigval]=eig(Kt/(nx-1));
9      eigval=real(diag(eigval));
10     nbeigval=length(find(abs(eigval)>tol)); % keeping only eigval higher than tol
11     eigvect=eigvect(:,1:nbeigval);
12     eigval=eigval(1:nbeigval);
13
14     % SORT THE EIGENVECTOR
15     [junk,rindices] = sort(-1*eigval);
16     eigval = eigval(rindices);
17     eigvect = eigvect(:,rindices);
18     for i=1:nbeigval % normalizing eigenvector
19         eigvect(:,i)=eigvect(:,i)/sqrt(eigval(i));
20     end;

```

Figure 4.6: Algorithm for SKPCA feature space construction in MATLAB environment.

Figure 4.7: Algorithm for test data’s SKPCA feature extraction in MATLAB environment.

necessary step in SKPCA algorithm, more detail of its operation refer to [92]. Concretely, the kernel we chosen is a 2 degree polynomial one; line 5 in Figure 4.6 shows its format.

Therefore, after the constructed SKPCA feature space, SKPCA features of a test image x can be calculated via $\sum_{i=1}^{837} \hat{\alpha}_i \hat{k}(x_i, x)$. Figure 4.7 shows test data feature extraction in MATLAB environment. Similar to construction step, the testing data should also be centralized first. From line 8 in Figure 4.7, we can find that test data’s SKPCA features are achieved by mapping the centralized test data into the constructed feature space. Figure 4.8 illustrated the total process of wavelet-based SKPCA feature extraction method.

Figure 4.8: The process of constructed sub-kernel PCA.

4.7 Conclusion

This chapter presents SKPCA feature extraction method working on Haar wavelet coefficients. Relative works in this aspect have been introduced. Moreover, the principle of kernel and its advantage, as well as the principle of KPCA have been illustrated. In this chapter much attention has been paid on SKPCA construction aspect. Detail of face representation using SKPCA is illustrated in the last part. The algorithm of wavelet-based SKPCA in MATLAB environment is also given to illustrate the process of construction.

Chapter 5

Example-based Face Detection System

In the previous two chapters, we have paid much attention on wavelet-based SKPCA feature extraction method, but a complete example-based face detection system will include another two operations: preprocessing and classification (see Figure 1.3). To understand the procedure of designing such a system, we must know the operations of each component in turn.

5.1 Preprocessing

In order to reduce the effect of illumination and noise, and enhance contrast of images, preprocessing step is very important for face detection. Usually normalization will be done in this step. There are many ways to normalize image, such as normalizing the face size and the face intensity, adjusting each image so that the standard deviation is similar, and normalizing the vectorized face and non-face images so that each is a unit vector. In our preprocessing step, we use the following two ways to

preprocess input images:

1. **Normalize to unit vector:** The value range of intensity images is $[0, 255]$. We use algebra method to adjust the range to $[0, 1]$.
2. **Histogram equalization:** This operation can enhance the contract of image, so that the imaging effects due to changes in illumination, brightness and differences in camera response curves can be compensated. The result of this processing is shown in Figure 5.1. After doing this step, images' standard deviations are around a constant.

Figure 5.1: Histogram equalization is used on the left image. We can observe from the right one the contract of a face is enhanced after this processing.

5.2 Classifier - Support Vector Machine

The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category. Here we adopted SVM classifier. In the following section, we will recall the basic notions of the theory of SVM.

Consider a two-class classifier for which the decision rule takes the form:

$$y = \text{sgn}\left(\sum_{i=1}^{N_s} \lambda_i K(\mathbf{x}, \mathbf{x}_i) + b\right), \quad (5.1)$$

where $\mathbf{x}, \mathbf{x}_i \in \mathbf{R}^d$, $\lambda_i, b \in \mathbf{R}$; $\mathbf{x}_i, \lambda_i, b$ and N_s are parameters and $\mathbf{x}$ is the vector to

be classified. The decision rule for a large family of classifiers can be cast in this functional form: for example, $K = \langle \mathbf{x} \cdot \mathbf{x}_i \rangle^p$ implements a polynomial classifier; $K = \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2 / \sigma^2)$ implements a radial basis function machine [9, 115].

The support vector algorithm is a principled method for training any learning machine whose decision rule takes the form (5.1): the only condition required is that the kernel K satisfies a general positivity constraint [9, 115]. In contrast to other techniques, the SVM training process determines the entire parameter set $\{\mathbf{x}_i, \lambda_i, b, N_s\}$; the resulting $\mathbf{x}_i, i = 1, \dots, N_s$ are a subset of the training set and are called *support vectors*.

Figure 5.2: This is an example of the architecture of support vector machines. Here 7, 1, 4, 1 are the 4 support vectors, and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the corresponding weights. After calculating the weights, a decision function is computed to class the input vector x : $f(x) = \text{sgn}(\sum_{i=1}^4 \lambda_i k(x, x_i) + b)$.

Support vector machines have a number of striking properties. The training procedure amounts to solving a constrained quadratic optimization problem, so the solution found is guaranteed to be the unique global minimum of the objective func-

Figure 5.3: Training and testing steps of face detection system. The bootstrapping training step will improve the system performance.

tion. SVM can be used directly to implement Structural Risk Minimization, in which the capacity of the learning machine can be controlled so as to minimize a bound on the generalization error. A support vector decision surface is actually a linear separation hyperplane in a high dimensional space. Figure 5.2 is the architecture example of SVM.

5.3 Example-based Face Detection System Training Strategy

Figure 5.3 shows the stages of our face detection system. In the initial training stage, we use MIT CBCL database, which include 2429 face training data and 4548 nonface

training data. In this process, three major steps exist: the first step is to construct SKPCA feature space (refer to Chapter 4). The second step is to mapped the Haar wavelet coefficients of training data to the feature space (use Equation (4.16)); and the third one is to train a two degree polynomial SVM classifier. The output of the training stage is the entire trained parameter set of SVM classifier, $\{\mathbf{x}_i, \lambda_i, b, N_s\}$. After the classifier is trained, it can be used to detect faces in arbitrary images. Figure 5.4 shows a slightly more elaborate diagram of the components of a face detection testing system. Obviously, before being input to a trained classifier, any testing image has to undergo the same preprocessing and feature extraction steps as the training stage, and at the same time rescaling and scanning of the test image.

Figure 5.4: Retrain and testing stage. In the retrain process, the input image is any image that doesn't contain faces. The false outputs will be used as input to train classifier. In the testing process, the output will determine whether there is a face in the image. If it has, then draw a frame around the face image.

5.3.1 Discussions

5.3.1.1 Reinforcement Learning

In the example-based classification, the concept of face class is clear, but the concept of nonface class is much more blurry. This is due to the fact that nonface class includes “the rest of the world”. Therefore, in order to overcome the definition

problem of this extremely large category, we adopt the “bootstrapping” training method, also called reinforcement learning, which is similar to [110]. Observed from Figure 5.3, scene images will be tested on the roughly trained face detection system. False positives will occur, then we manually label these false alarms as nonface class, and use them to retrain the classifier again. From the experimental results, the system performance increase gradually from 96.6% detection rate to 98.1% detection rate. This process will be repeated, until a satisfactory performance is achieved.

5.3.1.2 Multiple Detection Problem

In addition to the blurry definition of nonface class, multiple detection of the same face in nearby locations is another problem. Figure 5.5 illustrates multiple detection of the same face in nearby locations by the proposed system.

Figure 5.5: Multiple detection of the same face in nearby locations.

To avoid this problem, the position of the mass center of each connective region, which has reported detections of face, will be introduced as the final position of detected face. Mathematically, the definition of mass center can be defined as follow:

The centroid of a set of n point masses m_i located at positions x_i is

$$\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}. \quad (5.2)$$

Therefore, applying this concept in gray value image, we can find a region's mass center by using the position and gray value information of those region points:

$$\begin{cases} \bar{x} = \frac{\sum_{i=1}^n v_i x_i}{\sum_{i=1}^n v_i} \\ \bar{y} = \frac{\sum_{i=1}^m v_i y_i}{\sum_{i=1}^m v_i} \end{cases},$$

where v_i is the gray value of point (x_i, y_i) , $(\bar{x}, \bar{y})$ is the mass center position.

Usually, an image may contain more than one face, therefore, points that report detections of a same face should be attributed to same region. For example, there are two regions in Figure 5.6 (a) that report detection of two faces. The method to separate all these points into two regions is their 8-connection relationship [22]. Points in the same region should satisfy 8-connection relationship, which are all the connective points in a region. Therefore, using this idea, the points that report detections of faces are separated into two regions as in Figure 5.6 (a); this suggests that two different faces are contained in the same image. Thereafter, we can compute the mass center of each region to find the final position of a face, Figure 5.6 (b).

5.4 Conclusion

Using the proposed wavelet-based SKPCA feature extraction method, an example-based face detection system is proposed in this chapter. To improve its accuracy, reinforcement training is adopted in our training strategy. Moreover mass center conception is also used to solve the problem of multiple detections of a same face in nearby locations. The evaluation of its performance and comparisons with other

approaches will be given in the next chapter through experimental results.

Figure 5.6: (a) Multiple detections of two faces in two regions. (b) Mass centers of two different regions are the final positions of the detected faces.

Chapter 6

Experimental Results

This chapter presents the experiments and analyses of the example-based face detection system which has proposed in the precious chapter. The criteria for performance evaluation include:

1. The detection rate of the system should be maximized.
2. The false negative rate or false positive rate should be minimized based on the application purpose.
3. The robustness of the system should be large. For example, the scope testing on lighting condition and rotation should be good; faces with different expressions are able to be detected.
4. The computational cost and storage of the system should be small.

In this chapter, the training and testing data will be described first. And then the experimental results on reinforcement training will be presented, as well as:

1. comparisons of different feature extraction methods,
2. comparisons of testing set-1,

3. comparisons of different methods in face detection based on robust performance based on testing set-2 and set-3,
4. scope testing on different lighting conditions and rotation angles.

6.1 Training and testing data

In training process, the representative of training data is very important, typical positive and negative examples of faces and nonfaces are required in the training process. Here, we chose MIT CBCL face database, which was once used by Edgar Osuna et al. [69] and Sung et al. [110] in their face detection systems. This database includes 2429 faces and 4548 nonfaces training data and 24045 testing data. All the face images are upright or nearly upright frontal ones. And examples are given in Figure 6.1. The nearly upright faces are constructed from rotating by $\pm 5^\circ$ of the upright faces. Mirror reflections and zooming in and out by a constant factor are also processed to create more data [110]. Nonface training data, Figure 6.2, is gathered by randomly sampling patterns in images that do not contain face.

Figure 6.1: Examples from the database of faces used for training. The images are gray level of size 19×19 .

The testing data in CBCL are much diverse than training data. Seen from Figure 6.3, they contain some large rotation faces and low quality images.

Beside this test database, which we name set-1, we also construct another two

Figure 6.2: Examples from the database of nonfaces used for training. The images are gray level of 19×19 .

Figure 6.3: Examples of CBCL test database, set-1. This test set includes 472 face images and 23573 nonface images.

testing data sets. Set-2, consisting of all the frontal face images from the FERET database, contains mainly head and shoulder pictures as shown in Figures 6.4 and 6.5. Set-3 consists of images chosen from the MIT-CMU test sets. The chosen images contain different faces in clutter background. Examples are shown in Figure 6.6. In particular, set-2 consist of 736 images, and set-3 consists a total of 84 images which contain a total of 315 faces. Note that these testing data are much more diverse than training ones, therefore the classifier trained on a simple image set, yet works well on much complex images, displays robust generalization performance.

6.2 Experimental results

After the introduction of training and testing database, experimental results of wavelet-based SKPCA face detection classifier will be presented. First of all, let's give the performance result of reinforcement training.

Figure 6.4: Face detection examples from set-2 without wearing glasses. A square indicates a face region successfully detected. The resolution of the images is 256×384 .

Figure 6.5: Face detection examples from set-2 with glasses.

6.2.1 Performance of Reinforcement Training

In our example-based classification problem, due to the fact that nonface class includes “the rest of the world”. Therefore, although we have had the nonface database, we still want to enlarge the nonface data set by some typical nonface samples. Similar to [110], we used a trained classifier to classify scenical and ar-
chitectonical images. Then the false positive detections will be treated as nonface training data in next training cycle. Table 6.1 shows the training data’s cross-validation results in different retraining cycles. Conceptually, cross validation is

Figure 6.6: Face detection examples from set-3.

a model evaluation method. Concretely, we use 5-fold cross validation. Here the training data set, m in total, is randomly divided into 5 disjoint sets of equal size $\frac{m}{5}$. The system is trained 5 times, each time with a different set held out as validation set. The estimated performance is the mean of these 5 errors (more information can be referred to [16]). Through the estimated results, we can see that enlarging the nonface training examples can help to improve the detection rate on validation data.

Table 6.1: Performance comparison between different Classifiers. The original size of training data is 6977.

Times to retrain	Number of total retraining data	Detection rate
Retrain 1	7147	96.13%
Retrain 2	7201	96.92%
Retrain 3	7221	97.466%

Figure 6.7: A 19×19 mask to eliminate the near-cheeks pixels.

6.2.2 Performance comparison between different feature extraction methods

In this subsection, classification results of wavelet-based SKPCA SVM classifier will be presented. Firstly, we would like to compare the performance of the wavelet-based SKPCA feature extraction method with the performances of Schlkopf's KPCA and wavelet-based KPCA feature extraction method.

From the simulation results, the wavelet-based SKPCA feature extraction method takes only one fifteenth time of the Schlkopf's KPCA and wavelet-based KPCA feature extraction method, and has the highest detection rate based on set-1, which is 98.16% compared with 98.12% detection rate resulting from wavelet-based KPCA method and 97.6% detection rate resulting from Schlkopf's KPCA method. These results show that the wavelet-based SKPCA feature extraction method can speed up testing time with no loss in detection rate.

In order to get a whole sense of the advantage of the wavelet SKPCA feature extraction method, we will compare it with different feature extraction methods using the same SVM classifier. When extracting other features, like gray value, wavelet, PCA coefficients etc., we will use some more different preprocessing in order to improve the detection results of these methods. In addition to the former two preprocessing methods, normalizing to unit vector and histogram equalization, we will also do the following two preprocessings:

1. **Masking:** Similar to the preprocessing idea done by K. K. Sung *et al.* [110],

after observing the CBCL database, we would like to use a 19×19 mask (Figure 6.7) to eliminate the near-cheeks pixels, which may belong to background or have no important features. After masking, the dimension is reduced from 361 to 331.

2. **Rescale:** Because the raw values of images may vary excessively, thus there may exist greater numeric ranges dominating those in smaller numeric ranges. So when dealing with the comparison, we rescale the gray value to the range $[0, 1]$. We first calculate the maximum coefficient α and minimum coefficient β in all the corresponding coefficients including both face and nonface. Then the slope of this linear transform can be computed like this:

$$tg = 1/(\alpha - \beta + \varepsilon) \quad (6.1)$$

In order to avoid dividing by zero, we invite ε , a minimal integer, to calculate the slope. Obviously the truncation point is

$$b = -tg \times \beta. \quad (6.2)$$

Using these two parameters, we can map every image's coefficients to $[0, 1]$.

$$rescalecoe = tg \times originalcoe + b. \quad (6.3)$$

Table 6.2 shows the comparison results. Observed from the data, the propose method presents a good detection rate, especially reduces the false negative rate.

Table 6.2: Comparisons of different feature extractor. See from the data, our proposed method presents a good detection rate, especially reduces the false negative rate. The total amount of face testing data is 472.

Features	Detection Rate	False Negative Number
Wavelet-based SKPCA	98.16%	272
Wavelet-based KPCA	98.12%	286
Schlkopf's KPCA	97.6%	352
Mask Histogram equalization scale Gray Value	98.04%	340
Mask Gray Value	97.779%	380
Haar Wavelet Coefficients	98.10%	343
PCA Coefficients	97.55%	372

6.2.3 Experimental performance in testing data

In succession, we show the experimental results of our system on the 3 constructed datasets: set-1, set-2, set-3. Set-1, which comes from the same database as training set, contains 472 face testing data and 23573 nonface testing data. The image sizes are all 19×19. The ratio of the face and nonface testing data is close to a normal image which contains large background with fewer faces. These training and testing databases are also used by Edgar Osuna et al. [69] and Sung et al. [110]. Table 6.3 gives the testing results of our method and the two existed methods on testing data set-1.

Table 6.3: Comparative face detection performance of the Edgar Osuna et al., Sung et al. and Wavelet-based sub-KPCA SVM method on testing data set-1, which contains 472 faces and 23573 nonfaces images.

Method	Detection rate
Edgar Osuna et al.	97.1%
Sung et al.	94.6%
Wavelet-based SKPCA SVM	98.16%

The frontal faces in set-2 are all detected. Figure 6.4 shows some examples of the

Figure 6.8: Detection of large faces in set-3. The large face could be detected is up to 210×210 .

detected faces in set-2, where a square indicates a face region successfully detected. Figure 6.5 shows face detection examples from set-2 with glasses. The resolution of all these images is 256×384 .

To test the generalization performance of wavelet-based SKPCA method, a third testing data set, set-3, is created from the MIT-CMU test set [82]. Note that the training face images of our method are from only one database, but the test images in set-3 are from diverse sources: Some of the images are from the World Wide Web, some are scanned from photographs and newspaper pictures, and some are digitized from broadcast television [82]. Some images contain very small faces (Figure 6.9) or very large faces (Figure 6.8); some include old people faces (Figure 6.10); some have faces occluded by hand, drip-drop, glasses, etc (Figure 6.10); some consist of faces

Figure 6.9: Detection of small faces in set-3. The smallest size of faces can be detected is 11×11.

Figure 6.10: Detection of old people's faces in set-3. See from the images, old people's faces usually have wrinkle, and fleck. The face features in these people are quite different than the younger ones who have a "clear" faces.

Figure 6.11: Detection of occlude faces in set-3. In these images, faces features contain glasses, and mustache. Some of the faces are also occluded by water drops or by hands.

in different expressions and lighting conditions (Figure 6.12 and 6.16); yet some involve low quality face images (Figure 6.13). Experiments based on such a simple training set and such a diverse testing set should be able to test the generalization performance of the proposed method. The experimental results show that 305 faces were detected (The total number of faces in set-3 is 315.).

Figure 6.6 shows the results of detection of multiple frontal faces. The proposed method successfully detected all the faces except a face which is largely occluded by hair in the middle of the bottom line. In particular, all the 20 faces in Figure 6.6 are detected at the scales 22, 43, 48, and 180, respectively. Note that the scale 22 means that the original image is resized by a ratio, $\frac{19}{22}$. Figures 6.8 and 6.9 show the detector can detect faces ranging from scale 11 to 210.

Wavelet-based SKPCA method, trained on only simple data, can also detect old people faces, which have many wrinkle, and fleck. The detection of old people's faces is much more difficult than young people's, because the facial features of these people contain more noise than the younger "clear" ones. See from Figure 6.10, the proposed system is robust to faces occluded by hand, raindrop, glasses, etc. This good performance of the system is caused by the characteristic of SKPCA method, because the SKPCA can extract the most important principal components of faces.

Figure 6.12: Detection of faces in set-3 in different expressions.

Figure 6.13: Detection of blurred faces in set-3.

Figure 6.14: Rotate faces in set-3. However, because it is a classifier trained on frontal faces, its capacity to detect rotated face is limited.

Therefore, some of the noise in face images are discarded. Our face detection system was quite successful in detecting faces with different expressions. From Figure 6.12, we can see that no matter whether the people are talking, thinking, laughing, or crying, they can be detected successfully.

Figure 6.15: Scope testing on rotation angles in-plane and in-depth. A square indicates a face region successfully detected. (a) The largest angle of rotated face in-plane is around 18°. (b) The largest angle of rotated face in-depth is around 35°.

Our system also can detect rotated faces. Set-3, MIT-CMU test set, contains many rotated faces. See from the testing experiments, we can know that faces in certain

Figure 6.16: Detection of faces in set-3 in different light conditions.

Figure 6.17: Light source comes from frontal. The corresponding lighting intensity is showed in the above row. From left to right, light intensity is descend. When the light intensity becomes too low, false detection appears (see the fifth image).

rotation range can be detected. We construct two experiments to test the in-plane and in-depth rotation ranges using ORL database. From Figure 6.15 (a) and 6.15 (b), we know that this detection system can deal with the rotation tasks that are up to about 35° in depth and 18° in plane. The image in the upper left corner of Figure 6.12 shows an face with around 10° in-plane rotation. The image in the upper left corner of Figure 6.11 requires rotations around $(10^\circ, 6^\circ)$ in-depth and in-plane respectively. Figure 6.14 shows some of the faces that our system cannot detect.

Figure 6.16 shows the result of detecting faces in different lighting conditions. In

Figure 6.18: Light source comes from right side. The corresponding lighting intensity is showed in the above row. From left to right, the left side of the face becomes darker and darker.

Figure 6.19: Light source comes from left side. The corresponding lighting intensity is showed in the above row. When the light intensity is too low, the detection system will fail.

Figure 6.20: Light source comes from top. The corresponding lighting intensity is showed in the above row. System can detect all faces in the first five images.

Figure 6.21: Light source comes from the bottom. The corresponding lighting intensity is showed in the above row. The detection system doesn't work well in this case.

order to see the detection ability of proposed systems, we constructed additional experiments to test the scope of the system's range of application. See from Figures 6.17 to 6.20, we have simulated different light sources from frontal, sides, top and bottom. Because histogram equalization will equalize imbalance gray value of images, we discard the histogram equalization from the preprocessing step. From the detection results, we can conclude that the proposed wavelet-based SKPCA face detection system works quite well when light sources were coming from frontal, right, left, and top (Figures 6.17-6.21). False detections in these four cases won't appear until the average intensity of face region dropped to 82 (Table 6.4). Note that the intensity range of gray images is from 0 to 255. However, the experimental results of bottom light source is not satisfiable (Figure 6.20). According to the observation of training data, the lighting conditions coursing by bottom light source are not included in the training examples. In this case, example-based classification system cannot learned any information under such condition, therefore, face detection performance in this status would not be good. Deducing from this scope testing experiments, we can conclude that the performance of example-based face detection system depends on training examples to a great extent. Moreover, under the bottom light source condition, the intensity values of eyes region are small. Therefore, this region becomes much more darker compared to the mouth region (Figure 6.20). Consequently, the detail information of eyes is weakened. From here, we can

Table 6.4: The average intensities within face region of the corresponding lighting scope testing. The boldface indicates wrong classification.

Orientation of light sources	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Frontal	211	176	142	105	79	58
Right hand side	141	132	106	95	71	60
Left hand side	243	179	137	106	88	44
Top	184	175	157	125	106	101
Bottom	176	112	97	61	32	12

conclude that eyes are important features for face detection.

6.3 Conclusion

The system performance is evaluated by three different testing sets. Experimental results show that the proposed system have 98.16% accuracy and especially, it decreases the false negative rate. The robust performance of the proposed system in different rotation angles, noise and lighting conditions, etc. is also quite good.

Based on the obtained results, we can conclude that the proposed face detection method works well in images under conditions such as different lighting conditions, different facial expressions, and occluded conditions. However, its probability in detection of rotated faces is limited. The rotation range of faces that can be detected by our method is around $[0^\circ, 18^\circ]$ in-plane, and around $[0^\circ, 35^\circ]$ in-depth. Therefore, partial attention of the future work can be paid on the face detection in large angle problems.

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

Evidently the representative features that are invariant to different variations such as expression, occlusion, noise, and rotation are the most important factors in face detection. However, a lot of work are still needed to be done in this area. In this thesis, a robust face detection method in cluttered images is proposed and it has been proven to give good robust performance to deal with these variation. The approach here is based on representative features extracted by wavelet-based SKPCA method. Our approach does not make the assumption that the detected faces should have front-view or near-neutral facial expressions, and moreover we do not even control the background of faces. On the contrary, the testing data used in this thesis are very diverse. They include faces in different conditions and with different expressions.

Haar wavelet transformation is the first step of our proposed face detection system. Through this transformation, the detail of local facial features is preserved. The second step is the main contribution in the feature extraction aspect. In this step,

we proposed SKPCA, an extension of KPCA, to extract the global distribution structure of faces based on Haar wavelet coefficients. In particular, SKPCA can solve the computational and storage problems of KPCA via a simple and effective subset choosing method. It is also an effective global face representation method that concentrates face images to a certain place in the feature space while disperses the nonface images to different places.

Although the training data of proposed face detection system are homogeneous upright or near upright frontal face data, various kinds of testing data are included in our performance evaluation process. Concretely, they cover faces in different conditions, such as faces in different scales, lighting conditions, and faces with noise, different rotation angles, and occlusions. Obviously, the proposed system which was trained on a simple image set, yet works well on much complex images, displays robustness.

7.2 Recommendations

Although the proposed face detection system has a good performance, there are still some aspects that have not been well solved. One example is the speed problem. From the common characteristics of most vision-based detection tasks, the vast majority of the analyzed patterns in an image belongs to the background class. For example in Figure 7.1, the ratio of nonface to face patterns is 114000:2. Evidently, most of the background patterns in this image can be easily distinguished from the faces. Based on these characteristics, some speed up classification strategy can be applied. For example, simple classification criterion can be applied first to reject most of the easily distinguished nonface patterns in the image, and then more complicated criterion will be applied on the remaining face-like candidates. Via this coarse-to-fine strategy, the execution time of face detection can be reduced. On

Figure 7.1: Most of the analyzed patterns in this image (size 285×450) belongs to the background class. The ratio of nonface to face patterns is 114000:2.

the other hand, observed from experiments, the trained classifier can detect faces rotated up to about 35 degrees in-depth and about 18 degree in-plane. There still have expansion opportunity for this algorithm. Therefore, in order to extend face detection to multipose face detection, different classifiers trained on face database of different poses are needed. The future work can be summarized as below:

1. Speed up face detection system using hierarchical method. Different architecture of hierarchy will result in different speed and classification rate, therefore, the difficulty in building a hierarchy is how to choose corresponding algorithm to construct an optimal “coarse-to-fine” classifier.
2. We would like to extend our algorithm to detecting multipose faces. More classifiers will be used to cover the different poses of human faces. The reason for choosing more classifiers other than one is that the more examples of large angle pose of faces containing in the training database, the more faces distribution and nonfaces distribution will overlap. Therefore, if we separate large angle range face detection problem into several small angle range face detection, the overlap problem of face and nonface data in face detection will not be serious. So, simple classifier is able to deal with the face detection problem in this small pose angle range data.

Author's Publication

1. Xiaoling Zhu, Han Wang, Robust Face Detection in Cluttered Images. Submit to *Pattern Recognition Letter*, Nov 2004.

Bibliography

- [1] Y. Amit, D. Geman, and B. Jedynek. Efficient focusing and face detection. *Face Recognition: From Theory to Applications*, 163:124–156, 1998. H. Wechsler, P.J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, eds.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1998.
- [3] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [5] S. M. Bileschi and B. Heisele. Advances in component based face detection. In *Proc. of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.
- [6] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
- [7] G. Burel and D. Carel. Detection and localization of faces on digital images. *Pattern Recognition Letters*, 15(10):963–967, 1994.
- [8] C. J. C. Burges. Simplified support vector decision rules. In *13th International Conf. on Machine Learning*, pages 71–77, 1996.
- [9] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

- [10] C. J. C. Burges and B. Scholkopf. Improving the accuracy and speed of support vector learning machines. In *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. pp 375-381, MIT Press.
- [11] J. Cai, A. Goshtasby, and C. Yu. Detecting human faces in color iamges. In *Proc. 1998 Int'l Workshop Multi-Media Database Management Systems*, pages 124-131, 1998.
- [12] D. S. Chen and Z. K. Liu. A novel approach to detect and correct highlighted face region in color image. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 7-12, 2003.
- [13] D. Chetverikov and A. Lerch. Multiresolution face detection. *Theoretical Foundations of Computer Vision*, 69:131-140, 1993.
- [14] I. Craw, H. Ellis, and J. Lishman. Automatic extraction of face features. *Pattern Recognition Letters*, 5:183-187, 1987.
- [15] H. Drucker, D. Wu, and V. N. Vapnik. Face-texture model based on sgld and its application in face detection. *IEEE Transactions on Neural Networks*, 10(5):1048-1054, 1999.
- [16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Recognition (2nd Edition)*. John Wiley & Sons, New York, 2001.
- [17] R. Féraud. Pca, neural networks and estimation for face detection. *Face Recognition: From Theory to Applications*, 163:424-432, 1998. H. Wechsler, P.J. Phillips, V. Bruce, F. Fogelman-Soulie, and T.S. Huang, eds.
- [18] R. Féraud and O. Bernier. *Ensemble and Modular Approaches for Face Detection: A Comparison*. MIT Press, 1998. *Advances in Neural Information Processing Systems 10*, M.I. Jordan, M.J. Kearns, and S.A. Solla, eds., pp. 472-478.
- [19] R. Féraud, O. J. Bernier, J.-E. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(1):42-53, 2001.
- [20] Y. Freund. Boosting a weak learning algorithm by majority. *Inform. and Comput.*, 121(2):252-285, 1995. An extended abstract appeared in "Proceedings of the Third Annual Workshop on Computational Learning Theory, 1990.

- [21] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Trans. on Multimedia*, 1(3):264–277, 1999.
- [22] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, Upper Saddle River, New jersey 07458, 2001.
- [23] V. Govindaraju. Locating human faces in photographs. *Int'l J. Computer Vision*, 19(2):129–146, 1996.
- [24] V. Govindaraju, D.B. Sher, R.K. Srihari, and S.N. Srihari. Locating human faces in newspaper photographs. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 549–554, 1989.
- [25] V. Govindaraju, S.N. Srihari, and D.B. Sher. A computational model for face location. In *Proc. Third IEEE Int'l Conf. Computer Vision*, pages 718–721, 1990.
- [26] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, pages 41–46, 1995.
- [27] H. P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multimodal system for locating heads and faces. In *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pages 88–93, 1996.
- [28] C.-C. Han, H. Y. M. Liao, K. C. Yu, and L. H. Chen. Fast face detection via morphology-based pre-processing. In *Proc. Ninth Int'l conf. Image Analysis and Processing*, pages 469–476, 1998.
- [29] K. Hotta. View-invariant face detection method based on local pca cells. In *12th International Conference on Image Analysis and Processing*, pages 57–62, 2003.
- [30] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification.
- [31] R.-L. Hsu, A.-M. Mohamed, and A. K. Jain. Face detection in color images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.
- [32] J. Huang, S. Gutta, and H. Wechsler. Detection of human faces using decision trees. In *Proc. of the Second International Conference on Automatic Face and Gesture Recognition*, pages 248–252, 1996.

- [33] L. L. Huang, A. Shimizu, Y. Hagihara, and H. Kobatake. Face detection from cluttered images using a polynomial neural network. In *2001 International Conference on Image Processing*, pages 669–672, 2001.
- [34] X. Y. Jing, D. Zhang, and Y. Y. Tang. An improved lda approach. *IEEE Trans. Systems, Man and Cybernetics, Part B*, 34(5):1942–1951, 2004.
- [35] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods: Support Vector Learning*, pages 169–184, Cambridge, MA, 1999.
- [36] P. Juell and R. Marsh. A hierarchical neural network for human face detection. *Pattern Recognition*, 29(5):781–787, 1996.
- [37] A. Kadyrov and M. Petrou. The trace transform and its application. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(8):811–828, 2001.
- [38] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. First IEEE IntlInt'l Conf. Computer Vision*, pages 259–269, 1987.
- [39] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Fifth International Conference on Computer Vision*, pages 810–815, 1995.
- [40] S.-H. Kim, N.-K. Kim, S.C. Ahn, and H.-G. Kim. Object oriented face detection using range and color information. In *Proc. Third Int'l Conf. Automatic Face and Gesture Recognition*, pages 76–81, 1998.
- [41] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In *Proc. CVPR*, pages 762–767, Seattle, 1994.
- [42] Y. H. Kwon and N. da Vitoria Lobo. Face detection using templates. In *Proc. Int'l Conf. Pattern Recognition*, pages 764–767, 1994.
- [43] A. Lanitis, C.J. Taylor, and T.F. Cootes. An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.
- [44] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. Fifth IEEE int'l Conf. Computer Vision*, pages 637–644, 1995.

- [45] S. Z. Li. and Z. Q. Zhang. Floatboost learning and statistical face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(90):1112–1123, 2004.
- [46] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. International Conf. on Image Processing*, pages 22–25, 2002.
- [47] S. H. Lin, S. Y. Kung, and L. J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. Neural Networks*, 8(1):114–132, 1997.
- [48] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inform. and Comput.*, 108:212–261, 1994.
- [49] C. Liu. A bayesian discrimination features method for face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(6):725–740, 2003.
- [50] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):572–580, 2004.
- [51] E. Loutas, C. Nikou, and I. Pitas. An information theoretic approach to joint probabilistic face detection and tracking. In *Proc. International Conf. on Image Processing*, pages 22–25, 2002.
- [52] E. Loutas, I. Pitas, and C. Nikou. Probabilistic multiple face detection and tracking using entropy measures. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):128–135, 2004.
- [53] J. W. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using lda-based algorithms. *IEEE Trans. Neural Network*, 14(1):195–200, 2003.
- [54] J. B. Luo, C. W. Chen, and K. J. Parker. Face location in wavelet-based video compression for high perceptual quality videoconferencing. *IEEE Trans. on Circuits and Systems for Video Technology*, 6(4):411–414, 1996.
- [55] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
- [56] A. M. Martinez and A. C. Kak. Pca versus lda. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

- [57] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. on Geoscience and Remote Sensing*, 41(12):2947–2951, 2003.
- [58] B. Menser and F. Muller. Face detection in color images using principal components analysis. In *1999, Seventh International Conference on Image Processing and Its Applications*, pages 620–624, 1999.
- [59] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada. Facial pattern detection and color correction from television picture for newspaper printing. *J. Imaging Technology*, 16(5):165–169, 1990.
- [60] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of 6th International Conference on Computer Vision*, 1995.
- [61] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.
- [62] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.
- [63] A. N. Nefian and M. H. Hayes III. Face detection and recognition using hidden markov models. In *Proc. of International Conference on Image Processing*, pages 141–145, 1998.
- [64] A. N. Nefian and M. H. Hayes III. An embedded hmm-based approach for face detection and recognition. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3553–3556, 1999.
- [65] A. N. Nefian and M. H. Hayes III. Maximum likelihood training of the embedded hmm for face detection and recognition. In *Proc. of International Conference on Image Processing*, pages 33–36, 2000.
- [66] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 193–199, 1997.
- [67] E. Osuna, R. Freund, and F. Griosi. An improved training algorithm for support vector machines. In *Proc. of the 1997 IEEE Workshop on Neural Networks for Signal Processing 7*, pages 276–285, Amelia Island, FL, 1997.

- [68] E. Osuna, R. Freund, and F. Griosi. Support vector machines: Training and application. In *AI Memo 1602, Massachusetts Institute of Technology*, 1997.
- [69] E. Osuna, R. Freund, and F. Griosi. Training support vector machines: An application to face detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 130–136, 1997.
- [70] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proc. Sixth IEEE Int'l Conf. Computer Vision*, pages 555–562, 1998.
- [71] C. Papageorgiou and T. Poggio. A trainable system for object recognition. *Int'l J. Computer Vision*, 38(1):15–33, 2000.
- [72] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 84–91, 1994.
- [73] M. Petrou and A. Kadyrov. Affine invariant features from the trace transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):30–44, 2004.
- [74] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, pages 185–208, 1999. <http://www.research.microsoft.com/~jplatt/smo-book.pdf>.
- [75] M. Propp and A. Samal. Artificial neural network architectures for human face detection. *Intelligent Eng. Systems through Artificial Neural Networks*, 2, 1992.
- [76] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [77] J. R. Quinlan. Rule induction with statistical data - a comparison with multiple regression. *J. of the Operational Research Society*, 38:347–352, 1987.
- [78] J. R. Quinlan. Decision trees and decision-making. *IEEE Trans. Systems, Man and Cybernetics*, 20(2):339–346, 1990.
- [79] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan kaufmann, San mateo, 1993.
- [80] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286, 1989.

- [81] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In *Proc. Int. Conf. on Computer Vision*, pages 695–700, 2001.
- [82] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
- [83] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proc. IEEE Conference Computer Vision and Pattern Recognition*, pages 38–44, 1998.
- [84] T. Sakai, M. Nagao, and S. Fujibayashi. Line extraction and pattern detection in a photograph. *Pattern Recognition*, 1:233–248, 1969.
- [85] A. Samal and P.A. Iyengar. Human face detection using silhouettes. *Int'l J. Pattern Recognition and Artificial Intelligence*, 9(6):845–867, 1995.
- [86] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in news videos. *IEEE Multimedia*, 6(1):22–35, 1999.
- [87] Y. Freund R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [88] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.
- [89] B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods Support Vector Learning*. MIT Press, Cambridge, MA, 1999.
- [90] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Trans. Neural Networks*, 10:1000–1017, 1999.
- [91] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [92] B. Schölkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10:1299–1319, 1998.
- [93] H. Schneiderman. Learning a restricted bayesian network for object detection. In *Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–639–II–646, 2004.

- [94] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proc. of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 45–51, 1998.
- [95] H. Schneiderman and T. Kanade. A histogram-based method for detection of faces and cars. In *Proc. 2000 International Conference on Image Processing*, pages 504–507, 2000.
- [96] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1:746–751, 2000.
- [97] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Int'l J. of Computer Vision*, 56(3):151–177, 2004.
- [98] K. H. Seo, W. Kim, and C. Oh; J. J. Lee. Face detection and facial feature extraction using color snake. In *Proc. of the 2002 IEEE International Symposium on Industrial Electronics*, pages 457–462, 2002.
- [99] M. J. Seow, R. Gottumukkal, D. Valaparla, and K. V. Asari. A robust face recognition system for real time surveillance. In *Proc. of Coding and Computing Information Technology*, pages 631–635, 2004.
- [100] J. Shlens. A tutorial on principal component analysis-derivation, discussion and singular value decomposition.
- [101] P. Sinha. Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science*, 35(4):1735–1740, 1994.
- [102] S. A. Sirohey. Human face segmentation and identification. In *Technical Report CS-TR-3176, Univ. of Maryland*, 1993.
- [103] K. Sobottka and I. Pitas. Face localization and feature extraction based on shape and color information. In *Proc. IEEE Int'l Conf. Image Processing*, pages 483–486, 1996.
- [104] S. Srisuk, M. Petrou, W. Kurutach, and A. Kadyrov. Face authentication using the trace transform. In *Proc. of the 2003 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

- [105] E.J. Stollnitz, A.D. DeRose, and D.H. Salesin. Wavelets for computer graphics: A primer part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84, 1995.
- [106] E.J. Stollnitz, A.D. DeRose, and D.H. Salesin. Wavelets for computer graphics: A primer part 2. *IEEE Computer Graphics and Applications*, 15(4):75–84, 1995.
- [107] Q.B. Sun, W.M. Huang, and J.K. Wu. Face detection based on color and local symmetry information. In *Proc. Third Int’l Conf. Automatic Face and Gesture Recognition*, pages 130–135, 1998.
- [108] Z. H. Sun, G. Bebis, and R. Miller. Boosting object detection using feature selection. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 290–296, 2003.
- [109] K.-K. Sung and T. Poggio. Learning human face detection in cluttered scenes. *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*, Springer Verlag, 19:432–439, 1995.
- [110] 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.
- [111] X. G. Wang; X. O. Tang. A unified framework for subspace face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, 1999.
- [112] A. Tsukamoto, C.-W. Lee, and S. Tsuji. Detection and pose estimation of human face with synthesized image models. In *Proc. Int’l Conf. Pattern Recognition*, pages 754–757, 1994.
- [113] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.
- [114] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.
- [115] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

- [116] R. C. Verma, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1215–1228, 2003.
- [117] P. Viola. Robust real-time face detection. *Int'l J. of Computer Vision*, 57(2):137–154, 2004.
- [118] P. Viola and Michael Jones. Rapid object detection using a boost cascade of simple features. In *Proc. of the 2001 IEEE Computer Society Conference. 2001 CVPR*, pages 511–518, Computer Vision and Pattern Recognition, 2001.
- [119] 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.
- [120] C.-Y. Xu and J. L. Prince. Gradient vector flow: A new external force for snakes. In *Proc. of IEEE Computer Society Conference, Computer Vision and Pattern Recognition*, pages 66–71, 1997.
- [121] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang. Two-dimensional pca a new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.
- [122] M. H. Yang and N. Ahuja. Detecting human faces in color images. In *Proc. of 1998 International Conference Image Processing. 1998 ICIP*, pages 4–7, 1998.
- [123] K. C. Yow and R. Cipolla. A probabilistic framework for perceptual grouping of features for human face detection.
- [124] K. C. Yow and R. Cipolla. Detection of human faces under scale, orientation and viewpoint variations. In *Proc. 2nd Int. Conf. on Auto. Face and Gesture Recog*, Vermont, USA, 1996.
- [125] K. C. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15(9):515–522, 1998.
- [126] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *Int'l J. Computer Vision*, 8(2):99–111, 1992.