Sample selection and variance discriminant analysis for sample-based face detection

Yu, Wei

2009

Yu, W. (2009). Sample selection and variance discriminant analysis for sample-based face detection. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/14803>

<https://doi.org/10.32657/10356/14803>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**SAMPLE SELECTION AND VARIANCE DISCRIMINANT
ANALYSIS FOR SAMPLE-BASED FACE DETECTION**

YU WEI

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2009

SAMPLE SELECTION AND VARIANCE DISCRIMINANT ANALYSIS FOR SAMPLE-BASED FACE DETECTION

YU WEI

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

.....

Date

.....

Yu Wei

Acknowledgements

My research work received both financial and technical support from the School of Electrical & Electronic Engineering, Nanyang Technological University.

First and foremost, I would like to thank Assistant Prof. Jiang Xudong, my current supervisor, from the bottom of my heart, for his professional guidance, patience and encouragement. Without him, I cannot last to the end.

I would like to acknowledge my formal supervisor, Dr. Charayaphan, for introducing and suggesting face detection as the research topic for my thesis, and for sharing with me many hardware design experience for the research work. I also benefited from his insights into all the research aspects of face detection, machine learning and pattern recognition in this thesis through our discussions.

I would like to thank Prof Chan Kap Luk, Prof Gao Yongshen and Prof Alex Kot, who were the panel members for confirmation of Ph.D candidature, for their valuable advice and patient instruction throughout my thesis writing.

I would also like to thank Prof Chang Chip Hong, Dr Wu Jigang, and Dr. Tian Hui, who gave me very good advice and guidance on my research work. Thanks to all the teachers who taught me in NTU. During my tenure at NTU, I knew a lot of friends and classmates. We encouraged each other and discussed together. They are one of my best gains in NTU. Thanks to Gu Jiangmin, Liu Yongzhi etc. Special thanks to my junior, Xiong Bing. We had valuable and inspiring discussions on our projects and related issues.

I would also like to thank my family in China and my lovely wife, Zhang

Cangwen. They provided me good education and family care. They are always my power source. This thesis is dedicated to them. Finally, I thank all my friends, the students and staff in the CHIPES laboratory, and the SEL laboratory, for all their help, support, friendship during my study and research. Thanks to School of EEE in NTU to give me such a good chance to study here and thanks to NTU for providing me three-year scholarship.

Contents

Acknowledgements	i
List of Figures	xi
List of Tables	xiii
Summary	xiv
1 Introduction	1
1.1 Motivation	1
1.1.1 Practical Applications of Face Detection	2
1.1.2 Challenges in Face Detection	3
1.1.3 Motivation for Sample-Based Face Detection	6
1.1.4 Motivation for Sample Selection and Generation	8
1.1.5 Motivation for the Proposed Statistics-Based Method	9
1.2 Objectives	10
1.3 Major Contributions	10
1.4 Outline	11

2	Literature Review	12
2.1	Definitions	12
2.2	History of Face Detection	16
2.2.1	Face Detection Research	16
2.2.2	Techniques for Face Detection	18
2.2.3	Databases for Face Detection	24
2.3	Issues Related to Face Detection	28
2.3.1	Face Tracking	28
2.3.2	Multi-view Face Detection	28
2.4	General Procedures of Sample-Based Face Detection	31
2.4.1	General Pyramid Framework for Sample-Based Methods	31
2.4.2	A general procedure of training and constructing a sample-based classifier	37
2.5	Conclusions	37
3	Passive Sample Selection	39
3.1	Overview of Sample Selection	40
3.1.1	Motivation for Sample Selection in Face Detection	40
3.1.2	Bootstrap Review	42
3.1.3	Artificial Sample Generation	44
3.1.4	ICA	50
3.2	The Proposed Algorithm for Selection of Samples	53
3.2.1	The Proposed Method	54

3.2.2	Performance	56
3.2.3	Analysis of Passive Sample Selection	69
3.3	Conclusions	72
4	Active Sample Generation	75
4.1	Overview of Active Sample Generation and SVM	76
4.1.1	Active Sample Generation	76
4.1.2	Theories on SVM (Support Vector Machine)	78
4.1.3	S^3VM (Semi-Supervised SVM)	80
4.1.4	Active Support Vector Machine Learning	81
4.1.5	Support Vector Machine (SVM) for Face Detection . . .	82
4.2	Proposed Algorithm for Active Generation of Samples	85
4.3	Analysis of the Proposed Algorithm	90
4.4	Experimental Results	91
4.4.1	Artificial Example	91
4.4.2	Application to Face Detection	95
4.5	Conclusions	104
5	PLCB (PCA + LDA + CMFE + Bayes) for Face Detection	105
5.1	Statistics-Based Face Detection	105
5.1.1	The Bayesian Discriminating Features (BDF) Method . .	106
5.1.2	Face Detection Using Multimodal Density Models	111
5.2	Common Mean Feature Extraction (CMFE)	116
5.3	The Proposed PLCB Method for Face Detection	118

5.3.1	PLCB	118
5.3.2	Analysis	123
5.4	Performance	123
5.5	Conclusions	132
6	Conclusions and Recommendations for Future Work	133
6.1	Conclusions and Major Contributions	133
6.2	Suggestions for Future Work	134
	Appendix	136
	Author's Publications	138
	Bibliography	147

List of Figures

1.1	Faces in real world (from CMU Frontal Face Images database). .	4
1.2	The issue of precision in face detection.	7
2.1	Research in face image processing.	15
2.2	Multi-view faces [Yon01].	30
2.3	In-plane rotation invariant neural network [RBK97].	31
2.4	Face detection system based on the pyramid framework.	32
2.5	The number of sub-images in the pyramid framework.	34
2.6	The number of sub-images T generated in pyramid framework w.r.t step size S ($S_x = S_y$).	34
2.7	The framework for merging multiple detections from a single net- work from Figure 6 in [RBK98].	36
2.8	ANDing together the outputs from two networks over different positions and scales can improve detection performance from Fig- ure 7 in [RBK98].	36
2.9	General block diagram of training and constructing a face detec- tion system based on learning methods.	38
3.1	Original 16 face images from the ORL face database.	41

3.2	TICA processing of images in Figure 3.1. Grouped by top left 4, top right 4, bottom left 4 and bottom right 4 images.	42
3.3	Left: images of scenery containing no faces. Right: regions in the left image falsely detected as faces will be added into the set of non-face training samples.	44
3.4	Typical sample images: (a) Original image (b) Mirror image (c) Rotate +5 degree (d) Mirror image rotate +5 degree (e) Rotate -5 degree (f) Mirror image rotate -5 degree (g) Move up one pixel (h) Move left one pixel (i) Move down one pixel (j) Move right one pixel.	45
3.5	Clusters of the artificial expansion from 2 original face samples.	46
3.6	20 face samples represented by 2 main PCA features.	47
3.7	Clusters of the artificial expansion from samples in Figure 3.6. .	49
3.8	Crossover and mutation from [CCG04] (a) Each parent is converted into a sequence of observation vectors for crossover, (b) Crossover, (c) Mutation.	50
3.9	Eigen-face using PCA.	52
3.10	Comparisons between PCA and ICA.	52
3.11	Cascaded structure of the proposed sample selection technique. .	55
3.12	Equal Error Rate.	57
3.13	Random selection distribution using ECU and CMU data sets. .	59
3.14	Face detection EER against the number of training samples of random selection and PCA sample selection.	60
3.15	The 60 selected face patterns from ECU database.	61

3.16	Face detection EER against the number of training samples under various methods of sample selection.	64
3.17	The number of support vectors against the number of training samples under various methods of sample selection for face detection.	65
3.18	ROC curves of face detection by various methods using the ECU database.	66
3.19	ROC curves of face detection under various methods using the CMU database.	67
3.20	Face detection accuracies using original data and after redundancy database selection (data from Table 3.1.)	68
3.21	Curves of high density and low density.	70
4.1	Margin maximization explanation of SVM using a two-class classification.	79
4.2	(a) Solution found by robust linear program (b) Solution found by S^3VM [BD98].	82
4.3	(Taken from Figure 6 in [OFG ⁺ 97]) A circle represents a face and a square represents a non-face. On the decision hyper-plane between the two classes some of the support vectors are represented using sub-images.	83
4.4	(Taken from [OFG ⁺ 97]) Number of support vectors vs. number of samples.	84
4.5	An example of a non-linear mapping of a straight line between two SVs. (a) Samples in the feature space, (b) mapping samples in the version space.	87

4.6	An artificial example: two classes, class A and B.	91
4.7	The average performance of an SVM-based classifier recursively using active sample generation.	92
4.8	The true positive rate versus the false positive rate for each active sample generation.	94
4.9	The Euclidean distance, first generation.	95
4.10	The Euclidean distance, 10th generation.	96
4.11	The Kernel distance, first generation.	97
4.12	The Kernel distance, 10th generation.	98
4.13	(a) non-face sample; (l) face sample; (b)-(k) newly generated middle samples.	99
4.14	The performance of an SVM-based classifier recursively using ac- tive sample generation (trained by the CMU train data set). . .	101
4.15	The performance of an SVM-based classifier recursively using ac- tive sample generation (trained by the ECU data set).	102
4.16	Some generated face samples.	103
5.1	Common mean different standard deviation.	118
5.2	ROC of the performance trained by CMU train data set and tested by CMU test data set.	125
5.3	ROC of the performance trained by ECU data set and tested by CMU train data set.	126
5.4	ROC 1 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.	127

5.5	ROC 2 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.	128
5.6	ROC 3 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.	129
5.7	ROC 4 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.	130
5.8	ROC 5 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.	131

List of Tables

2.1	Face detection terminology.	13
2.2	Terminology for algorithmic testing.	14
2.3	Terminology for detection methods.	14
2.4	Terminology for face detection and tracking.	14
2.5	Common assumptions for face detection in different applications.	16
2.6	Papers from 1995 to the present on face detection and recognition.	17
2.7	Face detection methods.	24
2.8	Analysis of face detection methods.	25
2.9	Widely-used face databases.	26
2.10	Face tracking techniques.	29
3.1	Accuracies and number of Support Vectors (nSV) from different methods.	62
3.2	Redundancy database selection.	64
4.1	Simulation results of the proposed active generation method when using 2,000 ECU face images and CMU training database. . . .	100

4.2	Simulation results of the proposed active generation method when using ECU 4,000 face and 8,000 non-face images and CMU training database.	100
-----	--	-----

Summary

In recent years, face detection has been a very active area of research. These technologies can be applied to the domains of computer vision, pattern recognition, and machine learning. Among the many existing categories of face detection algorithms, the sample-based method is one of the most widely-used approaches. The essence of the sample-based method is to solve a two-class classification problem of face versus non-face. Many classification algorithms such as the Naïve Bayesian, Neural Network and Support Vector Machines (SVM) have been used for this purpose. This thesis showcases a research study into face detection technologies. This document is in two main parts. Firstly, in the sample preparation section, new passive sample selection and active sample generation algorithms are proposed to assist existing sample-based algorithms in solving the problem of face detection. Secondly, in the classification section, a new Bayesian-based classification method is proposed for face detection.

Sample-based algorithms have generally resulted in the best reported face detection performance. Sample-based methods in the thesis mean the methods that extract features or select features based on the machine learning from samples. A face detection algorithm that depends on sample-based approaches must consider various issues. The primary issues include how to determine a

suitable algorithm to construct the classifier, selecting representative samples and balancing training samples. One relevant approach to optimize face detection performance involves improving efficiency by selecting and adding useful samples into the training set without collecting new samples.

In this thesis, a new passive sample selection method and an active sample generation algorithm are proposed. These techniques are applied using sample-based algorithms. Empirical results show that the accuracy of face detection can be improved by 10 – 20% based on the new passive sample selection method with an artificially expanded database. Furthermore, the new active sample generation algorithm can further enhance the performance of a SVM classifier with the help of human experts. Naturally, the proposed algorithms can be also applied to solve other two-class classification problems.

In dealing with the problem of face detection by the classification, a new classification method, PLCB (PCA + LDA + CMFE + Bayes, where CMFE refers to Common Mean Feature Extraction), is proposed for the purpose of face detection and its performance is compared with a few well-known sample-based algorithms, i.e., SVM, LDA, Naïve Bayesian and MLP (Multi Layer Perception). Empirical results show that its accuracy is higher than classification algorithms currently used.

Chapter 1

Introduction

This chapter begins by outlining the motivation and objectives of the author’s research. Major research contributions are subsequently listed. An overview of the entire thesis is presented at the end of this chapter. The focus of this thesis is on passive and active sample selection methods for face detection. The term “passive” refers to those selection methods where only the available samples are used. In contrast, the term “active” refers to those where certain new samples are requested by the algorithm. New samples that match the required specifications would subsequently be labeled by an oracle (usually a human) who would add them to the training data set.

1.1 Motivation

Face detection is one of the most active research topics in the field of pattern recognition. This is due in large part to its relevance to real-world scenarios

(see section 1.1.1). Furthermore, as an example of a typical two-class classification problem, the methods developed for face detection can be applied to other two-class classification applications. Accordingly, it is of great interest to improve the performance of the current methods, which are mainly sample-based. In section 1.1.3, the motivation for sample-based face detection methods is given. Efficient selection of appropriate training samples is one of the most effective ways to improve performance. In section 1.1.4, the motivation for sample selection and generation is summarized. Details are given in Chapter 3 and Chapter 4. A new statistics-based method capable of achieving higher accuracy than other sample-based methods is proposed in Chapter 5. Its motivation is described in section 1.1.5.

1.1.1 Practical Applications of Face Detection

Face detection is the first key step in automated human face recognition in machine vision systems. It is a crucial step towards Automatic Target Recognition (ATR) or generic object detection/recognition. A large number of tasks that are today performed by humans could in the future be completed by these types of intelligent machine vision systems. Potential applications are numerous. These include the ability to encode face objects in MPEG-4 video, smart demonstration systems, improved auto-focus features in digital cameras, intelligent identification systems, and intelligent human search engines. Face detection has been marketed as a key selling point for digital cameras such as the Fujifilm FinePix Fxxx series, S6000fd, Sony W200 etc.

Face detection and tracking allow machines to interact with humans in a more fluid and responsive manner. These technologies thereby improve human-

machine interaction. For example, in an airport, passenger facial recognition could be used as a security measure. Such an approach would be significantly faster and less obtrusive than current document checks and fingerprinting procedures.

The human face is semi-rigid and face detection is generally considered a two-class classification problem. Therefore, methods developed for this application can be readily extended to other two-class classification scenarios. For example, based on the work in [VJ01], this Haar feature based algorithm was extended to a general object detection problem in OpenCV[Int].

1.1.2 Challenges in Face Detection

Face detection is a difficult challenge because of natural variability of face patterns:

- (a) Different poses (out-of-plane rotations) such as frontal views, side views, and faces that look up or down.
- (b) Orientation (in-plane rotation).
- (c) Surroundings such as beards, mustaches, earphones, microphones.
- (d) Facial expressions such as smiling, frowning, etc.
- (e) Facial size within an image.
- (f) Similar objects such as a real face and line drawing of a face.
- (g) Occlusions due to moving objects such as hands, other faces, and body parts in the foreground and background of the image.

Figure 1.1: Faces in real world (from CMU Frontal Face Images database).

(h) Variations in image quality due to issues such as sub-optimal lighting and inadequate focus, picture blurring, or unsuitable image resolution.

Figure 1.1 (from CMU Frontal Face Images database, vasc.ri.cmu.edu/idb/html/face/profile_images) illustrates the above points. The cases described are not uncommon. Even a human will occasionally find it difficult to identify faces without being given any clues. Therefore, many problems in face detection still remain to be resolved, some of which are discussed below:

(a) Robustness of face detection. Increasing numbers of methods are being proposed to solve the face detection problem. As shown in Table 2.5, most require certain assumptions. To apply the methods in the real world, more robust face detectors are needed.

- (b) Face representation. How to describe a typical face? Mathematical equations cannot readily describe such semi-rigid objects.
- (c) Scaling. How to deal with faces of different sizes? Nowadays, the general framework for face detection, which will be introduced in section 2.4.1, uses the brute force search to scale to different face sizes. Is there any scale-invariant feature that avoids the need for exhaustive search and thereby offers faster detection processing and improved accuracy?
- (d) Search strategy. How to spot faces? When a brute-force search strategy is applied, face detection is a rare event detection problem in the sense that only very few image regions among the millions contain faces [WRM03]. The classifier design problem is very challenging in that the detection rate must be very high in order to avoid missing any rare events. At the same time, the false positive rate must be very low (e.g. 10^{-6}) in order to avoid a proliferation of non-events. From a computational perspective, huge speed increases are possible if the sparsity of faces in the input set can be exploited.
- (e) Speed of detection. How to accelerate the process? This issue is closely related to choice of search strategy. A greedy search strategy or cascade structure can save time in solving a rare event detection problem. The latest research is mainly based on cascade structures. Another two important factors are the operating speed of the classifier itself and the time taken to extract facial features. The Haar-like feature is widely used because of its robustness and speed. The classifier trained by the Adaboost algorithm [VJ02] using Haar-like features deserves special attention because of its fast computation and high precision.

- (f) Precision. How to locate the faces precisely? It is difficult to locate the exact face position. In typical images, a shift of several pixels in various directions is very common as shown in Figure 1.2 [YKA02]. Which method can be used to solve this problem? If a soft template is used, more calculation is needed. Is such an approach worthwhile?
- (g) Post processing. How to combine detection results? In Rowley et al. [RBK98], a general post-processing method was proposed that is discussed in section 2.4.1. Is it the best post-processing technique available? Is there any other method that can be used to improve the performance of the whole system? After studying many methods, which of them can be adopted and can an efficient algorithm be found to fuse their results?

Undoubtedly, the focus on face detection is the overall classification performance. Even if one were to collect millions of training samples, the core research questions would still remain. First, how to improve performance after collecting a given number of training samples? Second, how to eliminate redundancy in the training data set to alleviate the imbalance problem of the training set and improve the classifier's performance? This thesis aims to address these issues.

1.1.3 Motivation for Sample-Based Face Detection

Face detection is currently a very active research field. Generally, face detection techniques can be categorized into two kinds, feature-based and sample-based.

Features-based methods in the thesis mean the methods that extract features from the image based on the human knowledge about the face and non-face patterns. Sample-based methods in the thesis mean the methods that extract

Figure 1.2: The issue of precision in face detection.

features or select features based on the machine learning from samples. Sample-based methods usually but not always use the information of the whole face image region to detect faces. The key parts of sample-based methods emphasize the learning algorithm and the importance of collecting of training samples. Currently, the best reported face detection performance is essentially achieved by sample-based methods, such as neural networks [RBK98] [RBK97] [PS92] [JM96] [LKL97], SVM [OFG⁺97] [LGL00] and the Adaboost algorithm [Hyv99] [WAHL04] [VJ02] [LKP03] [LZSZ02] [YH93]. Section 2.2.2 reviews all of these methods. Section 2.4.1 describes a general framework of sample-based methods in detail.

The sample-based methods can be widely applied in many areas as sample-based methods do not have to search for a proper choice of an image representation. Samples from parts of the whole data set are often easy to collect. Combined with some prior knowledge such as the samples distribution, the bet-

ter performance can be achieved by applying some techniques such as Bayesian rules [SE92]. The performance achieved by sample-based methods can be improved through efficient sample collection and sample utilization.

Sample-based methods are used in face detection scenarios for a number of reasons. Firstly, faces are semi-rigid and it is therefore impractical to apply analytical geometric or algebraic methods. Secondly, pure feature-based methods may not be able to extract reliable and discriminative features only based on human knowledge [MBN02]. Thirdly, Sample-based methods are especially appropriate when there is no prior knowledge about what are the reliable and discriminative features for the classification tasks. In such cases it is generally fairly easy to collect appropriate sample data.

1.1.4 Motivation for Sample Selection and Generation

Apart from the choice of the classifier and its training algorithm, sample-based face detection has several disadvantages:

- (I) The size of training data sets often needs to be large.
- (II) It may be difficult to obtain a suitable range of balanced training data samples.

(I) and (II) together provide the core motivation for this research work. Strong performance is tightly correlated with effective training. Accordingly, the key issues to be addressed are as follows:

1. *How do the number and type of training samples impact final performance?*
2. *How should one select an appropriate set of training samples from a large*

pool of data?

3. *If a training data set is insufficient, what strategies are appropriate for obtaining or generating new samples?*

The first question is very difficult to answer. In this thesis, the author attempts to answer this question by addressing questions 2 and 3. Chapter 3 describes a passive sample selection method that can be used to answer the second question. Chapter 4 proposes the use of an active sample generation method to address the third question.

1.1.5 Motivation for the Proposed Statistics-Based Method

PCA is used to reduce the number of dimensions to remove the unreliable dimensions and hence to alleviate the over-fitting problem. From the reliable PCA subspace, a smaller subspace in which the face and non-face classes are the most separable is needed to speed up the detection. LDA and its variants are widely applied in face recognition. For face detection, however, only a single feature can be extracted by LDA, which is far from sufficient for a reasonable classification. For two-class problem, one may think of the common-mean feature extraction (CMFE) method. It is a surprise that CMFE is not seen in face detection literature. Chapter 5 proposes a new discriminant analysis method that integrates PCA, LDA, CMFE and Bayes, called PLCB method.

1.2 Objectives

The overall objective of this thesis is to improve on the current state of the art in sample-based face detection, as measured primarily in terms of algorithmic performance.

In this context, the first goal is to identify new methods that can efficiently select appropriate training samples from available data sets. The second goal is to devise methods to generate new samples that can compensate for an inadequate set of training data. The third goal involves deriving a new classification method that can improve algorithmic accuracy independent of the first two goals.

1.3 Major Contributions

Two new selection methods, passive and active, are proposed for the purpose of pre-processing data. The passive method performs clustering on existing samples and selects only certain samples for further processing. The approach is particularly useful for artificially expanded data sets. This data reduction method helps to improve both accuracy and processing speed. In contrast, the active method makes use of original data to generate new samples by means of numerical computation in the margin space. New samples are then manually labeled before being inserted into the training data set.

A new statistics-based method named as PLCB (PCA+LDA+CMFE+Bayes, CMFE refers to Common Mean Feature Extraction) is proposed for face detection. It offers improved accuracy when compared with other sample-based

algorithms such as SVM (Support Vector Machine) and the Naïve Bayesian.

1.4 Outline

The thesis is organized as follows:

- Chapter 2 reviews existing work in the area of algorithms for face detection.
- Chapter 3 introduces the concept of passive sample selection.
- Chapter 4 describes active sample generation.
- Chapter 5 describes the new statistics-based method for face detection that has been developed as part of this thesis research.
- Chapter 6 offers conclusions and recommendations for future work.

Chapter 2

Literature Review

This chapter reviews the history and background of research in face detection. After that, general procedures of sample-based methods are introduced.

2.1 Definitions

Descriptions of a few formal terms used in the area of face detection are given in Table 2.1 below [HPV99]. Terms specific to performance evaluation appear in Table 2.2. Table 2.3 and Table 2.4 summarize terminology related to face detection and tracking methods.

Face detection: The goal of face detection is to determine whether or not there is a face in a given image and, if so, to locate the area and extent of the face(s).

Figure 2.1 shows research in the domain of face image processing. The lowest row is the “image and video level”. Face localization and verification are performed here on the basis of raw input image, while motion prediction and

Table 2.1: Face detection terminology.

Term	Description	Assumptions and conditions
Face Localization	Aims to determine the position of a single face in an image, a simplified face detection problem.	Input image contains faces.
Face Verification	An input image is verified to be a face or non-face image.	One or no face will be in the input image. If there is a face, it will be tightly cropped
Face Segmentation	Identify the location and actual shape of the complete face(s) appearing in a given image [MV02].	Identical to face detection.
Facial Feature Detection/Extraction	Detect the presence and location of features of the face such as eyes, nose, nostrils, eyebrows, mouth, lips, ears, etc.	Only one face is in the image.
Face Tracking	Continuously estimate the location and possibly the orientation of a face in an image sequence in real time.	Input image sequence contains a face.
Face Authentication	Verify the identity of an individual in an input image.	Inputs must include both face image and a predefined face template for comparison.
Face Recognition or Identification	Compare an input image against a database and report a match.	Inputs include a face image and a database of many faces
Facial Expression Analysis	Identifying human emotional states (happy, sad, disgusted, etc).	Input face image.
Face Synthesis (Animation)	Extract facial anatomy and track face motion. Generate data that can be used to [EPT ⁺ 98] construct a synthetic head.	extracted and then face motion is tracked.
Face Alignment and Pose Estimation	Fix all the facial landmarks in an image. Face alignment can make the face distribution statistically more compact and improve face recognition, modeling and synthesis.	Only one face is in the image.
Face Analysis	Analyze facial features extract information.	Input image features one face at known coordinates.
3D Face Reconstruction	Build a 3D model of a face from an image sequence.	An image sequence with one required face.

Table 2.2: Terminology for algorithmic testing.

Terms	Descriptions
Detection Rate	The ratio between the number of faces algorithmically detected and the number faces identified by a human.
False Negatives	Faces are missed, resulting in low detection rates.
False Positives	An image region is incorrectly identified as a face.
ROC	Receiver Operating Characteristic curves.

Table 2.3: Terminology for detection methods.

Terms	Descriptions
Knowledge-based	Encode human knowledge of what constitutes a typical face, usually focusing on relationships between different facial features.
Feature-based	Find structural features that are invariant to pose, viewpoint, or lighting conditions.
Template matching	Compute the correlation between an input image and several standard stored facial patterns to describe the face either as a whole or as a collection of features.
Appearance-based	Models or templates are learned from a set of training images, which should capture the representative variability of facial appearance.
Sample-based /Image-based	Models are learned from a set of training images or samples, which should capture the representative variability of face.

Table 2.4: Terminology for face detection and tracking.

Terms	Descriptions
Multi-view face detection	Face can be detected when it looks up or down and turns right or left, by means of comparison with a frontal view.
Scale invariant	Face size does not affect the results.
Orientation invariant	Face rotation does not affect the results.

Figure 2.1: Research in face image processing.

verification are based on the input raw video. Immediately above, at the “Face level”, are face detection and motion estimation, the results of which allow face tracking to take place. At the “Model level”, face information is abstracted into a facial map that can be used for feature extraction and 3D reconstruction. The uppermost row, known as the “Application level”, includes elements such as face synthesis (animation), facial feature extraction and face recognition or identification.

Different applications may require different algorithmic techniques, each of which may require a unique set of assumptions. Table 2.5 shows common assumptions for video-based and still image face detection.

Table 2.5: Common assumptions for face detection in different applications.

Face Detection	Video	Color	Viewpoint, perspective, occlusion, rotation, resolution, lighting conditions.
		Gray	
	Still Image	Color	
		Gray	

2.2 History of Face Detection

Face detection is an essential precursor to face recognition. The aim is to locate and extract the facial region(s) from the scene. The human face is a dynamic object with a high degree of variability in appearance. Over the past twenty years, a variety of techniques have been proposed. A survey of the algorithms for face detection partially taken from [MV02] and [HPV99] is provided in section 2.2.2. Certain algorithms could potentially be placed in multiple categories.

2.2.1 Face Detection Research

Early efforts in face detection date back to the early 1970s. Techniques in those days required rigid assumptions, including simple backgrounds and faces that were viewed from the front. When practical face recognition and video coding systems became a reality in the 1990s, researchers showed increased interest in face detection for two main reasons. Firstly, the facial signal process system can have a significant impact on human daily life. Secondly, researchers had new tools in the form of more mature face recognition technologies. Table 2.6 lists the numbers of published papers on face detection and recognition in the SCI index, IEEE journals and conferences from 1989 to the present (Searched from IEEE and SCI websites).

Many signal processing methods can be applied to face detection. The prob-

Table 2.6: Papers from 1995 to the present on face detection and recognition.

	face detection			face recognition		
	SCI Index	IEEE Journal	IEEE Conf	SCI Index	IEEE Journal	IEEE Conf
1989	-	0	0	-	0	6
1990	-	0	2	-	0	2
1991	-	0	0	-	0	4
1992	-	0	0	-	2	28
1993	-	0	2	-	5	26
1994	-	1	4	-	5	54
1995	2	0	4	75	6	55
1996	4	0	22	88	6	121
1997	8	1	9	100	17	100
1998	17	4	33	121	14	176
1999	15	2	39	122	17	159
2000	23	1	66	139	19	262
2001	34	2	58	185	16	198
2002	40	2	80	187	30	328
2003	86	4	94	307	31	315
2004	83	9	59	297	41	236
2005	230	5	104	879	54	499
2006	306	5	153	1068	57	661
2007	151	15	94	721	97	388
Total	960	48	777	4154	401	3428

lem of automated face detection is related to topics in machine intelligence. Important research directions in the realm of face detection include high-accuracy face detection, multi-view face detection, real-time and fast adaptive machine learning.

Factors to consider in face detection include:

- Target application domain: still images vs. video.
- Representation: holistic features, etc.
- Pre-processing: histogram equalization, etc.
- Cues: color, motion, depth, voice, etc.
- Search strategy: exhaustive, greedy, focus of attention, etc.
- Classifier design: ensemble, cascade.
- Post-processing: combining detection results.

2.2.2 Techniques for Face Detection

Face detection techniques can be categorized as either feature-based or sample-based. The former approach extracts features from the image based on the human knowledge about the face and non-face patterns and uses these to detect faces. If feature extraction is reliable and discriminative, it is generally relatively easy to subsequently separate face and non-face samples. Sample-based methods usually but not always use information from the entire face image region to detect faces. Sample-based methods may also extract features or select features based on the machine learning from samples.

The key parts of sample-based methods emphasize the learning algorithm and the importance of collecting of training samples.

All sample-based methods emphasize the learning algorithm and the importance of providing appropriate training samples. The aforementioned methods are summarized in Figure 2.7. Their advantages and disadvantages are summarized in Table 2.8.

Extracting edge information [OFG⁺97] requires an analysis of line drawings of faces. This is justified because the scale, view and orientation of edge features are invariant. Algorithms will often also include certain shape rules such as the ratio between facial height and width. A cost function is needed to perform such shape analysis. Edge feature detection is a low-complexity technique that should be used together with other methods.

The gray-level [GT94][GT93] method uses grayscale information alongside edge feature detection. It assumes that face regions become uniform at low resolutions. This is a disadvantage because the performance of this algorithm is consequently severely impacted by viewpoint and facial ornaments. The complexity depends on the required volume of computation.

Using the human skin color model to detect faces in an image is fast and practical. Under normal illumination, this technique can detect most faces with minimal computation. However, this method can only be used alongside other techniques.

Motion detection [LKP96][YB99] is another computationally efficient method for detecting objects in image sequences or videos. For moving faces, it is very accurate. However, the extracted regions often do not exactly define the complete face. Other algorithms, such as ellipse fitting, have to be employed after

the moving objects have been extracted. The degree of complexity depends on the exact algorithms used. Human skin, hair or other texture [AS93] information can also be used for face detection. Certain parameters are scale, view, and orientation invariant and therefore require few assumptions. The key elements associated with this approach are a texture model and a classifier to differentiate the texture features. The algorithm is not very complex.

The generalized measure [RWY95][RY98] uses the symmetric nature of facial features. It is orientation invariant but is best used for frontal view detection. The algorithm is not very complex.

Feature searching [GT94] involves detecting the most prominent facial features and estimating the likely locations of less evident ones. The detection step can tolerate variations in illumination, scale and orientation. If the prominent feature is the eye, the eye must be open. This technique can easily fail when there is a change in prominent features. Usually the computational load associated with this method is not high.

Group facial features in face-like constellations [YCoEoC96] can be assessed using robust modeling methods such as statistical analyses. These approaches are suitable for complex backgrounds and various facial poses because they require few assumptions. However, the algorithms are not always sufficiently robust and the computational load of this method is high.

Snakes [WYPY96], is an energy-minimizing spline algorithm. Its operation is guided by external constraint forces and is influenced by image forces that pull the analysis engine towards features such as lines and edges. If the energy function is defined appropriately, it can be incorporated alongside high-level processes thereby reducing the reliance on low-level snake-mediated feature identi-

fication. However, this algorithm may not converge to a face-like shape due to the presence of non-facial edges in the vicinity. In addition, substantial manual tuning of the algorithm may be inevitable in many cases. The complexity of the snake algorithm is not high.

Deformable templates [NHI98] use a parametric shape vector with a few degrees of freedom. This approach can locate facial boundaries very accurately. Usually the algorithm is used following edge detection. The evolution of a deformable template is sensitive to its initial position because of its fixed matching strategy. The complexity is high.

The point distribution model (PDM) [LHCT95][CH96] is a compact statistics-driven shape modeling algorithm. All relevant features can be located simultaneously without the need for a search stage. It is often used in face verification and requires minimal prior knowledge to build models. The complexity is low to medium.

Linear subspace methods include PCA and Eigen-faces [PT91] that offer good performance when faces are frontally viewed. The complexity is medium-high.

Neural networks [RBK98] can be trained to capture the complex conditional class densities associated with facial patterns. However, it is challenging to determine a suitable network architecture and substantial fine-tuning is essential to achieve acceptable performance. There is a risk of over-training, and long-term training may lead to over-fitting, which can negatively impact results. The complexity of this method largely depends on the architecture of the network.

Some statistical methods [CH96][SK00] are based on information theory, which uses probability rules to find an exact solution. However, an extensive

knowledge of probability is needed, together with substantial training. The performance level depends on the statistical model used. The complexity is usually not high.

The Support Vector Machine (SVM) [OFG⁺97][LGL00] is used as a classifier [Bur98][LGSL04] to discriminate between face and non-face images. It has good generalization ability, relatively low error rate and offers high speed. It is mainly used for face verification. The disadvantage is that it is hard to train because it requires large data sets. Training is time-consuming but the computational load is low. Researchers continue to actively explore this method and a detailed implementation of SVM in this thesis is described in section 4.1.5.

The facial regions method involves computing the states of a continuous density HMM (Hidden Markov Model [NHI98]). A generic facial model is first trained from a large number of images. This approach has an excellent detection rate, but careful positioning of facial features within the template is essential. Disadvantages include a need for prior probability information and the time-consuming training process.

The fuzzy logic method [WCY99] applies fuzzy operators, properties and inference rules to detect faces in images. Benitez et al. [BCR97] have that proved that multi-layer feed-forward neural networks are equivalent to a system that is based on fuzzy rules. Fuzzy logic can be easily mapped to human knowledge and rule sets. It is usually combined with other methods and uses fuzzy rules to identify features extracted by other methods.

The Adaboost algorithm, introduced by Freund and Schapire[FS97], was firstly used by Viola[VJ02] to select face features and train the classifier for face detection. Viola[VJ02] combined multiple trained weak classifiers to form a

strong classifier and it became the most recently developed successful method. With the help of a cascaded structure and an integral image speedup technique, it has been reported to be one of the fastest methods in face detection. Several researchers have developed alternative boosting algorithms for feature selection. Lienhart et al. [LKP03] experimentally evaluated various boosting algorithms and weak classifiers. Their results showed that the Gentle Adaboost algorithm and CART decision trees have the best performance. In an extension of their original work, Viola and Jones proposed an asymmetric Adaboost algorithm [VJ01] in which false negatives were penalized more than false positives. FloatBoost incorporates the Floating Search into the Adaboost algorithm to overcome the non-monotocity problems associated with Adaboost. With this improvement, multi-view faces in 320×240 pixels were detected at 5 frames per second on a Pentium III 700Mhz PC [LZSZ02][YH93]. Bo Wu et al [WAHL04] extended the discrete Adaboost algorithm to a real-valued version and claimed a multi-view face detection runtime of about 80ms on a Pentium IV 2.4 GHz PC. They cited a rotation-invariant multi-view face detection runtime of about 250ms for a 320×240 image on the same hardware. The method is available in OpenCV (the most widely used open source computer vision library).

The following list summarizes certain key published results in the field of face detection.

1. The detection rate is usually somewhere between 80% and 95%, depending on the test database used.
2. Sample-based methods are more robust thanks in part to their learning abilities.
3. Feature-based methods that rely on color and motion are more practical

Table 2.7: Face detection methods.

Face detection methods	1.Feature-based methods	1.1 Low-level analysis	1.1.1 Edges
			1.1.2 Gray-levels
			1.1.3 Color
			1.1.4 Motion
			1.1.5 Texture
			1.1.6 Generalized measure
		1.2 Feature analysis	1.2.1 Feature searching
			1.2.2 Constellation analysis
		1.3 Active shape models	1.3.1 Snakes
			1.3.2 Deformable templates
			1.3.3 Point distribution models (PDMs)
	2.Sample-based methods	2.1 Linear subspace methods (Eigen-face)	
		2.2 Neural networks	
		2.3 Statistical methods	2.3.1 Distribution-based
			2.3.2 Naïve Bayes Classifier
			2.3.3 Information -Theoretical Approach
		2.4 Support Vector Machine	
		2.5 Hidden Markov Model	
		2.6 Fuzzy logic	
		2.7 Adaboost (FloatBoost)	

for real-time systems.

4. The fastest sample-based methods are almost all based on the Adaboost algorithm (see above discussion).

2.2.3 Databases for Face Detection

Table 2.9 shows some standard face databases that are widely used for research into face detection and recognition. The ECU and CMU databases were used for this thesis.

The CMU database (also known as the MIT CBCL FACE DATABASE) is

Table 2.8: Analysis of face detection methods.

Methods	Performance & Advantage	Limitation & Disadvantage
Edges [OFG ⁺ 97]	Scale, view, orientation invariant.	Need cost functions for shape analysis. Must be combined with other methods.
Gray-levels	Orientation invariant.	May be effected by view & ornament.
Color	Can detect almost all faces.	Must be combined with other methods.
Motion [LKP96] [YB99]	Very accurate for moving faces.	The extracted regions do not exactly define the complete face. An ellipse fitting algorithm has to be employed.
Texture [AS93]	Scale, view, orientation invariant.	Need a texture model and a classifier to classify the key features.
Generalized measure [RWY95][RY98]	Orientation invariant.	Frontal views only.
Feature searching[GT94] [Sir01][JLH ⁺ 98]	Detection of prominent features is robust to illumi- nation, scale and orientation.	Can easily fail when there is a change in prominent features.
Constellation Analysis [YCoEoC96]	More stable in complex backgrounds and with various facial poses.	Hard to find robust models. Experimental results determine the final performance.
Snakes [WYPY96]	If a energy function is defined properly, it may incorporate high-level processes instead of low-level feature identification.	May not converge to a face-like shape due to the presence of other local edges. Significant manual parameter tuning may be inevitable
Deformable templates [NHI98]	Can locate a facial boundary accurately.	The evolution of the template is sensitive to its initial position be- cause of the fixed matching strategy.
Point distri- bution model [CH96]	Allows all the features to be located simultaneously. No need for feature searching.	Require prior knowledge to build model.
Linear subspace methods[PT91]	Efficient for detecting frontal faces.	Moderately time-consuming and challenging to use in multi-view.
Neural net- works[RBK98] [PS92][JM96] [LKL97] [RBK97]	System can be trained to capture the complex class conditional density of face patterns.	Hard to determine the architecture whi- ch requires substantial tuning to achieve exceptional performance. Overtraining or long-term training and initialization may negatively impact the results.
Statistical methods	Can find the ideal solution based on probability rules.	Need probability knowledge and training.
SVM[LGL00] [OFG ⁺ 97] [Bur98]	Good generalization ability. Moderately low error rates and runs fast.	Hard to train for large data sets.
Hidden Markov Model [NHI98]	Excellent detection rates.	Frontal-view. Prior probability.
Fuzzy logic [WCY99]	Easy to map rules to human knowledge.	Need to generate fuzzy rules from human knowledge.
Adaboost[VJ02] [ZZLZ02]	Relatively high detection rate and fast method.	Hard to train weak classifiers. Frontal view only.

Table 2.9: Widely-used face databases.

Data set	Location	Description
UMIST	http://images.ee.umist.ac.uk/danny/database.html	564 images of 20 people. Each covering a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. Each subject exists in their own directory labelled 1a,1b, ... 1t and images are numbered consecutively consistent with order of capture. The files are all in PGM format approximately 220×220 pixels in 256 shades of gray.
CMU Test Set (MIT CBCL)	http://vasc.ri.cmu.edu/idb/html/face/index.html and http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html	Includes five datasets. PIE Database, the CMU Pose, Illumination, and Expression (PIE) database. Frontal images, test images for algorithmic face detection task. Profile images, test images for the face detection task. Training images, including for human face detection. Facial expressions, CMU-Pitt facial expression database.
BioID Database	fau.pdacentral.com/pub/facedb/readme.html	1521 grayscale images with human faces, recorded under natural conditions, i.e. various illumination types & complex backgrounds. Eye positions have been manually adjusted for ease of performing accuracy calculations during face detecting. A formula is provided to normalize match/mismatch decision-making. All images have resolutions of 384×286 pixels.
Yale Test Set	http://cvc.yale.edu/projects/yalefaces/yalefaces.html	GIF format, gray-scale, 15 subjects, 11 per subject with different facial expressions and lighting: center-light, w/glasses, happy, left-light, w/no glasses, normal, right light, sad, sleepy, surprised, and wink.
AR-Face Database	rvl1.ecn.purdue.edu/v1/ARdatabase/ARdatabase.html	126 people (over 4,000 color images). Different facial expressions, illumination conditions and occlusions.
FERET	http://www.itl.nist.gov/iad/humanid/feret/feret_master.html	The Facial Recognition Technology (FERET) Database. This image database was assembled to support government monitored testing and evaluation of face recognition algorithms using standardized tests and procedures. The final data set consists of 14051 eight-bit grayscale images of human heads with views ranging from frontal to left and right profiles.
ECU	Not available online	A large and comprehensive face detection database at ECU (Edith Cowan University).
AT&T (formerly 'The ORL Database of Faces')	http://www.uk.research.att.com/facedatabase.html	Ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

a database of faces and non-faces that has been used extensively at the Center for Biological and Computational Learning at MIT. It is freely available for research use, and can be downloaded from <http://cbcl.mit.edu/software-datasets/FaceData2.html>. The CMU test database includes 472 face samples and 23,573 non-face samples, and its training database includes 2429 face samples and 4548 non-face samples. The face training set was generated for [Sun96]. The non-face training set was generated for [HPP00]. The test set is a subset of CMU Test Set 1 [RBK98]. Information about how this subset was chosen can be found in [HPP00].

Chung [Phu03] created, as part of his Ph.D. research, a large and comprehensive face detection database at Edith Cowan University. Compared with other existing face detection database, this database is unique in three important respects. Firstly, the database consists of color images, and therefore supports both color-based and intensity-based face detection algorithms. If face detection needs to be done in the grayscale domain, the images can be converted to grayscale. Secondly, the database consists of a large number of images (over 3,300 at the time of writing). These images vary significantly in terms of the graphic size, background, person, lighting condition, facial expression, and facial pose. Therefore, the database encapsulates a wide spectrum of face detection scenarios. Thirdly, the database contains manually produced ground-truth images. In this thesis, two sets of the ECU database (set 6 & set 7) are widely used including 9339 aligned face and 8951 non-face samples.

2.3 Issues Related to Face Detection

2.3.1 Face Tracking

If a face detection algorithm were fast enough to detect face(s) in every frame in real time, the technique would essentially be equivalent to face tracking. However, normally face tracking has its own unique characteristics, such as the need for initialization. Generally, in a video or image sequence, face tracking utilizes every frame of the video [JW01]. In contrast, static face detection naturally requires just one frame. Occlusion and shading often make static face detection methods very difficult. However, utilizing information from multiple image frames means face tracking algorithms can be more robust. In general, face tracking algorithms contain two stages: a prediction stage and a verification stage. The verification stage is usually computationally intensive as it involves a search within the image [BJ98]. Table 2.10 summarizes two methods of face tracking.

2.3.2 Multi-view Face Detection

Statistics show that approximately 75% of faces in home photos are non-frontal [KPC⁺99]. Therefore, it is also important to consider multi-view faces in image sequences and videos. There are 3 degrees of freedom that describe possible movement of the human face: tilting up and down, rotating to the left and to the right, and rotating into or out of the image plane.

Three methods have been proposed to address this situation. The first involves building a single detector that can process multiple different types of

Table 2.10: Face tracking techniques.

Methods	Description	Performance	Assumption	Comment (Disadvantage)	Complexity
Kalman Filter	Accurately predict the location of each target in the next frame in order to minimize the number of false matches and reduce the number of hypothesized trajectories.	Provide highly accurate position estimates.	Preset the initialized area.	Less robust because it lacks the ability to globally localize the object and to recover from localization errors [GBFK98].	$O(n^3)$
CAM-SHIFT [B ⁺ 98]	Continuous adaptive mean shift algorithm.	Fast, noise robust, distraction robust.	Preset the initialized area.	Affected by skin color, object occlusion and lighting variation.	$O(an)$, where a is some constant.

rotated image. However, the single detector is seldom adopted because it is very complex and improving its accuracy is a significant challenge.

The second method involves building several detectors, each of which deals with a specific view. A famous example is FloatBoost, proposed by Stan Z. Li et al. [LZSZ02][YH93]. This algorithm includes a detector-pyramid architecture designed to identify multi-view faces very efficiently. The pyramid adopts an integrated strategy of coarse-to-fine view decomposition, and simple-to-complex face/non-face classification.

The third method, as proposed by Yongmin Li [LGSL04][Yon01], is known as the “pose.” It generates an estimate of a given pattern before choosing only one of the view-based face detectors to determine whether the target image is a face.

The problem of multi-view face detection can be divided into multiple sub-

Figure 2.2: Multi-view faces [Yon01].

problems. The view space can be split into 8 segments: left profile, left frontal, right frontal, right profile in the horizontal direction (yaw), and upper and lower in the vertical direction (tilt), as shown in Figure 2.2. When constructing a view-dependent piecewise face detector, certain strategies [Yon01] may be adopted:

a) Faces are symmetrical about a vertical line across the nose-bone. As a result, the right half of a face can be obtained by mirroring the left half without losing general face characteristics. Only four detectors need be constructed, as illustrated in Figure 2.2.

b) Soft boundaries between segments, i.e. overlaps by 10° between neighboring segments, allow for seamless detection.

c) The separating angle is vertical 0° in tilt and horizontal $\pm 5^\circ$ in yaw. Use of this angle can separate one-eyed faces (profile) from two-eye faces (frontal).

Figure 2.3: In-plane rotation invariant neural network [RBK97].

Multi-view may be considered an out-of-plane rotation. In-plane rotation often occurs in video capture scenarios. One concept requires rotating the original image by a certain angle and then comparing it with the frontal-view face detector. This process is repeated as many times as necessary. To speed up the detection, Rowley etc [RBK97] proposed a rotation system that they termed a “Router network” to detect the possible direction of the face as shown in Figure 2.3.

2.4 General Procedures of Sample-Based Face Detection

2.4.1 General Pyramid Framework for Sample-Based Methods

A pyramid framework for the brute-force search is commonly used in sample-based face detection to solve the problem of face localization, i.e. the scale and location of face(s). An illustration of the pyramid framework for sample-based

Figure 2.4: Face detection system based on the pyramid framework.

face detection is shown in Figure 2.4. When the brute-force search strategy is used, face detection is a rare event detection problem because among the many image regions, only very few contain faces [WRM03]. The resulting classifier design problem is very challenging in that the detection rate must be very high in order to avoid missing any rare events. At the same time, the false positive rate must be very low (i.e. 10^{-6}) in order to avoid a proliferation of non-events.

Under the pyramid framework, every input image is cropped, at different positions and scales, into normalized sub-images. Each sub-image is then classified as face or non-face. The complexity of this framework is analyzed in the following paragraphs.

Suppose that the size of the input image is $x \times y$, the size of the normalized image used for classification is $x_0 \times y_0$, and the scaling factor is r . Let S_x and S_y be the step values in the x- and y-directions respectively. It can be shown that T , the total number of the sub-images generated by the input image, is:

$$T \leq \frac{xy}{S_x S_y} \frac{r^2 - \min([x_0/x]^2, [y_0/y]^2)}{r^2 - 1} \quad (2.1)$$

The derivation of the equation (2.1) is given in appendix A. Typical values of x_0 and y_0 are around 20 and S_x and S_y are in the range 1 to 5 pixels. Figure 2.5 shows the distribution of T generated by the pyramid framework when S_x and S_y are set to 1 based on equation (2.1). The x-coordinate represents the scaling factor r and the y-coordinate represents the size of the input image. When $r = 1.2$ and the input image size is 256×256 (see point A in Figure 2.5) T is a little over 200,000.

Accordingly, equation (2.1) becomes:

$$T \leq \frac{256 \times 256}{20 \times 20} \frac{r^2 - \left(\frac{20}{256}\right)^2}{r^2 - 1} = 163.84 \times \frac{r^2 - 0.0061}{r^2 - 1}$$

Frequently, one has to adjust the available parameters to reach a compromise between accuracy and processing time. Note, from equation (2.1), that the total number of the sub-images, T , is inversely proportional to the step values, S_x and S_y . Therefore, if S_x and S_y are chosen to be from 3 to 5, T will be reduced by 9 to 25 as shown in Figure 2.6 derived from equation (2.1).

Accordingly, equation (2.1) becomes:

$$T \leq \frac{x^2}{S^2} \frac{1.2^2 - \left(\frac{20}{x}\right)^2}{1.2^2 - 1} = \frac{3.273x^2 - 9.09}{S^2}$$

Even so, the real-time processing is still very difficult. In addition, based on [VJ02], the accuracy of face detection might reduce by 1-2% when r is

Figure 2.5: The number of sub-images in the pyramid framework.

Figure 2.6: The number of sub-images T generated in pyramid framework w.r.t step size S ($S_x = S_y$).

increased. There exist techniques such as hierarchical classification (rejection-based face detection) [EHOK02][VJ02], feature reduction [HSPP03], and coarse-to-fine classifiers [SB02] that can accelerate the processing.

Equally important, the post processing also affects performance of the entire detection system. Some well-known methods have been described by Rowley et al [RBK98]. Figure 2.7 demonstrates a general framework for merging multiple detections from a single network from Figure 6 in [RBK98]. Step *A* shows that the detections are recorded in an “output” pyramid. In Step *B*, the detections are “spread out” and a threshold is applied. After that, the centroids in scale and position are computed, and the regions contributing to each centroid are collapsed to single points in step *C*, which leaves only two detections in the output pyramid. Step *D* is to check the proposed face locations for overlaps. The final step *E* removes overlapping detections if they exist. In the example shown in this diagram, removing the overlapping detection eliminates what would otherwise be a false positive.

Another issue is how to improve accuracy after assembling several network outputs. Figure 2.8 shows that an “AND” operation together with the outputs from two networks over different positions and scales can improve detection performance (Ref. Figure 7 in [RBK98]). Nowadays, this step is seldom used in this stage because, firstly, more complex regression algorithms are used in the detection stage and secondly, only one network is usually employed due to the limitations of speed and storage resources.

Figure 2.7: The framework for merging multiple detections from a single network from Figure 6 in [RBK98].

Figure 2.8: ANDing together the outputs from two networks over different positions and scales can improve detection performance from Figure 7 in [RBK98].

2.4.2 A general procedure of training and constructing a sample-based classifier

A general block diagram of training and constructing a classifier-based face detection system is shown in Figure 2.9. In such a system, two procedures are always included: training and testing. During the testing procedure (the detection procedure), the input image is first scanned exhaustively, using 24×24 windows for example, in a cascade manner at different scales. The scanned windows are then passed through the preprocessing unit that performs basic operations such as histogram equalization, brightness and contrast adjustment. Finally, the feature vectors are extracted and used as the input of the classifier to judge whether the windows contain a face. The classifier is obtained from the training procedure that operates on the basis of a different learning method.

2.5 Conclusions

In summary, a number of sample-based methods have been applied to the problem of face detection. However, a little research has been carried out to investigate method efficiency and further accuracy improvements remain unexplored. The major issue discussed in this thesis involves the improvement of efficiency through judicious use of training sets (see Chapters 3 and 4). On the other hand, face detection is always viewed as a typical two-class classification application that can be used for testing and verifying the performance of any proposed new classification methods. In Chapter 5, a new method, termed PLCB, is proposed and compared with other state-of-art methods.

Figure 2.9: General block diagram of training and constructing a face detection system based on learning methods.

Chapter 3

Passive Sample Selection

The term “passive” in the title of this chapter refers to the fact that these methods make use of the only available information and query nothing from the outside, including the (human) oracle. Most of the existing learning methods treat learning as a passive process, i.e. they only utilize the information presented to them. Such schemes are accordingly called “Passive Learners”. In this chapter, a passive selection method is proposed to reduce samples in order to improve the accuracy and increase the speed of subsequent processing. An overview of sample selection is initially described and the empirical results are discussed at the end of this chapter.

3.1 Overview of Sample Selection

3.1.1 Motivation for Sample Selection in Face Detection

As mentioned in Chapter 2, work on face detection has been ongoing for many years and most of the best reported face detection algorithms are sample-based. SVM and the Adaboost algorithm are two popular examples. In typical sample-based applications, massive or even boundless data can be collected for training databases. However, most of the time, these methods are inadequate or imbalanced because the accuracy of many applications trained by using these databases may not meet performance requirements. Nevertheless, it remains relatively easy to collect face samples (as compared with many other contexts such as healthcare records). In the situation where face samples are not readily available, additional face images can be artificially generated as explained in section 3.1.3. In such a case, information redundancy exists at the beginning of the training stage in the expanded data set, and this may throttle the training procedure. Random selection from the expanded data set may result in more imbalance of the whole training set, which may adversely affect the classification performance.

Two general techniques could be used to solve the above problems. One is to generate some new high-efficiency samples. This technique is one of the major contributions of this thesis and will be elaborated on in Chapter 4. The other method is to efficiently select face samples that cover the expanded face samples as comprehensively as possible. In many reported research, training samples were simply randomly selected while not considering their completeness and distribution. Not enough studies have been reported that focus on how

Figure 3.1: Original 16 face images from the ORL face database.

to select samples from a large redundant database and whether the reduced database covers enough face space for appropriate training.

Feature selection can be used to reduce the input dimensions [BL97], which also reduces computational complexity. However, the feature extraction stage is considered one of the most difficult and critical tasks in many applications, including face detection. Sample selection is another technique commonly used to reduce the input dimensions. According to mathematical theory, selecting samples based on their distributions is the best choice. However, as often occurs in most applications, the exact distribution is unknown.

Manual sample selection is possible but tends to be time-consuming and subjective. A human's subjective feeling may very well be wrong, as shown in the following simple example. Sixteen 20×20 face images obtained from the ORL face database (<http://www.uk.research.att.com/facedatabase.html>) are shown in Figure 3.1. TICA (Topographic Independent Component Analysis) [HHI01] defined a topographic order where the simplest topographic order, a 1-D, 3×3 neighborhood relation, is applied to these sixteen 20×20 face images. The result of TICA shown in Figure 3.2 has four subspaces. The subspaces are not directly grouped by gender. It is stated in chapter two of [Yon01] that “*the*

Figure 3.2: TICA processing of images in Figure 3.1. Grouped by top left 4, top right 4, bottom left 4 and bottom right 4 images.

similarity between two images of the same person in different poses is less than the similarity between images of two different persons in the same pose". The result may be counter to human intuition and shows that subjective feeling can be inconsistent. Another direct inference of this example is that frontal-view based face detection and recognition methods may fail in multi-view face detection and recognition.

In section 3.2, a newly proposed technique, passive sample selection, is developed to tackle the problem of inadequate or imbalanced training data in face detection based on existing training databases.

3.1.2 Bootstrap Review

Bootstrap is a well-known sample-selection algorithm that is widely used in face detection to select "Non-face" samples that are highly relevant to the learning problem. It is also referred to as "Bootstrapping" or "Boot-strap". In [SP98], the Bootstrap strategy is identified as follows:

1. Start with a small and possibly highly non-representative set of "non-face" examples in the training database.

2. Train a face classifier to output a value of “1” for face samples and “0” for non-face samples using samples from the training database.
3. Run the trained face detector on a sequence of images with no faces. Collect all (or a random subset of) the non-face samples that the system wrongly classifies as faces (i.e. those that have an output value of > 0.5). Add these non-face samples to the training databases as new negative examples.
4. Return to Step 2.

Algorithm 1: Bootstrap Algorithm

The study in Sung and Poggio [SP98] concluded that the Bootstrap strategy may be “sub-optimal” in choosing new training samples. However, in practice, the technique is simple and effective for sifting through unmanageably large data sets.

The Bootstrap method is often applied to iteratively update the non-face set when starting with a relatively small data set as shown in Figure 3.3. The Bootstrap-based non-face selection scheme can lead to a significant increase in training time since each iteration will inevitably involve training a new classifier [GLZ01].

The proposed method described in this chapter has no conflict with the Bootstrap method, i.e. they can be applied in cascade. The motivation of the proposed method is to reduce the significant redundancy found in the artificial expanded face training data. In contrast, Bootstrap is often used to select more non-face training data. The active sample generation method proposed in the

Figure 3.3: Left: images of scenery containing no faces. Right: regions in the left image falsely detected as faces will be added into the set of non-face training samples.

next chapter aims to further improve the accuracy of an SVM classifier with the help of human experts

3.1.3 Artificial Sample Generation

Frontal-view face samples can be easily expanded using some simple geometrical transformations on image. Figure 3.4 (a) - (f) show the method described on page 81 of [Sun96]. In [LKP03], shifting the original image up/down or left/right led to the generation of four new patterns from just one original image (shown in Figure 3.4 (g) - (j)). Therefore, four independent dimensions (shift left-right, shift up-down, rotation ± 5 degree, and mirror) could be used to generate new patterns from the foundation image. If the image is shifted by one pixel, and rotated by five degrees, 53 new patterns can be generated. In this way, the data set could readily be artificially expanded 54 times. Other image processing techniques such as adding noise and non-uniform illumination can also be applied and these processing stages can be cascaded. The general rule is to make the results different from the original but still recognizable as

Figure 3.4: Typical sample images: (a) Original image (b) Mirror image (c) Rotate +5 degree (d) Mirror image rotate +5 degree (e) Rotate -5 degree (f) Mirror image rotate -5 degree (g) Move up one pixel (h) Move left one pixel (i) Move down one pixel (j) Move right one pixel.

faces. In summary, the expanded face database can be made much larger than the original.

Figure 3.5 shows two clusters of the above artificial expansion from 2 original face samples. Every sample will be expanded 54 times. The x-coordinate represents the first feature of PCA while the y-coordinate represents the second. In Figure 3.6, twenty face samples are selected and shown in the same manner as in Figure 3.5. Figure 3.7 shows the expansion from the samples of Figure 3.6. It is obvious that more information is embedded in this operation but at the same time much more redundancy is also generated.

In [LKL97], a similar method named “Training Pattern Generation” was used to expand the face database. It had three steps. The first step involved “virtual training patterns” that generated more samples via various affine transformations (e.g., rotation, scaling, shifting) and mirroring processes. The second step was “Positive/negative training patterns”. The idea was to manually eliminate those samples with low confidence. The third step was “Run-time negative

Figure 3.5: Clusters of the artificial expansion from 2 original face samples.

Figure 3.6: 20 face samples represented by 2 main PCA features.

pattern generation” that is the same as the Bootstrap technique described in section 3.1.2.

In [CCG04], re-sampling using Genetic Algorithms was applied to expand the face database as shown in Figure 3.8. According to [CCG04], the fitness evaluation was carried out by SNoW [YRA00] (Sparse Network of Windows). As shown in Figure 3.8, the parent face sample was broken down into smaller pieces without overlap. Features included the forehead, eye, nose, and mouth, as demonstrated in Figure 3.8(a). The crossover process is shown in Figure 3.8 (b). According to this method, mutation is accomplished by sharpening, blurring or lighting. The procedure of sharpening or blurring involves first obtaining a sub-image, about a quarter to half the size of its parent, from its parent. This image is then sharpened or blurred randomly; and finally the algorithms recombine the changed sub-image and the unchanged part to reproduce its child. To avoid including the trace that results from recombination, the intermediate solution is smoothed as shown in Figure 3.8 (c). As for lighting, the researchers used the same scheme as that in [Krü01]. However, the final samples were still manually checked and selected.

It is obvious that a large amount of redundancy will be present in the expanded database. This may impair the training procedure. Therefore, sample selection is essential to make sure that the expanded database will not be overly redundant.

To compare various methods, the author uses the original intensity, PCA [SK87], ICA architecture 1, and ICA architecture 2 [BS97] to represent the original images. The reason for experimenting with PCA features is that this technique was found to be successful when used for face recognition and detec-

Figure 3.7: Clusters of the artificial expansion from samples in Figure 3.6.

Figure 3.8: Crossover and mutation from [CCG04] (a) Each parent is converted into a sequence of observation vectors for crossover, (b) Crossover, (c) Mutation.

tion [BS97]. Performances of ICA1 and ICA2 are compared with that of PCA. ICA might represent face features better than PCA because PCA assumes that the distribution of face features must be orthogonal. However, ICA might not always perform better than PCA because the data may be insufficiently complete. A simulation result that addresses this issue will be discussed in section 3.2.2. According to [BMS02], ICA1 tries to find a set of spatially independent basis images while ICA2 tries to find a representation in which the coefficients used to code images are statistically independent.

3.1.4 ICA

General definition: ICA (Independent Component Analysis) of the random vector $\mathbf{x}$ consists of finding a linear transform $\mathbf{s} = \mathbf{W}\mathbf{x}$ so that the components $\mathbf{s}_i$ are as independent as possible, in the sense of maximizing some function $F(\mathbf{s}_1, \dots, \mathbf{s}_m)$ that measures independence [VJ02].

Statistically independent sources $\mathbf{s}_1, \dots, \mathbf{s}_m$ are mixed together, linearly or non-linearly, to produce the observed signals $\mathbf{x}_1, \dots, \mathbf{x}_m$. The observed signals may be expressed as:

$$\mathbf{x}_i = \sum_{j=1}^n a_{ij} \mathbf{s}_j = a_{i1} \mathbf{s}_1 + a_{i2} \mathbf{s}_2 + \dots + a_{in} \mathbf{s}_n; \quad 1 \leq i \leq m \quad (3.1)$$

or in matrix form, $\mathbf{x} = \mathbf{A}\mathbf{s}$

All observed signals are instances of the random vector $\mathbf{x}$. Both $\mathbf{A}$ and $\mathbf{s}$ must be estimated using only $\mathbf{x}$. It is impossible to solve for $\mathbf{A}$ and $\mathbf{s}$ using only the equation above. The key argument for the ICA model is non-Gaussianity. The Central Limit Theorem, a classical theory in probability, is invoked. The theory states that the distribution of a sum of independent random variables tends to move toward a Gaussian distribution under certain conditions. Therefore, a sum of two independent random variables usually has a distribution that is closer to Gaussian than either of the two original random variables.

According to the research in [HHI01], the ICA method is equal to a combination of “objective functions” and “optimization algorithms”. The statistical properties (e.g. consistency, asymptotic variance and robustness) of the ICA method depend on the choice of the objective function, and the algorithmic properties (e.g. convergence speed, memory requirements, and numerical stability) depend on the optimization algorithm used. There exist many ICA models based on various assumptions about the characteristics of the noise and the source densities [RE01].

In the late 1980s, PCA (Principal Component Analysis) was used by Sirovich and Kirby [SK87] to process images of human faces. Eigen-face, as shown in Figure 3.9, is a method based on PCA. The performance of PCA in face rep-

$$\begin{array}{c} \text{Face Image} \end{array} = C_1 \begin{array}{c} \text{Eigen-face 1} \end{array} + C_2 \begin{array}{c} \text{Eigen-face 2} \end{array} + C_3 \begin{array}{c} \text{Eigen-face 3} \end{array} + C_4 \begin{array}{c} \text{Eigen-face 4} \end{array} + C_5 \begin{array}{c} \text{Eigen-face 5} \end{array} + \dots$$

Figure 3.9: Eigen-face using PCA.

Figure 3.10: Comparisons between PCA and ICA.

representation is very good for aligned and scaled human faces. However, for non-aligned faces, its performance degrades rapidly; as a consequence, the search for new techniques began. ICA, which is one of the improved methods, was proposed by Barlett and Sejnowski [BS97]. Figure 3.10 shows the relationship between ICA and PCA. Two eigen-vectors from PCA are orthogonal and they represent the Gaussian distribution very well but they fail in the case of data that does not follow a Gaussian distribution. PCA may be considered a special case of ICA. When the distribution of data is orthogonal, ICA will be the same as PCA. Theoretically, ICA offers several advantages over PCA for face representation. First, ICA decorrelates higher-order statistics from the training signals, while PCA only decorrelates second-order statistics. Secondly, ICA basis vectors are better spatially localized than the PCA basis vectors, and these localized features generate better face representation [OFG⁺97]. Thirdly, certain independent components are less sensitive to variations such as facial expressions, small occlusions and pose variations. ICA has its own disadvantages, such as the need for heavy computational resources.

3.2 The Proposed Algorithm for Selection of Samples

In this section, a new method is proposed to select samples from the training database. The passive method performs clustering on existing samples. It is especially appropriate for artificially expanded samples and it selects center points using the k-means algorithm[SP98] for further processing. Using this method, it is possible to select face samples to cover the entire expanded training

data set as much as possible.

3.2.1 The Proposed Method

The author proposes performing clustering on the training samples and then selecting center points using the k-means algorithm so that these clusters are spread according to the distribution of the whole training set. As the entire training set distribution ends up being better than that of the original training set, the performance of the trained classifier could be improved without collecting additional data. K may be larger or smaller than the number of original samples, depending on the nature of the application. However, ideally, the best choice for K should be the same as the number of original samples when the database is artificially expanded from the original samples and considered to have K clusters.

To compare the performance of the various methods, the original image, PCA [PT91], ICA architecture 1, and ICA architecture 2 [BMS02] were applied to represent the original images. Then, the k-means algorithm was used to cluster the features of images so that these clusters were spread as widely as the whole training set. These methods are referred to as intensity selection, PCA selection, ICA1 selection, and ICA2 selection respectively. ICA1 tries to find a set of spatially independent basis images while ICA2 tries to find a representation by which the coefficients used to code images are statistically independent. The k-means algorithm is adopted here because of its simplicity. In the case of the k-means algorithm, it has been shown in [Hay94] that the codebook-vectors do not faithfully represent the underlying probability densities. Rather, they are distributed according to $p^{1/3}$ where p is the data density. However, the aim of

Figure 3.11: Cascaded structure of the proposed sample selection technique.

the proposed method is to cover the data sets as widely as possible, and not to only base outputs on the data density itself. The k-means algorithm is a method whose tradeoff between the low-density data and the high-density data is about one-third. Finding an optimized tradeoff value remains an unaddressed research topic. However, based on experimental results in section 3.2.2, the k-means algorithm has been shown to produce good performance.

A cascaded structure as shown in Figure 3.11 can be used to further improve the performance using the same number of samples. This structure can solve the problem of huge training data sets being generated from the artificial expansion process. The number of stages, n , can be unlimited and should match the number of approaches required for artificial generation as discussed in 3.1.3. The performance of every stage as analyzed in section 3.1.3 and the experiments shown in section 3.2.2 could thereby be improved without limit.

3.2.2 Performance

There are many tools available for SVM implementation such as Herosvm [DSK03], Svmlight [Joa99] etc. In this thesis, a widely used SVM tool, LibSVM [HCL03][CL01], was adopted for training and testing. In the SVM training stage, capacity parameter C who trades off complexity and empirical risk is empirically set to be 50. The other parameters used were the default values used in LibSVM. The settings of the experiments are for consistency and they may not be the best parameters.

In the evaluation experiments in Chapter 3 and Chapter 4, each image pattern, was processed by using histogram equalization and then quantized into 50 gray-scales (rather than the typical 256) similar to earlier work by Ming-Hsuan Yang (<http://vision.ai.uiuc.edu/mhyang>). These quantized image patterns are used as the input for the SVM classifiers and the input of the training data set. In the evaluation experiments in Chapter 5, the normalization is done by mean and standard deviation.

In the thesis, Equal Error Rate (EER) is used when using only one dimension data to show results. When adjusting a threshold to the output score, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) will change accordingly and intersect at a certain point as shown in Figure 3.12. The value of the FAR and the FRR at this point, which is of course the same for both of them, is called the Equal Error Rate (EER). In the thesis, EER is used to represent the accuracy at EER point.

In the first experiment, each image pattern, 20×20 pixels in size, was processed as described in section 4.1.5. Each pattern was reshaped into a vector of 400 dimensions and PCA was subsequently used to reduce it to 200 dimensions.

Figure 3.12: Equal Error Rate.

It is possible to use other dimensions and the 200 dimensions are used to be the same as ICA1 and ICA2. PCA output was used as the input to the k-means algorithm, which clustered the 200-dimension principal components of N images. In ICA1 and ICA2 selection, 500 training patterns from an available total of 4,000 were used to generate the 200 ICs (Independent components). The weights of the ICs from all 4,000 training patterns were then clustered using the k-means algorithm.

ECU database sets 5, 6 and 7 were used for the evaluation. As mentioned in Chapter 2, there were 9,339 face images in set 5 and 8,951 non-face images in sets 6 and 7. The author randomly selected 4,000 faces and 4,000 non-face images for training and used the remainder for testing. In typical images, the area of the face region was usually much smaller than that of the non-face zone. Accordingly, it is clear that the non-face testing data set should end up being

much larger than the face testing data. The CMU test database was also used for testing and the CMU training database was used for face redundancy testing thereafter. The LibSVM [HCL03][CL01] program was used for SVM training and testing.

Results generated using the proposed new method were compared against the results generated from random sample selection, which is commonly used for training learning algorithms. Because the random selection process was associated with a risk of relatively large variations in results among different training runs, the performance of the random selection engine was evaluated by repeating the process 20 times. Figure 3.13 shows variation inherent to the random selection process. 40 groups were generated and each group contains 100 random faces. The minimum, average, and maximum performances over the 20 runs were used as the criteria against which to measure the performance of the random selection as shown in Figure 3.14.

Large variations in the results of sample-based face detection methods can be attributed to an incomplete distribution of the training samples. This is a side-effect to the random sample selection method. For example, rare face patterns such as those that feature people with glasses or beards are relatively small in number (low sampling density) and thus there is a high probability that these would not be selected by the random selection algorithm. In contrast, the proposed method uses the k-means algorithm to achieve a more widely distributed training data set. This means that those infrequent samples with low probability are more likely to be included in the training set. Figure 3.15 shows 60 face patterns selected from the ECU database using the proposed selection method. Note that these patterns include faces of the young and

Figure 3.13: Random selection distribution using ECU and CMU data sets.

Figure 3.14: Face detection EER against the number of training samples of random selection and PCA sample selection.

Figure 3.15: The 60 selected face patterns from ECU database.

the old, men and women, people with and without glasses, and people with wide open mouths and those with closed mouths. Other variations among faces are due to illumination, scale, and expression. The proposed selection method generated comprehensive training samples, which enabled the learning machine to achieve a higher generalization accuracy.

Since the selected training samples are better distributed after using the clustering selection method as explained in section 3.2.1 and 3.2.3, the learning machine is able to achieve a higher accuracy than the random selection method. Figure 3.16 shows a performance comparison among random sample selection, PCA, ICA1, and ICA2 selection. The number of samples increases by a factor of 1.2. It is obvious that, for all methods, EER decreases with number of training samples.

Table 3.1: Accuracies and number of Support Vectors (nSV) from different methods.

Points	Accuracy(random selection)			Accuracy			nSV		
	Average	Max	Min	PCA	ICA1	ICA2	PCA	ICA1	ICA2
60	87.75%	91.33%	88.81%	86.56%	88.39%	88.73%	183	192	165
72	88.79%	91.72%	89.81%	89.51%	89.37%	89.60%	223	254	193
86	89.60%	92.02%	90.47%	90.01%	90.46%	90.13%	226	244	182
103	90.01%	92.27%	91.17%	92.37%	91.48%	91.90%	258	267	195
123	91.07%	93.25%	91.95%	92.78%	92.64%	92.32%	274	292	197
147	91.74%	93.46%	92.72%	93.45%	93.20%	93.38%	293	296	235
176	91.96%	94.16%	93.26%	93.98%	94.28%	93.55%	325	318	265
211	92.74%	94.71%	93.96%	94.97%	95.14%	93.67%	338	357	268
253	93.74%	95.09%	94.52%	95.11%	95.39%	94.47%	351	360	316
303	94.33%	95.52%	94.97%	95.71%	95.86%	95.23%	405	387	339
363	94.84%	95.83%	95.38%	96.12%	96.14%	95.54%	425	413	356
435	95.13%	96.35%	95.84%	96.38%	96.24%	95.82%	467	452	410
522	95.57%	96.53%	96.18%	96.42%	96.39%	96.34%	478	458	471
626	96.09%	96.83%	96.50%	97.09%	96.70%	96.70%	516	511	467
751	96.46%	97.11%	96.80%	97.29%	97.05%	96.86%	582	523	506
901	96.79%	97.46%	97.06%	97.32%	97.20%	97.05%	581	547	499
1081	97.08%	97.67%	97.31%	97.43%	97.48%	97.31%	618	618	586
1297	97.35%	97.90%	97.53%	97.63%	97.37%	97.24%	675	660	600
1556	97.57%	98.09%	97.75%	97.66%	97.56%	97.56%	685	689	683
1867	97.67%	98.22%	97.91%	97.67%	97.66%	97.72%	759	740	659
2240	97.88%	98.39%	98.08%	97.58%	97.83%	97.70%	791	795	754
2688	97.99%	98.51%	98.24%	97.86%	97.81%	97.70%	820	827	795

Figures 3.14, 3.16 and 3.17 are generated from the empirical data listed in Table 3.1. Note in Figure 3.14 that PCA performance results are similar to the maximum values achieved by random selection. This is especially true when the number of samples is in the range 2.5% - 20%. When the number of samples is very small (i.e. fewer than 100), EER of random selection is slightly lower. This is explained in section 3.2.3. The performance of PCA selection is close to the maximum performance of the random selection approach (see Figure 3.14) when the ratio between the selected samples and the total samples in the database reaches approximately 3-6%, or 100-250 of the 4,000 total samples. The performance of ICA2 selection is lower than that of ICA1 and PCA. ICA1 is slightly better than PCA selection when the number of samples is between 200 and 400, or around 5% to 10% of the total.

In Figure 3.18 and Figure 3.19, the ROC (Receiver Operating Characteristic) curves of the ECU and CMU databases are shown. The x-coordinate represents the false positive rate in non-face patterns and the y-coordinate represents the true positive rate in face patterns. Circles represent the samples selected from PCA and squares represent the results from random sample selection. The results from ICA1 and ICA2 selections are represented as star and triangle respectively. The plots are hard to view because the curves are overlapped. Note that in Figure 3.18 there are more circles and stars in the upper-left corner, which means higher detection accuracy. The same can be observed in Figure 3.19.

Comparing random sample selection against sample selection by PCA and the k-means algorithm, one concludes that, when the number of selected samples approaches the total number of face patterns, random selection produces a

Figure 3.16: Face detection EER against the number of training samples under various methods of sample selection.

Table 3.2: Redundancy database selection.

	CMU test set		ECU test set		CMU Accuracy	ECU Accuracy
	face	non-face	face	non-face		
Original set	25.42%	99.53%	46.56%	99.96%	62.48%	81.22%
Selection set	39.83%	97.74%	59.08%	99.50%	68.79%	84.14%
3 face samples	47.03%	96.56%	68.31%	99.47%	71.80%	85.63%
3 face & non-face samples	52.12%	98.57%	64.83%	99.83%	75.35%	87.59%

Figure 3.17: The number of support vectors against the number of training samples under various methods of sample selection for face detection.

Figure 3.18: ROC curves of face detection by various methods using the ECU database.

Figure 3.19: ROC curves of face detection under various methods using the CMU database.

Figure 3.20: Face detection accuracies using original data and after redundancy database selection (data from Table 3.1.)

slightly higher accuracy. This is demonstrated in Figure 3.14 and Figure 3.16. It is clear that clustering when the number of selected samples is not large enough (less than about 120) leads to lower performance than random selection. In this case, density-driven selection is the best approach. To test selection performance in the expanded database in cases of large redundancy, the CMU training sample sets were used to artificially generate 7,289 face samples from 2,429 original face samples. The CMU train non-face set contains 4,548 non-face samples. The performance improvement was clearly demonstrated in the results shown in Table 3.2 and Figure 3.20. EER refers to Equal Error Rate, which is defined as the average of accepted true and rejected false events. CMU EER decreased from 37.52% to 28.20% while the ECU EER value decreased from 18.78% to 12.41%. The results confirm that the greater the number of training samples, the higher the accuracy that can be achieved when the training samples are insufficient. Furthermore, the larger the number of original samples, the higher the redundancy and thus the better the performance of the selection algorithm.

3.2.3 Analysis of Passive Sample Selection

Training samples can be roughly separated into two classes, according to their distribution density in the data space. The classes contain high density samples and low density samples respectively. A trade-off between using the density distribution of the training samples and assuming a uniform sample space distribution should be considered for two reasons. Firstly, the estimated density distribution of training samples is not equal to the exact density distribution of sample space, i.e. the set of all possible face patterns in the face detection data space. Secondly, there are redundancies in the high density samples while infor-

Figure 3.21: Curves of high density and low density.

mation included in the low density samples may be very unique. Therefore, if the rate of the performance improvement contributed from low density samples to that from high density samples is higher than the rate of the probability of the low density samples to that of the high density samples, the samples in low density are more important. A more detailed discussion of this is provided in the following paragraphs.

Usually, in face detection applications, the detection rate monotonically increases with an increasing number of selected samples. This can be seen from the experimental results in Figures 3.14 and 3.16 in section 3.2.2. A threshold T can be defined at the position where the increase of the detection rate, i.e. the gradient at each point on the curve, becomes very small. This curve represents a general behavior of most training processes in face detection applications. The detection rate often increases much faster at low selected sample numbers than at high selected sample numbers.

When comparing the sample selection in high density area with the sample

selection in low density area, two curves, denoted as C_{high} and C_{low} , are shown in Figure 3.21. Let C_{high} represents the detection rate against added selected sample number for the high density samples. Similarly, let C_{low} represents the detection rate against added selected sample number for the low density samples. Referring to Figure 3.21, it is obvious that $T_{high} \ll T_{low}$ (T_{high} and T_{low} are the thresholds of T in C_{high} and C_{low}).

Let N indicate the number of the added selected sample. N_{high} and N_{low} represent the added selected samples from the high density samples and the low density samples. For $N_{high} > T_{high}$, the gradient $G_{high}(N_{high})$,

$$G_{high}(N_{high}) = \frac{C_{high}(N_{high}) - C_{high}(T_{high})}{(N_{high} - T_{high})} \quad (3.2)$$

will become increasingly small as N_{high} increases. Because $G_{low}(N_{low})$

$$G_{low}(N_{low}) = \frac{C_{low}(N_{low}) - C_{low}(T_{low})}{(N_{low} - T_{low})} \quad (3.3)$$

may be much larger than $G_{high}(N_{high})$, the inequality

$$\frac{G_{low}(N_{low})}{G_{high}(N_{high})} > \frac{P_{high}}{P_{low}} \quad (3.4)$$

will become true where P_{high} and P_{low} are the probabilities of high density samples and low density samples in the whole data space respectively. According to inequality (3.4) and consistent with the change of the generalization accuracy defined as

$$G_{Accuracy} = G_{low}(N_{low}) \times P_{low} + G_{high}(N_{high}) \times P_{high} \quad (3.5)$$

selecting low density data will be more informative and useful for sample-based methods than selecting high density data.

In the real world, training samples may be separated into more than two classes. More clusters may be chosen. However, the same conclusion will still hold: if the number of selected samples is large enough, sample selection according to distribution density in the sample space is not the best choice for achieving higher levels of accuracy. A trade-off should be considered when selecting from the low-density data and the high-density data. However, another conclusion could be drawn: if the number of selected samples is too small, selection according to density does become the best choice. In this situation, the proposed sample selection based on clustering will not outperform the random selection strategy. This is because clustering needs at least a certain number of samples to achieve good performance. This is also confirmed by the empirical results shown in section 3.2.2.

3.3 Conclusions

A sample selection method is proposed for face detection in this chapter. At first, face samples are artificially expanded N times. Subsequently, clustering is applied to select $1/N$ clusters. Empirical results show that this approach can improve the accuracy of sample-based face detection methods (SVM, as discussed in this chapter). Although SVM is a margin maximizing method, the margin cannot be fixed before the final support vector machine is determined. Using nonlinear kernel techniques, it is possible to locate the samples in the spatial domain that are in the margin of the mapped high-dimension domain.

That is why the author does not choose only those samples that are near the margin.

Usually, random selection leads to the selected samples being distributed according to the data density. However, it was shown that if the number of selected samples is large enough, sample selection according to distribution density in the sample space is not the best choice to achieve a higher accuracy. However, if the number of selected samples is small, selection according to density is the best choice. In face detection, the number of selected samples is very large and the expanded database is even larger. Therefore, the author proposes using clustering to select samples for face detection. Better performance was shown in section 3.2.2. A cascaded multi-stage structure clustering approach was also proposed to solve the problem of huge training data sets.

In the experiments, various features including intensity, PCA, ICA1 and ICA2, were used for clustering to select samples from a large face database. The results show that the selection clustering method improves accuracy, especially for intensity, PCA and ICA1 features when the ratio M/N is small. If a dimension reduction is not needed, PCA and ICA1 are not necessary because intensity offers the same performance. Note that ICA1 reduces the input from 400 to 200 dimensions and gives slightly better results in the selection range of 5-10% samples.

The samples in the training set determine the performance of sample-based face detection methods. Efficiently selecting training samples will definitely increase final performance. The proposed method still requires substantial computation. However, it separates the training stage into two parts, the pre-training part (i.e, the selection part). The work of selecting the training samples can be

carried out independent of the training part.

Chapter 4

Active Sample Generation

There was a South Korean movie called “JSA (Joint Security Area)” that showed a conflict in a security area between North Korea and South Korea. Sometimes the internal rules in different fields are very similar in this world. Similarly, the accuracy of classification also depends heavily on the ability to accurately identify the region common to both parties, known as “the margin” in the field of pattern recognition. In this chapter, the author proposes a new active sample generation algorithm that is used to generate new samples in the region of confusion, thereby improving the separation accuracy of the margin between two classes. The related work is reviewed at first and the empirical results are shown later.

4.1 Overview of Active Sample Generation and SVM

4.1.1 Active Sample Generation

Active sampling for pattern classification is an area in machine learning where, instead of learning only from given training data, the learning system actively requests specific training data that will maximize learning performance. If the requested training data are not available and the new training data are artificially generated, active sampling will be considered as active samples generation. Active learning was motivated by statistical experiments in which the very act of performing an experiment (acquiring a single training example) may incur significant cost [Fed72]. Cohn [CCTA94] and Hwang et al. [HOMR91] focused on active learning for pattern classification applications, with a common heuristic that required sampling a training example at or near a current estimated category boundary. They justified this approach by arguing that the functional approximation of the posterior probability is most uncertain near the category boundary. Existing applications randomly select training data near the decision hyper-plane, which is consistent with general applications of active learning in a multi-dimensional space [Sol96][KR90][WR92]. Other active learning strategies and applications, based on the concept of optimal experimental design, have been reported in [CL90][CGJ96][Mac92][Fuk96][PW91][WRM⁺02][KS93]. Recent active sampling strategies are essentially based on SVM due to its successful geometric justification for margin maximization [SC00] [Par04] [MMP04] [TK02].

In learning methods such as SVM, the accuracy with which a margin can be

identified is the most important factor determining the overall accuracy of the classification system. Support vectors, which are the points at the edge of the margins (a sample in feature space is represented as a point), are often more important than the internal points. Support vectors have successfully been used to reduce the number of required training samples [RTSB01][TLV⁺04]. Substantial work has been performed using Semi-Supervised SVM (S^3VM) [BD98] or SVM Active Learning [TK02] to improve the accuracy of inter-class margin identification. However, most of these methods need supplemental samples (either unlabeled or labeled samples). Minimal research effort has been dedicated to improving the accuracy of margins identification through active sample generation. In other words, little research has been carried out using original training samples to generate new unlabeled samples that are located at or near the margin. In [Par04], the author proposed an algorithm for linear separable applications. There the author mentioned that the nonlinear case may be broken into piecewise-linear cases. Alternatively, nonlinear decision boundaries may be determined by applying kernel functions. However, these linear equations cannot be applied to non-linear cases as the precondition is not satisfied. The motivation for active sample generation is derived from the way humans learn. People begin by learning from examples, after which they ask questions on material that is unclear. Can a trained classifier (i.e., one that is SVM-based) propose and request new samples after learning from original training samples? Can human experts assist trained classifiers to gradually improve their performance?

In this chapter, the author proposes a method for active learning that focuses on unlabeled points that are generated from support vectors in the margin between two classes. The proposed method can be used in any typical

two-class classification problem such as face detection. The main advantage of this method is that it does not need additional labeled or unlabeled samples. In many other applications (such as in the medical field), training databases are usually very small and unlabeled samples are difficult to collect. Even in the application of face detection where face and non-face samples are relatively easy to collect, searching for those samples that are sufficiently useful for a high-accuracy SVM-based classifier is difficult and time-consuming. The proposed method can help to alleviate this problem because the generated samples are located in the margin of the two classes. Therefore, if the generated samples are correctly labeled by an oracle, it is guaranteed in theory that the performance of the SVM-based classifier can be improved. The experimental results in section 4.4 confirm the author's analysis and conclude that the proposed method improves the performance of SVM in two-class classification applications.

4.1.2 Theories on SVM (Support Vector Machine)

The SVM (Support Vector Machine) was introduced by Boser, Guyon, and Vapnik [BGV92]. It has been successfully applied in many applications including bioinformatics, text and handwriting recognition [TK02][HS01][YC04].

Margin maximization is an effective way to explain the concept of SVM. The idea of SVM is to minimize the risk of over-fitting by choosing the maximal margin hyper-plane in a given feature space. Figure 4.1 shows an example of two-class classification using SVM. In theory, SVM is superior to methods based on the Empirical Risk Minimization (ERM) principle such as Artificial Neural Networks (ANNs). Specifically, the SVM method minimizes the upper bound on the VC dimension (Vapnik-Chervonenkis dimension) as opposed to ERM

Figure 4.1: Margin maximization explanation of SVM using a two-class classification.

methods that minimize the error on the training data.

Consider a simple case with two classes. Define the training vectors: $\mathbf{x}_i, i = 1, \dots, l$,

$$y_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is class 1} \\ -1 & \text{if } \mathbf{x}_i \text{ is class 2} \end{cases} \quad (4.1)$$

The SVM is formed by solving the quadratic programming problem:

$$\bar{\alpha} = \frac{1}{2} \arg \min_{\alpha} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^l \alpha_i \quad \text{subject to} \quad \begin{cases} 0 < \alpha_i < C, i = 1 \dots l \\ \sum_{i=1}^l \alpha_i y_i = 0 \end{cases} \quad (4.2)$$

All of the $\mathbf{x}_i$ corresponding to non-zero α_i are the Support Vectors (SVs).

The classifier is:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{\mathbf{x}_i \in SV_s} \bar{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{x}) + \bar{b}\right) \quad (4.3)$$

where

$$\bar{b} = -\frac{1}{2} \sum_{\mathbf{x}_i \in SV_s} \bar{\alpha}_i y_i [K(\mathbf{x}_m, \mathbf{x}) + K(\mathbf{x}_n, \mathbf{x})] \quad (4.4)$$

and $\mathbf{x}_m$ and $\mathbf{x}_n$ are different types of SVs. The Gaussian kernel is the most popular kernel function. Its equation is:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \quad (4.5)$$

The parameter C in equation (4.2), named as the Error Penalty, is used to set the trade-off between the margin width and the classification error. Making adjustments to this parameter through the use of cross-validation or empirical testing can help counter the over-fitting problem.

4.1.3 S^3VM (Semi-Supervised SVM)

In addition to regular induction, SVM can be used for transduction. In Transductive SVM, also known as S^3VM , uses both labeled training data and an additional unlabeled data set. The learning task involves assigning labels to the unlabeled data as accurately as possible. SVM can achieve good transductive performance by identifying the hyper-plane that maximizes the margin relative to both the labeled and unlabeled data. However, a substantial volume of unlabeled data may be required based on equation (4.6) and Figure 4.2 b.

$$\min_{\mathbf{w}, b, \eta, \varepsilon, z} \left\{ C \left[\sum_{i=1}^n \eta_i + \sum_{j=n+1}^{n+l} g(\mathbf{w} \cdot \mathbf{x}_j - b) \right] + \|\mathbf{w}\| \right\} \quad (4.6)$$

where η_i is a slack term added for each point so that if the point is misclassified, $\eta_i = 1$, and $g(\alpha)$ is the margin penalty function on unlabeled data $\mathbf{x}_j$, $j = n + 1, \dots, n + l$.

4.1.4 Active Support Vector Machine Learning

The goal in active learning [BEYL04] is to design and evaluate learning algorithms that can effectively choose which samples should be labeled by an oracle. SVM Active Learning is based on the theory of S^3VM . The active learning algorithm will extract the most informative unlabeled data (usually only a small percentage of the total unlabeled data) and subsequently query an oracle for correct labels.

Support Vector machine Active Learning applies support vector theory to identify the key criteria that should be used in selecting the query samples. In Tong and Koller[TLV⁺04], those samples in the version space that maximize the minimum distance to any of the delineating hyper-planes are used to choose the unlabeled data. Given a set of labeled training data and a Mercer kernel K , there is a set of hyper-planes that separate the data in the induced feature space F . This set of consistent hypotheses is regarded as the version space (refer to [Mit82]).

Nevertheless, the active learning method requires additional unlabeled data that must be statistically distributed in a way that mirrors the real distribution. For example, in the case of face detection, the techniques for generating new

Figure 4.2: (a) Solution found by robust linear program (b) Solution found by S^3VM [BD98].

samples in the space domain are introduced in section 3.1.3. The images can be altered to generate new, surrogate samples that can be requested from an oracle. The pixels in the images may be rotated, scaled, cropped or noisy. However, this method is blind and therefore generates substantial redundancy. The following sections will describe the proposed algorithm for generating new samples in accordance with margin theory. Using the proposed new technique, the performance of an SVM-based classifier may be improved.

4.1.5 Support Vector Machine (SVM) for Face Detection

The Support Vector Machine (SVM) has been used in many applications since it was first introduced by Boser, Guyon, and Vapnik [BGV92]. In 1997, Osuna et al. first applied SVM in face detection [OFG⁺97]. The brute-force search of the input image in the SVM method is almost the same as the search strategy

Figure 4.3: (Taken from Figure 6 in [OFG⁺97]) A circle represents a face and a square represents a non-face. On the decision hyper-plane between the two classes some of the support vectors are represented using sub-images.

of the sample-based method described in section 2.4.1. The only difference is the classifier that classifies the sub-images. In the case of SVM, every sub-image is identified as either a face or a non-face sample by a SVM-based classifier in which every support vector is also either a face or a non-face sample. Figure 4.3 is taken from Figure 6 in [OFG⁺97]. In this illustration, a circle represents a face sample and a square represent a non-face sample. Certain support vectors exist on the decision hyper-plane between the two classes. These support vectors are actually the sub-images used during face detection. Note that some of the non-face support vectors are very similar to those associated with face regions. This is the major motivation for the method proposed in Chapter 4. Figure 4.4, also taken from [OFG⁺97], shows the number of vectors versus the number of training samples.

Figure 4.4: (Taken from [OFG⁺97]) Number of support vectors vs. number of samples.

To speed up the computational performance of the SVM-based classifier in face detection, a cascade structure may be used. A manually-designed two node cascade structure was first proposed in the neural network-based detector developed by Rowley et al. [RBK98]. In [RTSB01], a set of reduced set vectors was calculated from the support vectors. Each reduced set vector can be interpreted as either a face or non-face template. Since these reduced set vectors are applied sequentially to the input pattern, they can be viewed as nodes in the cascade. An alternative cascade framework for SVM-based classifiers was proposed by Heisele et al. [HSPP03].

Another research direction that holds promise for accelerated SVM compute times focuses on how to simplify a given SVM. In [Bur96], Burges introduced a method by which a set of so-called reduced set vectors (RSVs) that approximate the decision function can be created. This method has been successfully

applied in the face detection domain - speedups on the order of 10 to 30 have been reported [OFG⁺97] while retaining full accuracy. Recently, Kienzle et al [KBFS04] proposed a method constraining the RSVs to have a special structure in which these RSVs can be evaluated via separable convolutions. The complexity of the RSV evaluation reduced from $O(h \times w)$ to $O(r \times (h + w))$, where $h \times w$ are the height and width of the target image and r is the rank of a $h \times w$ matrix and the matrix itself represents the impulse response, i.e., the filter mask of the input image. r is a small number that allows the user to select a balance between speed and accuracy. In this way, a given SVM containing near 7,000 SVs can achieve around 14 frames per second in a 320×240 image.

4.2 Proposed Algorithm for Active Generation of Samples

Due to the fact that samples located in the margin of the SVM classifier have the largest possibility to be errors, the accuracy of the SVM-based classifier will increase if performance at the margin can be improved. In order to improve accuracy, the only option is to add new samples since the SVM-based classifier is pre-optimized for margin maximization. However, it may be very difficult to collect additional samples that are sufficiently useful. The Bootstrap procedure proposed by Efron [Efr83] in 1983 (see section 3.1.2) is usually applied to collect more useful samples. Nevertheless, the method needs specific types of data. For example, in face detection, it may require more labeled images. Another disadvantage of the Bootstrap technique is that collecting false samples could be very challenging in cases when the classifier accuracy is already quite high and

therefore very few errors occur when processing large sets of unlabeled data.

The proposed active sample generation method makes use of the current samples in the training set to generate more useful training samples. An oracle will label the generated samples. For instance, if the oracle is a human being, the new samples are manually labeled. This new method alleviates the above two problems as it does not need any additional training data. Although it takes time to generate and label the new samples (especially if this requires human experts) offline, it is still worth the effort because the new samples are most likely to further improve the SVM-based classifier. The new method is useful especially when training samples are very difficult to collect. As mentioned before, this approach bears similarities to human learning. Students may ask teachers questions in class to help them improve their learning performance. The Bootstrap method is analogous to teachers asking students new questions to check their learning performance.

Support vectors from a trained SVM-based classifier are first used to create a training sample set from which new training samples can be generated. The primary justification is to reduce computational complexity. Secondly, points along the line connecting two support vectors from the two classes must by definition pass through the margin. The simple but famous “XOR” example is shown in Figure 4.5. There are 4 points labeled A to D in the plane. A, C belong to one class while B, D belong to the other class. Suppose that points B and D are support vectors and the most popular kernel, the Gaussian kernel, is being used to calculate the mapping in the high-dimensional space. Figure 4.5 (a) shows the samples in feature space, and its corresponding mapping in version space (a nonlinear mapping that is usually extremely hard to solve as

Figure 4.5: An example of a non-linear mapping of a straight line between two SVs. (a) Samples in the feature space, (b) mapping samples in the version space.

version space is usually high dimensional) is shown in Figure 4.5 (b). The points are individually discernable because the number of points on the straight line is not very large. The mapping is non-linear and results are in a very different data density.

The statement that “*points along the line connecting two support vectors from the two classes must by definition pass through the margin.*” merits further explanation. According to the theory of SVM, the kernel function K can be described as:

$$K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}) \quad (4.7)$$

where φ is the high-dimensional mapping function. Usually kernel function K is manually selected to be a continuous function. If $\mathbf{x} = \mathbf{y}$, equation (4.7) will become $K(\mathbf{x}, \mathbf{x}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}) = \|\varphi(\mathbf{x})\|^2$. In such a case, $\|\varphi(\mathbf{x})\|$ is continuous

because K is continuous and $\|\varphi(\mathbf{x})\|$ is always not less than 0. $\varphi(\mathbf{x})$ will be continuous if $\|\varphi(\mathbf{x})\|$ is continuous when $\varphi(\mathbf{x})$ is greater than 0 or smaller than 0. According to the above analysis, although it is not guaranteed that $\varphi(\mathbf{x})$ is always continuous, it is still a reasonable assumption that $\varphi(\mathbf{x})$ will likely be continuous for some given kernel function. Therefore, it can be stated that there exist two support vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, that are consistent with the continuous function $\varphi(\mathbf{x})$. If $\varphi(\mathbf{x}_1) \cdot \varphi(\mathbf{x}_2) < 0$, there must be a point ζ between $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $\varphi(\zeta) = 0$. This leads to the assertion that ζ must be in the margin based on the theory of SVM. After that, the two closest support vectors in the two classes are chosen to create a pair and a greedy search algorithm is used to search for points that are on the line segment that links the two SVs. Their mapping points (in version space) will be located in the margin. Equation (4.8) is a modification of equation (4.3) based on the explanation given above. θ can be manually adjusted between 0 to 1.

$$\Lambda = \arg \min_j \left\{ \text{abs} \left[\sum_{\mathbf{x}_i \in \text{SVs}} \bar{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{SV}_{\text{candidate}(j)}) + \bar{b} - \theta \right] \right\} \quad (4.8)$$

λ is the relative location of the new sample ranging from 0 to 1 and Λ is the final location calculated from equation (4.9) by numerical computation.

$$\mathbf{SV}_{\text{candidate}(j)} = \lambda_j \times \mathbf{SV}_{+1} + (1 - \lambda_j) \times \mathbf{SV}_{-1} \quad (4.9)$$

The best way to select the two SVs from the original labeled data is to find a pair of SVs that are the smallest possible distance apart from one another. The distance can be the Mahalanobis distance, the Kernel distance (defined as equation (4.5)) or the Euclidean distance. The approach used to choose the distance

function has been widely debated. The Kernel distance may be considered more appropriate for high dimensional applications such as face detection.

After the location of each pair is calculated, the new samples can be generated using the following equation (4.10):

$$\mathbf{SV}_{new} = \Lambda \times \mathbf{SV}_{+1} + (1 - \Lambda) \times \mathbf{SV}_{-1} \quad (4.10)$$

Equation (4.10) can be justified by the method of linear interpolation. The new generated sample is similar to the result of linearly interpolating between the two samples. Equation (4.10) is not necessarily the only equation capable of generating new samples, but it is used here because of its simplicity.

A greedy search algorithm can be used to search for Λ . In real applications, quantization errors should be taken into consideration. In summary, the proposed new active sample generation algorithm requires the following steps:

1. Train an SVM-based classifier from an existing training data set.
2. Extract SVs from the model and pair the relative SVs based on the smallest distance between them (using the Mahalanobis, Euclidean or Kernel distance).
3. Search for the location specified by equation (4.9).
4. Generate new unlabeled data using equation (4.8).
5. Label the new data manually and discard any data points that have low confidence.

6. Add the new labeled data into the original training data set and train a new SVM-based classifier. Return to step 2.

Algorithm 2: Algorithm for active sample generation

4.3 Analysis of the Proposed Algorithm

According to the theory of SVM analysis in section 4.1.2, the performance of the SVM-based classifier is determined by support vectors that are on the margin of two classes. The solution of equation (4.2) is very likely to be different once the new training samples are added into the margin. This is true because the newly added samples are those whose “ α_i ” values are not zero. Therefore, the SVs in equation (4.3) and (4.4) will also be modified. For the traditional statistics based methods (including the Naïve Bayesian), the proposed method will have little impact on improving their performance. This is because a small number of newly-generated samples may not change the distribution of the training data sufficiently to move it closer to the sample distribution of the real world. However, for the SVM itself, performance improvements are very likely since final performance is based on the accuracy of its boundary. Having more precise samples in the margin will allow boundary correction and improve accuracy, especially in those applications where there are very few support vectors. Experimental results in the next section confirm the performance improvement of the proposed active sample generation method in two-class classification applications.

Figure 4.6: An artificial example: two classes, class A and B.

4.4 Experimental Results

4.4.1 Artificial Example

This simple example uses a data set that was artificially generated using MATLAB. The two classes are defined as class A and class B, as shown in Figure 4.6. Every sample has two dimensions represented as (x,y) and obeys the following constraints.

$$A = \{(x,y) | 0 \leq x \leq 1 \text{ and } 0 \leq y \leq 1\}$$

$$B = \{(x,y) | x > 1 \text{ or } y > 1\}$$

The training sets of A and B are represented as TrainA and TrainB. The 40

Figure 4.7: The average performance of an SVM-based classifier recursively using active sample generation.

samples in each training set, i.e., set A and B, are randomly generated using the below-listed constraints.

$$TrainA = \{(x, y) | x^2 + y^2 \leq 1 \text{ and } x > 0, y > 0\}$$

$$TrainB = \{(x, y) | x > 1 \text{ or } y > 1, \text{ and } x < 1.2, y < 1.2\}$$

Note that the area of TrainA is smaller than the area of class A. Accordingly, there will be some errors between TrainA and Class B. The test sets of A and B are represented as TestA and TestB. To achieve the highest classification performance, 10,000 test samples in each set, i.e., set A and B, were generated separately as TestA and TestB using the constraints listed below.

Figure 4.7 shows the average performance of an SVM-based classifier recursively using active sample generation. The x axis represents the number of samples generated and the y axis represents the average performance. From this figure, it can be seen that after 10 generations, the performance is improved by approximately 4-7%. Figure 4.8 shows the ROC of true and false positive rates. The closer to the upper-left corner, the higher the accuracy. The trend of active sample generation is towards improved overall performance. Performance given Kernel-based distance calculations is not as high as the corresponding performance from the Euclidean calculations. This is mainly because this example is two-dimensional, which makes the Euclidean distance an obvious best choice. However, the Euclidean distance will not offer such impressive performance in the case of higher dimensional scenarios. In the following example, only the Kernel distance is applied to test face detection performance since the input dimension is very high. Ideally, the accuracy should monotonically increase. However, as shown in Figure 4.7 and Figure 4.8, accuracy does increase but not monotonically. This could be because of two reasons. One is that a fixed value of $C=50$, is used, which may not be the best choice. The fixed value is used for consistency. The other may be that, in this case, the new input is very close to the border (which is only a threshold) and accordingly there is no vague margin between the two classes.

Figures 4.9 - 4.12 show the simulation results using the Euclidean distance (Figures 4.9, 4.10) and the Kernel distance (Figures 4.11, 4.12). The points denoted by triangles or stars represent newly generated samples. Dots (TrainA) and crosses (TrainB) are the original training samples. The circled dots are the support vectors of class A and the circled crosses are the support vectors of class B. These figures show clearly how an increasing number of samples are generated

Figure 4.8: The true positive rate versus the false positive rate for each active sample generation.

Figure 4.9: The Euclidean distance, first generation.

in the margin as the algorithm progresses across multiple generations.

4.4.2 Application to Face Detection

As described in Chapters 1 and 2, face detection is a typical application of the two-class classification problem since any given sample will definitively be either a face or non-face. However, not all newly-generated samples should be added into the training data set because the new points may be very close to the border of the real world and thus very hard to label when using the same criterion applied by humans. To avoid labeling errors, only those informative

Figure 4.10: The Euclidean distance, 10th generation.

Figure 4.11: The Kernel distance, first generation.

Figure 4.12: The Kernel distance, 10th generation.

Figure 4.13: (a) non-face sample; (l) face sample; (b)-(k) newly generated middle samples.

samples with high confidence should be selected by the oracle (human). Figure 4.13 shows a simple example. Figure 4.13 (a) is an available non-face sample and (l) is a face sample that has the smallest Euclidean distance among the set of the trained face support vectors to Figure 4.13 (a). (b)-(k) represent middle samples that can be calculated. A human would classify sub-images (a)-(e) and (h)-(l) as samples with high confidence, while in the case of (f)-(g) it is hard to make a clear decision. If the newly generated sample using Λ from equation (4.8) is located in (a)-(e) or (h)-(l), it can be labeled confidently as a face or non-face sample. Otherwise, the newly generated sample must be thrown away because the SVM is a relatively noise-sensitive system. One would rather remove the uncertain sample than add an incorrect sample into the training data.

In the experiment, each pattern of dimensions 20×20 pixels was processed as described in section 3.2.2. Each pattern was reshaped to form a vector of 400 dimensions. The CMU training database was used for training. In order to balance the face and non-face classes, the ECU database was also utilized in these experiments. The author randomly selected 2,000 face samples from the ECU database and combined them with the CMU training database. The remainder of the ECU samples together with the CMU test database were used for testing. The LibSVM [HCL03] program was employed for both training and testing. Table 4.1 shows the simulation results for the CMU and ECU

Table 4.1: Simulation results of the proposed active generation method when using 2,000 ECU face images and CMU training database.

		Original SVM	After Active Generation
CMU Test Data	Negative accuracy	99.53% (23462/23573)	99.46% (23446/23573)
	Positive accuracy	25.21% (119/472)	27.12% (128/472)
ECU Test Data	Negative accuracy	99.96% (12946/12951)	99.67% (12908/12951)
	Positive accuracy	46.51% (3413/7339)	55.09% (4043/7339)

Table 4.2: Simulation results of the proposed active generation method when using ECU 4,000 face and 8,000 non-face images and CMU training database.

		Original SVM	After Active generation
CMU Test Data	Negative accuracy	83.67%(19723/23573)	83.78% (19750/23573)
	Positive accuracy	88.98% (420/472)	88.77% (419/472)
ECU Test Data	Negative accuracy	99.17% (12844/12951)	99.19% (12846/12951)
	Positive accuracy	98.12% (7201/7339)	98.28% (7213/7339)

databases. In this experiment, λ , which increases from 0.1 to 0.9, is shown as an example in Figure 4.13 (b)-(k). The smallest value is regarded as Λ . Note that the improvement is significant when the original SVM-based classifier has low accuracy. It can be seen that, when the accuracy of the SVM-based classifier is low, the active generation method can improve the positive accuracy by as much as 8.58% ($55.08 - 46.50 = 8.58\%$).

Table 4.2 shows a similar result ; in this case, the training data set used included 4000 face and 8000 non-face images. Due to the limits imposed by quantization errors in the face detection application, λ steps from 0.01 to 0.99 in this experiment. The smallest value is regarded as Λ . The accuracy of the SVM model is already high and it is consequently very difficult to improve its performance any further. Although only 54 new samples (all high confidence human-labeled data points) were generated, the proposed active sample generation technique did show a slight improvement compared to using only the original samples. Note that the variation in accuracy before and after active

Figure 4.14: The performance of an SVM-based classifier recursively using active sample generation (trained by the CMU train data set).

sample generation is not significant but the improvement is in both negative and positive accuracy for the testing database. Moreover, the proposed active sample generation method may also be implemented iteratively to further improve accuracy levels. A better performance might be achieved if equation (4.10) were modified to be a non-linear expression.

Figure 4.14 shows the accuracies when recursively calling the new active sample generation method. The training sample was the CMU train data set. The classifier used as the oracle had been trained by both the CMU train data set and the ECU data set. Note that accuracy was improved by as much as

Figure 4.15: The performance of an SVM-based classifier recursively using active sample generation (trained by the ECU data set).

0.6%.

Figure 4.15 shows the performance of an SVM-based classifier trained using the ECU data set when recursively calling the proposed active sample generation method. The classifier used as the oracle had been trained by both the CMU train data set and the ECU data set. Note that accuracy was improved by as much as 0.2% even when the EER was very high.

Figure 4.16 shows some face samples that were generated by the simulation program. Note that some figures are very hard to recognize as a face sample. Because SVM is a noise-sensitive algorithm, not all the new samples are suitable

Figure 4.16: Some generated face samples.

for use in the next training sessions. As previously explained, discarding some unsure samples is a better strategy than adding them into the training set. The justification is that although the newly generated samples are slightly distorted, they could effectively be considered to have arisen from poor conditions with a large amount of noise.

4.5 Conclusions

In this chapter, active sample generation from support vectors was used for two-class classification. Active learning methods such as S^3VM [BD98] had been previously used to choose samples from large unlabeled training samples. Niyogi and Sung [NS98] also proposed a strategy for active sample selection. All these nominally “active” methods selected new samples from existing unlabeled data sets. However, in the case of the proposed new method, new samples are generated from support vectors that were actually trained on the original samples. The experimental results show that the proposed method improves the performance of SVM in two-class classification problems such as face detection.

There remain many questions for further study. Firstly, only linear interpolation is used in this experiment, but other methods such as polynomial interpolation may offer substantially better performance. Secondly, the interpolation points are generated by two-class support vectors. It is unclear whether other samples can also be selected for use. Thirdly, the distance metric used in this chapter is the Euclidean distance. It may not be the best choice in this case. Fourthly, is the oracle indispensable or not? Can S^3VM or Active SVM be used with the newly generated samples? Fifthly, the support vector machine is relatively complex. Is it possible to use Reduced SVM [TLV⁺04] to simplify the SVM-based classifier? Finally, can researchers quantify the confidence of the generated samples? The author looks forward to exploring all these directions in the future.

Chapter 5

PLCB (PCA + LDA + CMFE + Bayes) for Face Detection

In this chapter, a new statistics-based face detection method, named PLCB (PCA + LDA + CMFE + Bayes), is proposed. Here, CMFE refers to Common Mean Feature Extraction. Analysis and empirical results show that PLCB can achieve performance comparable with other state-of-art methods in face detection.

5.1 Statistics-Based Face Detection

In general, statistics-based detection learns the statistical models of the face and non-face images, and then applies a two-class classification rule to discriminate between face and non-face patterns. This section presents the following methods based on statistical models: Bayesian Discriminating [Liu03] and LDA [YKA01] for frontal view face detection. The method proposed in this chapter, PLCB,

is also an example of a statistics-based method and is ideologically linked with the methods described below.

5.1.1 The Bayesian Discriminating Features (BDF) Method

The Bayesian Discriminating Features (BDF) Method for face detection was proposed by [Liu03]. The main idea of the method comes from 1) discriminating feature analysis of the input image, 2) statistical modeling of face and non-face classes, and 3) the application of the Bayesian classifier for multiple frontal face detection.

Firstly, the discriminating feature analysis combines the input image, its 1D Haar wavelet representation (equations (5.1), (5.2)[Liu03]), and its amplitude projections (equations (5.3) and (5.4)[Liu03]). Research [POP98] has shown that the 2D Haar wavelet representation is effective for human face and pedestrian detection. To improve efficiency, the 1D Haar wavelet representation is used to determine discriminating features. The amplitude projections, namely the column and row projections, capture the vertical symmetric distributions and the horizontal characteristics of human facial images. The amplitude projections (equations (5.3) and (5.4)) enhance the discriminating power for face detection.

$$I_h(i, j) = I(i + 1, j) - I(i, j) \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \quad (5.1)$$

$$I_v(i, j) = I(i, j + 1) - I(i, j) \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \quad (5.2)$$

$$\mathbf{x}_r(i) = \sum_{j=1}^n I(i, j) \quad 1 \leq i \leq m \quad (5.3)$$

$$\mathbf{x}_c(j) = \sum_{i=1}^m I(i, j) \quad 1 \leq j \leq n \quad (5.4)$$

Let $\mathbf{x}_h \in \mathbb{R}^{(m-1)n}$ and $\mathbf{x}_v \in \mathbb{R}^{m(n-1)}$ be the vectors formed by concatenating the rows (or columns) of $I_h(i, j)$ and $I_v(i, j)$, respectively. The vectors $\mathbf{x}$, $\mathbf{x}_h$, $\mathbf{x}_v$, $\mathbf{x}_r$, and $\mathbf{x}_c$ are normalized by subtracting the means of their components and dividing by their standard deviations, respectively. Let $\bar{\mathbf{x}}$, $\bar{\mathbf{x}}_h$, $\bar{\mathbf{x}}_v$, $\bar{\mathbf{x}}_r$ and $\bar{\mathbf{x}}_c$ be the normalized vectors. A new feature vector $\bar{\mathbf{y}}$ is defined as the concatenation of the normalized vectors: $\bar{\mathbf{y}} = (\bar{\mathbf{x}}^T, \bar{\mathbf{x}}_h^T, \bar{\mathbf{x}}_v^T, \bar{\mathbf{x}}_r^T, \bar{\mathbf{x}}_c^T)^T$ where T is the transpose operator and $N = 3mn$ is the dimensionality of the feature vector $\bar{\mathbf{y}}$. Finally, the normalized vector of $\bar{\mathbf{y}}$ defines the discriminating feature vector: $\mathbf{y} = (\bar{\mathbf{y}} - \boldsymbol{\mu})/\sigma$ where $\boldsymbol{\mu}$ and σ are the mean and the standard deviation of the components of $\bar{\mathbf{y}}$, respectively.

Secondly, statistical modeling of face and non-face classes essentially estimates the conditional probability density functions (PDFs) of the two classes. The face class is usually modeled as a multivariate normal distribution, while the non-face class is difficult to model due to the fact that it includes the rest of the world. The author derives a subset of the non-face class that lies closest to the face class, and then models this particular subset of non-faces as a multivariate normal distribution as shown in (equations (5.5), (5.6), (5.7) and (5.8)).

$$p(\mathbf{y}|\omega_f) = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}_f|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_f)^t \boldsymbol{\Sigma}_f^{-1} (\mathbf{y} - \boldsymbol{\mu}_f)\right\} \quad (5.5)$$

$$\ln[p(\mathbf{y}|\omega_f)] = -\frac{1}{2}\{(\mathbf{y} - \boldsymbol{\mu}_f)^t \boldsymbol{\Sigma}_f^{-1} (\mathbf{y} - \boldsymbol{\mu}_f) + N \ln(2\pi) + \ln |\boldsymbol{\Sigma}_f|\} \quad (5.6)$$

$$p(\mathbf{y}|\omega_n) = \frac{1}{(2\pi)^{N/2} |\Sigma_n|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_n)^t \Sigma_n^{-1} (\mathbf{y} - \boldsymbol{\mu}_n)\right\} \quad (5.7)$$

$$\ln[p(\mathbf{y}|\omega_n)] = -\frac{1}{2}\{(\mathbf{y} - \boldsymbol{\mu}_n)^t \Sigma_n^{-1} (\mathbf{y} - \boldsymbol{\mu}_n) + N \ln(2\pi) + \ln |\Sigma_n|\} \quad (5.8)$$

PCA (equations (5.9) and (5.10)) is used to reduce the number of dimensions and remove the effects of noise.

$$\mathbf{Z} = \Phi_f^t (\mathbf{y} - \boldsymbol{\mu}_f) \quad (5.9)$$

$$\mathbf{U} = \Phi_n^t (\mathbf{y} - \boldsymbol{\mu}_n) \quad (5.10)$$

After Eigen-decomposition (equations (5.11) and (5.12)), equations (5.6) and (5.8) become equations (5.13) and (5.14).

$$\Sigma_f = \Phi_f \Lambda_f \Phi_f^t \text{ with } \Phi_f \Phi_f^t = \Phi_f^t \Phi_f = \mathbf{I}_N, \Lambda_f = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\} \quad (5.11)$$

$$\Sigma_n = \Phi_n \Lambda_n \Phi_n^t \text{ with } \Phi_n \Phi_n^t = \Phi_n^t \Phi_n = \mathbf{I}_N, \Lambda_n = \text{diag}\{\lambda_1^{(n)}, \lambda_2^{(n)}, \dots, \lambda_N^{(n)}\} \quad (5.12)$$

$$\ln[p(\mathbf{y}|\omega_f)] = -\frac{1}{2}\left\{\sum_{i=1}^M \frac{z_i^2}{\lambda_i} + \ln\left(\prod_{i=1}^M \lambda_i\right) + \sum_{i=M+1}^N \frac{z_i^2}{\lambda_i} + \ln\left(\prod_{i=M+1}^N \lambda_i\right) + N \ln(2\pi)\right\} \quad (5.13)$$

$$\ln[p(\mathbf{y}|\omega_n)] = -\frac{1}{2}\left\{\sum_{i=1}^M \frac{u_i^2}{\lambda_i^{(n)}} + \ln\left(\prod_{i=1}^M \lambda_i^{(n)}\right) + \sum_{i=M+1}^N \frac{u_i^2}{\lambda_i^{(n)}} + \ln\left(\prod_{i=M+1}^N \lambda_i^{(n)}\right) + N \ln(2\pi)\right\} \quad (5.14)$$

A model developed by Moghaddam and Pentland [MP97] is adopted to estimate the remaining $N - M$ eigenvalues using the average of the remaining eigenvalues (equations (5.15) and (5.16)):

$$\rho = \frac{1}{N - M} \sum_{k=M+1}^N \lambda_k \quad (5.15)$$

$$\varepsilon = \frac{1}{N - M} \sum_{k=M+1}^N \lambda_k^{(n)} \quad (5.16)$$

Therefore, the conditional density function of the face and non-face class can be estimated as (equations (5.17) and (5.18)):

$$\begin{aligned} \ln[p(\mathbf{y}|\omega_f)] = & -\frac{1}{2} \left\{ \sum_{i=1}^M \frac{z_i^2}{\lambda_i} + \frac{\|\mathbf{y} - \boldsymbol{\mu}_f\|^2 - \sum_{i=1}^M z_i^2}{\rho} \right. \\ & \left. + \ln\left(\prod_{i=1}^M \lambda_i\right) + (N - M) \ln \rho + N \ln(2\pi) \right\} \end{aligned} \quad (5.17)$$

$$\begin{aligned} \ln[p(\mathbf{y}|\omega_n)] = & -\frac{1}{2} \left\{ \sum_{i=1}^M \frac{u_i^2}{\lambda_i^{(n)}} + \frac{\|\mathbf{y} - \boldsymbol{\mu}_n\|^2 - \sum_{i=1}^M u_i^2}{\varepsilon} \right. \\ & \left. + \ln\left(\prod_{i=1}^M \lambda_i^{(n)}\right) + (N - M) \ln \varepsilon + N \ln(2\pi) \right\} \end{aligned} \quad (5.18)$$

The final result is classified by equation (5.19):

$$\mathbf{y} \in \begin{cases} w_f & \text{if } P(\omega_f|\mathbf{y}) > P(\omega_n|\mathbf{y}) \\ w_n & \text{otherwise} \end{cases} \quad (5.19)$$

The major advantages of this method derive from:

1. The analysis of the discriminating features of the input image, its 1D Haar wavelet representation, and its amplitude projections;
2. Statistical modeling of the face class and the non-face class. Reducing the dimension of the feature vector to a very small number, M , which equaled 10 in the experiments;
3. The application of the Bayesian classifier with a modified decision rule for multiple frontal face detection;
4. The development of the single response criterion and the early exclusion criterion for computational efficiency.

The disadvantages of this method are as follows:

1. Although 1D Haar wavelet representation and image amplitude projections are useful, they do not increase the volume of information in the original images. A lot of redundancy is contained within the input features and this method generates feature dimensions that are three times the original image dimensions.
2. The model used by Moghaddam and Pentland [MP97] to estimate the remaining eigenvalue is not the best choice. This value will significantly impact final accuracy. The remaining eigenvalues may have good capability to classify two classes but they are all ignored in the context of this method. Based on the analysis in section 5.3.2, the proposed new method discussed in section 5.3.1 solves this problem and shows increased performance.

5.1.2 Face Detection Using Multimodal Density Models

The method proposed by [DS01] is considered one of the most accurate in face detection. Kohonen's self-organizing map (SOM) [Koh01] is first applied to divide the face samples into c_1 face sub-classes and the non-face samples into c_2 non-face sub-classes. In the experiments, $c_1 = 25$ and $c_2 = 25$. When making this decision, the author faced the typical "curse of dimensionality" problem [DS01], which states that the amount of data required to construct a reliable estimate of the true solution increases exponentially with dimension. If c_1 and c_2 are too small, the clustering results may be poor. On the other hand, there may not be enough samples in each sub-class to estimate the class-conditional density function well if a large number of classes is chosen. While Fisher's linear discriminant provides effective projections when two classes are unimodal or linearly separable, it may not be effective when classes can be characterized by a multi-modal density function. Hence the training set (using Kohonen's SOM algorithm) is decomposed into subclasses that can be characterized by a well-behaved density function, and subsequently multi-discriminant analysis is applied to the subclasses. After training, the final weight vector for each node is the centroid of the class, i.e., the prototype vector, which corresponds to the prototype of each class. The same procedure is applied to non-face samples. It is clear that face samples with different poses and under different lighting conditions (intensity increases from the lower right corner to the upper left corner) would be categorized into different classes. The SOM algorithm also places the prototypes in the two-dimensional feature map according to their adjacency relationships in the image space. In other words, prototype vectors corresponding to nearby points on the feature map grid also have nearby locations in the

high-dimensional image space (e.g., nearby prototypes have similar intensity and pose). As mentioned at the beginning of this section, the number of face and non-face sub-classes has a direct relation to the number of samples. The face (and non-face) samples are partitioned into several configurations, e.g., 4×4 , 5×6 , and 7×7 . It was confirmed experimentally that the SOM map with 5×5 grids performs best.

After that, the Fisher projection is computed based on all c ($c = c_1 + c_2$) classes to maximize the ratio of the between-class scatter (variance) and the within-class scatter (variance). Consider a c -class problem with N samples; let the between-class scatter matrix be defined as:

$$\mathbf{S}_B = \sum_{i=1}^c N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \quad (5.20)$$

and the within-class scatter matrix be defined as:

$$\mathbf{S}_W = \sum_{i=1}^c \sum_{\mathbf{x}_k \in \omega_i} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T = \sum_{i=1}^c \mathbf{S}_W^{(w_i)} \quad (5.21)$$

where $\boldsymbol{\mu}$ is the mean of all samples, $\boldsymbol{\mu}_i$ is the mean of class ω_i , $\mathbf{S}_{W_i}$ is the covariance of class ω_i , and N_i is the number of samples in class ω_i . The optimal projection $\mathbf{W}_{FLD}$ is chosen as the matrix with orthogonal columns that maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.,

$$\mathbf{W}_{FLD} = \arg \max_{\mathbf{W}} \frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_W \mathbf{W}|} \quad (5.22)$$

where $\mathbf{w}_i \mid i = 1, 2, \dots, m$ is the set of generalized eigenvectors of $\mathbf{S}_B$ and $\mathbf{S}_W$, corresponding to the m largest generalized eigenvalues $\lambda_i \mid i = 1, 2, \dots, m$. However, the rank of $\mathbf{S}_B$ is $c - 1$ or less because it is the sum of c matrices of rank one or less. Thus, the upper bound on m is $c - 1$ [DS01]. Similarly, the rank of $\mathbf{S}_W$ is at most $N - c$. For a set of N sample images of n pixels where N is usually smaller than n , the within-scatter matrix $\mathbf{S}_W \in \mathbb{R}^{n \times n}$ is always singular. This means that the projected within-scatter matrix can be zero if the projection matrix is not chosen properly.

The now-labeled training set is projected from a high-dimensional image space to a lower-dimensional feature space, and a Gaussian distribution is used to model the class-conditional density function for each class where the parameters are estimated using maximum likelihood. For detection, the conditional probability of each sample given by each class is computed, and the maximum likelihood principle is used to decide in which class the sample belongs.

While PCA is commonly used to project face patterns from a high-dimensional image space to a lower-dimensional feature space, one drawback is that it defines such a subspace that it has the greatest variance of the projected sample vectors among all the subspaces. However, such projection may not be effective for classification since large and unwanted variations may be retained. Consequently, the projected samples for each class may not be well clustered and instead, the samples may be smeared together [BHK⁺97][FCH98][HDR97]. Fisher's linear discriminant (FLD) is an example of a class-specific method that finds the optimal projection for classification. Rather than finding a projection that maximizes the projected variance, FLD determines a projection, $\mathbf{z} = \mathbf{W}_{FLD}\mathbf{x}$, that maximizes the ratio between the between-class scatter (variance) and the within-

class scatter (variance). As a result, classification is simplified in the projected space.

It is suggested in [BHK⁺97] [SW96] [ZCK98] that this problem can be avoided by first projecting the image set to a lower-dimensional space using PCA so that the resulting within-class scatter matrix $\mathbf{S}_W$ becomes nonsingular before computing the optimal projection $\mathbf{W}_{FLD}$. In other words, the image set is first projected from N -dimensional space to $(N - c)$ -dimensional space, and then the optimal projection matrix was computed by using equation (5.22). Let $\hat{\mathbf{x}} = \mathbf{W}_{PCA}^T \mathbf{x}$ where $\mathbf{W}_{PCA}$ is $n \times (N - c)$ matrix computed from (equation (5.23)).

$$\mathbf{W}_{PCA} = \arg \max_{\mathbf{W}} |\mathbf{W}^T \mathbf{S}_T \mathbf{W}| \quad (5.23)$$

where $\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$ is the total scatter matrix. Next, $\mathbf{W}_{FLD}$ is computed by using $\hat{\mathbf{x}}$. Consequently, $\mathbf{W}_{FLD}$ is an $(N - c) \times m$ matrix computed by (equation (5.24)).

$$\mathbf{W}_{FLD} = \arg \max_{\mathbf{W}} \frac{|\mathbf{W}^T \mathbf{W}_{PCA}^T \mathbf{S}_B \mathbf{W}_{PCA} \mathbf{W}|}{|\mathbf{W}^T \mathbf{W}_{PCA}^T \mathbf{S}_W \mathbf{W}_{PCA} \mathbf{W}|} \quad (5.24)$$

By using $\mathbf{W}_{FLD}$, $\mathbf{x}$ can be projected from an n -dimensional space to $\tilde{\mathbf{x}}$, $(c - 1)$ -dimensional vector, in a $(c - 1)$ -dimensional space spanned by nonzero eigenvectors. Once $\mathbf{W}_{FLD}$ is computed, the now-labeled training set is projected to the $(c - 1)$ dimensional feature space, i.e., $\tilde{\mathbf{x}} = \mathbf{W}_{FLD}^T \mathbf{x}$, and a Gaussian distribution is used to model each class conditional density (CCD) function, i.e., $P(\tilde{\mathbf{x}}|\omega_i) = N(\boldsymbol{\mu}_{\omega_i}, \boldsymbol{\Sigma}_{\omega_i})$, where $i = 1, \dots, c$. The parameters, $\boldsymbol{\mu}_{\omega_i}, \boldsymbol{\Sigma}_{\omega_i}$ of each CCD are the maximum likelihood estimates, i.e. equations (5.25) and

(5.26).

$$\tilde{\boldsymbol{\mu}}_{\omega_i} = \mathbf{W}_{FLD}^T \boldsymbol{\mu}_i \quad (5.25)$$

$$\tilde{\boldsymbol{\Sigma}}_{\omega_i} = \frac{1}{|\omega_i|} \mathbf{W}_{FLD}^T \mathbf{S}_{\mathbf{W}}^{(\omega_i)} \mathbf{W}_{FLD} \quad (5.26)$$

where $|\omega_i|$ is the number of samples in class ω_i . Each input image is scanned with a rectangular window to determine whether a face exists in the window. The decision rule for deciding whether an input window contains a face is based on maximum likelihood,

$$\omega^* = \arg \max_{i \in 1, \dots, c} P(\mathbf{x} | \omega_i) \quad (5.27)$$

Given $\tilde{\mathbf{x}}$, a face pattern is detected if ω^* is a class label belonging to a face subclass. Otherwise, a non-face pattern is detected. To detect faces at different scales, each input image is repeatedly sub-sampled by a factor of 1.2 and scanned through using 10 iterations.

The major advantages of this method come from 1). The linear projection method is applied, which has the potential to perform better than PCA in classification applications. The experimental results of classification show that this method performs well. 2). Multi-modal density models are applied so that each modality can better capture the variations in face patterns. Other methods do not maximize the classes' separation the same way FLD does.

The major disadvantages of this method come from 1). SOM is first applied to build multi-classes. No sound proof shows why SOM should be used in place

of other clustering methods. 2). It is unclear whether face detection follows a multi-modal density model or just a single Gaussian distribution. Therefore, the number of models can only be determined by experience.

5.2 Common Mean Feature Extraction (CMFE)

The CMFE method [HL98] was designed for the case in which multiple classes have an identical mean but do not share the same covariance matrix. Figure 5.1 shows a straightforward example of the common mean and differential standard deviation. It is very obvious that the two curves whose standard deviations are 0.5 and 0.1 have less common area than the two curves whose standard deviations are 0.25 and 0.35. In this common mean case, LDA fails to work as there is no mean-difference information in the data.

The Bhattacharyya distance measures the separability between two classes. When the face and non-face classes are in Gaussian distribution, the Bhattacharyya distance is given analytically by

$$D = \frac{1}{8}(\boldsymbol{\mu}_f - \boldsymbol{\mu}_n)^T \left(\frac{\boldsymbol{\Sigma}_f + \boldsymbol{\Sigma}_n}{2} \right)^{-1} (\boldsymbol{\mu}_f - \boldsymbol{\mu}_n) + \frac{1}{2} \ln \frac{|(\boldsymbol{\Sigma}_f + \boldsymbol{\Sigma}_n)/2|}{\sqrt{|\boldsymbol{\Sigma}_f| |\boldsymbol{\Sigma}_n|}}, \quad (5.28)$$

$\boldsymbol{\mu}_f$ and $\boldsymbol{\mu}_n$ are the mean vectors of the face and non-face training samples, respectively. $\boldsymbol{\Sigma}_f$ and $\boldsymbol{\Sigma}_n$ are the covariance matrices of the face and non-face training samples, respectively.

The first term of the Bhattacharyya distance is maximized in a subspace

obtained by LDA, i.e., solving the generalized eigenvalue decomposition problem

$$\mathbf{S}_B \Phi = (\Sigma_f + \Sigma_n) \Phi \Lambda \quad (5.29)$$

where Φ is the matrix from eigen-decomposition applied on $\mathbf{S}_B$ and $(\Sigma_f + \Sigma_n)$. $\mathbf{S}_B$ is the between-class scatter matrix defined in equation (5.20).

As face detection is a two-class problem, the rank of $\mathbf{S}_B$ is one at most. Thus, LDA (5.29) produces only a single feature, which is far from sufficient for a face detection task.

It is proven in [Fuk90] that the second term of the Bhattacharyya distance can be maximized in a subspace obtained by solving the generalized eigenvalue decomposition problem of covariance matrix pair of either (Σ_f, Σ_n) or (Σ_n, Σ_f) .

$$\Sigma_f \Phi = \Sigma_n \Phi \Lambda \quad (5.30)$$

The subspace that maximizes the second term of the Bhattacharyya distance is spanned by the generalized eigenvectors corresponding to the largest $\lambda_k + 1/\lambda_k$ [Fuk90], where λ_k is the generalized eigenvalue of matrix pair (Σ_f, Σ_n) or (Σ_n, Σ_f) . Obviously, λ_k is the ratio between the two class-conditional variances projected on the eigenvector, v_k^f/v_k^n or v_k^n/v_k^f . Thus the largest $\lambda_k + 1/\lambda_k$ maximizes the sum of the two ratios, $v_k^f/v_k^n + v_k^n/v_k^f$. As the example shown in Figure 5.1, it is very obvious that the two curves whose standard deviations representing the class-conditional variances v_k^f and v_k^n , are 0.5 and 0.1 have less common area than the two curves whose v_k^f and v_k^n are 0.25 and 0.35.

Figure 5.1: Common mean different standard deviation.

5.3 The Proposed PLCB Method for Face Detection

5.3.1 PLCB

The proposed method, named PLCB, is unique in that it uses PCA, LDA, CMFE and Bayesian techniques. As mentioned before, PCA [PT91], LDA [YKA01] and the Naïve Bayesian classifier [Liu03] have already been applied in face detection while CMFE has not yet been reported for face detection applications.

In the high dimensional image space, the available training samples are often not sufficient and/or not representative for a robust statistical estimation in all dimensions. Thus, image space is decomposed into a principal and its complementary subspaces based on the eigenvector decomposition for face recognition and detection tasks [MP97]. Mahalanobis distance is applied in the principal subspace as it is deemed to be a reliable subspace. In the complementary subspace, simple Euclidean distance is used because the variance of individual dimension in this subspace cannot be robustly estimated by the training samples. Similar idea of using Mahalanobis distance in the principal subspace and Euclidean distance in the complementary subspace is employed in the face detection framework proposed in [SP98]. An extended space containing image and its Harr wavelet features is also decomposed into a principal and its complementary subspaces to apply the Mahalanobis and Euclidean distances, respectively [Liu03]. These three approaches of face detection circumvent the over-fitting problem in the complementary subspace but there are still problems in merging these two distances. As eigenvalues in the complementary subspace are not well estimated, their average may not be a proper scaling factor of the distance [JMK06]. This could be a reason why these approaches often work with a much smaller principal subspace than the complementary subspace.

Instead of smoothing eigenvalues in the complementary subspace into a constant in [MP97, SP98, Liu03], PCA removes this unreliable subspace and is widely applied in face recognition. However, most approaches apply PCA only aimed at solving the singularity problem of the within-class scatter matrix for the subsequent LDA. For face detection, it is not difficult to collect a much larger number of samples than the image dimensionality. Hence, PCA is seldom applied in face detection. In fact, the role of PCA can be far beyond

solving the singularity problem of the scatter matrix if it is properly applied. A good example of applying PCA to improve the face identification accuracy can be found in [WT04]. Therefore, in the training stage, PCA is first used and the role of PCA is to remove the unreliable dimensions and hence to alleviate the over-fitting problem. PCA applies eigen-decomposition on the total scatter matrix $\mathbf{S}_T$, i.e., $\mathbf{S}_T = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T$, and keeps the m eigenvectors $\hat{\mathbf{\Phi}}$, $\hat{\mathbf{\Phi}} \in \mathbb{R}^{n \times m}$, corresponding to the m largest eigenvalues (n is the input dimension).

After removing unreliable dimensions, the data dimensionality is reduced. As the objective of this dimension reduction is to alleviate over-fitting problem, the resulting feature vector is in general not a compact feature set for an efficient (fast) classification. When using the general pyramid framework of sample-based methods as described in section 2.4.1, the detection could be very time consuming if it took a long time to classify a sub-image into face or non-face pattern. To achieve efficient (fast) face detection, a compact feature representation is needed.

The proposed PLCB discriminant analysis applies the generalized eigenvalue decomposition (5.29) and (5.30) in the reliable PCA subspace (After removing unreliable dimensions, the data dimensionality is reduced from n to m , $m < n$.) and extracts d ($d < m$) generalized eigenvectors $\tilde{\mathbf{\Phi}}$, where one eigenvector from LDA corresponding to the largest eigenvalue of (5.29) and $d-1$ eigenvector from (5.30) corresponding to the $d-1$ largest $\lambda_k + 1/\lambda_k$, where λ_k is the generalized eigenvalue of (5.30). Its purpose is to extract d discriminative features from the reliable PCA subspace.

Given an n -dimensional properly normalized image vector $\mathbf{x}$ (column vector),

a d -dimensional reliable and discriminative feature vector $\tilde{\mathbf{x}}$ is extracted by

$$\tilde{\mathbf{x}} = \mathbf{W}^T \mathbf{x} \quad (5.31)$$

where

$$\mathbf{W} = \hat{\Phi} \tilde{\Phi} \quad (5.32)$$

is the feature extraction matrix of size $n \times d$, $\hat{\Phi}$ is the PCA eigenvector matrix of size $n \times m$ and $\tilde{\Phi}$ is the proposed generalized eigenvector matrix of size $m \times d$.

In the last step, the minimum Mahalanobis distance classifier is applied and it is equal to the Naïve Bayesian classifier when the distribution of the features are Gaussian. A face pattern is detected in the d -dimensional feature space if

$$(\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}_n)^T \tilde{\boldsymbol{\Sigma}}_n^{-1} (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}_n) - (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}_f)^T \tilde{\boldsymbol{\Sigma}}_f^{-1} (\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}_f) > b \quad (5.33)$$

where b is a constant threshold determined by the compromise between positive and negative error rates in application and

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_f &= \mathbf{W}^T \boldsymbol{\mu}_f, & \tilde{\boldsymbol{\Sigma}}_f &= \mathbf{W}^T \boldsymbol{\Sigma}_f \mathbf{W} \\ \tilde{\boldsymbol{\mu}}_n &= \mathbf{W}^T \boldsymbol{\mu}_n, & \tilde{\boldsymbol{\Sigma}}_n &= \mathbf{W}^T \boldsymbol{\Sigma}_n \mathbf{W} \end{aligned} \quad (5.34)$$

To speed up the online face detection process, two eigenvalue decomposition processes are performed in the training stage:

$$\tilde{\boldsymbol{\Sigma}}_f = \Phi_f \Lambda_f \Phi_f^T \quad (5.35)$$

$$\tilde{\boldsymbol{\Sigma}}_n = \Phi_n \Lambda_n \Phi_n^T \quad (5.36)$$

Two feature normalization matrices are then obtained in the training phase as

$$\mathbf{U}_n = \Phi_n \Lambda_n^{-\frac{1}{2}} \quad (5.37)$$

$$\mathbf{U}_f = \Phi_f \Lambda_f^{-\frac{1}{2}} \quad (5.38)$$

which normalize the two class mean to

$$\check{\boldsymbol{\mu}}_n = \mathbf{U}_n^T \tilde{\boldsymbol{\mu}}_n, \quad \check{\boldsymbol{\mu}}_f = \mathbf{U}_f^T \tilde{\boldsymbol{\mu}}_f \quad (5.39)$$

In the online detection stage, the decision rule (5.33) is then simplified as

$$(\check{\mathbf{x}}_n - \check{\boldsymbol{\mu}}_n)^T (\check{\mathbf{x}}_n - \check{\boldsymbol{\mu}}_n) - (\check{\mathbf{x}}_f - \check{\boldsymbol{\mu}}_f)^T (\check{\mathbf{x}}_f - \check{\boldsymbol{\mu}}_f) > b \quad (5.40)$$

where

$$\check{\mathbf{x}}_n = \mathbf{U}_n^T \tilde{\mathbf{x}} \quad \text{and} \quad \check{\mathbf{x}}_f = \mathbf{U}_f^T \tilde{\mathbf{x}} \quad (5.41)$$

One major difference between the proposed method and the BDF method in section 5.1.1 is that the previous method used Haar features and expanded the sample dimension from 400 to 1200 while the proposed new method extracts d features from the original samples, where d is usually smaller than 100.

The difference in how LDA is applied to the method in section 5.1.2 and the proposed method is in that the proposed method uses only two classes. The method used in section 5.1.2 uses SOM to generate multiple classes for LDA. The rank of the between-class scatter matrix is up to one and thereby only one eigenvalue will be non-zero.

5.3.2 Analysis

Besides face detection, PLCB may also be applied to other two-class classification applications. The major advantages of PLCB come from:

Firstly, the dimension of samples is reduced from the original 400 to $d < 100$, which accelerates the subsequent processing.

Secondly, the Bayesian classifier ensures that the classification is the best when the distribution of the features are Gaussian. Although it may not be the best in theory when the distribution of features are non-Gaussian, the minimum Mahalanobis distance classifier usually still work well.

Thirdly, the reliable and discriminative features can be extracted by the generalized eigenvalue decomposition even when the face and non-face classes have an identical mean.

Fourthly, the linear projection method is applied such that PLCB has the potential to perform better than PCA in a given classification. The experimental results show that PLCB performs well. Compared with the method described in section 5.1.2, PLCB does not need SOM to build multi-classes.

Thanks to the above advantages, PLCB has shown a good classification capability in two-class classification applications.

5.4 Performance

SVM[Bur98][LGSL04], LDA[YKA01], Bayes[Liu03] and MLP (Multi Layer Perception) [RBK98] are chosen against which to evaluate the performance of PLCB. The databases described in section 2.2.3 are used here. The dimen-

sion reduction of PCA was empirically set to 100 and the dimension of feature extraction by generalized eigen-decomposition was empirically set to 50.

In the SVM training stage, capacity parameter C , which defines the tradeoff between complexity and empirical risk, was empirically set to 50, the same as experiments mentioned in other chapters. As described in [YKA01], both face and non-face were classified into 25 sub-classes by SOM and PCA was used to reduce dimension from 400 to 80(following the work [YKA01]). In the implementation of the Bayesian classifier, reducing the dimensionality of the feature vector to 10 made it the same as the value in [Liu03]. The network structure of MLP comprises three layers that include 10 units in the input layer, 25 hidden units in the hidden layer and one output unit in the output layer. Each unit in the input layer connects to one of the Principal Components. There are bias units connected to each of the hidden units and to the output unit. The network is fully connected. There is a connection between each input unit and each hidden unit, as well as between each hidden unit and the output unit. This leads to $10 \times 25 = 250$ input-to-hidden connections and 25 hidden-to-output connections. The activation function is sigmoid.

Figure 5.2 shows the ROC curve trained by the CMU train data set and tested by CMU test data set. PLCB shows the best performance among all the five methods.

Figure 5.3 shows the ROC curve trained by ECU data set and tested by CMU train data set. The performance of PLCB is much better than that of the other methods.

As MLP did not show a good performance in the previous experiments, the other three methods were compared with PLCB in a five-fold cross-validation

Figure 5.2: ROC of the performance trained by CMU train data set and tested by CMU test data set.

Figure 5.3: ROC of the performance trained by ECU data set and tested by CMU train data set.

Figure 5.4: ROC 1 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.

experiment. Figures 5.4, 5.5, 5.6, 5.7 and 5.8 show the experimental performance of the five-fold cross-validation as tested using the ECU database with SVM, LDA, Bayes and PLCB algorithms. 80% of the samples in the ECU database are used for training and the remaining 20% of the samples are used for testing. PLCB shows the best performance and SVM is the second.

In general, the performance of PLCB is better than SVM, Bayes, LDA and MLP based on the results of the above experiments.

Figure 5.5: ROC 2 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.

Figure 5.6: ROC 3 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.

Figure 5.7: ROC 4 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.

Figure 5.8: ROC 5 of the performance trained using 80% of ECU data set and tested using the remaining 20% of the ECU data set.

5.5 Conclusions

A new two-class classification method, termed PLCB (PCA + LDA + CMFE + Bayes), is proposed in this chapter. CMFE is first applied on face detection application because it provides new feature extraction criteria. With the help of CMFE, the most discriminant feature between face class and non-face class is likely to be extracted in the common mean case where LDA fails to work as there is no mean-difference information in data. Consequently, empirical results show that the new method improved the accuracy of face detection.

Chapter 6

Conclusions and Recommendations for Future Work

6.1 Conclusions and Major Contributions

In this thesis, sample-based face detection is studied with respect to different aspects of the two-class classification problem. Firstly, in the sample preparation section, new passive sample selection and active sample generation algorithms are proposed to assist sample-based algorithms to better solve the problem of face detection. Secondly, in the classification aspect, a new classification method named PLCB is proposed for face detection. Its performance is evaluated in this context. The main contributions of the author's work are as follows:

1. A clustering sample selection method is proposed for face detection. Analyses and empirical results show that it can improve the accuracy of sample-based

face detection methods (SVM in this thesis). A cascaded clustering structure is also proposed to solve the problem of huge training data sets. Various features, including intensity, PCA, ICA1 and ICA2, were used and compared.

2. Active sample generation from support vectors was used for two-class classification. In contrast with other active learning methods such as S^3VM , our proposed method generated new samples from the support vectors that had been trained using the original samples. The experimental results show that the proposed method improves the performance of SVM in two-class classification problems such as face detection.

3. A new classification method, PLCB, is proposed and its performance is compared with other well-known sample-based algorithms. Analysis and empirical results show that its accuracy is higher than other classification algorithms commonly used for face detection.

6.2 Suggestions for Future Work

Based on the research presented in this thesis, some recommendations for future work are as follows:

Firstly, how to create the best distributed training data by efficiently selecting features in the spatial or frequency domains is still an open topic in the field of face detection. Many issues remain unresolved. The samples in the training set determine the performance of sample-based face detection methods. Efficient selection of training samples will undoubtedly increase final performance. The k-means algorithm was adopted because of its simplicity. More researches on choosing the clustering methods such as k-NN or other sampling methods

could achieve better performance. These will comprise the main areas of the author's future research.

Secondly, there are still many questions for further study in respect of active sample generation. (1) The linear interpolation method was used for this experiment. Might other methods such as polynomials offer better performance? (2) The interpolation points were generated by two-class support vectors. It is not obvious whether or not other samples could also be selected. (3) The distance measurement used in this thesis was Euclidean. It may not be the best choice in many applications. (4) The oracle may or may not be indispensable. Can S^3VM or Active SVM be applied to the newly generated samples? (5) The support vector machine is relatively complex. Is it possible to use a Reduced SVM [TLV⁺04] to simplify the SVM-based classifier? (6) Can researchers quantify the confidence of generated samples?

Appendix

A. The derivation of equation 2.1

Suppose that the size of the input image is $x \times y$, the size of the normalized image used for classification is $x_0 \times y_0$, and the scaling factor is r . Let S_x and S_y be the step values in x- and y-directions respectively.

The scaling time N can be calculated by:

$$N = \log_r \left[\min \left(\frac{x_0}{S_x}, \frac{y_0}{S_y} \right) \right]$$

When the scaling factor becomes $(\frac{1}{r^2})^n$, the number of the sub-images generated by the scaled image is:

$$T_n = \frac{x_n y_n}{S_x S_y} \leq \frac{x_0 y_0}{S_x S_y} \left(\frac{1}{r^2} \right)^n$$

It can be shown that T , the total number of the sub-images generated by the input image, is:

$$\begin{aligned}
T &= \sum_{n=1}^N T_n \\
T &\leq \sum_{n=1}^N \left[\frac{x_0 y_0}{S_x S_y} \left(\frac{1}{r^2} \right)^n \right] \\
T &\leq \frac{x_0 y_0}{S_x S_y} \sum_{n=1}^N \left(\frac{1}{r^2} \right)^n \\
T &\leq \frac{x_0 y_0}{S_x S_y} \frac{r^2 - \left(\frac{1}{r^2} \right)^N}{r^2 - 1} \\
T &\leq \frac{x_0 y_0}{S_x S_y} \frac{r^2 - \left(\frac{1}{r^2} \right)^{\log_r \left(\min \left(\frac{x_0}{S_x}, \frac{y_0}{S_y} \right) \right)}}{r^2 - 1} \\
T &\leq \frac{x_0 y_0}{S_x S_y} \frac{r^2 - \min \left(\left(\frac{x_0}{S_x} \right)^2, \left(\frac{y_0}{S_y} \right)^2 \right)}{r^2 - 1}
\end{aligned}$$

$$p(w_f|y) - p(w_n|y) > b$$

$$p(w_f)p(y|w_f) - p(w_n)p(y|w_n) > b$$

$$(y - \mu_f)^t \Sigma_f^{-1} (y - \mu_f) - (y - \mu_n)^t \Sigma_n^{-1} (y - \mu_n) > b$$

Author's Publications

Journals

- [J1] Wei Yu, Xudong Jiang, Bing XIONG, "Data preparation for sample-based face detection", *Int. J. Computer Applications in Technolog*, Accepted.
- [J2] Bing XIONG, Ce ZHU, Charayaphan CHAROANSACK and Wei YU, "An FPGA Prototyping of Gabor-wavelets Transform for Motion Detection," *International Journal of Computer Sciences and Engineering Systems*, Vol. 1 No. 1, pp65-70.

Conferences

- [C1] Yu Wei and Charoensak Charayaphan, "Non-Iterative ICA for Detecting Motion in Image Sequences", in *International Conference on Neural Information Processing 2002, Simulated Evolution and Learning 2002 & Fuzzy Systems and Knowledge Discovery 2002*, 18-11-2002 to 22-11-2002.
- [C2] Yu Wei and Charoensak Charayaphan, "FPGA Implementation of Non-Iterative ICA for Detecting Motion in Image Sequences", 7th International Conference on Control, Automation, Robotics and Vision (ICARCV 2002), 02-12-2002 to 05-12-2002.
- [C3] Xiong Bing, Yu Wei and Charayaphan Charoensak, "Face contour tracking in video", International conference on image processing(ICIP2004), 1021-1024.

- [C4] Xiong Bing, Yu Wei and Charayaphan Charoensak, "Face contour tracking in video and its applications", IASTED International Conference on Computer Graphics and Imaging, accepted.
- [C5] Xiong Bing, Yu Wei and Charayaphan Charoensak, "Gradient orientation based face detection", ICARCV 2004, accepted.
- [C6] Yu Wei, Xiong Bing and Charoensak Charayaphan, "FPGA implementation of AdaBoost algorithm for detection of face biometrics", IEEE International Workshop on Biomedical Circuits and Systems, 2004.
- [C7] Xiong Bing, Yu Wei and Charoensak Charayaphan, "Face contour extraction using SNAKE", IEEE International Workshop on Biomedical Circuits and Systems, 2004.
- [C8] Yu Wei, Xiong Bing and Charoensak Charayaphan, "Multi-weight diameter bisection method for ellipse detection", IEEE International Workshop on Biomedical Circuits and Systems, 2004.
- [C9] Charayaphan Chareonsak, Farook Satta, Yu Wei, and Xiong Bing, "Design of FPGA hardware for a real-time blind source separation of fetal ECG signals", IEEE International Workshop on Biomedical Circuits and Systems, 2004.
- [C10] Xiong Bing, Yu Wei and Charoensak Charayaphan, "Automatic focusing technique for face detection and face contour tracking", IEEE International Workshop on Biomedical Circuits and Systems, 2004.

Other

- [O1] Yu Wei and Charoensak Charayaphan, "Non-Iterative ICA for Detecting Motion in Image Sequences" chapter 16 of Book "Computational Intelligence for Modeling and Predictions" published by Springer.

Bibliography

- [AS93] MF Augusteijn and TL Skufca. Identification of human faces through texture-based feature recognition and neural network technology. *IEEE International Conference on Neural Networks*, pages 392–398, 1993.
- [B⁺98] G.R. Bradski et al. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(2):12–21, 1998.
- [BCR97] JM Benitez, JL Castro, and I. Requena. Are artificial neural networks black boxes? *Neural Networks, IEEE Transactions on*, 8(5):1156–1164, 1997.
- [BD98] K. Bennett and A. Demiriz. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, 11:368–374, 1998.
- [BEYL04] Y. Baram, R. El-Yaniv, and K. Luz. Online Choice of Active Learning Algorithms. *The Journal of Machine Learning Research*, 5:255–291, 2004.
- [BGV92] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [BHK⁺97] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, et al. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.
- [BJ98] M.J. Black and A.D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.
- [BL97] A. Blum and P. Langley. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
- [BMS02] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *Neural Networks, IEEE Transactions on*, 13(6):1450–1464, 2002.
- [BS97] M.S. Bartlett and T.J. Sejnowski. Independent components of face images: A representation for face recognition. *4th Joint Symposium on Neural Computation Proceedings*, 1997.
- [Bur96] C.J.C. Burges. Simplified support vector decision rules. *13th International Conference on Machine Learning*, pages 71–77, 1996.

- [Bur98] C.J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [CCG04] J. Chen, X. Chen, and W. Gao. Expand training set for face detection by GA re-sampling. *Automatic Face and Gesture Recognition, Proceedings. Sixth IEEE International Conference on*, pages 73–78, 2004.
- [CCTA94] D.A. Cohn, J.D. Cowan, G. Tesauro, and J. Alspector. Neural Network Exploration Using Optimal Experiment Design. *Advances in Neural Information Processing Systems*, 6:679–686, 1994.
- [CGJ96] D.A Cohn, Z. Ghahramani, and MI Jordan. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.
- [CH96] A.J. Colmenarez and T.S. Huang. Maximum Likelihood Face Detection. *Automatic Face and Gesture Recognition, Proceedings, Second IEEE International Conference on*, pages 222–224, 1996.
- [CL90] D.A Cohn and Richard L. Training connectionist networks with queries and selective sampling. *Advances in Neural Information Processing Systems*, 2, 1990.
- [CL01] C.C. Chang and C.J. Lin. Training ν -Support Vector Classifiers: Theory and Algorithms. *Neural Computation*, 13(9):2119–2147, 2001.
- [DS01] Hart P.E. Duda, R.O. and D.G. Stork. *Pattern Classification*. Wiley/Interscience, New York, 2001.
- [DSK03] J.X. Dong, C.Y. Suen, and A. Krzyzak. A Fast SVM Training Algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(3):367–384, 2003.
- [Efr83] B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.
- [EHOK02] M. Elad, Y. Hel-Or, and R. Keshet. Rejection based classifier for face detection. *Pattern Recognition Letters*, 23(12):1459–1471, 2002.
- [EPT⁺98] M. Escher, I. Pandzic, N.M. Thalmann, D. Thalmann, and F. Bossen. Face Synthesis in the VIDAS project. *Comtec*, 78(1):14–19, 1998.
- [FCH98] BJ Frey, A. Colmenarez, and TS Huang. Mixtures of local linear subspaces for face recognition. *Computer Vision and Pattern Recognition, Proceedings. IEEE Computer Society Conference on*, pages 32–37, 1998.
- [Fed72] V.V. Fedorov. *Theory of optimal experiments*. Academic Press New York, 1972.
- [FS97] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. second ed. Academic Press, 1990.
- [Fuk96] K. Fukumizu. Active learning in multilayer perceptrons. *Advances in Neural Information Processing Systems*, 8:295–301, 1996.

- [GBFK98] J.S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. *Intelligent Robots and Systems, Proceedings, IEEE/RSJ International Conference on*, 2, 1998.
- [GLZ01] L. Gu, S.Z. Li, and H.J. Zhang. Learning probabilistic distribution model for multiview face detection. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
- [GT93] G. Yang and T.S. Huang. Finding Human Faces in a Picture? *Proceedings of the 5th International Conference on Computer Analysis of Images and Patterns*, pages 766–769, 1993.
- [GT94] G. Yang and T.S. Huang. Human face detection in a complex background. *Pattern Recognition*, 27(1):53–63, 1994.
- [Hay94] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1994.
- [HCL03] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. *National Taiwan University, Tech. Rep.*, July, 2003.
- [HDR97] GE Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. *Neural Networks, IEEE Transactions on*, 8(1):65–74, 1997.
- [HHI01] A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic Independent Component Analysis. *Neural Computation*, 13(7):1527–1558, 2001.
- [HL98] P.F. Hsieh and D. Landgrebe. Linear feature extraction for multiclass problems. *Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS'98. 1998 IEEE International*, 4, 1998.
- [HOMR91] J.N.C. Hwang, JJ Oh, S. Marks, and II R.J. Query-based learning applied to partially trained multilayer perceptrons. *Neural Networks, IEEE Transactions on*, 2(1):131–136, 1991.
- [HPP00] B. Heisele, T. Poggio, and M. Pontil. *Face Detection in Still Gray Images*. Number 1687. Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.
- [HPV99] N. Herodotou, KN Plataniotis, and AN Venetsanopoulos. Automatic location and tracking of the facial region in color video sequences. *Signal Processing: Image Communication*, 14(5):359–388, 1999.
- [HS01] S. Hua and Z. Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001.
- [HSPP03] B. Heisele, T. Serre, S. Prentice, and T. Poggio. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition*, 36(9):2007–2017, 2003.
- [Hyv99] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2(94-128):3–1, 1999.
- [Int] Intel. Open Source Computer Vision Library (OpenCV). <http://opencvlibrary.sourceforge.net>.

- [JLH⁺98] S.H. Jeng, H.Y.M. Liao, C.C. Han, M.Y. Chern, and Y.T. Liu. Facial feature detection using geometrical face model: an efficient approach. *Pattern Recognition*, 31(3):273–282, 1998.
- [JM96] P. Juell and R. Marsh. A Hierarchical Neural-Network for Human Face Detection. *Pattern Recognition*, 29(5):781–787, May 1996.
- [JMK06] X. D. Jiang, B. Mandal, and A. Kot. Enhanced maximum likelihood face recognition. *Electronics Letters*, 42(19):1089–1090, September 2006.
- [Joa99] T. Joachims. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods-Support Vector Learning*, pages 169–184, 1999.
- [JW01] J.Zhang and W.Lin. Real time image sequence segmentation using curve evolution? *Proceeding of SPIE*, 4303, 2001.
- [KBFS04] W. Kienzle, G. Bakır, M. Franz, and B. Schölkopf. Face Detection—Efficient and Rank Deficient. *Neural Information Processing System*, 2004.
- [Koh01] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.
- [KPC⁺99] A. Kuchinsky, C. Pering, M.L. Creech, D. Freeze, B. Serra, and J. Gwizdka. FotoFile: a consumer multimedia organization and retrieval system. *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 496–503, 1999.
- [KR90] W. Kinzel and P. Ruján. Improving a network generalization ability by selecting examples. *Europhysics Letters*, 13:473, 1990.
- [Krü01] V. Krüger. *Gabor Wavelet Networks for Object Representation*. Springer, 2001.
- [KS93] Y. Kabashima and S. Shinomoto. Incremental learning with and without queries in binary choice problems. *Neural Networks, Proceedings of International Joint Conference on (IJCNN’93-Nagoya)*, 2, 1993.
- [LGL00] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. *Automatic Face and Gesture Recognition, Proceedings, Fourth IEEE International Conference on*, pages 300–305, 2000.
- [LGSL04] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5):413–427, 2004.
- [LHCT95] A. Lanitis, A. Hill, TF Cootes, and CJ Taylor. Locating facial features using genetic algorithms. *International Conference on Digital Signal Processing*, pages 520–525, 1995.
- [Liu03] C. Liu. A Bayesian Discriminating Features Method for Face Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(6):725–740, 2003.
- [LKL97] S.H. Lin, S.Y. Kung, and L.J. Lin. Face recognition/detection by probabilistic decision-based neuralnetwork. *Neural Networks, IEEE Transactions on*, 8(1):114–132, 1997.
- [LKP96] CH Lee, JS Kim, and KH Park. Automatic human face location in a complex background. *Pattern Recognition*, 29:1877–1889, 1996.

- [LKP03] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *DAGM 25th Pattern Recognition Symposium*, 2003.
- [LZSZ02] S.Z. Li, Z.Q. Zhang, H. Shum, and H.J. Zhang. FloatBoost learning for classification. *Neural Information Processing System*, 15, 2002.
- [Mac92] D.J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.
- [MBN02] LC Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. *Data Mining, Proceedings. IEEE International Conference on (ICDM)*, pages 306–313, 2002.
- [Mit82] T. Mitchell. Generalization as search. *Artificial Intelligence*, 28:203–226, 1982.
- [MMP04] P. Mitra, CA Murthy, and SK Pal. A probabilistic active support vector learning algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):413–418, 2004.
- [MP97] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):696–710, 1997.
- [MV02] F. Marques and V. Vilaplana. Face segmentation and tracking based on connected operators and partition projection. *Pattern Recognition*, 35(3):601–614, 2002.
- [NHI98] AV Nefian and MH Hayes III. Face detection and recognition using hidden Markov models. *International Conference on Image Processing (ICIP)*, 1, 1998.
- [NS98] P. Niyogi and K.K. Sung. Epsilon focusing strategy for active example selection. *Knowledge-Based Systems*, 10(7):441–447, 1998.
- [OFG⁺97] E. Osuna, R. Freund, F. Girosi, et al. Training support vector machines: an application to face detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 24, 1997.
- [Par04] J.M. Park. Convergence and application of online active sampling using orthogonal pillar vectors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1197–1207, 2004.
- [Phu03] S.L. Phung. *Automatic human face detection in color images*. PhD thesis, Edith Cowan University, Perth, Australia, 2003.
- [POP98] CP Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. *Sixth International Conference on Computer Vision*, pages 555–562, 1998.
- [PS92] M. Propp and A. Samal. Artificial neural network architecture for human face detection. *Intell. Eng. Syst. Artificial Neural Networks*, 1:535–540, 1992.
- [PT91] A. Pentland and M. Turk. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

- [PW91] M. Plutowski and H. White. Active selection of training examples for network learning in noiseless environments. *Dept. Computer Science, UCSD, TR*, pages 90–011, 1991.
- [RBK97] H.A. Rowley, S. Baluja, and T. Kanade. *Rotation Invariant Neural Network-based Face Detection*. Tech Report CMU-CS-97-201, Carnegie Mellon University, School of Computer Science, 1997.
- [RBK98] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):23–38, 1998.
- [RE01] S. Roberts and R. Everson. *Independent Component Analysis: Principles and Practice*. Cambridge University Press, 2001.
- [RTSB01] S. Romdhani, P. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. *Computer Vision, Proceedings, Eighth IEEE International Conference on (ICCV)*, pages 695–700, 2001.
- [RWY95] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119–130, 1995.
- [RY98] D. Reisfeld and Y. Yeshurun. Preprocessing of Face Images: Detection of Features and Pose Normalization. *Computer Vision and Image Understanding*, 71(3):413–430, 1998.
- [SB02] H. Sahbi and N. Boujemaa. Coarse-to-Fine Support Vector Classifiers for Face Detection. *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3-Volume 3*, 2002.
- [SC00] G. Schohn and D.A Cohn. Less is more: Active learning with support vector machines. *Proceedings of the Seventeenth International Conference on Machine Learning*, 282:285–286, 2000.
- [SE92] BE Stuckman and EE Easom. A comparison of Bayesian/sampling global optimization techniques. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(5):1024–1032, 1992.
- [Sir01] S. Sirohey. Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition*, 34(7):1367–1391, 2001.
- [SK87] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.
- [SK00] H. Schneiderman and T. Kanade. A Statistical Model for 3D Object Detection Applied to Faces and Cars. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:746–751, 2000.
- [Sol96] P. Sollich. Learning from minimum entropy queries in a large committee machine. *Physical Review E*, 53(3):2060–2063, 1996.
- [SP98] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):39–51, 1998.

- [Sun96] K.K. Sung. *Learning and Example Selection for Object and Pattern Recognition*. PhD thesis, Cambridge, MA, 1996.
- [SW96] D.L. Swets and J.J. Weng. Using discriminant eigenfeatures for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):831–836, 1996.
- [TK02] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.
- [TLV⁺04] T. Theoharides, G. Link, N. Vijaykrishnan, MJ Irwin, and W. Wolf. Embedded hardware face detection. *VLSI Design, Proceedings. 17th International Conference on*, pages 133–138, 2004.
- [VJ01] P. Viola and M. Jones. Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. *Neural Information Processing Systems*, 2001.
- [VJ02] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.
- [WAHL04] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real Adaboost. *Automatic Face and Gesture Recognition, Proceedings, Sixth IEEE International Conference on*, pages 79–84, 2004.
- [WCY99] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(6):557–563, 1999.
- [WR92] TLH Watkin and A. Rau. Selecting examples for perceptrons. *Journal of Physics A: Mathematical and General*, 25:113–121, 1992.
- [WRM⁺02] M.K. Warmuth, G. Ratsch, M. Mathieson, J. Liao, and C. Lemmen. Active learning in the drug discovery process. *Advances in Neural Information Processing Systems*, 14, 2002.
- [WRM03] J. Wu, J.M. Rehg, and M.D. Mullin. Learning a Rare Event Detection Cascade by Direct Feature Selection. *GVU Technical Report; GIT-GVU-03-16*, 2003.
- [WT04] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, September 2004.
- [WYPY96] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida. Face and Facial Feature Extraction from Color Image. *Automatic Face and Gesture Recognition, Proceedings, Second IEEE International Conference on*, pages 345–350, 1996.
- [YB99] L. Yin and A. Basu. Integrating active face tracking with model based coding. *Pattern Recognition Letters*, 20(6):651–657, 1999.
- [YC04] Z.R. Yang and K.C. Chou. Bio-support vector machines for computational proteomics. *Bioinformatics*, 20:5, 2004.
- [YCoEoC96] K.C. Yow, R. Cipolla, Department of Engineering, and University of Cambridge. *Feature-based Human Face Detection*. University of Cambridge, Department of Engineering, 1996.

- [YH93] G. Yang and T.S. Huang. Finding Human Faces in a Picture. *Proceedings of the 5th International Conference on Computer Analysis of Images and Patterns*, pages 766–769, 1993.
- [YKA01] M.H. Yang, D.J. Kriegman, and N. Ahuja. Face detection using multimodal density models. *Computer Vision and Image Understanding*, 84(2):264–284, 2001.
- [YKA02] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):34–58, 2002.
- [Yon01] Li Yongmin. *Dynamic Face Models: Construction and Application*. PhD thesis, the University of London, 2001.
- [YRA00] M.H. Yang, D. Roth, and N. Ahuja. A SNoW-Based Face Detector. *Advances in Neural Information Processing Systems*, 12, 2000.
- [ZCK98] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. *Face Recognition: From Theory to Applications*, pages 73–85, 1998.
- [ZZLZ02] Z.Q. Zhang, L. Zhu, SZ Li, and H.J. Zhang. Real-time multi-view face detection. *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 142–147, 2002.