Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

EYE DETECTION USING DISCRIMINATORY FEATURES AND AN EFFICIENT SUPPORT VECTOR MACHINE

by
Shuo Chen

Accurate and efficient eye detection has broad applications in computer vision, machine learning, and pattern recognition. This dissertation presents a number of accurate and efficient eye detection methods using various discriminatory features and a new efficient Support Vector Machine (eSVM).

This dissertation first introduces five popular image representation methods — the gray-scale image representation, the color image representation, the 2D Haar wavelet image representation, the Histograms of Oriented Gradients (HOG) image representation, and the Local Binary Patterns (LBP) image representation — and then applies these methods to derive five types of discriminatory features. Comparative assessments are then presented to evaluate the performance of these discriminatory features on the problem of eye detection.

This dissertation further proposes two discriminatory feature extraction (DFE) methods for eye detection. The first DFE method, discriminant component analysis (DCA), improves upon the popular principal component analysis (PCA) method. The PCA method can derive the optimal features for data representation but not for classification. In contrast, the DCA method, which applies a new criterion vector that is defined on two novel measure vectors, derives the optimal discriminatory features in the whitened PCA space for two-class classification problems. The second DFE method, clustering-based discriminant analysis (CDA), improves upon the popular Fisher linear discriminant (FLD) method. A major disadvantage of the FLD is that it may not be able to extract adequate features in order to achieve satisfactory performance, especially for two-class problems. To address this problem, three CDA

models (CDA-1, -2, and -3) are proposed by taking advantage of the clustering technique. For every CDA model a new between-cluster scatter matrix is defined. The CDA method thus can derive adequate features to achieve satisfactory performance for eye detection. Furthermore, the clustering nature of the three CDA models and the nonparametric nature of the CDA-2 and -3 models can further improve the detection performance upon the conventional FLD method.

This dissertation finally presents a new efficient Support Vector Machine (eSVM) for eye detection that improves the computational efficiency of the conventional Support Vector Machine (SVM). The eSVM first defines a Θ set that consists of the training samples on the wrong side of their margin derived from the conventional soft-margin SVM. The Θ set plays an important role in controlling the generalization performance of the eSVM. The eSVM then introduces only a single slack variable for all the training samples in the Θ set, and as a result, only a very small number of those samples in the Θ set become support vectors. The eSVM hence significantly reduces the number of support vectors and improves the computational efficiency without sacrificing the generalization performance. A modified Sequential Minimal Optimization (SMO) algorithm is then presented to solve the large Quadratic Programming (QP) problem defined in the optimization of the eSVM.

Three large-scale face databases, the Face Recognition Grand challenge (FRGC) version 2 database, the BioID database, and the FERET database, are applied to evaluate the proposed eye detection methods. Experimental results show the effectiveness of the proposed methods that improve upon some state-of-the-art eye detection methods.

EYE DETECTION USING DISCRIMINATORY FEATURES AND AN EFFICIENT SUPPORT VECTOR MACHINE

by
Shuo Chen

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

January 2013

APPROVAL PAGE

EYE DETECTION USING DISCRIMINATORY FEATURES AND AN EFFICIENT SUPPORT VECTOR MACHINE

Shuo Chen

Dr. Chengjun Liu, Dissertation Advisor — Date
Associate Professor of Computer Science, NJIT

Dr. James McHugh, Committee Member — Date
Professor of Computer Science, NJIT

Dr. David Nassimi, Committee Member — Date
Associate Professor of Computer Science, NJIT

Dr. Usman W. Roshan, Committee Member — Date
Associate Professor of Computer Science, NJIT

Dr. Edip Niver, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Shuo Chen

Degree: Doctor of Philosophy

Date: January 2013

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2013
- Master of Engineering in Computer Science and Technology,
  Hangzhou Dianzi University, Hangzhou, Zhejiang, China, 2007
- Bachelor of Engineering in Computer Science and Technology,
  Zhengzhou University, Zhengzhou, Henan, China, 2004

Major: *Computer Science*

Presentations and Publications:

S. Chen and C. Liu, "A New Efficient SVM and Its Application to Accurate and Efficient Eye Localization", *Artificial Intelligence*, 2012 (under review).

S. Chen and C. Liu, "Clustering-based Discriminant Analysis for Eye Detection", *IEEE Transactions on Image Processing*, 2012 (under review).

S. Chen and C. Liu, "Eye Detection Using Discriminatory Haar Features and A New Efficient SVM", *Image and Vision Computing*, 2012 (under review).

S. Chen and C. Liu,"Various Discriminatory Features for Eye Detection," in *Cross Disciplinary Biometric Systems*, C. Liu and V. Mago Eds., Springer-Verlag, pp.183-203, 2012.

S. Chen and C. Liu, "Eye Detection Using Color, Haar Features, and Efficient Support Vector Machine," in *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition: Advancing Technologies*, V.K. Mago and N. Bhatia Eds., IGI Global, Hershey, PA, pp.286-309, 2012.

S. Chen and C. Liu, "Fast Eye Detection Using Different Color Spaces," *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC'11)*, Anchorage, Alaska, October 9 - 12, 2011.

S. Chen and C. Liu, "Precise Eye Detection Using Discriminating HOG Features," *14th International Conference on Computer Analysis of Images and Patterns (CAIP'11)*, Seville, Spain, August 29 - 31, 2011.

S. Chen and C. Liu, "A New Efficient SVM and Its Application to Eye Detection," *2011 International Conference on Neural Networks (IJCNN'11)*, San Jose, California, July 31 - August 5, 2011.

S. Chen and C. Liu, "Discriminant Analysis of Haar Features for Accurate Eye Detection," *15th International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'11)*, Las Vegas, Nevada, July 18-21, 2011.

S. Chen and C. Liu, "Eye Detection Using Color Information and a New Efficient SVM," *IEEE Fourth International Conference on Biometrics: Theory, Applications, and Systems (BTAS'10)*, Washington DC, September 27-29, 2010.

"Precise Eye Detection Using Discriminatory HOG Features", The Seventh Annual Graduate Student Research Day (GSRD'11), New Jersey Institute of Technology, Newark, New Jersey, November 11, 2011.

"Eye Detection Using Color, Haar Features, and eSVM", The Sixth Annual Graduate Student Research Day (GSRD'10), New Jersey Institute of Technology, Newark, New Jersey, November 4, 2010.

"Eye Detection Using Color Information and a New Efficient SVM", IEEE Fourth International Conference on Biometrics: Theory, Applications, and Systems (BTAS'10), Washington DC, September 27, 2010.

To my beloved grandparents and parents, without whom I am nothing.

To my beloved wife, whose love and support brighten everyday.

ACKNOWLEDGMENT

First of all, I would like to express my sincere appreciation to my dissertation advisor, Prof. Chengjun Liu, for his tremendous guidance with this dissertation. Ever since I joined his research group, he has been an inexhaustible source of ideas, support, information, and energy. What he means to me is not only a dissertation advisor but a life mentor. I will always be grateful for the patience, encouragement, confidence, and kindness he has given me during the past five years.

Secondly, I am extremely grateful to Prof. James McHugh, Prof. David Nassimi, Prof. Usman Roshan, and Prof. Edip Niver, for serving on my committee. They have provided me invaluable advice, immense support, and unlimited encouragement throughout the dissertation process. It is their vast knowledge and generous help that make this dissertation possible. In addition, I would like to specially thank Prof. David Nassimi, who provided continuous assistantship to help me complete my degree.

I would also like to extend my thanks to all the faculty and staff in the Department of Computer Science at New Jersey Institute of Technology for any help they have given me. I am also thankful to all my friends at NJIT, including Zhiming Liu, Venkata Gopal Edupuganti, Jichao Sun, Shengyan Gao, Xiguo Ma, Wei Wang. They brought fun and joy into the tough Ph.D. study.

Lastly, but most importantly, my deepest gratitude goes to my beloved family. I am grateful to my grandparents, Jizu Chen and Guiying Zhang. They planted a seed of science in my heart when I was very young. They have taught me how to live a good life through one's own endeavor. I could never have accomplished all I have without their neverending encouragement, unconditional support, and endless love. I thank my parents, Xinsheng Chen and Shanfeng Liu. They have given me everything that they had. Their love and support are my motivation to keep moving

forward. I am so blessed to be able to share this accomplishment with them. I also owe many thanks to my wife, Fang Hou. She is the sunshine to my every day. Her love makes my life full of beauty and meaning. I am so fortunate to have her behind my back. I would also like to thank my uncles, my aunts, and my in-laws. They have been always there whenever I need help and advice. They have always stayed with me, supported me, and been confident in me in every hard time throughout my life.

TABLE OF CONTENTS

Chapter **Page**

1 INTRODUCTION 1

1.1 Motivation and Background 1

1.2 Topics Overview 4

2 VARIOUS IMAGE REPRESENTATIONS FOR EYE DETECTION 7

2.1 Gray-scale and Color Image Representations 7

2.2 Haar Wavelet Image Representation 9

2.3 Histograms of Oriented Gradients 12

2.4 Local Binary Patterns 13

3 DISCRIMINANT COMPONENT ANALYSIS 15

3.1 Principal Component Analysis 16

3.2 Discriminant Component Analysis 17

3.3 Experiments 19

3.3.1 Overview of the DCA-based Eye Detection method 20

3.3.2 Database 21

3.3.3 Effectiveness Evaluation of the DCA Method 22

3.3.4 Comparative Assessment of the PCA and DCA Features 26

3.4 Conclusion 33

4 CLUSTERING-BASED DISCRIMINANT ANALYSIS 34

4.1 Background 35

4.2 Clustering-based Discriminant Analysis 37

4.2.1 CDA-1 Model 38

4.2.2 CDA-2 Model 42

4.2.3 CDA-3 Model 45

4.3 Experiments 46

4.3.1 Evaluation of the Size of Extracted Features 47

TABLE OF CONTENTS
(Continued)

Chapter **Page**

4.3.2 Evaluation of the Number of Clusters 49

4.3.3 Evaluation of the CDA-based Eye Detection Method 52

4.3.4 Comparison with State-of-the-art Methods 55

4.4 Conclusion 57

5 EFFICIENT SUPPORT VECTOR MACHINE 58

5.1 Background 59

5.2 Support Vector Machine 60

5.3 Efficient Support Vector Machine 62

5.4 Modified Sequential Minimal Optimization Algorithm 67

5.4.1 An Analytic Solution to the Smallest QP Problem 67

5.4.2 A Heuristic Approach for Choosing Multipliers 69

5.5 Accurate and Efficient Eye Detection Using eSVM 70

5.5.1 The Eye Candidate Selection Stage 71

5.5.2 The Eye Candidate Validation Stage 76

5.6 Experiments 76

5.6.1 Evaluation of the eSVM Method 77

5.6.2 Evaluation of the eSVM-based Eye Detection Method 84

5.6.3 Comparison with Recent Methods 90

5.7 Conclusion 92

6 CONCLUSION AND FUTURE WORK 93

REFERENCES 96

LIST OF TABLES

Table		Page
3.1	Parameter Settings and the Feature Size of the PCA and the DCA Features	25
3.2	Performance Comparison under Different Similarity Measures (ED Stands for the Euclidean Distance and DR Stands for the Detection Rate)	32
4.1	Parameter Settings of the FLD-, NDA-, and CDA-based Eye Detection Methods	52
4.2	Comparison of the True Positive Rate (TPR) of the FLD-, NDA-, and CDA-based Eye Detection Methods at the False Positive Rate (FPR) of 0.1	53
4.3	The detection Rate and Detection Accuracy of the FLD-, NDA-, and CDA-based Methods. The Detection Rate is for the Normalized Error $e \leq 0.07$. The mean$(\cdot)$ and std$(\cdot)$ Represent the Mean and the Deviation of the Detection Pixel Error with respect to the Direction Specified by the Parameter, Respectively (DR Stands for the Detection Rate)	55
4.4	Comparison of the Eye Detection Performance with State-of-the-Art Methods (e Stands for the Normalized Error)	56
5.1	Result Comparisons between the Conventional SVM and the eSVM with the Linear and RBF Kernels	79
5.2	Performance Assessment of the SVM and the eSVM	81
5.3	Data Set Description and Parameter Settings	82
5.4	Performance Assessment of the SVM, the RSVM, and the eSVM (T Stands for Time in Seconds)	83
5.5	Efficiency Comparison between the SVM and the eSVM	85
5.6	Performance of the SVM and the eSVM within Five Pixel Localization Error	88
5.7	Performance of Final Eye Detection within Five Pixel Localization Error under Different Q	88
5.8	Comparisons of the Eye Detection Performance for Different Methods on the FERET Database (e Stands for the Normalized Error)	91

LIST OF FIGURES

Figure **Page**

2.1 The RGB image and the gray-scale image in the first row. The red, green, and blue components of the RGB color image representation in the middle row. The Y, Cb, and Cr components of the YCbCr color image representation in the bottom row. 8

2.2 The 64 2D Haar basis functions for V^3. White, black, and gray represent 1, -1, and 0, respectively, and for simplicity the basis functions are not scaled. 10

2.3 (a) the gray-scale face image; (b) the gradient norm; (c) cell splitting; (d) the gradient orientation of the cell marked by the red square in (c); (e) the histogram ($K = 12$) of (d). 11

2.4 Examples of the gray-scale images in the top row and their corresponding LBP image representations in the bottom row. 14

3.1 An example of PCA based feature extraction for the two-fold classification. The distributions of both classes follow an ellipse shape. The yellow area represents one class while the blue area represents another. The gray area represents the overlap of these two classes. 17

3.2 System architecture of the DCA-based eye detection method. 20

3.3 Example eye strip images from the FRGC version 2 database with the spatial resolution of 55×128.The top three rows show the colored image, whereas the bottom three rows show the corresponding gray-scale image after illumination normalization. 21

3.4 Example eye (the top row) and non-eye (the bottom row) training images that are normalized to 20×40. The images are preprocessed by illumination normalization. 22

3.5 Different selection of the basis vectors for the PCA method and the DCA method, respectively. 23

3.6 The distribution of the eye and non-eye training images using two most significant PCA features (a) and two most significant DCA features (b), respectively. 24

3.7 Eye detection performance of the PCA and the DCA gray-scale features, respectively. 27

3.8 Eye detection performance of the PCA and the DCA YCbCr color features, respectively. 28

LIST OF FIGURES
(Continued)

Figure **Page**

3.9 Eye detection performance of the PCA and the DCA Haar features, respectively. 29

3.10 Eye detection performance of the PCA and the DCA HOG features, respectively. 30

3.11 Eye detection performance of the PCA and the DCA LBP features, respectively. 31

4.1 The between-cluster matrices of CDA-1, -2, and -3, respectively. The figure shows a three-class problem and each class is further divided into three clusters. M_0 represents the grand mean, whereas $M_q^{(p)}, p, q = 1, 2, 3$, represents the mean vector of the qth cluster from class ω_p. $\mathbf{X}_j^{(2)}, j = 1, 2, \cdots, 6$, represents six data samples from class ω_2. (a) The between-cluster scatter matrix of CDA-1 measures the scatter of the mean vector from each cluster with respect to the grand mean. (b) The between-cluster scatter matrix of CDA-2 measures the scatter of each feature vector from one class with respect to the mean vector of its nearest cluster from otherwise classes. (c) The between-cluster scatter matrix of CDA-3 measures each feature vector from one class with respect to mean vectors of all the clusters from otherwise classes. 37

4.2 System architecture of the CDA-based eye detection method. 47

4.3 The detection performance of the CDA-1 as the size of features varies. . 48

4.4 The detection performance of the CDA-2 as the size of features varies. . 48

4.5 The detection performance of the CDA-3 as the size of features varies. . 49

4.6 The detection performance of the CDA-1 as the number of clusters varies. 50

4.7 The detection performance of the CDA-2 as the number of clusters varies. 50

4.8 The detection performance of the CDA-3 as the number of clusters varies. 51

4.9 The ROC curves of the FLD-, NDA-, and CDA-based eye detection methods. 53

4.10 The detection rate of the FLD-, NDA-, and CDA-based eye detection methods over different normalized errors. 54

5.1 Illustration of the Conventional soft-margin SVM in the two dimensional space, where the two classes are presented by the solid and open circles, respectively. All five samples on the wrong side of their margin are pulled onto their boundaries to become support vectors. 61

LIST OF FIGURES
(Continued)

Figure **Page**

5.2 Illustration of the eSVM in the two dimensional space, where the two classes are presented by the solid and open circles, respectively. Only one of the five samples on the wrong side of their margin is pulled onto its boundary to become a support vector. 66

5.3 System architecture of the eSVM-based eye detection method. 71

5.4 The eye-tone distribution in the YCbCr color space. The skin pixels are represented in red, the eye region pixels are in blue, and the pupil-center pixels are in green. 72

5.5 Example eye strip images in the YCbCr color space, where Y is represented in red, Cb in green, and Cr in blue. 73

5.6 The percentage of the real eye representations as the number of the selected eye candidates varies. 74

5.7 Examples of good (a) and bad (b) eye candidate selection results. 75

5.8 Example of the eye pair selection scheme. 75

5.9 The support vectors and the separating boundaries of the conventional SVM and the eSVM with the linear and RBF kernels on the *Ripley* data set, respectively. The dashed lines/curves depict the ± 1 margins around the separating boundary. 78

5.10 Recall and precision of the SVM- and the eSVM-based methods as Q varies. 86

5.11 (a) Performance comparison of the final eye localization under different Q. (b) Distribution of eye localization pixel errors for final eye localization when $Q = 3$. 89

5.12 Examples of the eye detection results using the eSVM based eye detection method. 90

CHAPTER 1

INTRODUCTION

1.1 Motivation and Background

Eye detection has broad applications in computer vision, machine learning, and pattern recognition. However, finding an accurate and efficient solution to eye detection is really challenging. Example challenges include large variations in image illumination, skin color (white, yellow, and black), facial expression (eyes open, partially open, or closed), as well as scale and orientation. Additional challenges include eye occlusion caused by eye glasses or long hair, and the red eye effect due to the photographic effect. All these challenging factors increase the difficulty of accurate and efficient eye-center detection.

Eye detection therefore has attracted much attention, and numerous eye detection methods have been proposed. Current eye detection methods can be classified into three categories [94]: the template based methods, the feature based methods, and the appearance based methods. For the template based methods, a sliding window is moved over the whole image to find the best match with a pre-designed generic eye template. The eye template is usually built upon either the prior knowledge or a large eye database. Jorge et. al. [40] applied a deformable eye template to detect eyes. This template is represented by two distinct geometrical entities: a circumference, that defines the iris contour; and two parabolas, one concave and other convex, that define respectively the above and below contours of the eye. The geometry shape of the eye template is controlled by a set of eleven parameters that allow its change in scale, position, and orientation. Rurainsky & Eisert [76] presented an adaptive eye template that is controlled by only four position parameters. This small number of parameters limits the range of changes and subsequently limits the number of

possible template shapes. Besides these artificial eye templates, Moriyama et. al. [61] presents natural eye templates taken from real persons. These eye templates are designed upon a large eye database and thus in various orientations, sizes and illuminations. A pre-processing step is necessary to align and normalize these eye templates in order to improve the detection accuracy. Some other template based eye detection methods can be found in [25], [87], and [43].

The feature based eye detection methods focus on the characteristics of eyes, such as the shape, the color distribution, and the intensity gradient information around eye regions. Among these characteristics the circular shape of the iris is a typical one. Wan Mohd Khairosfaizal and Nor'aini [86] presented an eye detection method by searching the circular shape over a face image. This method starts with applying a sharpening filter to enhance edges of objects in an image. The Circle Hough Transform (CHT) is then applied to search the circular patterns in the edge image. Feng and Yuan [28], as well as Zhou and Geng [92], presented a number of eye detection methods using projection functions. These methods are based on the observation that the eye boundaries have more significant intensity variance than other areas of a face image. Three projection functions, the Integral Projection Function (IPF), the Variance Projection Function (VPF), and the Generalized Projection Function (GPF), are presented in their work. The GPF is finally proved to be optimal for eye detection under the illumination variations. Chen & Liu [20] presented an eye detection method using different color channels. This method first roughly locates the eye boundaries in the YCbCr color space [79], and then further detects the eye-center in the HSV color space [79]. In addition to above methods, a special type of illumination, active near-infrared (IR) illumination [94], is widely used in the feature based eye detection method. The IR illumination is able to produce the dark or bright pupil effect, which can enhance the characteristics of eyes and hence may improve the detection performance.

The appearance based eye detection methods detect eyes based on their photometric appearance. These methods usually need to first train a classifier based on a training data set, and the detection is then achieved via a two-fold classification process. Various image representations, other than the simple gray-scale image representation, are widely used in the appearance based methods. These image representations may be able to extract discriminatory features that are suitable for classifier design and improve the classification performance. Kroon et al. [41] presented a probabilistic eye localization method using the multi-scale Local Binary Patterns (LBP) image representation. Nguyen et al. [62] proposed an energy-based framework to jointly perform relevant feature weighting and SVM parameter learning for facial feature detection. Jin et al. [38] proposed an eye detection algorithm that integrates the characteristics of single eye and eye-pair images. Campadelli et al. [12] presented an eye detection method that uses two SVMs trained on properly selected Haar wavelet coefficients. Everingham and Zisserman [27] investigated three approaches for eye detection: a regression approach for directly minimizing eye location error, a Bayesian approach for eye and non-eye modeling, and an AdaBoost approach for training a discriminative eye detector. Some other state-of-the-art appearance based eye detection methods can be found in [91], [88], [37], [34], [35], [22], [5], [6], [63], [13], and [81].

Even though numerous eye detection methods have been proposed, many problems still exist, especially in detection accuracy and efficiency under challenging image conditions. This dissertation presents a number of accurate and efficient eye detection methods using discriminatory features and a new efficient Support Vector Machine (eSVM).

1.2 Topics Overview

This dissertation aims to design accurate and efficient eye detection methods using discriminatory features and a new efficient Support Vector Machine (eSVM). Chapter 2 first introduces five image representation methods that are widely used in pattern recognition and computer vision. Chapter 3 and Chapter 4 then present respectively two Discriminatory Feature Extraction (DFE) methods, the Discriminant Component Analysis (DCA) and the Clustering-based Discriminant Analysis (CDA), to extract the discriminatory features for eye detection. The DCA method improves upon the conventional Principal Component Analysis (PCA) method, whereas the CDA method improves upon the conventional Fisher Linear Discriminant (FLD) method. Next, Chapter 5 presents a new efficient Support Vector Machine (eSVM) for eye detection to improve the computational efficiency of the conventional Support Vector Machine (SVM). Finally, Chapter 6 concludes this dissertation and depicts the future research directions. An overview of Chapters 2, 3, 4, 5, and 6 is given in the follows.

Chapter 2 introduces five image representation methods: the gray-scale image representation, the color image representation, the 2D Haar wavelet image representation, the Histograms of Oriented Gradients (HOG) image representation, and the Local Binary Patterns (LBP) image representation. These five image representations are then applied to derive five types of discriminatory features in Chapter 3. Comparative assessments are presented to evaluate the performance of these discriminatory features on the problem of eye detection.

Chapter 3 proposes a Discriminant Component Analysis (DCA) method to extract discriminatory features for eye detection. The DCA method improves upon the popular Principal Component Analysis (PCA) method. It starts with a PCA process followed by a whitening transformation. A discriminant analysis is then performed on the whitened PCA space. A set of DCA basis vectors, based on the novel definition of the cluster-measure vector and the separation-measure vector, as

well as a new criterion vector, is defined. The DCA features are then derived in the subspace spanned by these DCA basis vectors. Experiments on the Face Recognition Grand Challenge (FRGC) database show that the DCA features significantly enhance the discriminating power of various image representations and hence improve the eye detection performance.

Chapter 4 proposes a clustering-based discriminant analysis (CDA) method to extract discriminatory features for eye detection. The CDA method improves upon the Fisher Linear Discriminant (FLD) method. One major disadvantage of the FLD is that it may not be able to extract adequate features in order to achieve satisfactory performance, especially for two class problems. Three CDA models (CDA-1, -2, and -3) are proposed by taking advantage of the clustering technique. For every CDA model a new between-cluster scatter matrix is defined. The CDA method thus can derive adequate features to achieve satisfactory performance for eye detection. Furthermore, the clustering nature of the three CDA models and the nonparametric nature of the CDA-2 and -3 models can further improve the detection performance upon the conventional FLD method. Experiments on the FRGC and the BioID database show the feasibility of the proposed three CDA models and the improved performance over some state-of-the-art eye detection methods.

Chapter 5 proposes a new efficient Support Vector Machine (eSVM) for eye detection that improves the computational efficiency of the conventional Support Vector Machine (SVM). The eSVM first defines a Θ set that consists of the training samples on the wrong side of their margin derived from the conventional soft-margin SVM. The Θ set plays an important role in controlling the generalization performance of the eSVM. The eSVM then introduces only a single slack variable for all the training samples in the Θ set, and as a result, only a very small number of those samples in the Θ set become support vectors. The eSVM hence significantly reduces the number of support vectors and improves the computational efficiency without

sacrificing the generalization performance. The optimization of the eSVM is implemented using a modified Sequential Minimal Optimization (SMO) algorithm to solve the large Quadratic Programming (QP) problem. Experiments on several diverse data sets show that the eSVM significantly improves the computational efficiency upon the conventional SVM while achieving comparable generalization performance to or higher performance than the SVM. Furthermore, experiments on the FRGC and the FERET database show that the eSVM based eye detection method can achieve real-time eye detection speed and better eye detection performance than some recent eye detection methods.

Chapter 6 summaries the research achievements and contributions of this dissertation and depicts the future research directions.

CHAPTER 2

VARIOUS IMAGE REPRESENTATIONS FOR EYE DETECTION

Since various image representation methods are introduced into computer vision, it has been shown that these image representations can derive better recognition performance than the basic gray-scale image representation. This chapter briefly reviews five popular image representations: the gray-scale image representation, the color image representation [51, 48, 53, 52], the 2D Haar wavelet image representation [85], the Histograms of Oriented Gradients (HOG) image representation [23], and the Local Binary Patterns (LBP) image representation [64, 65]. These five image representations are then applied in the following chapters to derive five types of discriminatory features, and comparative assessments are presented to evaluate the performance of these discriminatory features on the problem of eye detection.

2.1 Gray-scale and Color Image Representations

Gray-scale image is a common image representation method. Each pixel of the image carries an intensity value varying from black at the weakest intensity to white at the strongest. A gray-scale vector may be formed by placing all the intensity values in a column to represent the image for pattern classification.

An alternative to the gray-scale image representation is the color image representation. Some widely used color image representations include the YCbCr color image representation, the YIQ color image representation, the HSV color image representation, and the $I_1I_2I_3$ color image representation [79]. A number of novel hybrid color image representations have been proposed recently. Since this dissertation does not focus on exploring and comparing different color image representations, only YCbCr color image representation is discussed in this dissertation, which has been shown

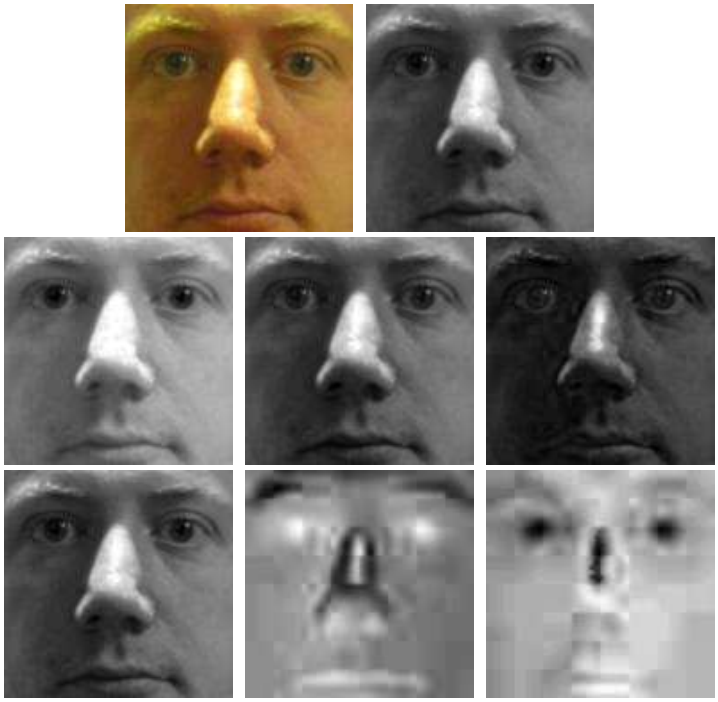


Figure 2.1 The RGB image and the gray-scale image in the first row. The red, green, and blue components of the RGB color image representation in the middle row. The Y, Cb, and Cr components of the YCbCr color image representation in the bottom row.

effective for eye detection [19]. The YCbCr color image representation contains three color components: luminance (Y), chrominance blue (Cb), and chrominance red (Cr). The YCbCr color image representation is defined as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.4810 & 128.5530 & 24.9660 \\ -37.7745 & -74.1592 & 111.9337 \\ 111.9581 & -93.7509 & -18.2072 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

Figure 2.1 shows an example of the RGB image, the gray-scale image, the red, green, and blue components of the RGB color image representation, and the Y, Cb, and Cr components of the YCbCr color image representation.

2.2 Haar Wavelet Image Representation

Haar wavelet image representation has been widely used in objection detection [85]. The 2D Haar wavelet transform is defined as the projection of an image onto the 2D Haar basis functions [11]. The attractive characteristics of the 2D Haar basis functions enhance local contrast and facilitate feature extraction in many target detection problems, such as eye detection, where dark pupil is in the center of colored iris that is surrounded by white sclera. The 2D Haar basis functions can be generated from the one dimensional Haar scaling and wavelet functions.

The Haar scaling function $\phi(x)$ may be defined as follows [11], [71]:

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

A family of functions can be generated from the basic scaling function by scaling and translation [11], [71]:

$$\phi_{i,j}(x) = 2^{i/2}\phi(2^i x - j) \tag{2.3}$$

As a result, the scaling functions $\phi_{i,j}(x)$ can span the vector spaces V^i, which are nested: $V^0 \subset V^1 \subset V^2 \subset \cdots$ [59].

The Haar wavelet function $\psi(x)$ may be defined as follows [11], [71]:

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

The Haar wavelets are generated from the mother wavelet by scaling and translation [11], [71]:

$$\psi_{i,j}(x) = 2^{i/2}\psi(2^i x - j) \tag{2.5}$$

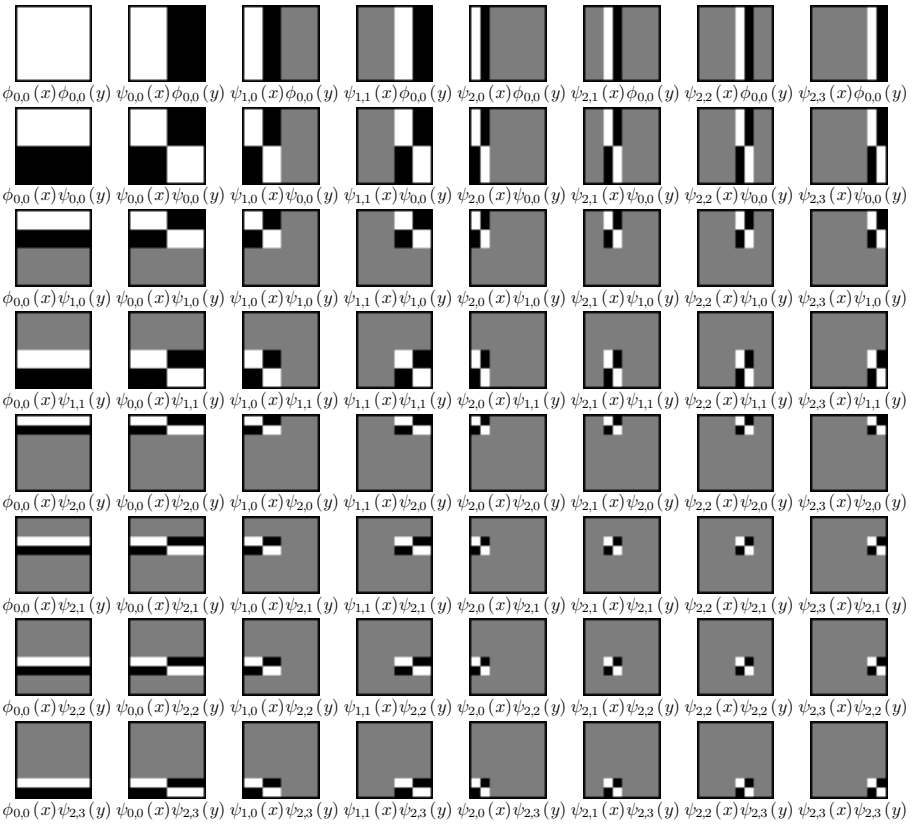


Figure 2.2 The 64 2D Haar basis functions for V^3. White, black, and gray represent 1, -1, and 0, respectively, and for simplicity the basis functions are not scaled.

The Haar wavelets $\psi_{i,j}(x)$ span the vector space W^i, which is the orthogonal complement of V^i in V^{i+1}: $V^{i+1} = V^i \oplus W^i$ [11], [71].

The 2D Haar basis functions are the tensor product of the one dimensional scaling and wavelet functions [8]. For example, for V^3, where $V^3 = V^0 \oplus W^0 \oplus W^1 \oplus W^2$, the 2D Haar basis consists of 64 basis functions. Figure 2.2 displays the 64 2D Haar basis functions for V^3, where white, black, and gray represent 1, -1, and 0, respectively. Note that for simplicity the basis functions in Figure 2.2 are not scaled. Figure 2.2 reveals that the 2D Haar basis functions include a set of scaled and shifted box type functions that encode the differences in average intensities among the regions in different scales. Specifically, the 2D Haar basis functions contain mainly three types of representations in the two dimensional space: (i) two horizontal neighboring regions

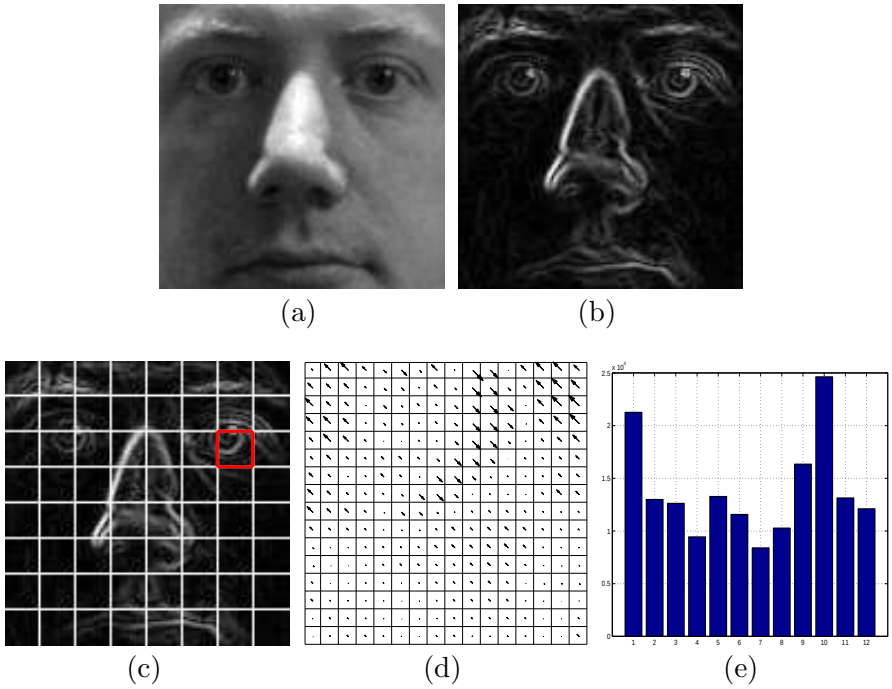


Figure 2.3 (a) the gray-scale face image; (b) the gradient norm; (c) cell splitting; (d) the gradient orientation of the cell marked by the red square in (c); (e) the histogram ($K = 12$) of (d).

for computing the difference between the sum of the pixels within each of them, (ii) two vertical neighboring regions for computing the difference between the sum of the pixels within each of them, and (iii) four neighboring regions for computing the difference between the diagonal pairs of the regions. Note that the first basis function is for computing the average of the whole image.

One advantage of the 2D Haar wavelet image representation is that the projection of an image onto the 2D Haar basis functions, which really is inner products of an image vector with the Haar basis functions, can be efficiently computed by just several integer additions and subtractions instead of the time-consuming floating point multiplications [85].

Algorithm 1 Overview of the HOG image representation method.

Step1: Compute the horizontal and vertical gradient of the input image by convolving it with a derivative mask.

Step2: Compute both norm and orientation of the gradient. Let G_h and G_v denote the horizontal and vertical gradient, respectively. The norm N_G and orientation O_G at the point (x, y) are computed as follows:

$N_G(x, y) = \sqrt{G_h(x, y)^2 + G_v(x, y)^2}$ (see Figure 2.3(b)),

$O_G(x, y) = arctan\frac{G_h(x,y)}{G_v(x,y)}$ (see Figure 2.3(d)).

Step3: Split the image into cells (see Figure 2.3(c)). Compute the histogram for each cell (see Figure 2.3(e)). Suppose the histogram is divided into K bins based on the orientation, the value of the i-th bin V_i for cell C is computed as follow:

$$V_i = \sum_{(x,y)\in C} \{N_G(x, y), O_G(x, y) \in Bin_i\}.$$

Step4: Normalize all histograms within a block of cells.

Step5: Concatenate all normalized histograms to form the HOG feature vector.

2.3 Histograms of Oriented Gradients

The Histograms of Oriented Gradients (HOG) image representation, which is inherited from the Scale Invariant Feature Transform (SIFT) [58], is originally applied to human detection [23]. The basis idea of HOG rests on the observation that the local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions. The HOG image representation is derived based on a series of well-normalized local histograms of image gradient orientations in a dense grid [23]. In particular, the image is firstly divided into a number of small cells. For each cell, a local histogram of gradient directions or edge orientations is accumulated over the pixels of the cell. All histograms within a block of cells are then normalized to reduce the effect of the illumination variation.

The blocks can be overlapped with each other for performance improvement. The final HOG image representation is formed by concatenating all normalized histograms into a single vector. Algrithm 1 shows the details of the HOG method.

2.4 Local Binary Patterns

In recent years, Local Binary Patterns (LBP) has been applied to many pattern recognition problems, such as face detection and recognition, scene and image texture classification [64, 65]. The gray-scale invariant property of the LBP image representation makes it a powerful tool for text description. The basic LBP labels the pixels of a gray-scale image by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as an 8-bit-code binary number. Specifically, given the central pixel (x_c, y_c) and its surrounding pixels $(x_s, y_s), s = 0, 1, \cdots, 7$, the labeled image can be defined as follows:

$$LBP(x_c, y_c) = \sum_{s=0}^{7} 2^s f(I(x_c, y_c) - I(x_s, y_s)) \tag{2.6}$$

where $I(\cdot)$ denotes the intensity value and $f(\cdot)$ is defined as follows:

$$f(u) = \begin{cases} 1, & \text{when } u \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Figure 2.4 shows some examples of the gray-scale images and their corresponding LBP images. The LBP description usually is the histogram of these LBP images.

Two extensions of the basic LBP image representation are further developed [2], [1]. The first extension allows LBP to define on the neighborhood of any size by using circular neighborhood and bilinear interpolation of the pixel values. The second extension defines a concept of uniform patterns. An LBP operator, when viewed as a

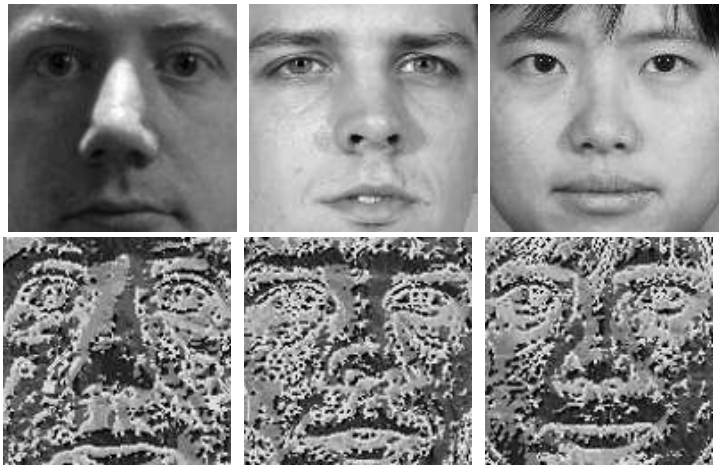


Figure 2.4 Examples of the gray-scale images in the top row and their corresponding LBP image representations in the bottom row.

circular bit string, is considered uniform if there are at most one transmission from 0 to 1 and one from 1 to 0. Based on these two extensions, LBP is commonly described as: $LBP^{u2}_{P,R}$, where $u2$ means using only uniform patterns and (P, R) denotes P sampling points on a circle of radius R.

In order to enhance the performance of the LBP image representation, one usually first divides an image into a number of regions and applies the LBP operator to each region. The regions can have different sizes and overlap with each other. The enhanced LBP image representation is then derived by concatenating the histograms from all the regions.

CHAPTER 3

DISCRIMINANT COMPONENT ANALYSIS

One drawback of image representations is that they usually reside in a high dimensional space. However, low dimensionality is especially important for learning, as the number of training samples required for attaining a given level of performance grows exponentially with the dimensionality of the vector space. The Principal Component Analysis (PCA) [30] is an optimal feature extraction and dimensional reduction method for signal or image representation in the sense of mean square error. However, it does not extract the optimal discriminatory features for classification [49]. In contrast to the case of pattern classification, where one need to decide between a relatively small number of classes, the detection problem requires to differentiate between the object class and the rest of the world. As a result, the extracted features for object detection must have discriminating power to handle the cluttered scenes that the object is presented within.

This chapter proposes a Discriminant Component Analysis (DCA) method, which improves upon the PCA method, to extract discriminatory features for eye detection. The DCA method starts with a PCA procedure followed by a whitening transformation. A discriminant analysis is then performed on the whitened PCA space. A set of DCA basis vectors, based on the novel definition of the cluster-measure vector and the separation-measure vector, as well as a new criterion vector, is defined. The DCA features are then derived in the subspace spanned by these DCA basis vectors. Experiments on the Face Recognition Grand Challenge (FRGC) database show that the DCA features significantly enhance the discriminating power of various image representations and hence improve the eye detection performance.

3.1 Principal Component Analysis

The PCA [30] is an optimal feature extraction and dimensional reduction method for signal or image representation in the sense of mean square error. Specifically, let $\mathcal{X} \in \mathbb{R}^N$ be an image representation pattern vector in an N dimensional space, and $\mathcal{S} \in \mathbb{R}^{N \times N}$ be the covariance matrix of $\mathcal{X}$. The covariance matrix $\mathcal{S}$ can be defined as follows:

$$\mathcal{S} = \varepsilon\{[\mathcal{X} - \varepsilon(\mathcal{X})][\mathcal{X} - \varepsilon(\mathcal{X})]^t\} \tag{3.1}$$

where $\varepsilon(\cdot)$ is the expectation operator. The covariance matrix can be factorized into the following form according to [30]:

$$\mathcal{S} = \Phi\Lambda\Phi \tag{3.2}$$

where $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]$ is an orthogonal eigenvector matrix and $\Lambda = diag\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ is a diagonal eigenvalue matrix with diagonal elements in decreasing order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. The PCA features $\mathcal{Y}$ are then extracted:

$$\mathcal{Y} = P^t\mathcal{X} \tag{3.3}$$

where $P = [\phi_1, \phi_2, \cdots, \phi_m]$, $m < N$, and $P \in \mathbb{R}^{N \times m}$.

The PCA method takes the mean-square error as its criterion to extract features. That is, the PCA is to search an optimal projection matrix (i.e., P) which generates a set of PCA features with minimum mean-square error. The mean-square error, even though, can preserve the optimal representation of the original features, it can not preserve any information of the class separation. Figure 3.1 gives an example, which shows two distributions with two variables $X = (x_1, x_2)$ from two independent classes. These two variables are highly correlated with each other as shown in Figure 3.1. In terms of PCA, the principle component ϕ_1 with larger eigenvalue produces a smaller

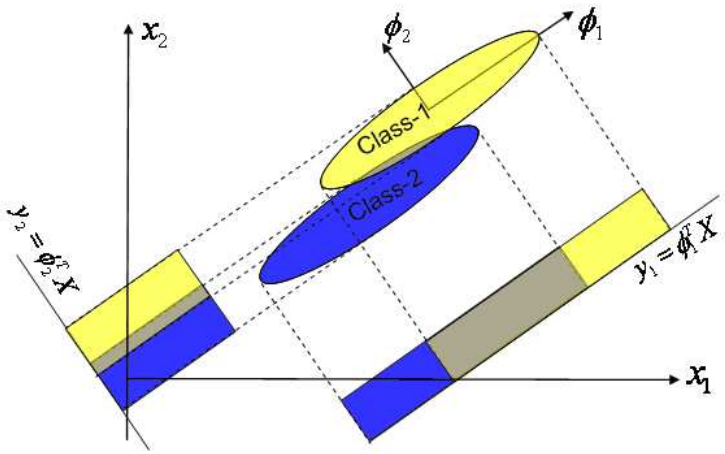


Figure 3.1 An example of PCA based feature extraction for the two-fold classification. The distributions of both classes follow an ellipse shape. The yellow area represents one class while the blue area represents another. The gray area represents the overlap of these two classes.

mean-square error than the principal component ϕ_2. As a result, the selection of y_1 is a better vector than y_2 to represent the vectors of these distributions. However, as shown in Figure 3.1, if the two distributions are projected onto ϕ_1, the two classes are heavily overlapped (the gray area), which indicates that they are hard to be separated. In contrast, if they are projected on to ϕ_2, the two classes are well separated with little overlap. Therefore, for classification purpose, y_2 is a better feature than y_1. Above observation reveals that PCA may not be able to extract the optimal discriminating features for classification.

3.2 Discriminant Component Analysis

This section presents a Discriminant Component Analysis (DCA) method for two-class problems, which improves upon the PCA method, to extract discriminatory features for classification. The DCA method starts with a PCA procedure followed by a whitening transformation. A discriminant analysis is then performed on the whitened PCA space. A set of DCA basis vectors, based on the novel definition of the cluster-measure vector and the separation-measure vector, as well as a new

criterion vector, is defined through this analysis. The DCA features are then derived in the subspace spanned by these DCA basis vectors.

In particular, the DCA method first applies a whitening transformation to sphere the covariance matrix of the PCA features $\mathcal{Y}$. The whitening transformation matrix is defined as follows:

$$W = P\Delta^{-1/2} \tag{3.4}$$

where $\Delta = diag(\lambda_1, \lambda_2, \cdots, \lambda_m)$, $W = [W_1, W_2, \cdots, W_m]$, and $W \in \mathbb{R}^{N \times m}$. The whitening transformation not only eliminates the correlation between variables but also normalizes the deviation of each variable.

The DCA method next defines two measure vectors, the cluster-measure vector and the separation-measure vector, as well as a criterion vector, in order to select the most discriminatory projection vectors from W defined in Equation 3.4. These selected vectors then form the DCA subspace, in which the DCA features reside. Towards that end, the cluster-measure vector, $\alpha \in \mathbb{R}^m$, and the separation-measure vector, $\beta \in \mathbb{R}^m$, are defined as follows:

$$\alpha = P_1 \sum_{i=1}^{n_1} s(W^t x_i^{(1)} - W^t M_1) + P_2 \sum_{i=1}^{n_2} s(W^t x_i^{(2)} - W^t M_2) \tag{3.5}$$

$$\beta = P_1 s(W^t M_1 - W^t M) + P_2 s(W^t M_2 - W^t M) \tag{3.6}$$

where P_1 and P_2 are the prior probabilities, n_1 and n_2 are the number of samples, and $x_i^{(1)}$ and $x_i^{(2)}$ are the pattern vectors of the first and the second classes, respectively. M_1, M_2, and M are the mean vectors of the two classes, and the grand mean, respectively. The $s(\cdot)$ function defines the absolute value of the elements of the input vector. The significance of these new measure vectors is that the cluster-measure

vector, $\alpha \in \mathbb{R}^m$, measures the clustering capability of the projection vectors in W, whereas the separation-measure vector, $\beta \in \mathbb{R}^m$, measures the separating capability of the vectors in W.

In order to choose the most discriminatory projection vectors, a new criterion vector $\gamma \in \mathbb{R}^m$ is defined as follows:

$$\gamma = \beta./\alpha \tag{3.7}$$

where ./ is element-wise division. The value of the elements in γ indicates the discriminatory power of their corresponding projection vectors in W: the larger the value is, the more discriminatory power the corresponding vector in W possesses. The DCA method therefore chooses the top p projection vectors, $W_{i1}, W_{i2}, \cdots, W_{ip}$, in W corresponding to the p largest values in γ to form the DCA basis vectors T:

$$T = [W_{i1}, W_{i2}, \cdots, W_{ip}] \tag{3.8}$$

where $T \in \mathbb{R}^{N \times p}$ and $p < m$. The DCA features thus reside in the feature space spanned by these DCA basis vectors. The DCA features are defined as follows:

$$\mathcal{Z} = T^t \mathcal{X} \tag{3.9}$$

The DCA method therefore captures the most discriminatory features of the original image representation pattern vectors in a low dimensional space.

3.3 Experiments

This section evaluates the effectiveness of the DCA method over the PCA method on the problem of eye detection. Five types of DCA (PCA) features are derived from the five types of image representations discussed in Chapter 2. Comparative assessments

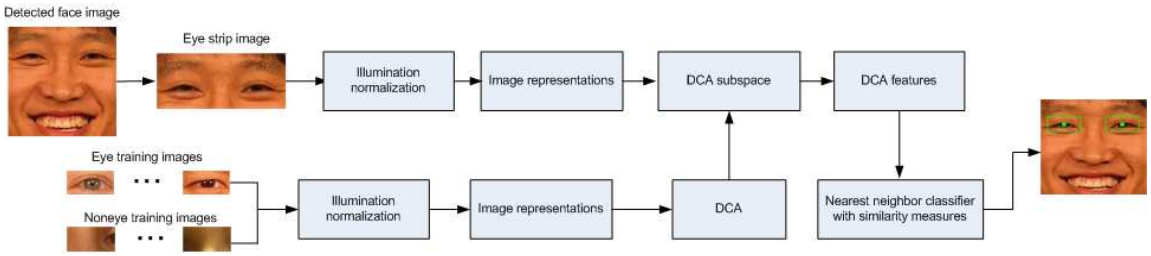


Figure 3.2 System architecture of the DCA-based eye detection method.

among these features are also presented in this section to evaluate their performance on eye detection.

3.3.1 Overview of the DCA-based Eye Detection method

Figure 3.2 shows the architecture of the DCA-based eye detection method. First, the Bayesian Discriminating Features(BDF) method [47] is applied to detect a face from an image and normalizes the detected face to a predefined size. Second, some geometric constraints are applied to extract an eye strip from the upper portion of the detected face. Illumination variations are then attenuated by means of an illumination normalization procedure that consists of Gamma correction, difference of Gaussian filtering, and contrast equalization as applied in [55] and [54]. Third, the image representations are derived from the eye strip image and then the DCA method is applied to extract the DCA features. Finally, the nearest neighbor classifier with different similarity measures are applied for classification to detect eyes. Three kinds of similarity measures are used to fully evaluate the performance of the DCA features. They are L_1 similarity measure δ_{L_1}, L_2 similarity measure δ_{L_2}, and cosine similarity measure δ_{cos}, which can be defined as follows:

$$\delta_{L_1}(\mathbf{X}, \mathbf{Y}) = \sum_i |\mathbf{X}_i - \mathbf{Y}_i| \tag{3.10}$$

Figure 3.3 Example eye strip images from the FRGC version 2 database with the spatial resolution of 55×128.The top three rows show the colored image, whereas the bottom three rows show the corresponding gray-scale image after illumination normalization.

$$\delta_{L_2}(\mathbf{X}, \mathbf{Y}) = (\mathbf{X} - \mathbf{Y})^t(\mathbf{X} - \mathbf{Y}) \tag{3.11}$$

$$\delta_{cos}(\mathbf{X}, \mathbf{Y}) = \frac{-\mathbf{X}^t\mathbf{Y}}{||\mathbf{X}||\ ||\mathbf{Y}||} \tag{3.12}$$

where $||\cdot||$ denotes the norm operator. Usually there are multiple detections around the pupil center. The average of these multiple detections is eventually chosen as the eye location.

3.3.2 Database

The experiments run on 12,776 Face Recognition Grand Challenge (FRGC) images from the FRGC version 2 database [56], [48]. Note that the FRGC images possess challenge properties, such as large variations in illumination, in skin color (white,

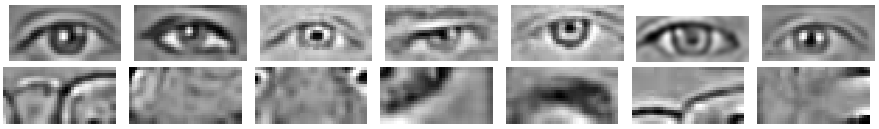


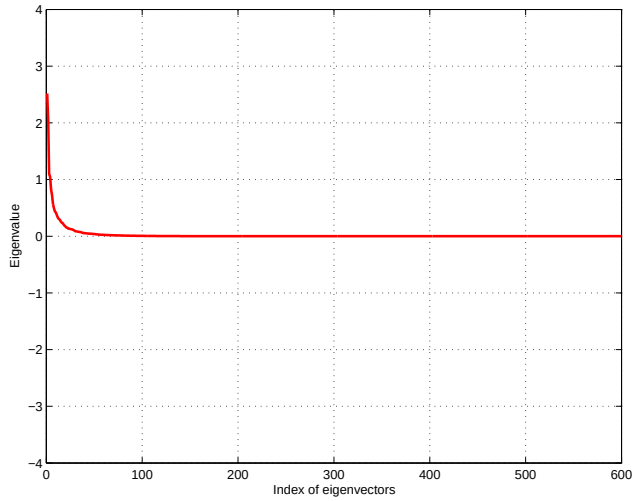
Figure 3.4 Example eye (the top row) and non-eye (the bottom row) training images that are normalized to 20×40. The images are preprocessed by illumination normalization.

yellow, and black), in facial expression (eyes open, partially open, or closed), as well as in scale and orientation. Additional challenges include eye occlusion caused by eye glasses or long hair, and the red eye effect due to the photographic effect. All these challenge factors increase the difficulty of accurate eye detection. Figure 3.3 shows some example eye strip images from the FRGC database with the spatial resolution of 55×128.

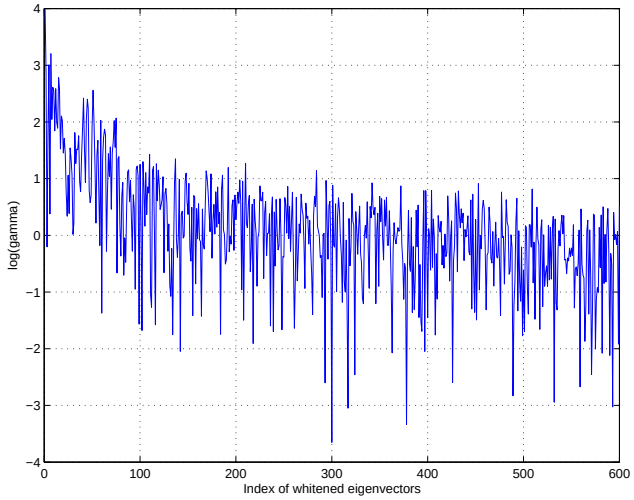
The training data collected from various sources contains 3,000 pairs of eyes and 12,000 non-eye patches in the experiments. The size of the eye and non-eye training images is normalized to 20×40. Figure 3.4 shows some example eye and non-eye training images after illumination normalization.

3.3.3 Effectiveness Evaluation of the DCA Method

The DCA method, different from the PCA method that defines the basis vectors as the eigenvectors corresponding to the largest eigenvalues, defines the basis vectors as the whitened eigenvectors corresponding to the largest values in the criterion vector γ in Equation 3.7. Given the 2D Haar wavelet image representation as the original image representation pattern vector, Figure 3.5 shows an example of the different selections of the basis vectors for the PCA method and the DCA method, respectively. Note that the 2D Haar basis functions for V^5 are used in this experiment. As $V^5 = V^0 \oplus W^0 \oplus W^1 \oplus W^2 \oplus W^3 \oplus W^4$, the length of the original 2D Haar pattern vector is $1,024$.



(a) PCA

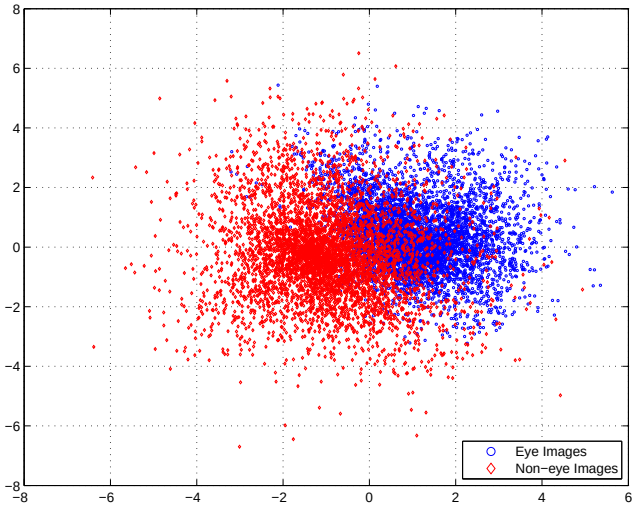


(b) DCA

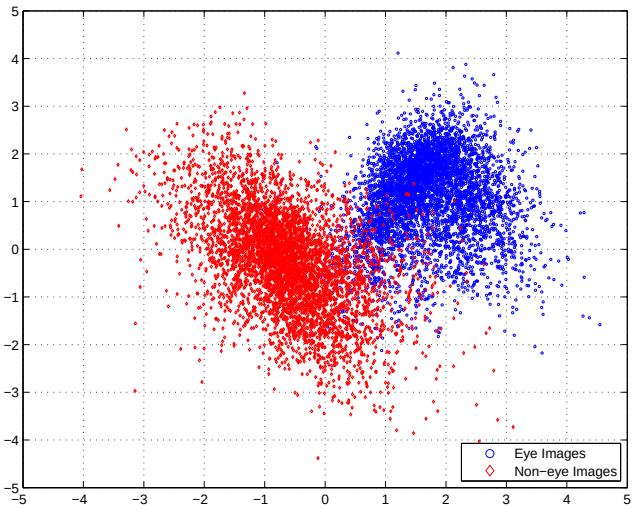
Figure 3.5 Different selection of the basis vectors for the PCA method and the DCA method, respectively.

Figure 3.5 reveals that the basis vectors for the PCA method always correspond to the lowest indexed eigenvectors as they correspond to the largest eigenvalues. The basis vectors for the DCA method, on the other hand, do not depend on such a natural order, as they are chosen based on the value of the elements in the criterion vector γ.

As discussed in Section 3.2, the DCA features preserve better discriminating capability than the PCA features. If only the most two significant basis vectors are



(a) PCA



(b) DCA

Figure 3.6 The distribution of the eye and non-eye training images using two most significant PCA features (a) and two most significant DCA features (b), respectively.

taken into consideration, the distribution of the eye and non-eye training images can be visualized in the 2D space spanned by these two basis vectors so as to give an intuitive view of the discriminatory power of the PCA features and the DCA features. Figure 3.6(a) and Figure 3.6(b) show the distributions of the eye and non-eye training images using the most tow significant PCA features and the most two significant DCA features, respectively. Figure 3.6(a) reveals that the two classes (eye and non-eye) are

Table 3.1 Parameter Settings and the Feature Size of the PCA and the DCA Features

image representation	feature size			comments
	original	PCA	DCA	
gray-scale	800	80	80	The intensity values of the gray-scale image are used.
YCbCr color	2,400	120	120	The three color component images Y, Cb, and Cr are used.
Haar	1,024	80	80	32×32 2D Haar wavelets at four scales are used.
HOG	1,296	80	80	1-D centered derivative $[-1, 0, 1]$ is used to compute the gradients. The size of each cell is 4×4 pixels and the histogram is evenly divided into 6 bins over $0° - 180°$. Each block contains 3×3 cells and blocks are overlapped with each other by two-thirds in a sliding fashion. L2 normalization is used for block normalization scheme.
LBP	472	80	80	The detection window (or training image) is evenly divided into four non-overlapped regions. $LBP^{u2}_{8,1}$ is applied to each region.

highly overlapped with each other when the PCA method are used, whereas Figure 3.6(b) reveals that the two classes are only slightly overlapped with each other when the DCA method are used. The distributions of the eye and non-eye training images

in Figure 3.6(a) and Figure 3.6(b) thus indicate that the DCA method derives more discriminatory features than the PCA method.

3.3.4 Comparative Assessment of the PCA and DCA Features

This subsection comparatively assesses the detection performance of the five types of PCA and DCA features derived from the five image representations: the gray-scale image representation, the YCbCr color image representation, the 2D Haar wavelet image representation, the HOG image representation, and the LBP image representation. The detection performance is evaluated using the normalized eye detection error [37], which is defined as the detection pixel error normalized by the interocular distance. For fair comparison, the size of DCA features are set equal to that of PCA features. The parameter settings and the size of the five types of PCA and DCA features are shown in Table 3.1.

Figure 3.7 — Figure 3.11 show the eye detection performance of these five types of PCA and DCA features, respectively. The top figures show the eye detection rate versus the normalized eye detection error. The bottom figures show the distribution of the eye detection pixel errors when the optimal similarity measure is applied to the PCA and the DCA methods. Note that in the bottom figures the more eye detections with small pixel errors, the more accurate the corresponding eye detection method is. For example, the top figure in Figure 3.9 shows that DCA Haar features derive better eye detection rate than that of the PCA Haar features regardless of the similarity measures used; the bottom figure in Figure 3.7 shows that the PCA Haar features derive the best eye detection results using the L_1 similarity measure, whereas the DCA Haar features derive the best eye detection results using the L_2 similarity measure. Furthermore, the bottom figure also shows that the DCA Haar features generate more detections than the PCA Haar features with small pixel errors, which

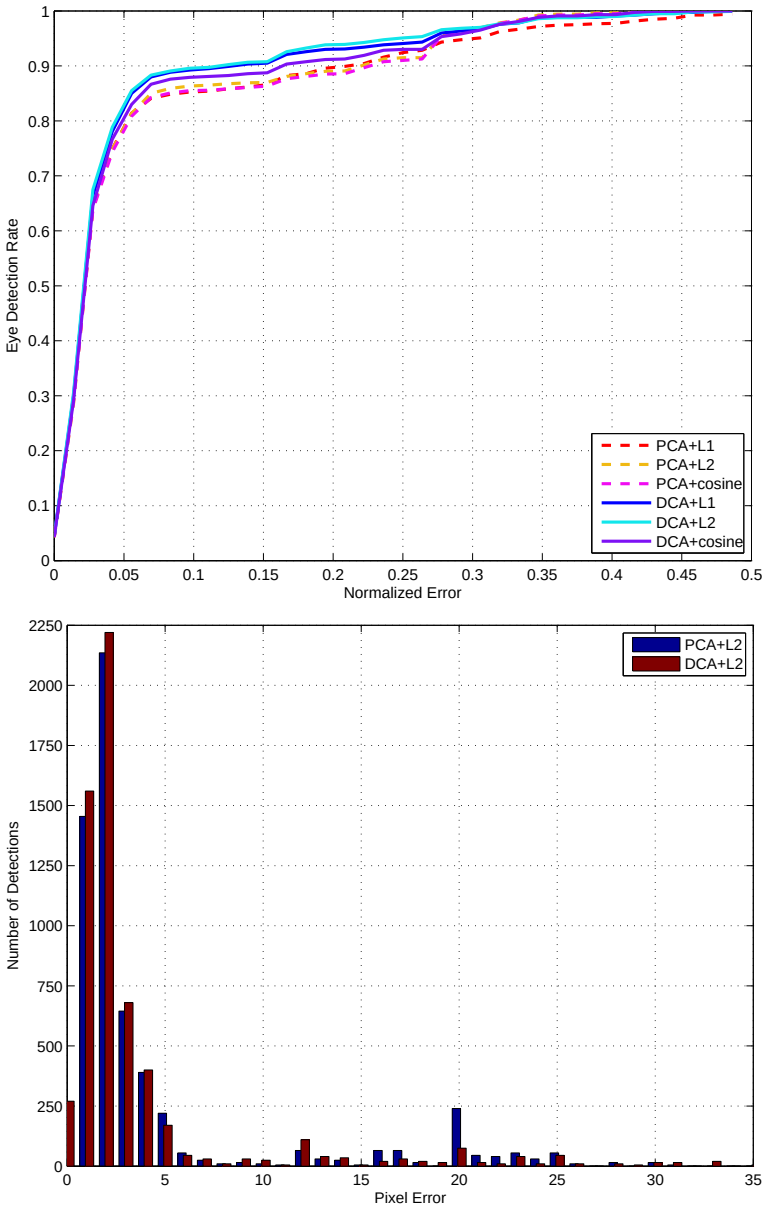


Figure 3.7 Eye detection performance of the PCA and the DCA gray-scale features, respectively.

indicates that the DCA Haar features achieves better detection accuracy than the PCA Haar features.

Specifically, Table 3.2 lists the average pixel errors and the eye detection rate for these five types of the PCA features and the DCA features, respectively. The mean and the standard deviation of the absolute errors in the X and the Y coordinates as

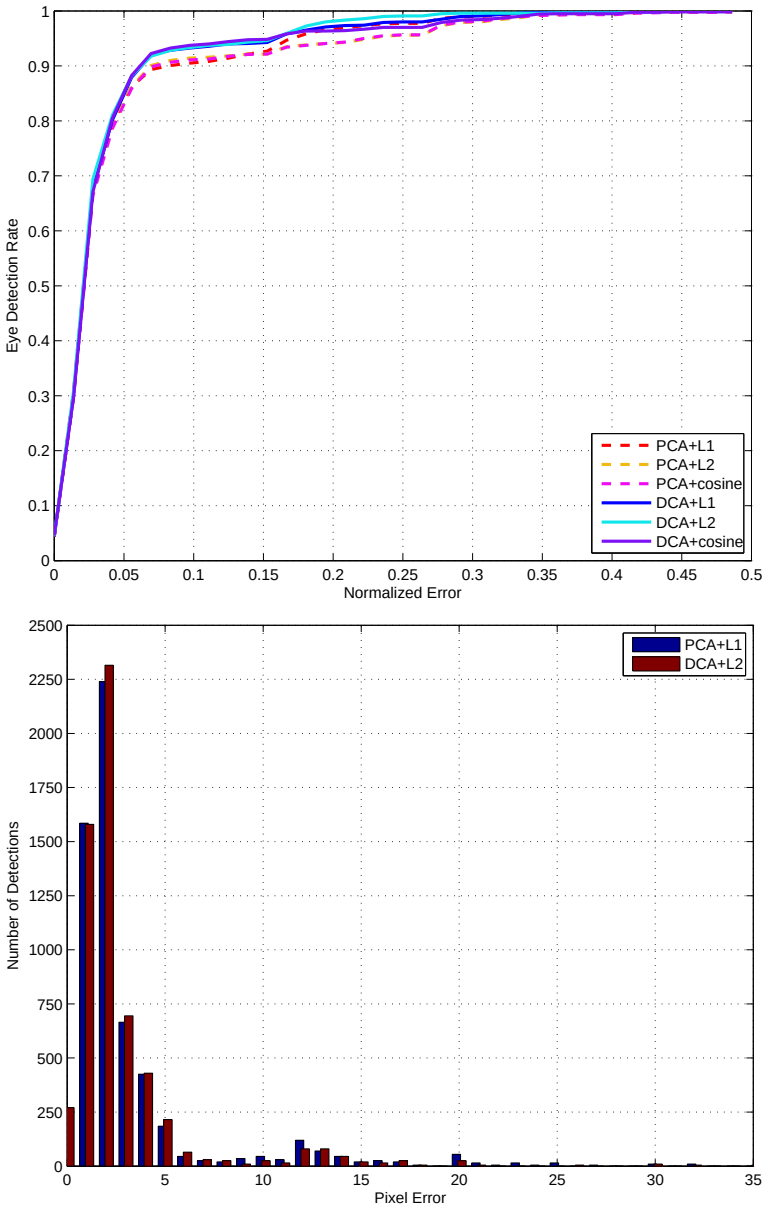


Figure 3.8 Eye detection performance of the PCA and the DCA YCbCr color features, respectively.

well as the mean of the errors in the Euclidean distance are listed in the table. Note that the detection rate shown in Table 3.2 represents the percentage of the correct detections within five pixels of the ground truth.

Figure 3.7 — Figure 3.11, and Table 3.2 reveal that the performance of the DCA features is consistently better than that of the PCA features regardless of the image

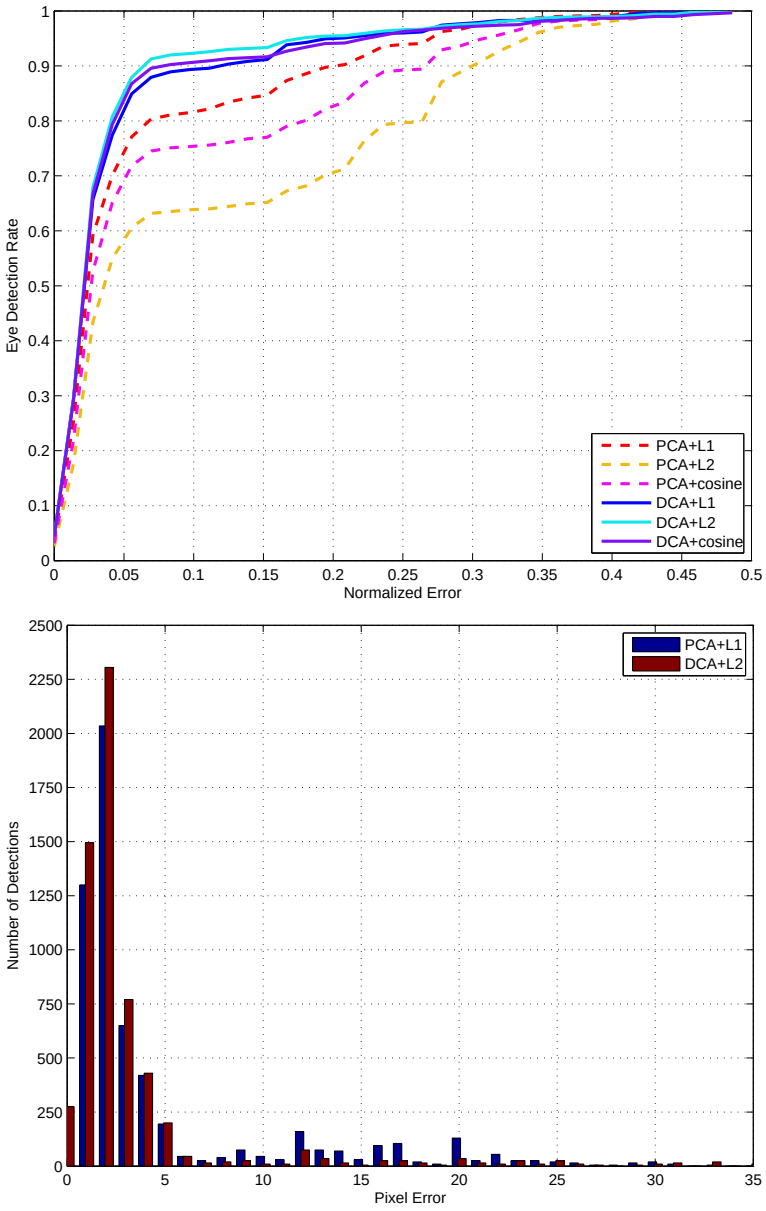


Figure 3.9 Eye detection performance of the PCA and the DCA Haar features, respectively.

representations and similarity measures used. For the gray-scale features, the YCbCr color features, the HOG features, and the LBP features, the DCA method just slightly improves the eye detection performance, since the performance by the PCA method already reaches a very high level. However, for the Haar features, the DCA method significantly improves the eye performance upon the PCA method. Take the Haar

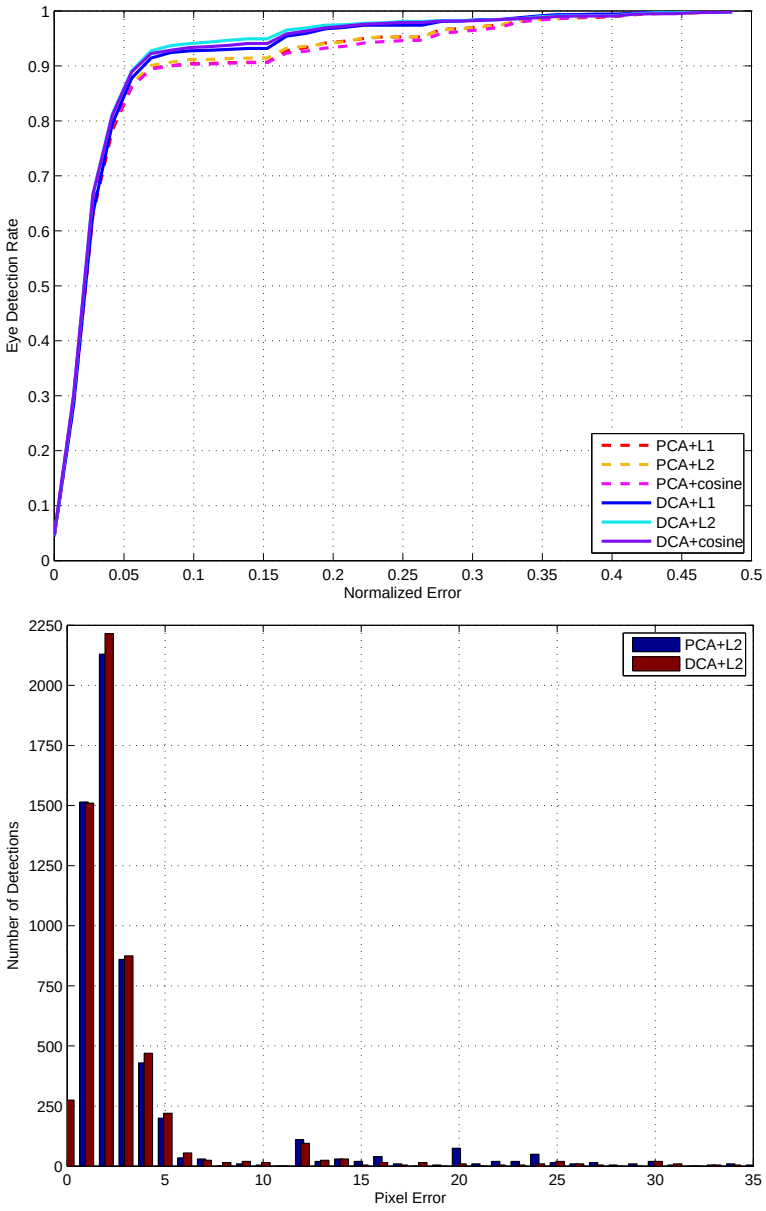


Figure 3.10 Eye detection performance of the PCA and the DCA HOG features, respectively.

features for an example. The DCA Haar features, as indicated in Table 3.2, improve the detection rate of the PCA Haar features by 7.59% using the L_1 measure, 28.08% using the L_2 measure, and 15.08% using the cosine measure, respectively. Regarding the eye detection accuracy, the DCA Haar features reduce the average eye detection

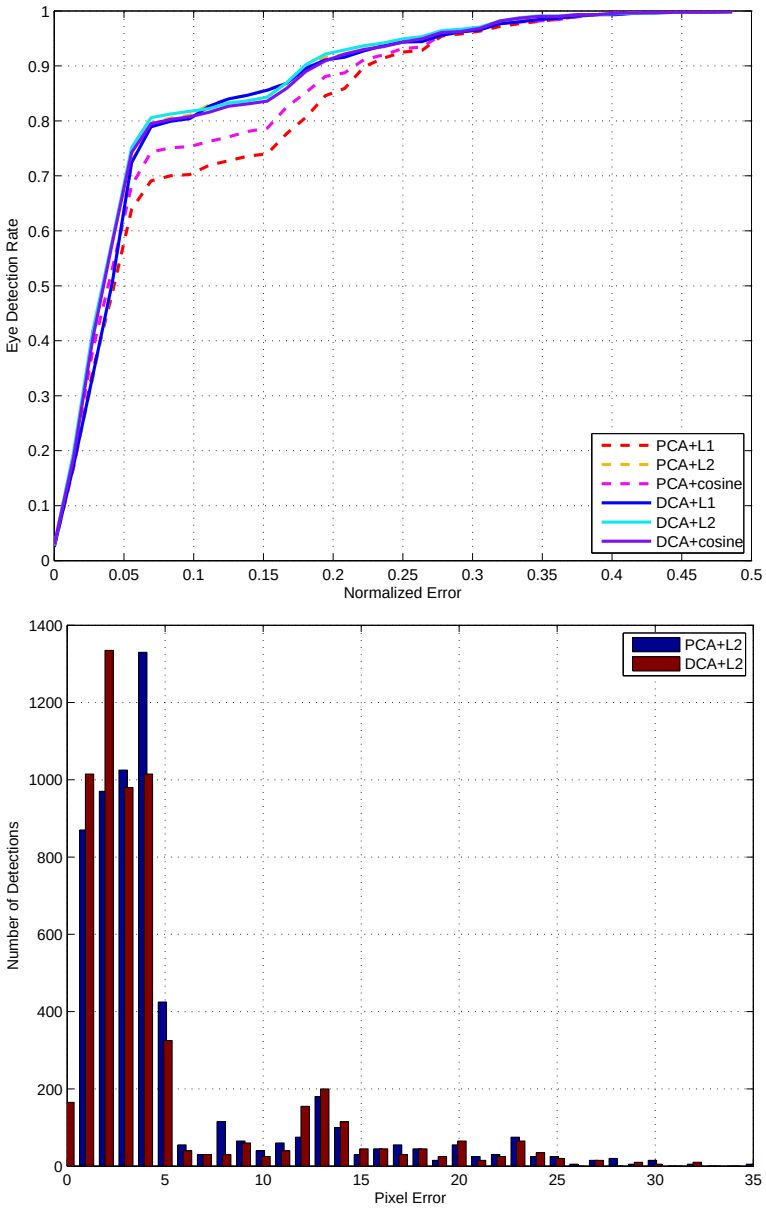


Figure 3.11 Eye detection performance of the PCA and the DCA LBP features, respectively.

error of the PCA Haar features by 1.1 pixels using the L_1 measure, 5.13 pixels using the L_2 measure, and 2.46 pixels using the cosine measure, respectively.

Figure 3.7, Figure 3.8, Figure 3.9, Figure 3.10, Figure 3.11, and Table 3.2 further reveal that the DCA HOG features achieve the best eye detection performance,

Table 3.2 Performance Comparison under Different Similarity Measures (ED Stands for the Euclidean Distance and DR Stands for the Detection Rate)

Features	Method	mean(x)	std(x)	mean(y)	std(y)	mean(ED)	DR
gray-scale	PCA+L1	3.22	4.89	2.64	5.89	4.87	84.08%
	PCA+L2	2.47	2.69	2.94	6.17	4.58	85.00%
	PCA+cosine	2.62	3.17	2.95	6.16	4.71	84.25%
	DCA+L1	2.68	3.51	2.05	5.11	4.02	88.00%
	DCA+L2	2.64	3.50	1.98	4.90	3.91	88.33%
	DCA+cosine	2.50	2.74	2.58	5.86	4.30	86.67%
color	PCA+L1	2.56	3.05	1.44	3.51	3.42	89.33%
	PCA+L2	2.26	2.31	2.11	5.03	3.68	90.08%
	PCA+cosine	2.27	2.27	2.13	5.07	3.71	89.83%
	DCA+L1	2.37	2.43	1.39	3.42	3.19	92.00%
	DCA+L2	2.41	2.68	1.15	2.83	3.06	91.75%
	DCA+cosine	2.24	2.24	1.65	4.20	3.29	92.25%
Haar	PCA+L1	3.09	3.77	2.83	5.56	4.94	80.33%
	PCA+L2	3.57	4.50	6.71	8.81	8.65	63.17%
	PCA+cosine	3.04	3.66	4.46	7.40	6.31	74.50%
	DCA+L1	2.67	3.33	1.85	4.70	3.84	87.92%
	DCA+L2	2.48	3.23	1.67	4.34	3.52	91.25%
	DCA+cosine	2.71	3.90	1.92	4.74	3.85	89.58%
HOG	PCA+L1	2.68	3.34	1.99	4.90	3.91	89.58%
	PCA+L2	2.58	3.29	1.93	4.81	3.79	90.08%
	PCA+cosine	2.69	3.58	2.02	4.87	3.93	89.42%
	DCA+L1	2.51	2.98	1.52	3.92	3.43	91.42%
	DCA+L2	2.37	2.70	1.36	3.75	3.19	92.75%
	DCA+cosine	2.45	2.88	1.41	3.88	3.29	92.25%
LBP	PCA+L1	4.51	5.29	3.75	5.59	6.83	69.08%
	PCA+L2	3.87	4.68	3.06	4.58	5.58	79.58%
	PCA+cosine	4.11	5.02	3.29	5.20	6.12	74.33%
	DCA+L1	3.83	4.60	3.16	4.68	5.62	78.92%
	DCA+L2	3.83	4.78	2.55	4.36	5.25	80.58%
	DCA+cosine	4.06	5.03	2.53	4.32	5.43	79.50%

followed in order by the DCA YCbCr color features, the DCA Haar features, the DCA gray-scale features, and the DCA LBP features.

3.4 Conclusion

This chapter presents a Discriminant Component Analysis (DCA) method, which improves upon the popular Principal Component Analysis (PCA) method, to extract discriminatory features for eye detection. The DCA method starts with a PCA followed by a whitening transformation. A discriminant analysis is then performed on the whitened PCA space. A set of DCA basis vectors, based on the novel definition of the cluster-measure vector and the separation-measure vector, as well as a new criterion vector, is defined through this analysis. This chapter then apply the DCA method to derive five types of the DCA features from five different image representations introduced in Chapter 2. Experiments on the FRGC version 2 database show that the DCA method is able to improve the discriminatory power of the PCA method and hence improves the eye detection performance. Furthermore, the experimental results also reveal that the DCA HOG features achieve the best eye detection performance, followed in order by the DCA YCbCr color features, the DCA Haar features, the DCA gray-scale features, and the DCA LBP features.

CHAPTER 4

CLUSTERING-BASED DISCRIMINANT ANALYSIS

Fisher linear discriminant (FLD) [30] [29] is a popular tool of discriminant analysis for feature extraction and classification. Since it was introduced, a number of its variants have been proposed and widely used in numerous fields of pattern recognition [49] [50] [78] [7] [57] [80] [90] [17] [93] [89]. However, the FLD and most of its variants shares a major disadvantage that they may not be able to extract adequate features in order to achieve satisfactory performance, especially for two class problems. This is caused by the property that the between-class scatter matrices S_b of the FLD and its variants are generally not full rank. For any L class problem, the FLD can only derive at most $L-1$ valid features. Thus, for two-class problems, the FLD can only derive a single valid feature, which is significantly inadequate for achieving satisfactory performance.

To address this problem, this chapter proposes three clustering-based discriminant analysis (CDA) models. The first CDA model, CDA-1, divides each class into a number of clusters by means of the k-means clustering technique. In this way, a new within-cluster scatter matrix S_w^c and a new between-cluster scatter matrix S_b^c are defined. The rank of the S_b^c increases as the number of clusters increases, and therefore the CDA-1 can derive adequate features for achieving satisfactory performance. The CDA-1 works well especially when inherent multi-models are presented in each class and the k-means clustering technique can properly identify the clusters. Take the task of eye detection as an example. It requires to differentiate between the eye class and the non-eye class, i.e. "the rest of the world". On one hand, the non-eye class indeed involves multi-models to represent different objects and scenes in "the rest of the world"; on the other hand, the eye class may contains multi-models as well, to represent different kinds of eyes such as open eyes, closed eyes, eyes with glasses, etc.

Motivated by the work of nonparametric discriminant analysis (NDA) in [31], this chapter further proposes another two CDA models, CDA-2 and CDA-3. The fundamental of the CDA-2 and CDA-3 is a clustering-based nonparametric form of the between-cluster scatter matrices N-S_b^c. These between-cluster scatter matrices N-S_b^c are full rank, and consequently both CDA-2 and CDA-3 can derive adequate features for classification. Furthermore, the nonparametric nature of the between-cluster scatter matrices inherently leads to the derived features that preserve the structure important for classification. The difference between CDA-2 and CDA-3 is that the former computes the between-cluster matrix N-S_b^c on a local basis whereas the latter computes the between-cluster matrix N-S_b^c on a global basis.

This chapter then evaluates these three CDA models on the problem of eye detection. Experiments on the Face Recognition Grand Challenge (FRGC) database and on the BioID face database [37] show the feasibility of the proposed three CDA models and the improved performance over some state-of-the-art eye detection methods.

4.1 Background

The principle of discriminant analysis is to find an optimal linear projection that is effective for reducing the feature dimensionality and preserving the class separability. Fisher linear discriminant (FLD) [30] [29] is a popular tool of discriminant analysis. The FLD uses the within-class and between-class scatter matrix to formulate a criteria of the class separability. The FLD projection is then defined to maximize the criteria.

Let $\mathbf{X}$ denote the feature vector and L denote the number of classes. Let $\omega_i, i = 1, 2, \cdots, L$ and $N_i, i = 1, 2, \cdots, L$ denote the classes and the number of feature vectors within each class, respectively. Let $M_i, i = 1, 2, \cdots, L$ and M_0 be the means of the classes and the grand mean. The within-class scatter matrix shows the scatter

of feature vectors around their respective class mean vectors, which can be defined as follow:

$$S_w = \sum_{i=1}^{L} P(\omega_i) E\{(\mathbf{X} - M_i)(\mathbf{X} - M_i)^t | \omega_i\} \tag{4.1}$$

where $P(\omega_i)$ is a prior probability. The between-class scatter matrix shows the scatter of the class mean vectors around the grand mean, which can be defined as follow:

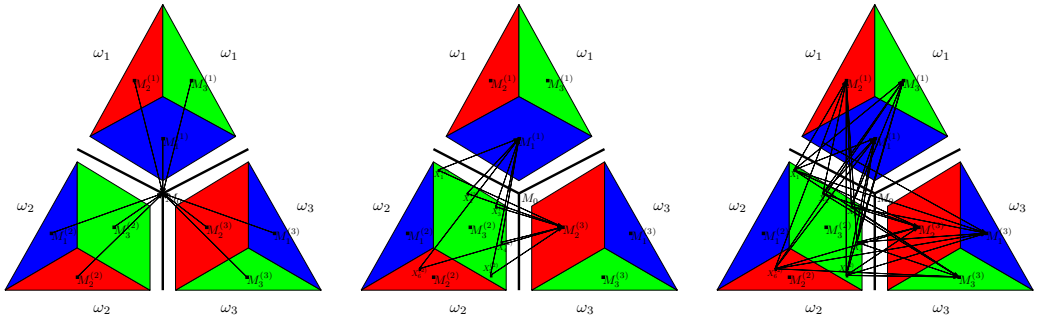
$$S_b = \sum_{i=1}^{L} P(\omega_i)(M_i - M_0)(M_i - M_0)^t \tag{4.2}$$

The FLD projection W is then defined to maximize the criteria as follow:

$$J(W) = \frac{|W^t S_b W|}{|W^t S_w W|} \tag{4.3}$$

and, mathematically, this criteria is maximized when W consists of the leading eigenvectors of $S_w^{-1} S_b$. Usually, to avoid the singularity of S_w, the principal component analysis (PCA) is applied before the FLD to reduce the high dimensional feature vector into a low dimensional one.

A major disadvantage of the FLD is that it may not be able to extract adequate features in order to achieve satisfactory performance, especially for two class problems. This is caused by the property that the between-class scatter matrix S_b of the FLD is generally not full rank. As indicated in Equation 4.2, the rank of S_b is at most $L-1$ for any L class problem, and consequently the rank of $S_w^{-1} S_b$ is at most $L-1$ as well. Therefore, there are at most $L-1$ valid eigenvectors of $S_w^{-1} S_b$, which means the FLD can only derive at most $L-1$ valid features for any L class problem. For two class problems, the FLD can only derive a single valid feature, which is significantly inadequate for achieving satisfactory performance.



(a) CDA-1 (b) CDA-2 (c) CDA-3

Figure 4.1 The between-cluster matrices of CDA-1, -2, and -3, respectively. The figure shows a three-class problem and each class is further divided into three clusters. M_0 represents the grand mean, whereas $M_q^{(p)}, p, q = 1, 2, 3$, represents the mean vector of the qth cluster from class ω_p. $\mathbf{X}_j^{(2)}, j = 1, 2, \cdots, 6$, represents six data samples from class ω_2. (a) The between-cluster scatter matrix of CDA-1 measures the scatter of the mean vector from each cluster with respect to the grand mean. (b) The between-cluster scatter matrix of CDA-2 measures the scatter of each feature vector from one class with respect to the mean vector of its nearest cluster from otherwise classes. (c) The between-cluster scatter matrix of CDA-3 measures each feature vector from one class with respect to mean vectors of all the clusters from otherwise classes.

Fukunaga [31] initiated the study on nonparametric discriminant analysis (NDA) to address this problem. The NDA maintains the within-class scatter matrix S_w the same with that of the FLD, but defines a nonparametric form of the between-class scatter matrix S_b using the nearest neighbor techniques. Since the NDA was introduced, a number of its variants have been proposed [44] [9] [45] [72] [32] [73]. All of these variants consistently follow the idea of the nearest neighbors to define their NDAs.

4.2 Clustering-based Discriminant Analysis

This section presents three clustering-based discriminant analysis (CDA) models, CDA-1, CDA-2, and CDA-3, to address the problem of inadequate features derived from the FLD.

4.2.1 CDA-1 Model

The CDA-1 model divides each class into a number of clusters by means of the k-means clustering technique. In this way, a new within-cluster scatter matrix S_w^c and a new between-cluster scatter matrix S_b^c are defined. The rank of the S_b^c increases as the number of clusters increases, and therefore the CDA-1 can derive sufficient valid features for achieving satisfactory performance.

Formally, the CDA-1 first uses the k-means clustering technique to divide the feature vectors from each class into k clusters so as to minimize the within-cluster sum of squares. The algorithm of the k-means clustering applying to the feature vectors within each class is described in Algrithm 2 in detail.

After the mean vector of each cluster from every class is derived, the new formulation of the within-class scatter matrix S_w^C is defined as follow:

$$S_w^c = \sum_{p=1}^{L}\{P(\omega_p)\sum_{j=1}^{k}\frac{|C_j^{(p)}|}{N_p}E\{(\mathbf{X}-M_j^{(p)})(\mathbf{X}-M_j^{(p)})^t|\omega_p\}\} \tag{4.4}$$

and the new formulation of the between-class scatter matrix S_b^C (Figure 4.1(a)) is defined as follow:

$$S_b^c = \sum_{p=1}^{L}\{P(\omega_p)\sum_{j=1}^{k}\frac{|C_j^{(p)}|}{N_p}(M_j^{(p)}-M_0)(M_j^{(p)}-M_0)^t\} \tag{4.5}$$

where $|C_j^{(p)}|$ denotes the number of feature vectors in the jth cluster of the class ω_p, and $M_j^{(p)}$ denotes the mean vector of the jth cluster of the class ω_p. Note that as k decreased to one, $M_j^{(p)}$ converges to M_i. Thus, Equation 4.4 and Equation 4.5 is a generalization of Equation 4.1 and Equation 4.2, respectively.

There are two advantages of the CDA-1 model. First, the rank of the S_b^c is increased compared with S_b. Recall that the rank of the S_b of the FLD is upper bound

Algorithm 2 The k-means clustering algorithm applying to the feature vectors within each class.

Input: the feature vectors $\mathbf{X}_j^{(i)}$, which denotes the jth feature vector in class ω_i, where $i = 1, 2, \cdots, L$, and $j = 1, 2, \cdots, N_i$.

Output: the mean vectors $M_p^{(i)}$, which denotes the mean vector of pth cluster in class ω_i, where $i = 1, 2, \cdots, L$, and $p = 1, 2, \cdots, k$.

FOR $i = 1, 2, \cdots, L$

- $t = 0$.
- Let $C_{p,t}^{(i)}, p = 1, 2, \cdots, k$, denotes the set of feature vectors from the pth cluster in class ω_i in the tth iteration. First, randomly assign each feature $X_j^{(i)}$ to a set $C_{p,t}^{(i)}$.
- Initialization: $M_{p,t}^{(i)} = \frac{1}{|C_{p,t}^{(i)}|} \sum_{X_j^{(i)} \in C_{p,t}^{(i)}} X_j^{(i)}$.
- REPEAT
  - Reassignment Step: reassign each feature vector to the cluster with the closest mean as follow:

    $$C_{p,t+1}^{(i)} = \{X_j^{(i)} : ||X_j^{(i)} - M_{p,t}^{(i)}|| \leq ||X_j^{(i)} - M_{p^*,t}^{(i)}|| \text{ for all } p^* = 1, 2, \cdots, k.\}$$

  - Update Step: calculate the new means to be the centroids of the feature vectors in the clusters as follow:

    $$M_{p,t+1}^{(i)} = \frac{1}{|C_{p,t+1}^{(i)}|} \sum_{X_j^{(i)} \in C_{p,t+1}^{(i)}} X_j^{(i)}.$$

  - $t = t + 1$
- UNTIL the algorithm converges when $M_{p,t}^{(i)}$ is unchanged.
- $M_p^{(i)} = M_{p,t}^{(i)}$.

END FOR

by $L-1$ for any L class problem and thus at most $L-1$ features can be derived. In comparison, the rank of the S_b^c is upper bound by $k \times L - 1$. The S_b^c can even be full rank when enough large k is set. Therefore, more features can be derived for classification and the performance will be improved. Second, the clustering algorithm can find inherent multi-models in each class and further improve the performance. Take the task of eye detection as an example. It requires to differentiate between the eye class and the non-eye class, i.e. "the rest of the world". On one hand, the non-eye class indeed involves multi-models to represent different objects and scenes in "the rest of the world"; on the other hand, the eye class may contains multi-models as well, to represent different kinds of eyes such as open eyes, closed eyes, eyes with glasses, etc. The CDA-1 features are able to preserve the class separability among these multi-models and thus achieve better performance as indicated in Section 4.3.

After computing S_w^c and S_b^c, the CDA-1 project matrix is the leading eigenvectors of $(S_w^c)^{-1}S_b^c$. To avoid the singularity problem, PCA is first applied before the CDA-1. Furthermore, inspired by the enhanced Fisher linear discriminant model (EFM) in [49], the CDA-1 procedure is decomposed intoa simultaneous diagonalization of the two within-cluster and between-cluster scatter matrices to improve the generalization performance of the CDA-1. The simultaneous diagonalization is stepwise equivalent to two operations as pointed out by Fukunaga [30]: whitening the within-cluster scatter matrix and applying PCA to the between-cluster scatter matrix using the transformed data. The CDA-1 should preserve a proper balance, during the stepwise process, between the need that the selected eigenvalues account for most of the spectral energy of the raw data (for representational adequacy), and the requirement that the eigenvalues of the within-class scatter matrix (in the reduced PCA space) are not too small (for better generalization performance) [49].

Finally, the detailed algorithm of the CDA-1 is given as follows:

1. Compute the PCA features as: $\mathbf{P} = \Phi^t \mathbf{X}$, where Φ is the leading eigenvectors of the mixture scatter matrix of $\mathbf{X}$.

2. In the feature space $\mathbf{P}$, follow the Algrithm 2 to divide each classes into k clusters and compute the mean vector of each cluster.

3. In the feature space $\mathbf{P}$, compute the S_w^c as indicated by Equation 4.4.

4. Whiten the S_w^c as follows:

$$\begin{gathered} S_w^c \Psi = \Psi \Lambda \quad \text{and} \quad \Psi^t \Psi = I \\ \Lambda^{-1/2} \Psi^t S_w^c \Psi \Lambda^{-1/2} = I \end{gathered} \tag{4.6}$$

   where Ψ and Λ are the eigenvectors and the diagonal eigenvalue matrices of S_w^C, respectively. Then compute the whitening transformed features $\mathbf{Y}$ with respect to S_w^c as follow:

$$\mathbf{Y} = \Lambda^{-1/2} \Psi^t \mathbf{P} \tag{4.7}$$

5. In the feature space Y, compute the S_b^c as indicated by Equation 4.5.

6. Diagonalize the S_b^c as follows:

$$S_b^c \Theta = \Theta \Gamma \quad \text{and} \quad \Theta^t \Theta = I \tag{4.8}$$

   where Θ and Γ are the eigenvectors and the diagonal eigenvalue matrices of S_b^c, respectively. Then the CDA-1 features $\mathbf{Z}$ are now defined as follows:

$$\mathbf{Z} = \Theta^t \mathbf{Y} \tag{4.9}$$

   Finally, the overall transformation matrix of the CDA-1 can be defined as:

$$T = \Phi \Psi \Lambda^{-1/2} \Theta \tag{4.10}$$

4.2.2 CDA-2 Model

The CDA-1 model significantly increases the number of the derived features, but the number is still upper bound by $k \times L - 1$. Even though the S_b^c can be full rank when enough large k is set, it takes the risk of impairing the inherent multi-models in each class. Motivated by the work of nonparametric discriminant analysis (NDA) in [31], this subsection further proposes the CDA-2 model. The fundamental of the CDA-2 model is a clustering-based nonparametric form of the between-cluster scatter matrix N-S_b^c. The N-S_b^c is full rank, and consequently CDA-2 can derive adequate features for classification. Furthermore, the nonparametric nature of the between class scatter matrix inherently leads to the derived features that preserve the structure important for classification.

Specifically, the between-class scatter matrix N-S_b^c of the CDA-2 is given on a local basis, which measures the scatter of each feature vector from one class with respect to the mean vector of its nearest cluster from otherwise classes (Figure 4.1(b)). The N-S_b^c is defined as:

$$N\text{-}S_b^c = \sum_{i=1}^{L} \frac{P(\omega_i)}{N_i} \sum_{\substack{p=1 \\ p \neq i}}^{L} \sum_{j=1}^{N_i} \mathrm{w}^{(p)}(\mathbf{X}_j^{(i)})(\mathbf{X}_j^{(i)} - M^{(p)}(\mathbf{X}_j^{(i)}))(\mathbf{X}_j^{(i)} - M^{(p)}(\mathbf{X}_j^{(i)}))^t \tag{4.11}$$

where $M^{(p)}(\mathbf{X}_j^{(i)})$ denotes the mean vector of the nearest cluster from the class ω_p to the feature vector $\mathrm{X}_j^{(i)}$, and $\mathrm{w}^{(p)}(\mathbf{X}_j^{(i)})$ is a weighting function, which can be defined as:

$$\mathrm{w}^{(p)}(\mathbf{X}_j^{(i)}) = \frac{\min\{d^\alpha(\mathbf{X}_j^{(i)}, M^{(i)}(\mathbf{X}_j^{(i)})), d^\alpha(\mathbf{X}_j^{(i)}, M^{(p)}(\mathbf{X}_j^{(i)}))\}}{d^\alpha(\mathbf{X}_j^{(i)}, M^{(i)}(\mathbf{X}_j^{(i)})) + d^\alpha(\mathbf{X}_j^{(i)}, M^{(p)}(\mathbf{X}_j^{(i)}))} \tag{4.12}$$

where α is a control parameter between zero and infinity, and $d(\mathbf{X}_j^{(i)}, M^{(i)}(\mathbf{X}_j^{(i)}))$ and $d(\mathbf{X}_j^{(i)}, M^{(p)}(\mathbf{X}_j^{(i)}))$ denote the Euclidean distance from a feature vector $\mathbf{X}_j^{(i)}$ to the mean vector of its nearest cluster from itself class ω_i and from the otherwise class

ω_p, respectively. The value of the weighting function is close to 0.5 when the feature vector is near the class boundary and drops off to 0.0 as feature vector moves away from the boundary. This property allows the feature vectors near the class boundary, which preserve the classification structure, to contribute more to the between-class scatter matrix N-S_b^c. Note that α is set to 2 in the experiments for the CDA-2 model.

From Equation 4.11, following findings are observed. First, due to the application of the clustering technique, the CDA-2, as CDA-1 does, is capable of finding the inherent multi-models in each class and improve the performance for those case where the multi-models are indeed present.

Second, the CDA-2 can derive more features than the CDA-1. The CDA-2 makes use of all the feature vectors, in stead of only the cluster centroids, in the definition of the N-S_b^c. Thus, the N-S_b^c is full rank. More features can be derived for the classification, and the performance may be improved as more information is included.

Third, the CDA-2 is capable of effectively preserving the classification structure. Either the FLD or the CDA-1 defines the between-class(cluster) scatter matrix only based on the mean vectors of the classes(clusters). It fails to involve the feature vectors on the class boundary which preserve the classification structure. In comparison, the CDA-2 takes account of all the feature vectors, including those on the class boundary, to define the between-cluster scatter matrix. More importantly, considering that the feature vectors far from the class boundary may distort the classification structure, the CDA-2 introduces a weighting function to emphasize the effect of those feature vectors near to the class boundary but to de-emphasize the effect of those far from the class boundary. As indicated in Figure 4.1(b), the data samples $\mathbf{X}_1^{(2)}, \mathbf{X}_2^{(2)}, \cdots, \mathbf{X}_5^{(2)}$ fall on the class boundary and are given a bigger weight to emphasize their effect, whereas the data sample $\mathbf{X}_6^{(2)}$ is far away from the boundary and is given a smaller

weight to de-emphasize its effect. In this way, the CDA-2 effectively preserves the classification structure.

Please note that the CDA-2 has two major difference from the nearest neighbor based nonparametric discriminant analysis (NDA) in [31]. First, the NDA only works on the two class problems, whereas the CDA-2 works on multi-class (equal to or bigger than two) problems. Second, the between-class scatter matrix of the NDA measures the scatter of feature vectors with respect to the mean vectors of their k nearest neighbors. This kind of measurement takes the risk of suboptimal performance, since only the local scatters with a very small amount of feature vectors are utilized and much information is lost in the learning procedure. By contrast, the CDA-2 takes advantage of the inherent multi-models in each class. The between-cluster scatter matrix of the CDA-2 measures the scatter of feature vectors with respect to the mean vectors of their nearest clusters from otherwise classes. The CDA-2 serves as a better representation of the scatter of the feature vectors with respect to the otherwise clusters(classes).

Finally, the detailed algorithm of the CDA-2 is given as follows:

1. Compute the PCA features as: $\mathbf{P} = \Phi^t \mathbf{X}$, where Φ is the leading eigenvectors of the mixture scatter matrix of $\mathbf{X}$.

2. In the feature space $\mathbf{P}$, follow the Algrithm 2 to divide each classes into k clusters and compute the mean vector M of each cluster.

3. In the feature space $\mathbf{P}$, compute the S_w^c as indicated by Equation 4.4.

4. Whiten the PCA features and the mean vector of each cluster with respect to S_w^c as: $\mathbf{Y} = \Lambda^{-1/2}\Psi^t\mathbf{P}$ and $M' = \Lambda^{-1/2}\Psi^t M$, where Λ and Ψ are the eigenvalues and eigenvectors of S_w^c.

5. In the feature space Y, find the nearest cluster to each feature vector from the otherwise classes. Compute the $\mathrm{w}^{(p)}(\mathbf{X}_j^{(i)})$ as indicated by Equation 4.12.

6. In the feature space Y, compute $N\text{-}S_b^c$ as indicated by Equation 4.11.

7. The CDA-2 features are then defined as: $Z = \Theta^t \mathbf{Y}$, where Θ are the eigenvectors of the $N\text{-}S_b^c$.

4.2.3 CDA-3 Model

The between-cluster scatter matrix $N\text{-}S_b^c$ of the CDA-2 is defined on a local basis, which measures the scatter of each feature vector from one class with respect to the mean vector of its nearest cluster from otherwise classes. One limitation of this definition is that the $N\text{-}S_b^c$ of the CDA-2 just takes account of the nearest cluster to each feature vector but ignores the contributions of other clusters. However, the fact is that the different clusters may contribute differently when one measures the scatter of each feature vector. Therefore, if all clusters are taken into account when one measures the scatter of each feature vector, the between-cluster scatter matrix may preserve the classification structure from different points of view, and hence may improve the classification performance.

Inspired by this idea, the CDA-3 model is proposed based on a new formulation of the between-cluster scatter matrix $N\text{-}S_b^c$. The $N\text{-}S_b^c$ of the CDA-3 is defined on a global basis, which measures the scatter of each feature vector from one class with respect to the mean vectors of all the clusters from otherwise classes (Figure 4.1(c)). The $N\text{-}S_b^c$ of the CDA-3 is defined as follows:

$$N\text{-}S_b^c = \sum_{i=1}^{L} \frac{P(\omega_i)}{N_i} \sum_{\substack{p=1 \\ p \neq i}}^{L} \sum_{q=1}^{k} \sum_{j=1}^{N_i} \mathrm{w}_q^{(p)}(\mathbf{X}_j^{(i)})(\mathbf{X}_j^{(i)} - M_q^{(p)})(\mathbf{X}_j^{(i)} - M_q^{(p)})^t \tag{4.13}$$

where $M_q^{(p)}$ denotes the mean vector of the qth cluster from the class ω_p, and $\mathrm{w}_q^{(p)}(\mathbf{X}_j^{(i)})$ is a weighting function, which can be defined as:

$$\mathrm{w}_q^{(p)}(\mathbf{X}_j^{(i)}) = \frac{\min\{d^\alpha(\mathbf{X}_j^{(i)}, M_q^{(i)}), d^\alpha(\mathbf{X}_j^{(i)}, M_q^{(p)})\}}{d^\alpha(\mathbf{X}_j^{(i)}, M_q^{(i)}) + d^\alpha(\mathbf{X}_j^{(i)}, M_q^{(p)})} \tag{4.14}$$

where $d(\mathbf{X}_j^{(i)}, M_q^{(i)})$ and $d(\mathbf{X}_j^{(i)}, M_q^{(p)})$ denote the Euclidean distance from a feature vector $\mathbf{X}_j^{(i)}$ to the mean vector of the qth cluster from itself class ω_i and from the otherwise class ω_p, respectively. Note that α is set to 2 in the experiments for the CDA-3 model.

The CDA-3 possesses all the advantages of the CDA-2. It is capable of finding the inherent multi-models in each class. The between-cluster scatter matrix is full rank and thus can derive sufficient features for satisfactory performance. And it can effectively preserve the classification structure due to the introduction of the weighting function.

The algorithm to derive the CDA-3 features is similar with that of the CDA-2. It is not explicitly presented here.

4.3 Experiments

The CDA method is then applied to the problem of eye detection and the effectiveness of the three CDA models is fully evaluated in this section. Figure 4.2 shows the architecture of the CDA-based eye detection method, which is similar with the DCA-based eye detection method introduced in Section 3.3.1. This method integrates the 2D Haar wavelets for image representation, the CDA for discriminatory feature extraction, and the nearest neighbor rule with similarity measures for classification. Note that considering both the accuracy and efficiency performance, only 2D Haar wavelet image representation is used in this method. The training and testing data sets are the same as those introduced in Section 3.3.2.

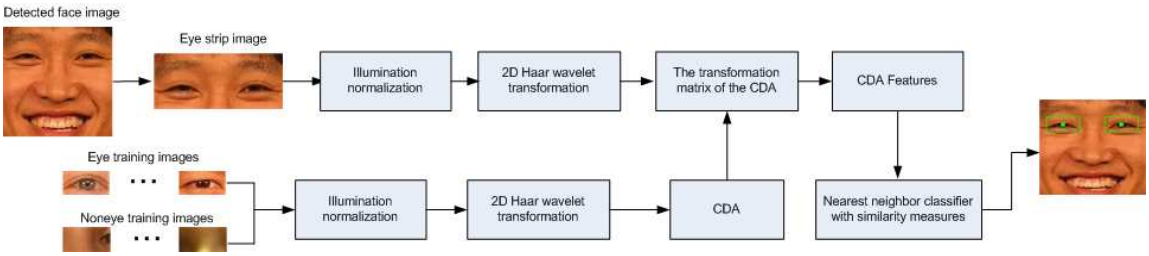


Figure 4.2 System architecture of the CDA-based eye detection method.

There are two key parameters involved in the CDA method: the size of extracted features (m) and the number of clusters (k). This section first evaluates the effect of these two parameters on the three CDA models. The evaluation of the best eye detection performance with the optimal parameters comes after the parameter evaluation. The comparison is made with the FLD and the NDA methods in order to show the performance improvement of the CDA method. Please note that when tuning the parameters only 600 FRGC images are used, but the final eye detection performance with the optimal parameters are given based on the whole FRGC database.

4.3.1 Evaluation of the Size of Extracted Features

The size of the original 2D Haar wavelet patten vectors in the experiments is 1,024. There are two points involving the size determination of the features in the process of the CDA: the size of the intermediate PCA features (m_1) and the size of the final CDA features (m_2). It is hard to exhaustively evaluate all the possible combinations of m_1 and m_2. Considering the aspect of the efficiency, m_1 is only evaluated in the range of 50 and 300. For simplicity, the maximum size of the final CDA features (m_2) that can be derived as the size of the PCA features varies is always chosen. Specifically, for CDA-1, m_2 is set equal to $2 \times k - 1$; for CDA-2 and CDA-3, m_2 is set equal to m_1. Please note that the number of clusters (k) is temporarily set to $m_1/2$

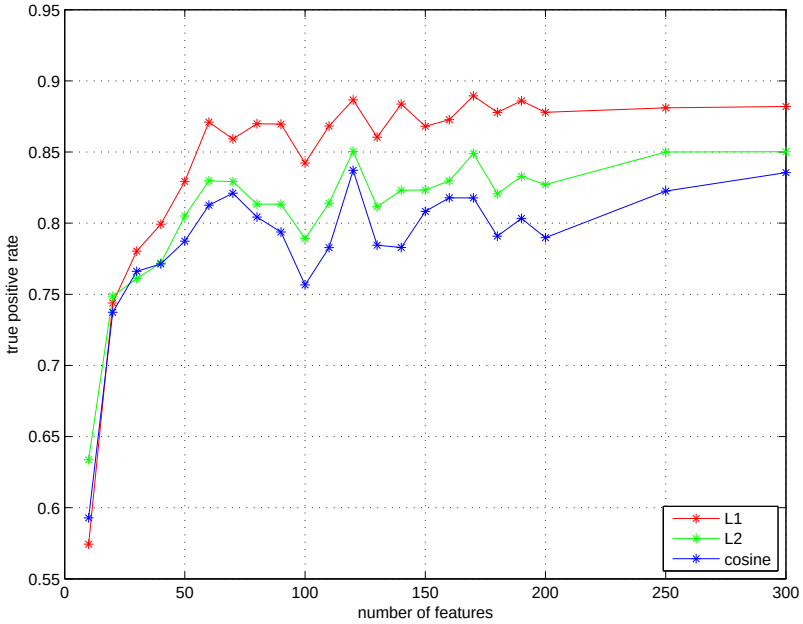


Figure 4.3 The detection performance of the CDA-1 as the size of features varies.

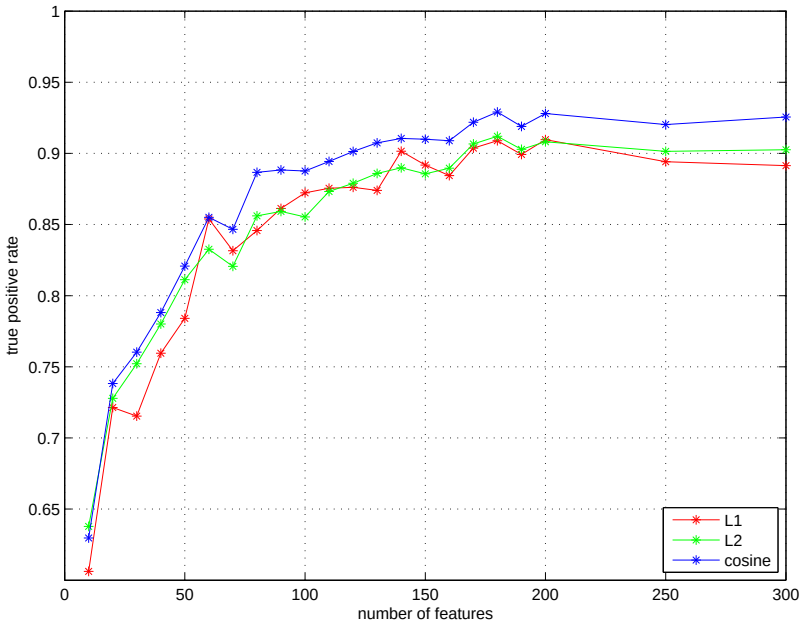


Figure 4.4 The detection performance of the CDA-2 as the size of features varies.

when evaluating the effect of the size of features on CDA. The complete evaluation of the effect of the number of clusters (k) on CDA will be given in the next subsection.

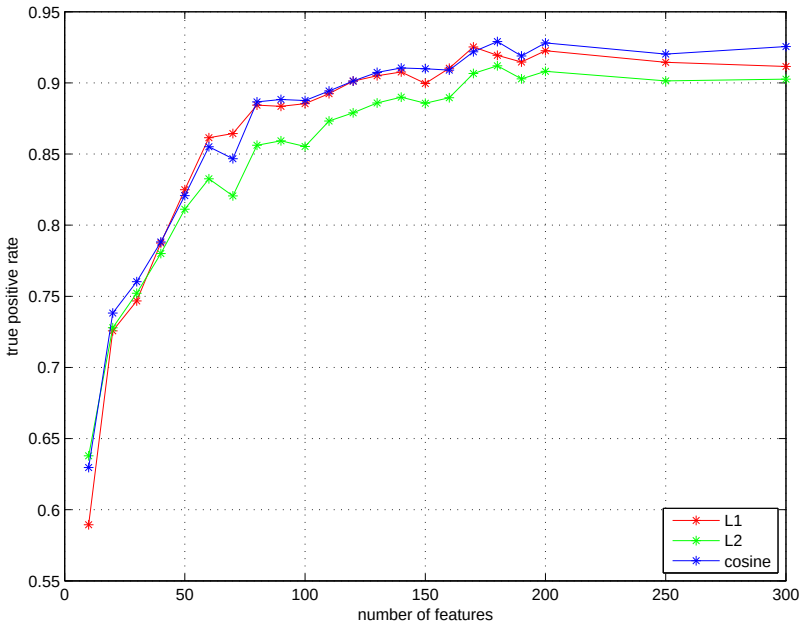


Figure 4.5 The detection performance of the CDA-3 as the size of features varies.

Figure 4.3 – Figure 4.5 show the performance comparison of the CDA-1, CDA-2, and CDA-3, respectively, as the m_1 and m_2 varies. Specifically, Figure 4.3 – Figure 4.5 show the true positive rate of the CDA-1, CDA-2, and CDA-3, respectively, at the false accept rate of 0.1 for the detection normalized error $e \leq 0.07$. Note that the normalized error of 0.07 is a significantly strict criteria, which can be considered that the detected eye center is inside the eye pupil.

Figure 4.3 – Figure 4.5 reveal that the detection performance is affected by the size of the derived features. The CDA-1 reaches the best performance by using 170 features and L_1 similarity measure, while both the CDA-2 and the CDA-3 reach the best performance by using 180 features and cosine similarity measure.

4.3.2 Evaluation of the Number of Clusters

The number of clusters (k) is evaluated between a small value ($k = 5$) and the half value of the size of PCA features ($k = m_1/2$). There are two reasons to just evaluate the number of clusters in above range: (i) the clusters inherently represent

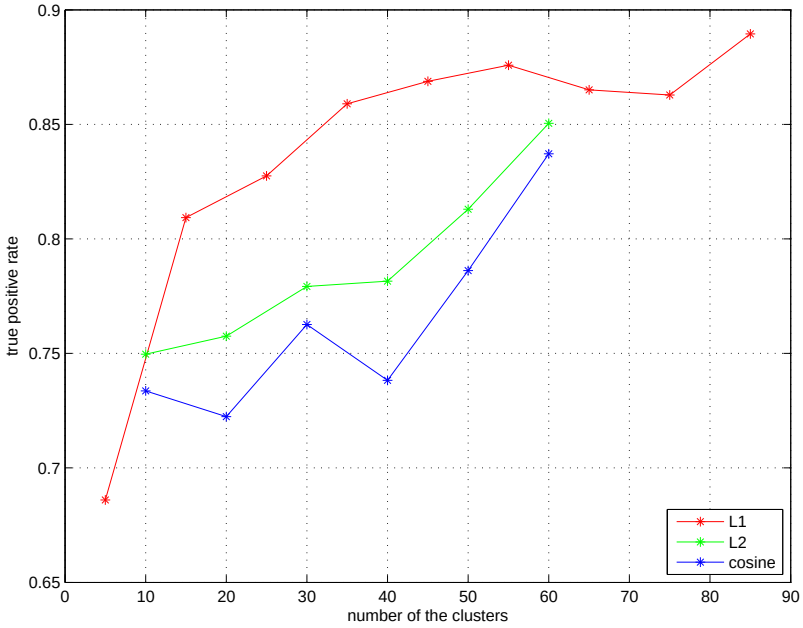


Figure 4.6 The detection performance of the CDA-1 as the number of clusters varies.

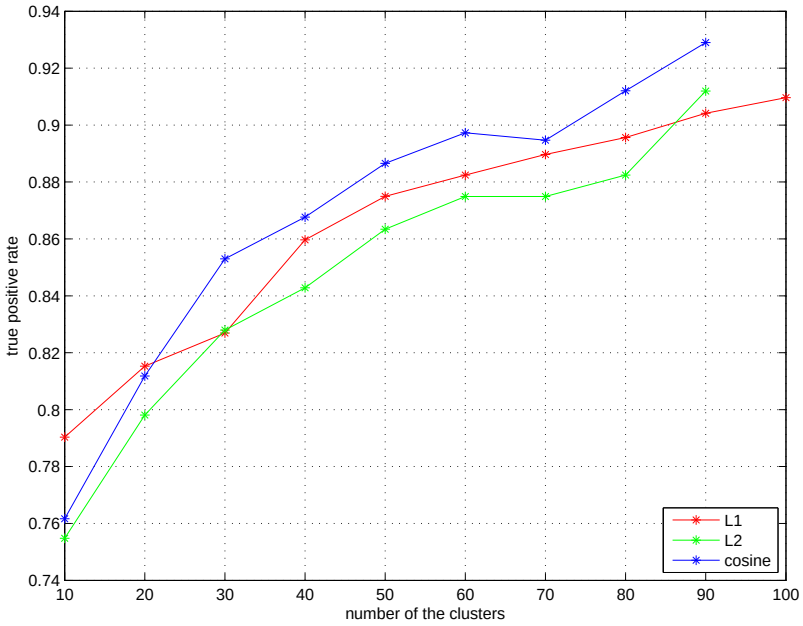


Figure 4.7 The detection performance of the CDA-2 as the number of clusters varies.

the multi-models of each class, and the number of the multi-models of each class should be neither too small nor too large; and (ii) for a two class problem, the

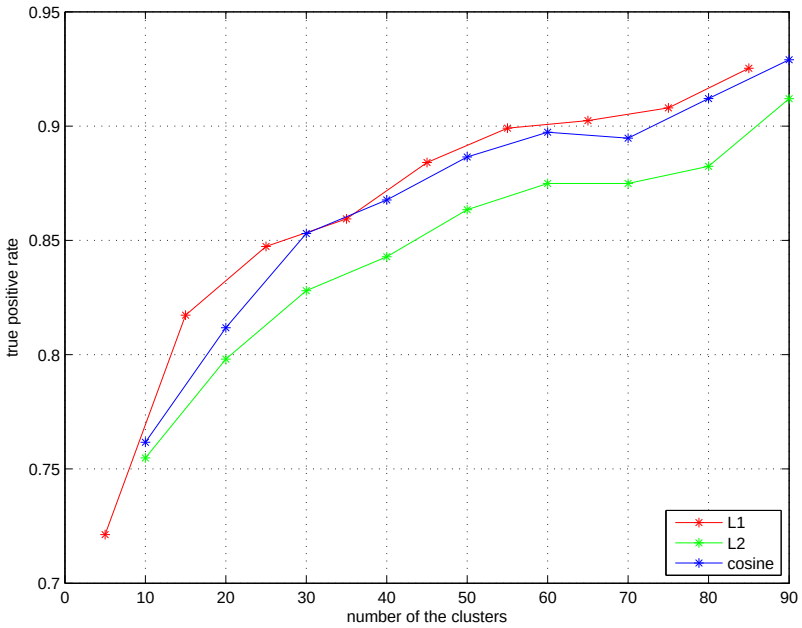


Figure 4.8 The detection performance of the CDA-3 as the number of clusters varies.

within-cluster matrix of CDA-1 is upper bound by m_1 in terms of Equation 4.4, whereas the between-cluster matrix is upper bound by the smaller value of m_1 and $2 * k - 1$ in terms of Equation 4.5; both the within-cluster and the between-cluster matrices of CDA-2 and CDA-3 are upper bound by m_1 in terms of Equations 4.4, 4.11, and 4.13. Therefore, even if more clusters ($k > m_1/2$) are built up, there are still at most m_1 features derived from the CDA, which means that only the first significant $m_1/2$ clusters of each class contribute to the process of the CDA.

Figure 4.6 - Figure 4.8 show the performance comparisons of the CDA-1, CDA-2, and CDA-3, respectively, as the number of clusters (k) varies. For simplicity, the effect of the clusters is only evaluated based on the size of features that gives the best performance as discussed in the previous subsection. Specifically, the CDA-1 evaluates the effect of the clusters using 169 ($m_1 = 170, m_2 = 169$) features for L_1 similarity measures and 119 ($m_1 = 120, m_2 = 119$) features for both L_2 and cosine similarity measures, respectively; the CDA-2 evaluates the effect of the clusters using 200 ($m_1 = 200, m_2 = 200$) features for L_1 similarity measure and 180 ($m_1 = 180, m_2 =$

Table 4.1 Parameter Settings of the FLD-, NDA-, and CDA-based Eye Detection Methods

Method	#PCA Features	#Discriminatory Features	#Nearest Neighbors	#Clusters	Similarity Measure
FLD	150	1	-	-	L_1
NDA	150	150	5	-	L_2
CDA-1	170	169	-	85	L_1
CDA-2	180	180	-	90	cosine
CDA-3	180	180	-	90	cosine

180) features for both L_2 and cosine similarity measures, respectively; and the CDA-3 evaluates the effect of the clusters using 170 ($m_1 = 170, m_2 = 170$) features for L_1 similarity measure and 180 ($m_1 = 180, m_2 = 180$) features for both L_2 and cosine similarity measures, respectively. In addition, note that Figure 4.6 - Figure 4.8, as Figure 4.3 – Figure 4.5 do, show the true positive rate of the CDA-1, CDA-2, and CDA-3, respectively, at the false accept rate of 0.1 for the detection normalized error $e \leq 0.07$, as the number of clusters (k) varies.

Figure 4.6 - Figure 4.8 reveal that the performance of all three CDA models is enhanced as the number of clusters increases and approaches to $m_1/2$. Specifically, the CDA-1 reaches the best performance by using 85 clusters of each class and L_1 similarity measure, while both the CDA-2 and the CDA-3 reach the best performance by using 90 clusters of each class and cosine similarity measure.

4.3.3 Evaluation of the CDA-based Eye Detection Method

This subsection evaluates the eye detection performance of the CDA-based method in comparison with the FLD- and NDA-based methods. Note that the experiments are carried on the 12,776 FRGC images. For the best performance, the parameter settings are shown in Table 4.1.

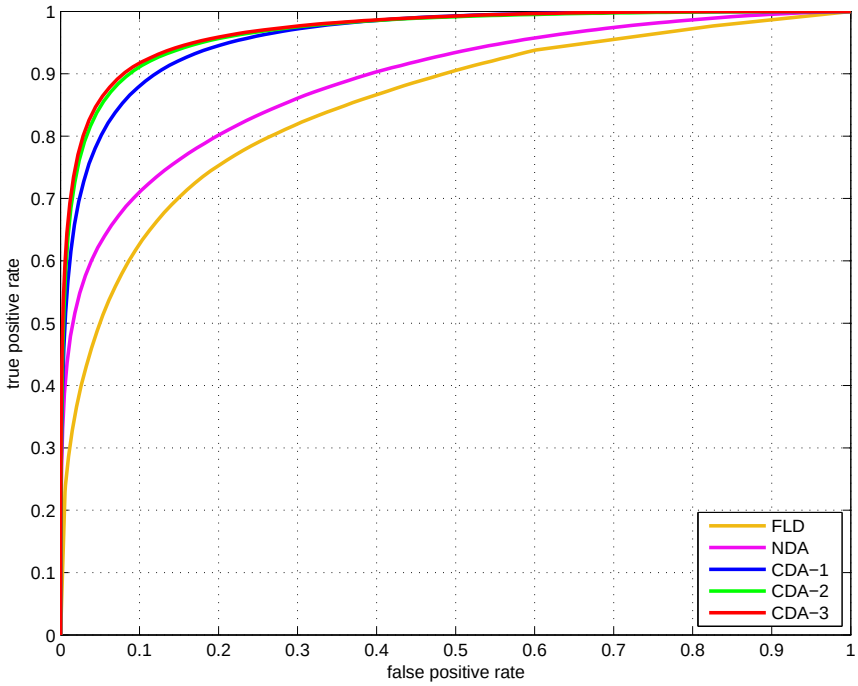


Figure 4.9 The ROC curves of the FLD-, NDA-, and CDA-based eye detection methods.

Table 4.2 Comparison of the True Positive Rate (TPR) of the FLD-, NDA-, and CDA-based Eye Detection Methods at the False Positive Rate (FPR) of 0.1

Method	TPR at FPR of 0.1
FLD	63.35%
NDA	71.27%
CDA-1	87.76%
CDA-2	91.35%
CDA-3	91.87%

Figure 4.9 shows the Receiver Operating Characteristic (ROC) curves of the FLD-, NDA-, and CDA-based eye detection methods. Note that the ROC curves are drawn for the detection normalized error $e \leq 0.07$. Figure 4.9 reveals that the CDA-based methods (CDA-1, CDA-2, and CDA-3) significantly improve the eye detection performance in comparison with the FLD- and NDA-based methods. The

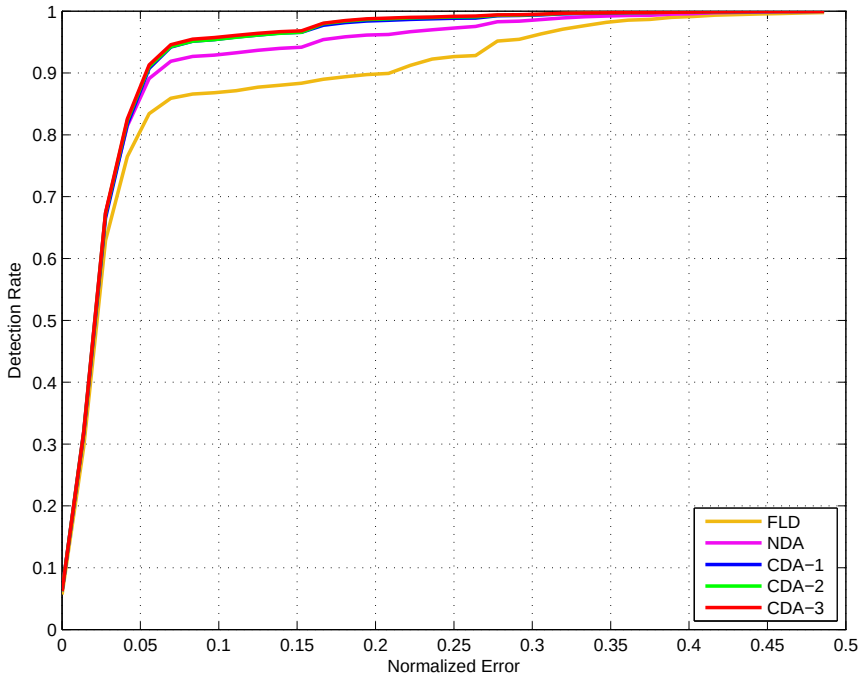


Figure 4.10 The detection rate of the FLD-, NDA-, and CDA-based eye detection methods over different normalized errors.

CDA-3 gives the best performance, followed in order by the CDA-2, the CDA-1, the NDA, and the FLD. Table 4.2 shows the true positive rate (TPR) of these methods at the false positive rate (FPR) of 0.1. It reveals that the CDA-3, which gives the best performance, improves the TPR of the FLD by 28.52% and the NDA by 20.60%, respectively. Even the CDA-1, which gives the lowest TPR among the three CDA models, improves the TPR of the FLD by 24.41% and the NDA by 16.49%, respectively.

If the detected eye location is eventually chosen as the average of the multiple detection around the pupil center (usually there are multiple detections around each pupil center), Figure 4.10 show the detection rate over the different normalized detection errors (e). In Figure 4.10, the horizontal axis represents the normalized detection error, and the vertical axis represents the corresponding detection rate. Figure 4.10 reveals that the CDA-based methods consistently outperform the FLD- and the NDA-based methods in terms of the detection rate. The three CDA mod-

Table 4.3 The detection Rate and Detection Accuracy of the FLD-, NDA-, and CDA-based Methods. The Detection Rate is for the Normalized Error $e \leq 0.07$. The mean$(\cdot)$ and std$(\cdot)$ Represent the Mean and the Deviation of the Detection Pixel Error with respect to the Direction Specified by the Parameter, Respectively (DR Stands for the Detection Rate)

Method	mean(x)	std(x)	mean(y)	std(y)	mean($\sqrt{x^2+y^2}$)	DR
FLD	2.70	3.68	2.66	5.86	4.49	85.90%
NDA	2.47	3.24	1.32	3.44	3.24	91.89%
CDA-1	2.27	2.46	1.05	2.49	2.84	94.20%
CDA-2	2.30	2.62	0.97	2.19	2.80	94.27%
CDA-3	2.27	2.56	0.94	2.10	2.75	94.58%

els, CDA-1, CDA-2, and CDA-3, have comparable detection rates, and the CDA-3 performs slightly better than CDA-2 and subsequently the CDA-2 performs slightly better than the CDA-1.

In order to further show the superiority of the CDA-based methods, Table 4.3 explicitly shows the detection rate (for the normalized error $e \leq 0.07$) and the detection accuracy (i.e., the average detection pixel errors in the Euclidean distance) of these eye detection methods. For complete assessment, the detection accuracy on both the horizontal (x) and the vertical (y) directions are shown as well in Table 4.3. Table 4.3 show the improvement of the CDA-based methods on both the detection rate and the detection accuracy over the nonCDA-based methods. If one focuses on that more eyes are detected within a criteria (e.g., $e \leq 0.07$), the CDA-3 gives the best detection rate, which is 94.58%; if one focuses on the minimum average detection pixel error, the CDA-3 still the best detection accuracy, which is 2.75.

4.3.4 Comparison with State-of-the-art Methods

In order to show the robustness of the proposed CDA-based eye detection method and to compare it with the state-of-the-art eye detection methods, experiments on

Table 4.4 Comparison of the Eye Detection Performance with State-of-the-Art Methods (e Stands for the Normalized Error)

method	database	$e \leq 0.05$	$e \leq 0.10$	$e \leq 0.25$
Jesorsky (2001) [37]	BioID	40.00%	79.00%	91.80%
Hamouz (2004) [34]	BioID	50.00%	66.00%	70.00%
Hamouz (2005) [35]	BioID	59.00%	77.00%	93.00%
Cristinacce (2004) [22]	BioID	56.00%	**96.00%**	98.00%
Asteriadis (2006) [5]	BioID	74.00%	81.70%	97.40%
Bai (2006) [6]	BioID	37.00%	64.00%	96.00%
Niu (2006) [63]	BioID	78.00%	93.00%	95.00%
Campadelli (2006) [12]	BioID	62.00%	85.20%	96.10%
Campadelli (2009) [13]	BioID	80.70%	93.20%	95.30%
Valenti (2008) [81]	BioID	**84.10%**	90.85%	**98.49%**
CDA-3	BioID	**87.25%**	**94.87%**	**99.21%**
CDA-3	FRGC	**87.79%**	**95.81%**	**99.17%**

the BioID face database [37] are also implemented. The BioID database consists of 1521 frontal face images of 23 subjects.

The methods that are compared with include those used by Jesorsky et al. [37], Hamouz et al. [34], [35], Cristinacce et al. [22], Asteriadis et al. [5], Bai et al. [6], Niu et al. [63], Campadelli et al. [12], [13], and Valenti et al. [81]. All above methods reported the performance on the BioID database and applied the same normalized error criterion to evaluate the performance.

Table 4.4 shows the performance comparison between the CDA-based method and the other methods mentioned above for the normalized error of 0.05, 0.10, and 0.25, respectively. Note that only the CDA-3 model is used in this comparison, which gives the slightly better detection performance than the CDA-1 and the CDA-2 models. The results derived from the CDA-based method and the best results reported by other methods are highlighted in bold text. Note that for the performance

which is in-explicitly reported by the authors, the results are estimated from the graphs in the literature.

Table 4.4 reveals that for the normalized error of 0.05 and 0.25, the proposed CDA-based method outperforms all other state-of-the-art methods listed in Table 4.4. For the normalized error of 0.10, the CDA-based method has comparable performance to the best results. Table 5.8 also shows the detection performance of the CDA-based method on the FRGC database. The performance on the BioID and FRGC database is very close to each other, which indicates the robustness of the CDA-based method.

4.4 Conclusion

This chapter presents a clustering-based discriminant analysis (CDA) method, which improves upon the Fisher Linear Discriminant (FLD) method, to extract discriminatory features for eye detection. Three CDA models (CDA-1, -2, and -3) are proposed by taking advantage of the clustering technique. For every CDA model a new between-cluster scatter matrix is defined. The CDA method thus can derive adequate features to achieve satisfactory performance for eye detection. Furthermore, the clustering nature of the three CDA models and the nonparametric nature of the CDA-2 and -3 models can further improve the detection performance upon the conventional FLD method. Experiments on the FRGC and the BioID face database show that (i) the CDA method significantly improves the performance of the conventional discriminant analysis methods, and (ii) the proposed CDA-based eye detection method achieves good eye detection performance and outperforms some state-of-the-art eye detection methods. In particular, the CDA-3 based eye detection method gives the detection rate of 94.58% and the detection accuracy of 2.75 pixel error in average.

CHAPTER 5

EFFICIENT SUPPORT VECTOR MACHINE

This chapter proposes a new efficient Support Vector Machine (eSVM) for eye detection that improves the computational efficiency of the conventional Support Vector Machine (SVM). The eSVM first defines a Θ set that consists of the training samples on the wrong side of their margin derived from the conventional soft-margin SVM. The Θ set plays an important role in controlling the generalization performance of the eSVM. The eSVM then introduces only a single slack variable for all the training samples in the Θ set, and as a result, only a very small number of those samples in the Θ set become support vectors. The eSVM hence significantly reduces the number of support vectors and improves the computational efficiency without sacrificing the generalization performance. The optimization of the eSVM is implemented using a modified Sequential Minimal Optimization (SMO) algorithm to solve the large Quadratic Programming (QP) problem. Experiments on several diverse data sets show that the eSVM significantly improves the computational efficiency upon the conventional SVM while achieving comparable generalization performance to or higher performance than the SVM.

An accurate and efficient eye detection method is then presented based on the eSVM method. This eSVM-based eye detection method consists of the eye candidate selection stage and the eye candidate validation stage. The selection stage selects the eye candidates in an image through a process of eye color distribution analysis in the YCbCr color space. The validation stage applies first 2D Haar wavelets for multi-scale image representation, the PCA for dimensionality reduction, and finally the eSVM for classification. Experiments on the FRGC and the FERET database

show that the eSVM-based eye detection method can reach real-time eye detection speed and better eye detection accuracy than some state-of-the-art methods.

5.1 Background

Support Vector Machine (SVM) [82], [83] has gained a great deal of attention due to its generalization performance. Since it was introduced, SVM has become a popular method in machine learning, object detection and recognition, as well as in various prediction and regression problems [62], [36], [24], [26], [39], [74], [14], [33]. However, when the recognition or regression problem becomes complex, the number of support vectors tends to increase, which leads to increasing the model complex. As a result, the SVM becomes less efficient due to the expensive computation cost of its decision function, which involves an inner product of all the support vectors for the linear SVM and a kernel computation of all the support vectors for the kernel SVM.

A number of simplified SVMs have been proposed to address the inefficiency problem (i.e., the large number of support vectors) of the conventional SVM. Burges [10] proposed a method, which computes an approximation to the decision function using a reduced set of support vectors, to reduce the computation complexity of the decision function by a factor of ten. This method was then applied to handwritten digits recognition [77] and face detection [75]. The authors in [42], [46] presented a new Reduced Support Vector Machine (RSVM) as an alternative to the standard SVM for improving computational efficiency [42], [46]. The RSVM generates a nonlinear kernel based on a separating surface (decision function) by solving a smaller optimization problem using a subset of training samples. The RSVM successfully reduces the model complexity. Other new SVM models include the υ-SVM [18], the simplified SVM [66], and the mirror classifier based SVM [16]. One drawback of these new SVMs is that they tend to reduce classification accuracy when improving the computational efficiency.

5.2 Support Vector Machine

This section briefly reviews the conventional SVM method, followed by the analysis of the factors causing the inefficiency problem (i.e., the large number of support vectors) of the conventional SVM.

Let the training set be $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$ indicate the two different classes, and l is the number of the training samples. When the training samples are linearly separable, the conventional SVM defines an optimal separating hyperplane, $\mathbf{w}^t\mathbf{x} + b = 0$, by minimizing the following functional:

$$\begin{aligned} &\min_{\mathbf{w},b} \tfrac{1}{2}\mathbf{w}^t\mathbf{w}, \\ &\textit{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \cdots, l. \end{aligned} \tag{5.1}$$

When the training samples are not linearly separable, the conventional soft-margin SVM determines the soft-margin optimal hyperplane by minimizing the following functional:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi_i} \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_{i=1}^{l}\xi_i \ , \\ &\textit{subject to} \quad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \cdots, l. \end{aligned} \tag{5.2}$$

where $\xi_i \geq 0$ are slack variables and $C > 0$ is a regularization parameter.

The Lagrangian theory and the Kuhn-Tucker theory are then applied to optimize the functional in Equation 5.2 with inequality constraints [84]. The optimization process leads to the following quadratic convex programming problem:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l}\alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j \\ &\textit{subject to} \quad \sum_{i=1}^{l} y_i\alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, ...l \end{aligned} \tag{5.3}$$

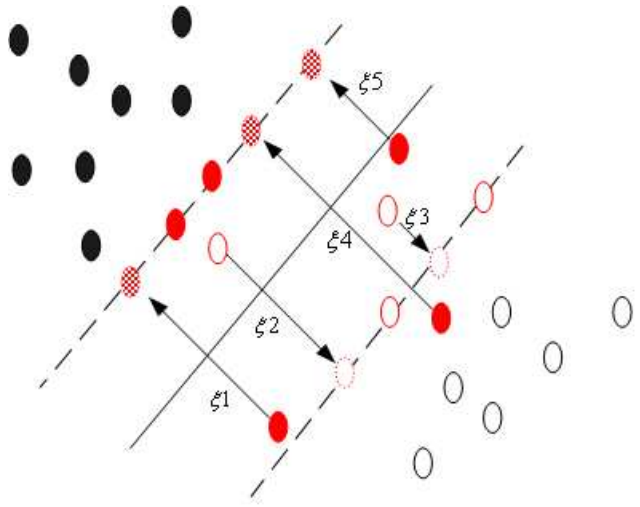


Figure 5.1 Illustration of the Conventional soft-margin SVM in the two dimensional space, where the two classes are presented by the solid and open circles, respectively. All five samples on the wrong side of their margin are pulled onto their boundaries to become support vectors.

From the Lagrangian theory and the Kuhn-Tucker theory, we also have:

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = \sum_{i \in SV} y_i \alpha_i \mathbf{x}_i \tag{5.4}$$

where SV is the set of Support Vectors (SVs), which are the training samples with nonzero coefficients α_i. The decision function of the SVM is therefore derived as follows:

$$f(x) = sign(\mathbf{w}\mathbf{x} + b) = sign(\sum_{i \in SV} y_i \alpha_i \mathbf{x}_i \mathbf{x} + b) \tag{5.5}$$

Equation 5.5 reveals that the computation of the decision function involves an inner product of all the support vectors. (Note that the computation of the decision function for the kernel SVM involves a kernel computation of all the support vectors.) Therefore, the computation cost of the decision function is proportional to the number of the support vectors. When the number of the support vectors is large, the computation cost of the inner product will become expensive and the computational

efficiency of the conventional soft-margin SVM will be compromised. According to the Kuhn-Tucker theory, we have the following conditions for the conventional soft-margin SVM:

$$\alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i] = 0, \quad i = 1, 2, \cdots, l. \tag{5.6}$$

Equation 5.6 shows that if $\alpha_i \neq 0$, then $y_i(\mathbf{w}^t\mathbf{x}_i+b)-1+\xi_i = 0$. Therefore, the training samples that satisfy $y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i = 0$ are support vectors for the conventional soft-margin SVM. The intuitive interpretation of the support vectors is that they are the training samples that lie on their boundaries or the samples pulled onto their boundaries by the slack variables ξ_i as shown in Figure 5.1. In fact, Figure 5.1 shows that all the training samples on the wrong side of their margin become support vectors because of the slack variables, which pull the training samples onto their boundaries to make them support vectors. As a complex pattern classification problem often has a large number of the training samples on the wrong side of their margin, the number of support vectors becomes quite large, which leads to the inefficiency problem of the conventional soft-margin SVM.

5.3 Efficient Support Vector Machine

To address the inefficiency problem of the conventional soft-margin SVM, this section presents a new SVM, the efficient SVM (eSVM). As discussed above, the conventional soft-margin SVM usually derives a large number of support vectors for the non-separable case, because all the training samples on the wrong side of their margin become support vectors as the slack variables pull these samples to their boundaries. The eSVM, however, reduces the number of support vectors significantly, because only a small number (can be as few as one) of the training samples on the wrong side of their margin are pulled to their boundaries to become support vectors. The eSVM first defines a Θ set that consists of the training samples on the wrong side of their margin

derived from the conventional soft-margin SVM. The Θ set plays an important role in controlling the generalization performance of the eSVM. The eSVM then introduces only a single slack variable for all the training samples in the Θ set, and as a result, only a very small number of those samples in the Θ set are pulled to their boundaries and become support vectors. As the number of support vectors reduced, the eSVM improve the computational efficiency upon the conventional soft-margin SVM.

Specifically, the eSVM optimizes the following functional:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\xi \ , \\ &subject\ to\ \ y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 \ , \ i \in \Omega - \Theta \\ &\qquad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi \ , \ i \in \Theta \ , \ \xi \geq 0 \end{aligned} \tag{5.7}$$

where Θ is the set of the training samples on the wrong side of their margin derived from the conventional soft-margin SVM, and Ω is the set of all the training samples. Compared with the conventional soft-margin SVM that defines the slack variables with different values, the eSVM specifies a fixed value for all the slack variables. The first inequality constraint in Equation 5.7 ensures that the training samples on the right side of their margin in the conventional soft-margin SVM are still on the right side in the eSVM. The second inequality constraint in Equation 5.7 ensures that only a small number of training samples in the Θ set becomes support vectors due to the introduction of the single slack variable that pulls most of the training samples in the Θ set beyond their margin to the right side and thus become non-support vectors. The significance of the eSVM is to simulate the maximal margin separating boundary of the conventional SVM by using much fewer support vectors.

Further analysis on the first inequality constraint in Equation 5.7 reveals that this constraint makes the eSVM to maintain a similar maximal margin separating boundary with that of the conventional SVM. The definition of the separating boundary $\mathbf{w}^t\mathbf{x}+b=0$ is associated with the definition of the maximal margin $\mathbf{w}^t\mathbf{x}+b=\pm 1$.

Given the separating boundary $\mathbf{w}^t\mathbf{x}+b=0$, the maximal margin is fixed to $\mathbf{w}^t\mathbf{x}+b=\pm 1$, and vice versa. The first inequality constraint in Equation 5.7 ensures that the training samples on the right side of their margin in the conventional soft-margin SVM are still on the right side in the eSVM. If the eSVM derives a separating boundary that is significantly different from the one derived by the conventional SVM, this constraint will not be satisfied. As the eSVM derives a similar separating boundary with the conventional SVM, the maximal margin of the eSVM is thus similar with that of the conventional SVM as well – neither degrade nor upgrade much from that of the SVM. Consequently, the eSVM inherits the advantage of generalization performance of the conventional SVM, and has comparable classification performance with the SVM.

The second inequality constraint in Equation 5.7 plays the significant role of reducing the number of support vectors. This constraint ensures that only a small number of training samples in the Θ set becomes support vectors due to the introduction of the single slack variable that pulls most of the training samples in the Θ set beyond their margin to the right side and thus become non-support vectors. It is possible that support vectors falling on the margin of the eSVM are a little bit more than those of the SVM. However, majority of support vectors for the SVM come from the samples on the wrong side of their margin. The samples on the margin only contribute to a small portion of support vectors for the SVM, since the chance that samples happen to fall on the margin is significantly lower than the chance that samples fall onto the right side or wrong side of the margin. Therefore, even though the number of support vectors falling on the margin may increase a little bit for the eSVM, the number of support vectors on the wrong side of the margin significantly decreases compared with the SVM, and consequently, the total number of support vectors for the eSVM is still significantly less than that of the conventional SVM.

As the optimization problem of the eSVM defined in Equation 5.7 is different from that of the conventional SVM, its corresponding dual mathematical problem

after applying the Lagrangian theory and the Kuhn-Tucker theory is also different from the one derived in the conventional soft-margin SVM. In particular, let $\alpha_1, \alpha_2, \cdots, \alpha_l \geq 0$ and $\mu \geq 0$ be the Lagrange multipliers, the primal Lagrange functional is defined as follows:

$$\begin{aligned}\mathcal{F}(\mathbf{w}, b, \xi, \alpha_i, \mu) &= \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\xi - \sum_{i\in\Omega-\Theta} \alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i - b) - 1] \\ &\quad - \sum_{i\in\Theta} \alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i - b) + \xi - 1] - \mu\xi \\ &= \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + (C - \sum_{i\in\Theta} \alpha_i - \mu)\xi - \sum_{i\in\Omega} \alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i - b) - 1]\end{aligned} \tag{5.8}$$

Next, we maximize the primal Lagrange functional $\mathcal{F}(\mathbf{w}, b, \xi, \alpha_i, \mu)$ with respect to $\mathbf{w}$, b, and ξ as follows:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i\in\Omega} \alpha_i y_i \mathbf{x}_i = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i\in\Omega} \alpha_i y_i \mathbf{x}_i \tag{5.9}$$

$$\frac{\partial \mathcal{F}}{\partial b} = \sum_{i\in\Omega} \alpha_i y_i = 0 \tag{5.10}$$

$$\frac{\partial \mathcal{F}}{\partial \xi} = C - \sum_{i\in\Theta} \alpha_i - \mu = 0 \tag{5.11}$$

Then, we derive a convex quadratic programming model by substituting Equations 5.9, 5.10, and 5.11 into Equation 5.8 as follows:

$$\begin{aligned}&\max_{\alpha} \sum_{i\in\Omega} \alpha_i - \tfrac{1}{2} \sum_{i,j\in\Omega} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j \\ &\textit{subject to} \quad \sum_{i\in\Omega} y_i\alpha_i = 0, \quad \left(\sum_{i\in\Theta} \alpha_i\right) \leq C, \quad \alpha_i \geq 0,\ i \in \Omega\end{aligned} \tag{5.12}$$

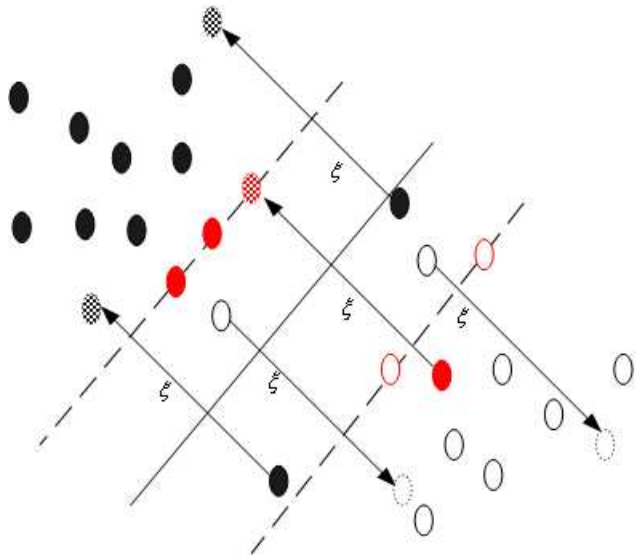


Figure 5.2 Illustration of the eSVM in the two dimensional space, where the two classes are presented by the solid and open circles, respectively. Only one of the five samples on the wrong side of their margin is pulled onto its boundary to become a support vector.

Furthermore, we have the following constraints from the Kuhn-Tucker theory:

$$\begin{aligned} \alpha_i[y_i(\mathbf{w}^t\mathbf{x}+b)-1] &= 0,\ i \in \Omega - \Theta \\ \alpha_i[y_i(\mathbf{w}^t\mathbf{x}+b)-1+\xi] &= 0,\ i \in \Theta \end{aligned} \tag{5.13}$$

Equation 5.13 shows that if $\alpha_i \neq 0$, then either $y_i(\mathbf{w}^t\mathbf{x}+b)-1=0, i \in \Omega-\Theta$ or $y_i(\mathbf{w}^t\mathbf{x}+b)-1+\xi=0, i \in \Theta$. The training samples that satisfy either $y_i(\mathbf{w}^t\mathbf{x}+b)-1=0, i \in \Omega-\Theta$ or $y_i(\mathbf{w}^t\mathbf{x}+b)-1+\xi=0, i \in \Theta$ are support vectors for the eSVM. Therefore, the intuitive interpretation of the support vectors is the training samples that lie on their boundaries or the samples pulled onto their boundaries by the slack variable ξ as shown in Figure 5.2. As all the slack variables in the eSVM have the same value, Figure 5.2 also reveals that only a small number (can be as few as one) of the training samples on the wrong side of their margin (i.e., samples in the Θ set) are pulled onto their boundaries to become support vectors. As a matter of fact, Figure 5.2 shows that only one training sample that is farthest away from their

boundaries are pulled back to become support vector, while the others are not support vectors because they are pulled to the right side of their margin but they do not fall onto the boundaries.

5.4 Modified Sequential Minimal Optimization Algorithm

Training a SVM requires a solution to a very large Quadratic Programming (QP) optimization problem. The Sequential Minimal Optimization (SMO) algorithm [70] is a popular and efficient tool to solve the large QP problem defined in the SVM. The SMO algorithm breaks a large QP problem into a series of the smallest possible optimization problems, which can be solved analytically without resorting to a time-consuming iterative process [70]. As the QP problem of the eSVM defined in Equation 5.12 is different from that of the conventional soft margin SVM, this section presents a modified SMO algorithm for training the eSVM.

The modified SMO algorithm, as the SMO does, consists of two major steps: an analytical solution to the smallest QP problem with two Lagrange multipliers, and a heuristic approach for choosing which two multipliers to optimize. The next two subsections present these two steps in details, respectively.

5.4.1 An Analytic Solution to the Smallest QP Problem

Let α_s and α_t, for one of the smallest quadratic programming problems, be the two Lagrange multipliers to be optimized while the other α_i's are fixed. First, the SMO algorithm derives the unconstrained maximum value for α_s and α_t:

$$\begin{aligned} \alpha_t^{new} &= \alpha_t^{old} + \frac{y_t(g_s^{old} - g_t^{old})}{\eta} \\ \alpha_s^{new} &= \gamma - \Delta\alpha_t^{new} \end{aligned} \tag{5.14}$$

where $\Delta = y_s y_t$, $\gamma = \alpha_s^{old} + \Delta\alpha_t^{old}$, $\eta = 2\mathbf{x}_s\mathbf{x}_t - \mathbf{x}_s\mathbf{x}_s - \mathbf{x}_t\mathbf{x}_t$, $g_s^{old} = y_s - \mathbf{x}_s^t\mathbf{w}^{old}$, and $g_t^{old} = y_s - \mathbf{x}_t^t\mathbf{w}^{old}$. Note that for the initialization step, α^{old} can be set to 0.

Then, the two Lagrange multipliers α_s^{new} and α_t^{new} should be checked if they satisfy the inequality constraints defined in Equation 5.12:

$$\alpha_i \geq 0, i \in \Omega, \left(\sum_{i\in\Theta} \alpha_i\right) \leq C$$

If the two Lagrange multipliers α_s^{new} and α_t^{new} in Equation 5.14 do not satisfy the above inequality constraints, their values need to be adjusted. Depending on the values of α_s^{new}, α_t^{new}, and Δ, there are several cases to consider:

1. If $\Delta = 1$, then $\alpha_s^{new} + \alpha_t^{new} = \gamma$
   (a) If $\alpha_s^{new}(\alpha_t^{new}) < 0$, then $\alpha_s^{new}(\alpha_t^{new}) = 0$, $\alpha_t^{new}(\alpha_s^{new}) = \gamma$;
   (b) If $s \in MV$, $t \notin MV$, and $\alpha_s^{new} > C - \sum_{i\in MV, i\neq s} \alpha_i^{old}$, then $\alpha_s^{new} = C - \sum_{i\in MV, i\neq s} \alpha_i^{old}$, $\alpha_t^{new} = \gamma - \alpha_s^{new}$;
   (c) If $s \notin MV$, $t \in MV$, and $\alpha_t^{new} > C - \sum_{i\in MV, i\neq t} \alpha_i^{old}$, then $\alpha_t^{new} = C - \sum_{i\in MV, i\neq t} \alpha_i^{old}$, $\alpha_s^{new} = \gamma - \alpha_t^{new}$;
   (d) If $s \in MV$ and $t \in MV$, since $\alpha_s^{new} + \alpha_t^{new} = \alpha_s^{old} + \alpha_t^{old}$, there is no effect on $\sum_{i\in MV} \alpha_i$. So α_s^{new} and α_t^{new} don't need to adjust;
   (e) If $s \notin MV$ and $t \notin MV$, since both α_s^{new} and α_t^{new} would not affect $\sum_{i\in MV} \alpha_i$, they don't need to adjust.
2. If $\Delta = -1$, then $\alpha_s^{new} - \alpha_t^{new} = \gamma$
   (a) If $\alpha_s^{new} < 0$, then $\alpha_s^{new} = 0$, $\alpha_t^{new} = -\gamma$;
   (b) If $\alpha_t^{new} < 0$, then $\alpha_t^{new} = 0$, $\alpha_s^{new} = \gamma$;
   (c) If $s \in MV$, $t \notin MV$, and $\alpha_s^{new} > C - \sum_{i\in MV, i\neq s} \alpha_i^{old}$, then $\alpha_s^{new} = C - \sum_{i\in MV, i\neq s} \alpha_i^{old}$, $\alpha_t^{new} = \alpha_s^{new} - \gamma$;
   (d) If $s \notin MV$, $t \in MV$, and $\alpha_t^{new} > C - \sum_{i\in MV, i\neq t} \alpha_i^{old}$, then $\alpha_t^{new} = C - \sum_{i\in MV, i\neq t} \alpha_i^{old}$, $\alpha_s^{new} = \alpha_t^{new} + \gamma$;

(e) If $s \in MV$, $t \in MV$, and $\alpha_s^{new} + \alpha_t^{new} > C - \sum_{i \in MV, i \neq s,t} \alpha_i^{old}$, then $\alpha_s^{new} = \frac{1}{2}(C - \sum_{i \in MV, i \neq s,t} \alpha_i^{old} + \gamma)$, $\alpha_t^{new} = \frac{1}{2}(C - \sum_{i \in MV, i \neq s,t} \alpha_i^{old} - \gamma)$;

(f) If $s \notin MV$ and $t \notin MV$, they don't need to adjust.

5.4.2 A Heuristic Approach for Choosing Multipliers

The conventional SMO algorithm [70] applies some independent heuristics to choose which two Lagrange multipliers to optimize jointly at every step. This heuristic process is implemented by means of two loops: the outer loop selects the first α_i that violates the Kuhn-Tucker conditions, while the inner loop selects the second α_i that maximizes $|E_2 - E_1|$, where E_i is the prediction error on the ith training sample.

Complexity analysis reveals that the conventional SMO algorithm, at every step of the heuristic process, takes $O(l^2)$ to choose the Lagrange multipliers, where l is the number of the training samples. This process is time-consuming if the number of the training samples is very large. This subsection presents an improved heuristic process that chooses the two α_i's simultaneously: each time the two α_i's that violate the Kuhn-Tucker conditions most seriously are chosen. The new heuristic process thus takes $O(l)$ to choose the Lagrange multipliers at every step.

In order to determine the pair of α_i's that violate the Kuhn-Tucker conditions most seriously, the Kuhn-Tucker conditions of the eSVM should be further analyzed. From Equation 5.13, the Kuhn-Tucker conditions of the eSVM can be decomposed as follows:

1. When $i \in V - MV$

   (a) If $\alpha_i > 0$ and $y_i = 1$, then $b = y_i - \omega^t \phi(x_i)$;

   (b) If $\alpha_i > 0$ and $y_i = -1$, then $b = y_i - \omega^t \phi(x_i)$;

   (c) If $\alpha_i = 0$ and $y_i = 1$, then $b \geq y_i - \omega^t \phi(x_i)$;

   (d) If $\alpha_i = 0$ and $y_i = -1$, then $b \leq y_i - \omega^t \phi(x_i)$.

2. When $i \in MV$

(a) If $\alpha_i > 0$ and $y_i = 1$, then $b \leq y_i - \omega^t \phi(x_i)$;

(b) If $\alpha_i > 0$ and $y_i = -1$, then $b \geq y_i - \omega^t \phi(x_i)$;

(c) If $\sum_{j \in MV} \alpha_j < C$ and $y_i = 1$, then $b \geq y_i - \omega^t \phi(x_i)$;

(d) If $\sum_{j \in MV} \alpha_j < C$ and $y_i = -1$, then $b \leq y_i - \omega^t \phi(x_i)$.

As a result, the pair of α_i's that violate the Kuhn-Tucker conditions most seriously can be determined as follows:

$$
\begin{aligned}
s = argmax(& \\
& \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i \geq 0, y_i = 1, i \in \Omega - \Theta\}, \quad \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i > 0, y_i = -1, i \in \Omega - \Theta\}, \\
& \{y_i - \mathbf{w}^t \mathbf{x}_i | \textstyle\sum_{j \in \Theta} \alpha_j < C, y_i = 1, i \in \Theta\}, \quad \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i > 0, y_i = -1, i \in \Theta\}. \\
&) \\
t = argmin(& \\
& \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i > 0, y_i = 1, i \in \Omega - \Theta\}, \quad \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i \geq 0, y_i = -1, i \in \Omega - \Theta\}, \\
& \{y_i - \mathbf{w}^t \mathbf{x}_i | \textstyle\sum_{j \in \Theta} \alpha_j < C, y_i = -1, i \in \Theta\}, \quad \{y_i - \mathbf{w}^t \mathbf{x}_i | \alpha_i > 0, y_i = 1, i \in \Theta\}. \\
&)
\end{aligned}
\tag{5.15}
$$

5.5 Accurate and Efficient Eye Detection Using eSVM

This section presents an accurate and efficient eye detection method by applying the eSVM together with color information and 2D Haar wavelets. Figure 5.3 shows the architecture of the proposed eye detection method. First, the Bayesian Discriminating Features(BDF) method [47] is applied to detect a face from an image and normalize the detected face to a predefined size (128×128 in the experiments). Second, some geometric constraints are used to extract an eye strip from the upper portion of the detected face (the size of the eye strip is 55×128 in the experiments). Third, a new eye detection method is applied that consists of two stages: the eye candidate selection stage and the eye candidate validation stage. Specifically, the selection stage rejects 99% of the pixels through an eye color distribution analysis in the YCbCr color space

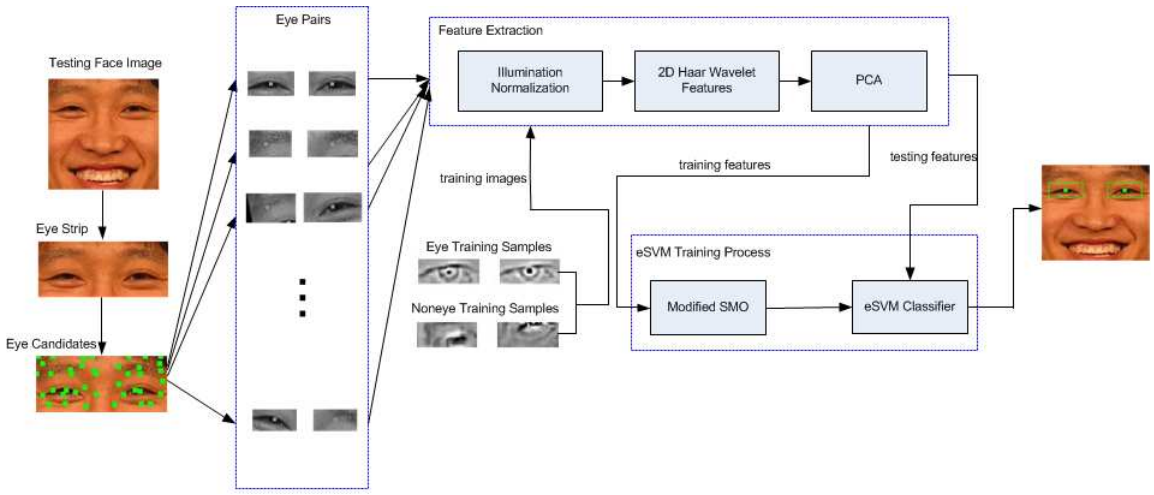


Figure 5.3 System architecture of the eSVM-based eye detection method.

[79], while the remaining 1% of the pixels are further processed by the validation stage. The validation stage applies illumination normalization through Gamma correction, Difference of Gaussian (DoG) filtering, and contrast equalization, 2D Haar wavelets for multi-scale image representation, PCA for dimensionality reduction, and the eSVM for classification to detect the center of the eye. The next two subsections present in details the eye candidate selection and validation stages, respectively.

5.5.1 The Eye Candidate Selection Stage

The conventional sliding window based eye detection methods exhaustively classify all the pixels in an image from left to right, top to bottom to locate the eyes. The excessive number of the pixels over an image significantly slows down the classifier-based eye detection methods. A novel eye candidate selection stage is therefore proposed in this subsection to first dramatically reduce the number of eye candidates, which will be further validated by the classifier-based methods.

Specifically, the eye candidates are chosen through an eye color distribution analysis in the YCbCr color space [79]. In the YCbCr color space, as indicated by

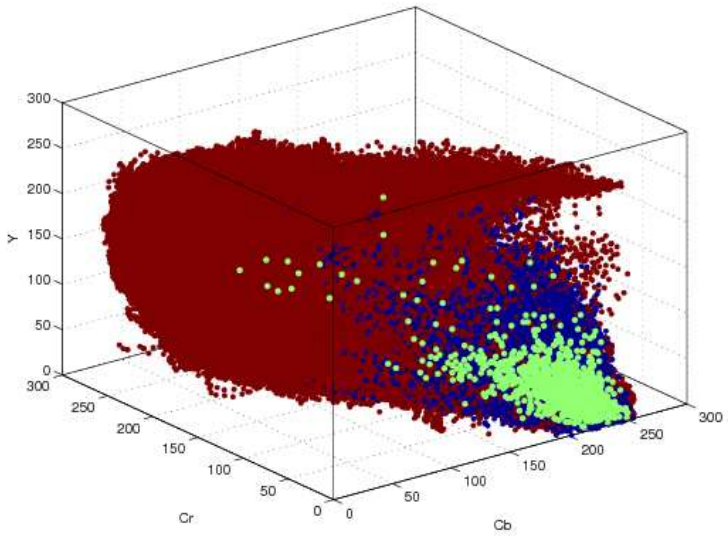


Figure 5.4 The eye-tone distribution in the YCbCr color space. The skin pixels are represented in red, the eye region pixels are in blue, and the pupil-center pixels are in green.

Equation 2.1, the RGB components are separated into luminance (Y), chrominance blue (Cb), and chrominance red (Cr) [79]. It is observed that in the eye region, especially around the pupil center, pixels are more likely to have higher values in chrominance blue (Cb) and lower values in chrominance red (Cr) when compared with those pixels in the skin region. It is also observed that the luminance (Y) of the pixels in the eye region is much darker than those in the skin region. To illustrate these findings, a mount of pixels are collected from random skin patches, eye regions, as well as pupil centers from 600 face images. The number of pixels randomly collected from the skin patches and the eye regions are 4,078,800 and 145,200, respectively. And the number of pixels corresponding to the pupil centers is 1,200 as there are two eyes in each face image. Figure 5.4 shows the eye-tone distribution in the YCbCr color space, where the skin pixels are represented in red, the eye region pixels are in blue, and the pupil-center pixels are in green. Figure 5.4 reveals that the eye-centers, which are represented by green, are clustered in the corner with higher Cb values but lower Cr and Y values. Figure 5.5 shows some eye strip examples in the YCbCr color

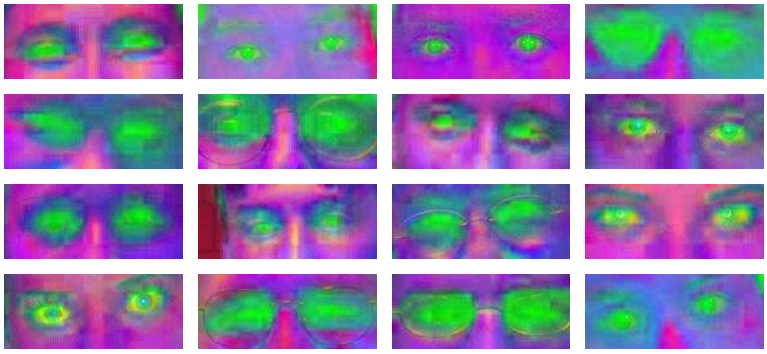


Figure 5.5 Example eye strip images in the YCbCr color space, where Y is represented in red, Cb in green, and Cr in blue.

space, where Y is represented in red, Cb in green, and Cr in blue. One can see from Figure 5.5 that the eye regions tend to have high green values and low blue values.

Motivated by these findings, a new method for eye candidate selection is presented. The idea of the eye candidate selection method is to define a weight for each pixel based on its Y, Cb, and Cr values and rank the pixels according to their weights. In particular, the weight of pixel (i, j) is defined as follows:

$$weight(i,j) = \sum_{i-2,j-2}^{i+2,j+2} [Cb(i,j) + (255 - Cr(i,j)) + (255 - Y(i,j))] \tag{5.16}$$

The first K pixels with maximum weights are therefore considered as the eye candidates. Figure 5.6 evaluates the performance of the proposed eye candidate selection method by randomly selecting 2,000 eye strip images from the FRGC database. The horizontal axis represents the number of selected eye candidates (i.e., K), and the vertical axis the percentage of the real eye representations. Please note that the real eye is considered represented if any of those eye candidates is within five pixels from the ground truth. Figure 5.6 shows that only 60 ($K = 60$) candidates per image through the eye color distribution analysis, which account for just 0.85% of the pixels over an image, can represent over 99% of the real eye locations on average. As a result,

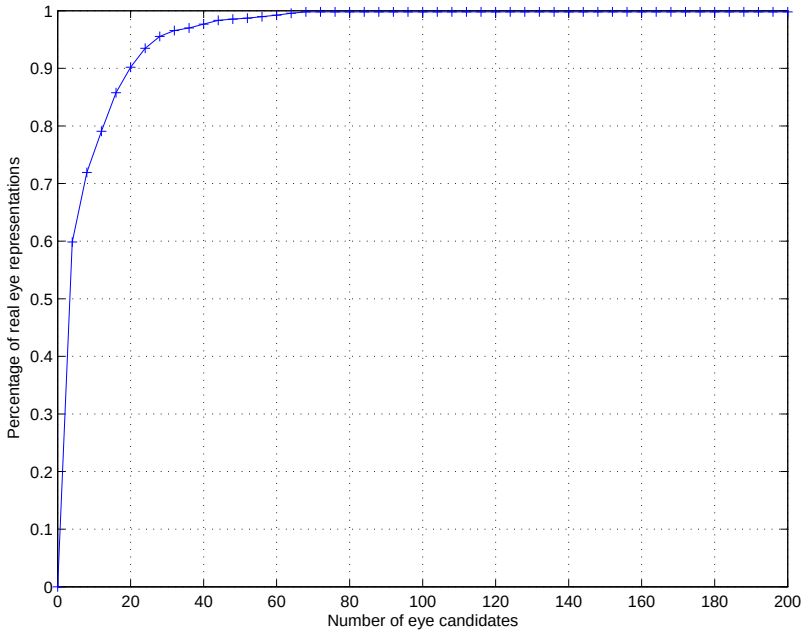
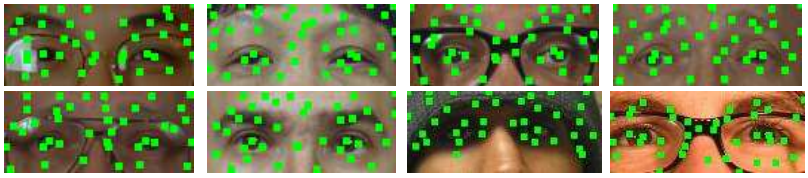


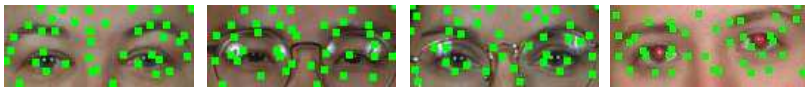
Figure 5.6 The percentage of the real eye representations as the number of the selected eye candidates varies.

the significance of the eye candidate selection stage is that more than 99% (1-0.85%) of the pixels over an image are rejected in this stage whereas only the remaining 1% of the pixels are further processed by the classifier-based validation stage. In comparison with the conventional sliding window method, the candidate selection stage dramatically reduces the number of eye candidates that will be validated by the classifier-based methods and hence significantly improves the efficiency of the eye detection system. Figure 5.7 shows some examples of good and bad eye candidate selection results on the FRGC database. Please note that the result is considered bad if there is no pixel within five pixel distance from the ground truth chosen by the candidate selection method.

Another advantage of the eye candidate selection stage is that it can further improve the efficiency and accuracy of the eye localization system by considering the left and right eye candidates in pair to address the problem of scaling and rotation. Although all the training eye images are rotated upright and normalized to



(a) Good eye candidate selection examples



(b) Bad eye candidate selection examples

Figure 5.7 Examples of good (a) and bad (b) eye candidate selection results.

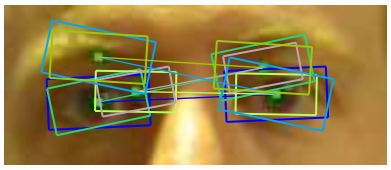


Figure 5.8 Example of the eye pair selection scheme.

a fixed size, the testing images may vary in terms of size and orientation. Even though the traditional methods try to overcome these difficulties by searching a number of predefined scales and orientations, they are either time consuming or the incremental steps too large to cover the continuous scaling and rotation values well. The eye candidate selection method solves the eye scaling and orientation problem by introducing an eye pair selection schema that considers the eye candidates in pairs. In particular, all the eye candidates are divided into the left and the right eye candidates according to their relative positions in the eye strip. The left and right eye candidates are then formed in pairs, and the distance and angle of the binocular line of each eye pair is used to normalize and rotate the eye candidates to the predefined size and the upright orientation, respectively. Figure 5.8 shows an example of the eye pair selection scheme. The eye pair marked by the dark blue rectangles is considered the most suitable scale and orientation. Note that some eye pairs can be removed if the binocular distance is too small or too large, or the angle is too large.

5.5.2 The Eye Candidate Validation Stage

So far, the eye candidate selection stage selects a small number of eye candidates from each image. The next stage will validate these eye candidates to find the real eyes from them. As shown in Figure 5.3, the eye candidate validation stage applies illumination normalization, 2D Haar wavelets for multi-scale image representation, PCA for dimensionality reduction, and the eSVM for detection of the center of the eye. Note that when eSVM is used for classification, the sign of its decision function only determines the class membership of the samples. It is reasonable that a number of candidates around the eye center will be classified into the eye category. In order to determine the final eye center location, the eSVM decision values, instead of the signs of the decision function, are used to first select Q eye candidates with bigger decision values. After the Q candidates are selected, following steps are introduced: first, for each eye candidate, consider an $n \times n$ square centered at this eye candidate; second, compute the summation of the eSVM decision values of all the eye candidates within this $n \times n$ square, and assign this summation to the eye candidate; finally, select the eye candidate with the highest summation as the center of the eye. Note that Q is determined empirically, and the next section will discuss the choice of Q and its effect on the performance of the eSVM classifier.

5.6 Experiments

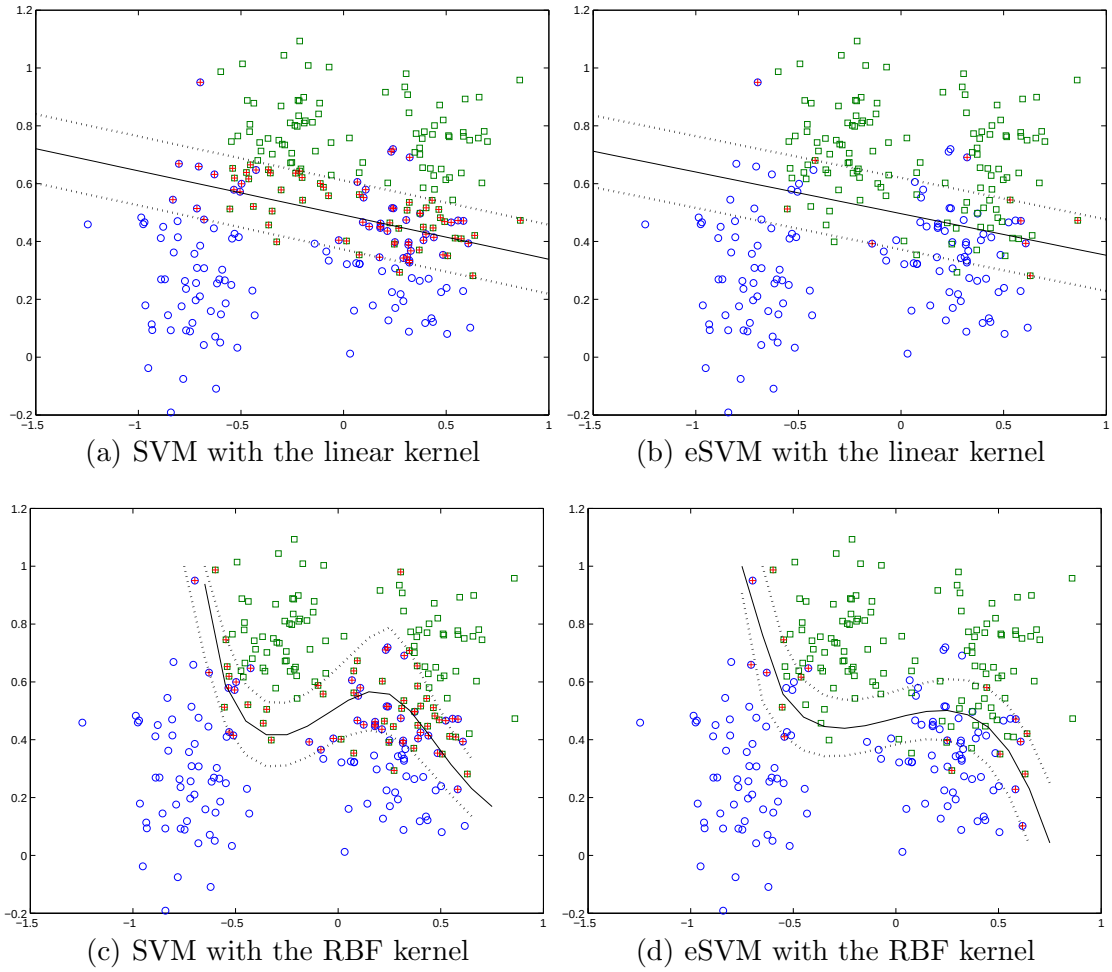
This section evaluates the proposed eSVM method as well as its application to accurate and efficient eye detection. In particular, this section first fully evaluates the eSVM method on a number of data sets that are from different classification problems and widely used by other SVM researchers. The first experiment runs on a synthetic data set [95], [66], [15] to give an intuitive view in the two dimensional space of the eSVM in comparison with the SVM. The second experiment runs on 17 data sets from the UCI Adult benchmark collection and the Web Classification collection [96], [97],

[69] to evaluate the performance of the eSVM over the SVM. The third experiment runs on 6 large-scale data sets [96] to compare the eSVM with other simplified SVM methods, such as the Reduced SVM (RSVM) [46]. The experimental results show that the eSVM method performs better than the conventional soft-margin SVM and the RSVM methods in terms of classification accuracy and computation efficiency. This section then evaluates the eSVM-based eye detection method on the Face Recognition Grand Challenge (FRGC) version 2 database [67] and the FERET database [68]. The experimental results show that the proposed eye localization method achieves real-time eye detection speed and better eye detection performance than some recent eye detection methods.

5.6.1 Evaluation of the eSVM Method

This subsection first evaluates the performance of the eSVM on a synthetic data set [95], the *Ripley* data set, which is also used in [66] and [15]. The advantage of applying this data set comes from the intuitive visualization of the experimental results in the two dimensional space where the data resides, as the data has only two attributes. The *Ripley* data set defines a two-class non-separable problem, and the number of training and testing samples is 250 and 1,000, respectively. In the experiments, two runs are performed with different kernels and parameter settings – one run applies a linear kernel with the regularizing parameter $C = 100$, while the other run applies an RBF kernel $K(x_i, x_j) = e^{-r\|x_i - x_j\|^2}$ with the regularizing parameter $C = 100$ and power $r = 1$. Note that the parameter settings are the same as those in [66] and [15].

Figure 5.9 plots the training samples, the support vectors, and the separating boundaries of the conventional SVM and the eSVM with the linear and RBF kernels, respectively. In particular, Figure 5.9 shows that the proposed eSVM has similar separating boundaries to the conventional SVM using either linear or RBF kernels. The significance of this finding reveals that both eSVM and the conventional SVM



(a) SVM with the linear kernel (b) eSVM with the linear kernel

(c) SVM with the RBF kernel (d) eSVM with the RBF kernel

Figure 5.9 The support vectors and the separating boundaries of the conventional SVM and the eSVM with the linear and RBF kernels on the *Ripley* data set, respectively. The dashed lines/curves depict the ± 1 margins around the separating boundary.

have similar generalization performance. Another finding reveals that the number of support vectors for the eSVM is much smaller than that for the conventional SVM as shown in Figure 5.9. Specifically, Figure 5.9(a) and Figure 5.9(c) show that the support vectors for the conventional SVM, which are represented by red crosses, are the samples that are on the wrong side of their margin. The number of support vectors thus is large due to the fact that many training samples are on the wrong side of their margin for the non-separable problem. In contrast, for the eSVM, only a small number of the training samples on the wrong side of their margin, as shown

Table 5.1 Result Comparisons between the Conventional SVM and the eSVM with the Linear and RBF Kernels .

method	#SV	slope	y-intercept	rate performance	time (ms)
SVM – linear	89	-0.152	0.491	89.7 (897/1000)	14.29
eSVM – linear	10	-0.149	0.496	89.7 (897/1000)	6.42
SVM – RBF	78	-	-	89.7 (897/1000)	37.41
eSVM – RBF	19	-	-	90.2 (902/1000)	11.52

in Figure 5.9(b) and Figure 5.9(d), become support vectors. The number of support vectors for the eSVM is thus much smaller than that for the conventional SVM.

Table 5.1 shows the comparison of the SVM and the eSVM on the number of the support vectors, the slope and the y-intercept of the separating boundaries using the linear kernel, as well as the classification rate and running time for the testing data set. In particular, the eSVM reduces the number of support vectors by 88.76% and 75.64%, when compared with the conventional SVM using the linear and RBF kernels, respectively. Consequently, the running time of the eSVM is also reduced compared with the SVM when same kernel is applied. The similar slope and the y-intercept values of the separating boundaries between the eSVM and the conventional SVM using a linear kernel indicate that they define similar separating boundaries, and hence have comparable generalization performance. Specifically, the classification rate of the eSVM on the testing data set is the same with that of the SVM using the linear kernel, but the classification rate of the eSVM is 0.5% higher than that of the SVM when using the RBF kernel.

This subsection then compares the performance of the eSVM and the SVM for accuracy and efficiency using 17 publicly available data sets – 9 from the UCI Adult benchmark collection and 8 from the Web Classification Collection [96], [97], [69]. The UCI Adult benchmark collection, also known as "Census Income" data set, is designed to predict whether a household has an income greater than $50,000. Each

record in the UCI Adult collection contains 123 features. The Web Classification collection is used for text categorization problem. It collects the keywords from the web page as the attributes and classifies whether a web page belongs to a category or not. Each record in the Web collection contains 300 features extracted from a web page.

The parameters of the conventional SVM are set the same as those in [69], which are chosen to optimize accuracy on a validation set as done in [69]. Specifically, only the RBF kernel $K(x_i, x_j) = e^{-r\|x_i - x_j|^2}$ is used. For the Adult data sets, the regularizing parameter C is set to 1 and the power r of the RBF kernel is set to 0.005. For the Web data sets, the regularizing parameter C is set to 5 and the power r of the RBF kernel is set to 0.005. For fair comparisons, the parameters of the eSVM are set the same as those of the conventional SVM.

Table 5.2 shows the experimental results of the SVM and the eSVM on the 17 data sets in terms of the number of support vectors (#SV), the classification running time, and the classification rate. Table 5.2 first reveals that the number of support vectors of the eSVM is significantly less than that of the SVM, and consequently the classification speed of the eSVM is much faster than that of the SVM. High computational efficiency is the primary contribution of the eSVM over the SVM. As discussed in Section 5.3, the computational efficiency of the SVM depends on the number of support vectors. Given a classification problem, the larger the number of support vector is, the lower the computational efficiency becomes. The eSVM improves the computational efficiency of the SVM by reducing the number of support vectors. Table 5.2 shows that the number of support vectors of the SVM, as the number of training samples increases, varies in a large range from 785 to 12,165 for the Adult data sets and from 231 to 2,547 for the Web data sets, respectively. This is consistent with the analysis in Section 5.3 that the number of support vectors increases dramatically as the problem becomes more complex, as all the training samples on

Table 5.2 Performance Assessment of the SVM and the eSVM

data set	#training samples	#testing samples	#SV		time (s)		rate (%)	
			SVM	eSVM	SVM	eSVM	SVM	eSVM
Adult1a	1,605	30,956	785	63	17.09	1.90	82.66	82.65
Adult2a	2,205	30,296	1,105	72	27.56	2.38	83.46	83.41
Adult3a	3,185	29,376	1,451	87	33.99	2.62	83.55	83.53
Adult4a	4,781	27,780	2,091	97	51.44	2.91	83.74	83.86
Adult5a	6,414	26,147	2,741	111	56.77	2.90	84.06	84.10
Adult6a	11,221	21,341	4,480	156	76.36	3.11	84.06	84.10
Adult7a	16,101	16,461	6,324	184	85.53	2.78	84.43	84.48
Adult8a	22,697	9,865	8,728	225	69.83	2.10	84.89	84.87
Adult9a	32,562	16,281	12,165	265	156.15	3.72	84.89	84.81
Web1a	2,477	47,272	231	65	6.65	2.69	97.33	97.38
Web2a	3,470	46,279	300	65	8.20	2.72	97.36	97.39
Web3a	4,912	44,837	361	95	9.36	3.38	97.44	97.49
Web4a	7,366	42,383	510	116	11.97	3.60	97.73	97.85
Web5a	9,888	39,861	629	120	14.09	3.54	97.80	97.84
Web6a	17,188	32,561	1,079	136	18.40	3.30	98.15	98.17
Web7a	24,692	25,057	1,444	164	18.67	2.84	98.28	98.34
Web8a	49,749	14,951	2,547	289	19.80	2.83	98.44	98.53

the wrong side of their margin become support vectors due to the introduction of the slack variables for the conventional soft-margin SVM method. However, the number of support vectors of the eSVM, as the number of training samples increases, varies in a smaller range from 63 to 265 for the Adult data sets and from 65 to 289 for the Web data sets, respectively. Consequently, the classification speed of the eSVM is much faster than that of the SVM. Take the A9a from the Adult data sets for an example. The number of support vectors of the SVM is 12,165 and the running time is 156.15 seconds. In comparison, the number of support vectors of the eSVM is only 265, which is 97% less than that of the SVM, and the running time is only 3.72 seconds, which is 41 times faster than that of the SVM.

Table 5.3 Data Set Description and Parameter Settings

data set	#training samples	#testing samples	#class	#features	(C, r)		
					SVM	RSVM	eSVM
dna	2,000	1,186	3	180	$2^4, 2^{-6}$	$2^2, 2^{-6}$	$2^4, 2^{-6}$
satimage	4,435	2,000	6	36	$2^4, 2^0$	$2^3, 2^0$	$2^4, 2^0$
letter	15,000	5,000	26	16	$2^4, 2^2$	$2^5, 2^1$	$2^4, 2^2$
shuttle	43,500	14,500	7	9	$2^{11}, 2^3$	$2^{11}, 2^3$	$2^{11}, 2^3$
ijcnn1	49,990	91,701	2	22	$2^1, 2^1$	$2^0, 2^0$	$2^1, 2^1$
protein	17,766	6,621	3	357	$2^1, 2^{-3}$	$2^1, 2^{-3}$	$2^1, 2^{-3}$

Table 5.2 also reveals that the eSVM has comparable classification performance to — sometimes a little bit lower and sometimes a little bit higher than — that of the conventional SVM. As discussed in Section 5.3, the eSVM improves the computational efficiency upon the SVM without sacrificing its generalization performance. The eSVM achieves this by simulating the maximal margin separating boundary of the conventional SVM using fewer support vectors. Therefore, the eSVM maintains a similar separating boundary with the SVM, and subsequently has comparable classification performance with the SVM. Table 5.2 shows that the eSVM has a little bit higher classification rate than that of the SVM for 12 out of the 17 data sets (e.g., A4a and W8a), and a little bit lower rate than that of the SVM for the remaining 5 data sets (e.g., A1a and A9a). The difference on the classification rate between the SVM and the eSVM is in the range of -0.08% (for A9a) and +0.12% (for W4a).

This subsection finally compares the eSVM method with other simplified SVM methods, such as the Reduced SVM (RSVM) [46]. 6 publicly available large-scale data sets [96] are used as done in [46]: *dna*, *satimage*, *letter*, and *shuttle*, *ijcnn1*, and *protein*. The first four data sets are from the Statlog collection, the fifth data set is from the 2001 IJCNN challenge competition, and the last one is from the UCI collection. The feature values of the samples in the data sets are normalized to [-1, 1]

Table 5.4 Performance Assessment of the SVM, the RSVM, and the eSVM (T Stands for Time in Seconds)

data set	SVM			RSVM			eSVM		
	#SV	rate	T	#SV	rate	T	#SV	rate	T
dna	973	95.45	2.39	372	92.33	1.52	503	95.86	1.03
satimage	1,611	91.3	2.50	1,826	90	11.4	299	91.7	0.58
letter	8,931	97.78	28.93	13,928	95.9	149.77	522	97.98	1.73
shuttle	280	99.92	1.65	4,982	99.81	74.82	96	99.95	0.81
ijcnn1	5,200	96.14	227.68	200	96.77	6.36	82	97.02	4.60
protein	17,424	68.51	589.58	596	66.24	35	2,866	69.15	99.38

as done in [46]. Only the RBF kernel $K(x_i, x_j) = e^{-r|x_i - x_j|^2}$ is applied as done in [46]. The regularizing parameter C and the power r of the RBF kernel are set the same as those in [46] for the conventional soft-margin SVM and the RSVM, respectively. For fair comparisons, the parameters of the eSVM are set the same as those of the conventional soft-margin SVM. Table 5.3 shows the number of training samples, the number of testing samples, the number of classes, and the number of features for each data set, as well as the parameter settings for the SVM, the RSVM, and the eSVM, respectively.

Table 5.4 shows the experimental results of the SVM, the RSVM, and the eSVM on the 6 data sets in terms of the number of support vectors (#SV), the classification accuracy (rate), and the running time (T). Note that the results for the RSVM are from the best reported results in [46].

Table 5.4 first reveals that the number of support vectors for the eSVM is much smaller than that for the SVM and the RSVM on average. Although the eSVM generates a little bit more support vectors than the RSVM for the *dna* and *protein* data sets, it outperforms the RSVM in the other four data sets. Take the *letter* data set for an example, the eSVM generates 522 support vectors, while the conventional soft-margin SVM and the RSVM generate 8,931 and 13,928 support

vectors, respectively, in comparison. On the average, the eSVM reduces the number of support vectors by 87.31% and 80.06%, respectively, when compared with the conventional soft-margin SVM and the RSVM methods. As a result, the eSVM method displays higher computational efficiency than both the conventional soft-margin SVM and the RSVM methods. On the average, the eSVM is 7.9 times faster than the SVM. Note that the running time for the RSVM listed in Table 5.4 is from the paper [46], where the RSVM may be implemented and run on different system environment, hence, the detailed comparisons on running time between the RSVM and the eSVM are not made.

Table 5.4 also reveals that the eSVM achieves better classification accuracy than both the conventional soft-margin SVM and the RSVM methods. Note that four different implementations of the RSVM method are reported in [46] with different classification results, and the best results are selected to show in Table 5.4. The experimental results on the 6 data sets demonstrate that the RSVM method reduces the number of support vectors at the expense of accuracy to some extent. The classification accuracy for the RSVM method, on the average, is 1.34% lower than that for the conventional soft-margin SVM. The eSVM, on the other hand, not only significantly reduces the number of support vectors but also improves the classification accuracy. In particular, Table 5.4 shows that the average classification rate of the eSVM is 0.43% higher than that of the SVM method and 1.77% higher than that of the RSVM method, respectively.

5.6.2 Evaluation of the eSVM-based Eye Detection Method

This subsection evaluates the effectiveness and the efficiency of the eSVM-based eye detection method on the FRGC database. The training and testing data sets are the same as those introduced in Section 3.3.2. The detection performance are evaluated in terms of recall and precision. Recall, which is also known as the true positive rate,

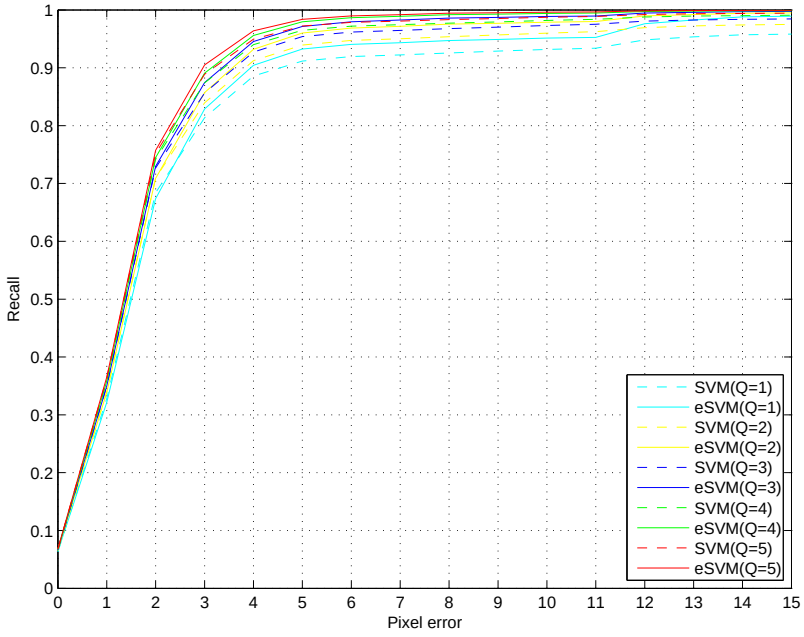
Table 5.5 Efficiency Comparison between the SVM and the eSVM

method	#SV	detection time (s)	detection time per image (s)
Haar-SVM	9,615	38,072	2.98
Haar-eSVM	267	1,916	0.15

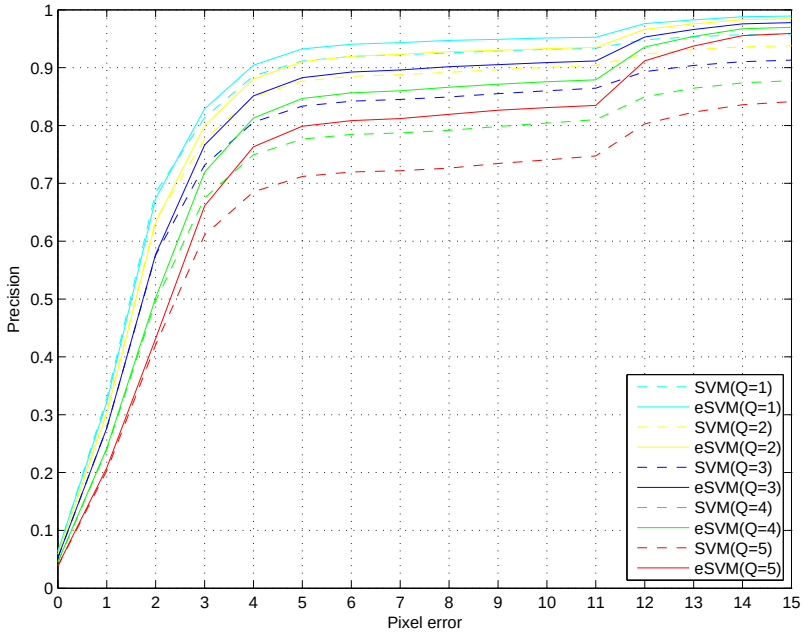
is defined as the number of the true positives divided by the sum of the true positives and false negatives. Precision is defined as the number of the true positives divided by the sum of the true positives and false positives. A robust detection system normally possesses the property of both high recall and precision.

The parameters of the eye localization system are optimized on a validation set by considering both accuracy and efficiency. Specifically, 60 candidates are chosen through the eye candidate selection stage. 2D Haar basis functions for V^5 are used to derive the 2D Haar wavelet features. As $V^5 = V^0 \oplus W^0 \oplus W^1 \oplus W^2 \oplus W^3 \oplus W^4$, the size of the 2D Haar wavelet features is 1,024. 80 eigenvectors out of 1,024 Haar features are derived using the PCA approach. Only the RBF kernel $K(x_i, x_j) = e^{-r\|x_i - x_j|^2}$ is used. The parameter r is set to 0.0125. The regularizing parameter C is set to 1.

As discussed in Section 5.3, the main advantage of the eSVM over the SVM is the computational efficiency. Therefore, this subsection first evaluates the computational efficiency of the eSVM in comparison with the SVM. Table 5.5 shows the comparison of the computational efficiency between the SVM and the eSVM using the FRGC database. Actually, Table 5.5 reveals that the eSVM significantly reduces the number of support vectors and as a result increases the detection speed. In particular, the number of support vectors of the eSVM is 97.23% less that that of the SVM. As the number of support vectors decreases, the detection time is reduced. The SVM takes 2.98 seconds (0.33 images per second) on average to process each image. The eSVM, in comparison, significantly improves the computational efficiency to real-time eye



(a) Recall



(b) Precision

Figure 5.10 Recall and precision of the SVM- and the eSVM-based methods as Q varies.

detection. Specifically, the eSVM, which takes 0.15 seconds (6.67 images per second) on average to process each image, is 20 times faster than the SVM.

This subsection then evaluates the classification performance under the difference choice of Q between the SVM and the eSVM classifiers. As discussed in the end of Section 5.5.2, the first Q left and right eye candidates with the largest decision values of the SVM (or eSVM) classifier are treated as the detected eyes. Figure 5.10 shows the comparison of eye detection performance of the SVM and the eSVM in terms of recall and precision using the FRGC database, respectively. The performance is evaluated as the Q varies from 1 to 5. When $Q > 5$, although recall can further increase, precision decreases dramatically. The horizontal axis represents the localization pixel errors, and the vertical axis denotes the accumulated distribution, which means recall (or precision) of eyes with smaller pixel error than the corresponding horizontal value. Please note that when $Q = 1$, the recall is equal to the precision according to their definition.

Figure 5.10 shows that the performance of the eSVM tends to be better than that of the SVM. In terms of recall, the performance of the eSVM is higher than SVM on average by 1.28% when $Q = 1$. As Q increases, the difference of the performance between SVM and eSVM becomes smaller. When $Q = 5$, the performance of the eSVM is only 0.42% higher than that of the SVM on average. In terms of precision, the difference in performance between SVM and eSVM is more significant. When $Q = 1$, the performance of the eSVM is 1.28% higher than the SVM on average. The difference increases as Q increases. When $Q = 5$, the performance of the eSVM is 5.25% higher than SVM on average.

Figure 5.10 reveals as well the relationship between the value of Q and the eye detection performance. In particular, Figure 5.10 shows that as the value of Q becomes larger, recall increases and precision decreases. As a matter of fact, if the value of Q is further increased, recall of detections within five pixels of the ground truth can be more than 99%. Precision, however, decreases dramatically to as low as 65%.

Table 5.6 Performance of the SVM and the eSVM within Five Pixel Localization Error

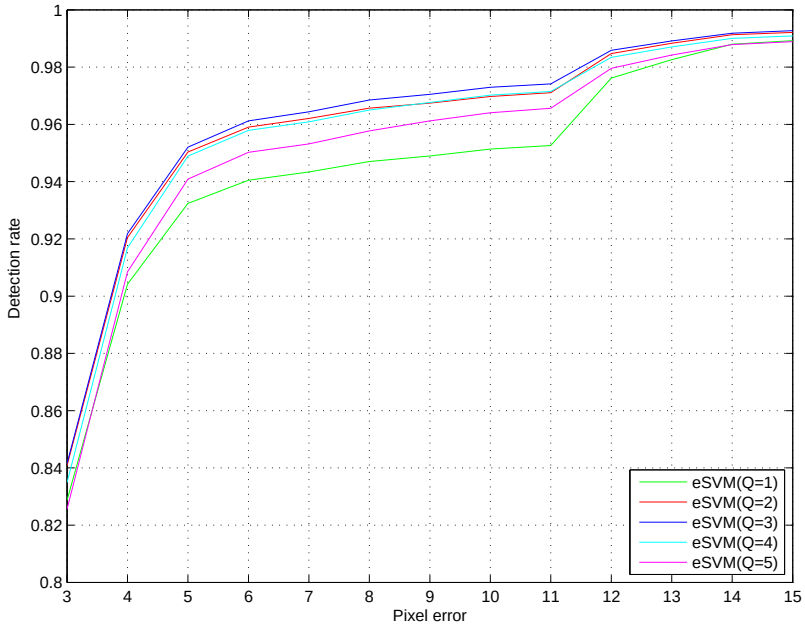
methods	Recall					Precision				
	Q=1	Q=2	Q=3	Q=4	Q=5	Q=1	Q=2	Q=3	Q=4	Q=5
SVM	91.16	93.94	95.40	96.48	97.21	91.16	87.67	83.36	77.65	71.19
eSVM	93.24	96.02	97.14	97.94	98.39	93.24	91.05	88.27	84.68	79.88

Table 5.7 Performance of Final Eye Detection within Five Pixel Localization Error under Different Q

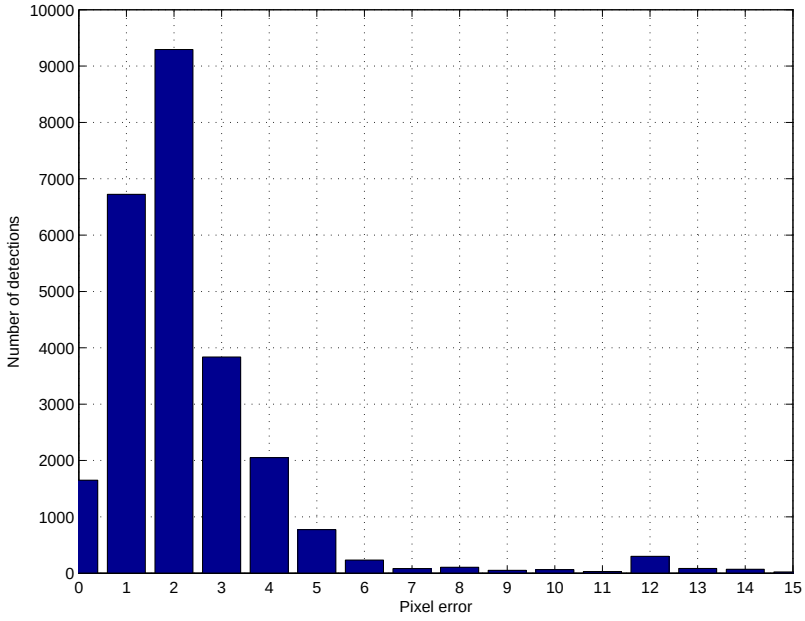
method	Q=1	Q=2	Q=3	Q=4	Q=5
Haar-eSVM	93.24	95.03	95.21	94.89	94.09

If the eyes are considered correctly detected when the Euclidean distance between the detected eye center and the ground truth is within five pixels, Table 5.6 lists the specific recall and precision for the SVM and the eSVM using the FRGC database. Table 5.6 only lists the performance for $Q <= 5$, since precision will dramatically decrease when $Q > 5$. First, Table 5.6 shows that the performance of the eSVM is better than that of the SVM. Second, Table 5.6 demonstrates the promising performance of the eye detection method: for the Haar-eSVM, for an example, recall is 96.02% whereas precision is 91.05%, when $Q = 2$.

This subsection next evaluates the final eye-center localization performance using the FRGC database following the steps introduced at the end of Section 5.5.2. In particular, the Haar-eSVM is applied, which yields better eye detection performance as shown in Figure 5.10 and Table 5.6. Only the recall criterion (i.e., detection rate) is applied, since precision is equal to recall in the case of the single detection for each eye. There are two parameters in the final eye localization: one is the size of the square (i.e., n) and the other is the value of Q, which means how many multiple detections are allowed to choose the final eye center location. Based on the size of the pupil of the normalized training eye sampled, n is set to five. For Q, the optimal



(a)



(b)

Figure 5.11 (a) Performance comparison of the final eye localization under different Q. (b) Distribution of eye localization pixel errors for final eye localization when $Q = 3$.

choice is searched between one and five, since precision will dramatically decrease if Q is greater than five. Figure 5.11(a), which shows the final detection rate as

Q varies from one to five, indicates that the eye detection performance peaks when $Q = 3$. Table 5.7 shows specific final eye detection rate for each Q value if the eye is considered to be detected correctly when the Euclidean distance between the detected eye center and the ground truth is within five pixels. The eye detection performance when $Q = 3$ is 95.21%.

Therefore, three left and right eye candidates are used, respectively, to determine the final eye location. Figure 5.11(b) shows the distribution of the Euclidean distance of detected eyes compared with the ground truth. The average Euclidean distance between the detected eyes and the ground truth is about 2.61 pixels. Figure 5.12 shows some examples of the eye detection results using the eSVM-based method.

5.6.3 Comparison with Recent Methods

In order to assess the robustness of the proposed eSVM based eye detection method and compare with some recent eye detection methods, experiments on another color face database, the FERET database [68], are implemented. The FERET database contains over 3,300 frontal color face images of nearly 1,000 subjects.

The methods that are compare with include the HOG descriptor based method by Monzo et al. [60], the hybrid classifier method by Jin et al. [38], the general-to-specific method by Campadelli et al. [12], and a facial identification software — Verilook [98], [60]. All the above methods applied the normalized error to evaluate

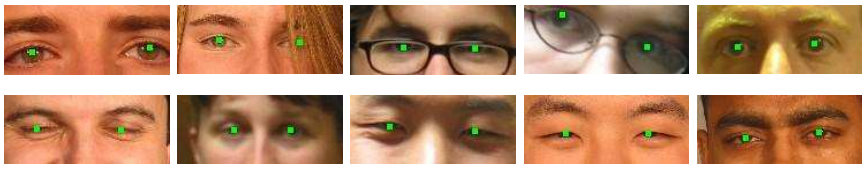


Figure 5.12 Examples of the eye detection results using the eSVM based eye detection method.

Table 5.8 Comparisons of the Eye Detection Performance for Different Methods on the FERET Database (e Stands for the Normalized Error)

method	$e \leq 0.05$	$e \leq 0.10$	$e \leq 0.25$
Monzo [60]	78.00%	96.20%	99.60%
Jin [38]	55.10%	93.00%	99.80%
Campadelli [12]	67.70%	89.50%	96.40%
Verilook [98]	74.60%	96.80%	99.90%
the eSVM based method	82.22%	94.25%	98.82%

the performance, which is defined as the detection pixel error normalized by the interocular distance.

Table 5.8 shows the performance comparison between the eSVM based method and the methods mentioned above for the normalized error of 0.05, 0.10, and 0.25, respectively. Table 5.8 reveals that for the normalized error of 0.05, the detection accuracy of the proposed eSVM based method is 4.22% higher than the best result reported by the other methods; for the normalized errors of 0.10 and 0.25, the detection accuracy of the proposed eSVM based method is 2.55% and 1.08% lower than the best results reported by other methods, respectively. Note that the normalized errors of 0.10 and 0.25 are considered loose criteria which may not be appropriate for evaluating the precise eye detection methods. As a matter of fact, the normalized error of 0.05 is a strict criterion and appropriate for evaluating the precise eye detection methods.

Regarding the efficiency, not many papers report the execution time of their methods. As a matter of fact, speed is an important factor in the real-word application of an eye localization method. Campadelli [12] presented an SVM based eye localization method and reported the execution time of 12 seconds per image (Java code running on a Pentium 4 with 3.2GHz). In comparison, the average execution time of the proposed method is only 0.15 seconds per image due to the application of the eSVM (MATLAB code running on a Pentium 3 with 3.0GHz). In fact, the execution

time can be further significantly reduced if some faster programming languages (like Java or C/C++) and multi-thread techniques are applied.

5.7 Conclusion

This chapter first proposes an efficient Support Vector Machine (eSVM) to address the inefficiency problem of the conventional SVM. The eSVM, which introduces a single value for all the slack variables corresponding to the training samples on the wrong side of their margin, defines a much smaller set of support vectors and hence improves the computational efficiency without sacrificing the generalization performance. A modified Sequential Minimal Optimization (SMO) algorithm is then presented to solve the large Quadratic Programming (QP) problem defined in the eSVM. This chapter then presents an accurate and efficient eye detection method using the eSVM method. This eye detection method consists of the eye candidate selection stage and the eye candidate validation stage. The selection stage selects the eye candidates in an image through a process of eye color distribution analysis in the YCbCr color space. The validation stage applies first 2D Haar wavelets for multi-scale image representation, the PCA for dimensionality reduction, and finally the eSVM for classification. Experiments on several diverse data sets show that the eSVM significantly improves the computational efficiency upon the conventional SVM while achieving comparable generalization performance to or higher performance than the SVM. Furthermore, experimental results on the FRGC and the FERET database reveal that the proposed eye detection method achieves real-time eye detection speed and better eye detection performance than some recent eye detection methods.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This dissertation focuses on eye detection using various discriminatory features and a new efficient Support Vector Machine (eSVM). The main contributions of this dissertation are listed below:

- A new Discriminant Component Analysis (DCA) method, which improves upon the popular Principal Component Analysis (PCA) method, is proposed to extract discriminatory features for eye detection. The PCA method can derive the optimal features for data representation but not for classification. The DCA method, in contrast, can derive the discriminatory features in the whitened PCA space for two-class classification problems. The DCA features thus are capable of improving the discriminating power of the PCA features and enhancing the eye detection performance.

- A clustering-based Discriminant Analysis (CDA) method, which improves upon the Fisher Linear Discriminant (FLD) method, is proposed to extract discriminatory features for eye detection. One major disadvantage of the FLD is that it may not be able to extract adequate features in order to achieve satisfactory performance, especially for two class problems. Three CDA models, CDA-1, CDA-2, and CDA-3, are proposed, which take the full advantage of the k-means clustering technique. For every CDA model a new between-cluster scatter matrix is defined. The CDA method thus can derive adequate features to achieve satisfactory performance for eye detection. Furthermore, the clustering nature of the three CDA models and the nonparametric nature of the CDA-2 and -3 models can further improve the detection performance upon the conventional FLD method.

- Comparative assessment of five types of discriminatory features derived from five popular image representations is presented for the problem of eye detection.

- A new efficient Support Vector Machine (eSVM) is proposed for eye detection that improves the computational efficiency of the conventional SVM without sacrificing the generalization performance. A modified Sequential Minimal Optimization (SMO) algorithm is then presented to solve the large Quadratic Programming (QP) problem defined in the eSVM. The eSVM is then applied to the problem of eye detection and achieves real-time eye detection speed and better eye detection performance than some recent methods.

The future work lies in the following two aspects:

- Regarding the clustering-based Discriminant Analysis (CDA) method, there are two further concerns. First, this dissertation simply uses the same number of clusters for each class when defining the within-class and between-class scatter matrices. However, the real world applications may contain unbalanced data for each class (e.g., 200 training samples for one class whereas 20,000 for another) and unbalanced inherent multi-models for each class (e.g., 10 inherent multi-models for one class whereas 200 for another). Therefore, the CDA method may be further improved if some advanced clustering techniques can be applied to automatically build up the unbalanced clusters for each class separately. Second, for the CDA-2 and CDA-3, even though the between-class scatter matrices follow the nonparametric nature, the within-class scatter matrices still follow the parametric nature. It is worthwhile to explore the effect of the nonparametric form of the within-class scatter matrix on the performance of the CDA.

- Regarding the efficient Support Vector Machine (eSVM), one direction of the future work focuses on exploring the relationship between different kernel functions and the performance of the eSVM. This dissertation mainly uses the RBF

kernel to evaluate the performance as recommended in [83]. However, it is shown that different kernel functions often lead to different classification results. Currently, kernel selection has become a very popular research area and much work has been carried out [3] [21] [4]. The future work concentrates on how the different kernels affect the performance of the eSVM and what kernel is the optimal one for the eSVM.

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face descriptor with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(28):2037–2041, 2006.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *8th European Conference on Computer Vision (ECCV'04)*, Prague, Czech Republic, May 11-14, 2004.

[3] C. Allauzen, C. Cortes, and M. Mohri. Large-scale training of svms with automata kernels. In *15th International Conference on Implementation and Application of Automata*, Winnipeg, MB, Canada, August 12-15, 2010.

[4] C. Allauzen, C. Cortes, and M. Mohri. SVM optimization for lattice kernels. In *Mining and Learning with Graphs*, 2010.

[5] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas. An eye detection algorithm using pixel to edge information. In *the 2nd International Symposium on Control, Communications, and Signal Processing*, Marrakech, Morocco, March 13-15, 2006.

[6] L. Bai, L. Shen, and Y. Wang. A novel eye location algorithm based on radial symmetry transform. In *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, August 20-24, 2006.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[8] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms i. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

[9] M. Bressan and J. Vitrià. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[10] C.J.C. Burges. Simplified support vector decision rule. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML '96)*, Bari, Italy, July 3-6, 1996.

[11] C.S. Burrus, R.A. Gopinath, and H. Guo. *Introduction to wavelets and wavelet transforms: A Primer*. Prentice-Hall, 1998.

[12] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *2006 British Machine Vision Conference (BMVC'06)*, Edinburgh, UK, September 4-7, 2006.

[13] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye and mouth localization. *International Journal of Pattern Recognition and Aritificial Intelligence*, 23(3):359–377, 2009.

[14] L. Cao. Support vector machines experts for time series forecasting. *Neurocomputing*, 51:321–339, 2003.

[15] J.H. Chen and C.S. Chen. Reducing svm classification time using multiple mirror classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(2):1173–1183, 2004.

[16] J. Chen and C. Chen. Reducing svm classification time using multiple mirror classifers. *IEEE Transcations on Systems, Man, and Cybernetics*, 34(2):1173–1183, April 2004.

[17] L. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 20-26, 2005.

[18] P.H. Chen, C.J. Lin, and B. Scholkopf. A tutorial on υ-support vector machines. *Applied Stochastic Models in Business and Industry*, 21:111–136, 2005.

[19] S. Chen and C.J. Liu. Eye detection using color information and a new efficient svm. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS'10)*, Washington DC, USA, September 27-29, 2010.

[20] S. Chen and C. Liu. Fast eye detection using different color spaces. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC'11)*, Anchorage, Alaska, October 9-12, 2010.

[21] C. Cortes. Invited talk: Can learning kernels help performance? In *26th ACM International Conference on Machine Learning (ICML'09)*, Montreal, Quebec, Canada, June 14-18, 2009.

[22] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In 15^{th} *British Machine Vision Conference (BMVC'04)*, London, England, September 7-9, 2004.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 20-26, 2005.

[24] M. A. Davenport, R. G. Baraniuk, and C. Scott. Tuning support vector machines for minimax and neyman-pearson classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1888–1898, 2010.

[25] J.Y. Deng and F.P. Lai. Region-based template deformation and masking for eye-feature extraction and description. *Pattern Recognition*, 30(3):403–419, March 1997.

[26] M. Eckhardt, I. Fasel, and J. Movellan. Towards practical facial feature detection. *Internatioanl Journal of Pattern Recognition and Artificial Intelligence*, 23(3):379–400, 2009.

[27] M. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG'06)*, Southampton, UK, April 10-12, 2006.

[28] G.C. Feng and P.C. Yuan. Various projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters*, 19(9):899–906, 1998.

[29] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[30] K. Fukunaga. Introduction to statistical pattern recognition, 1990. Academic Press.

[31] K. Fukunaga and J.M. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):671–678, 1983.

[32] Z. Gu, J. Yang, and L. Zhang. Push-pull marginal discriminant analysis for feature extraction. *Pattern Recognition Letters*, 31(15):2345–2352, 2010.

[33] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

[34] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, and H. Kälviäinen. Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model. In *6th IEEE International Conference on Automatic Face and Gesture Recognition(FG'06)*, Seoul, Korea, May 17-19, 2004.

[35] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, and J. Matas. Feature-based affine-invariant localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1490–1495, 2005.

[36] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *2001 IEEE International Conference on Computer Vision (ICCV'01)*, Vancouver, British Columbia, Canada, July 7-14, 2001.

[37] O. Jesorsky, K.J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *Third International Conference on Audio- and Video-Based Biometric Person Authentication*, Halmstad, Sweden, June 6-8, 2001.

[38] L.Z. Jin, X.H. Yuan, S. Satoh, J.X. Li, and L.Z. Xia. A hybrid classifier for precise and robust eye detection. In *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, August 20-24, 2006.

[39] T. Joachims, F. Informatik, and L. Viii. Text categorization with support vector machines: Learning with many relevant features, 1997.

[40] F. Jorge, S. Carvalho, J. Manuel, and R. S. Tavares. Eye detection using a deformable template in static images. In *1st ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, Porto, Portugal, October 17-19, 2007.

[41] B. Kroon, S. Maas, S. Boughorbel, and A. Hanjalic. Eye localization in low and standard definition content with application to face matching. *Computer Vision and Image Understanding*, 113(4):921–933, 2009.

[42] Y.J. Lee and O.L. Mangasarian. Rsvm: Reduced support vector machines. In *1st SIAM International Conference on Data Mining*, Chicago, IL, April 5-7, 2001.

[43] Y. Li, X.L. Qi, and Y.J. Wang. Eye detection by using fuzzy template matching and feature-parameter-based judgement. *Pattern Recognition Letters*, 22(10):1111–1124, August 2001.

[44] Z. Li, D. Lin, and X. Tang. Nonparametric discriminant analysis for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):755–761, 2009.

[45] Z. Li, W. Liu, D. Lin, and X. Tang. Nonparametric subspace analysis for face recognition. In *2005 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, June 20-26, 2005.

[46] K.M. Lin and C.J. Lin. A study on reduced support vector machine. *IEEE Transactions on Neural Networks*, 14(6):1449–1559, 2003.

[47] C. Liu. A Bayesian discriminating features method for face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):725–740, 2003.

[48] C. Liu. Learning the uncorrelated, independent, and discriminating color spaces for face recognition. *IEEE Transactions on on Information Forensics and Security*, 3(2):213–222, 2008.

[49] C. Liu and H. Wechsler. Robust coding schemes for indexing and retrieval from large face databases. *IEEE Transactions on on Image Processing*, 9(1):132–137, 2000.

[50] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transactions on on Image Processing*, 11(4):467–476, 2002.

[51] C. Liu and J. Yang. ICA color space for pattern recognition. *IEEE Transactions on Neural Networks*, 20(2):248–257, 2009.

[52] C. Liu. Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 28(5):725–737, 2006.

[53] C. Liu. The Bayes decision rule induced similarity measures. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 29(6):1086–1090, 2007.

[54] Z. Liu and C. Liu. Fusion of the complementary discrete cosine features in the yiq color space for face recognition. *Computer Vision and Image Understanding*, 111(3):249–262, 2008.

[55] Z. Liu and C. Liu. A hybrid color and frequency features method for face recognition. *IEEE Transactions on Image Processing*, 17(10):1975–1980, 2008.

[56] Z. Liu and C. Liu. Fusion of color, local spatial and global frequency information for face recognition. *Pattern Recognition*, 43(8):2882–2890, 2010.

[57] M. Loog and R. Duin. Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):732–739, 2004.

[58] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[59] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[60] D. Monzo, A. Albiol, J. Sastre, and A. Albiol. Precise eye localization using hog descriptors. *Machine Vision and Applications*, 22(3):471–480, 2011.

[61] T. Moriyama, T. Kanade, J. Xiao, and J.F. Cohn. Meticulously detailed eye region model and its application to analysis of facial images. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 28(5):738–752, 2006.

[62] M.H. Nguyen, J. Perez, and F.D.L.T. Frade. Facial feature detection with optimal pixel reduction svm. In *8th IEEE International Conference on Automatic Face and Gesture Recognition (FG'08)*, Amsterdam, The Netherlands, September 17-19, 2008.

[63] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. 2d cascaded adaboost for eye localization. In *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, August 20-24, 2006.

[64] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[65] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[66] E. Osuna and F. Girosi. Reducing the run-time complexity of support vector machines, 1998.

[67] P.J. Phillips, P.J. Flynn, and T. Scruggs. Overview of the face recognition grand challenge. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, June 20-26, 2005.

[68] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[69] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[70] J.C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998. MIT Press.

[71] P. Porwik and A. Lisowska. The haarwavelet transform in digital image processing: Its status and achievements. *Machine graphics & vision*, 13(1):79–98, 2004.

[72] X. Qiu and L. Wu. Face recognition by stepwise nonparametric margin maximum criterion. In *10th IEEE International Conference on Computer Vision (ICCV'05)*, Beijing, China, October 17-20, 2005.

[73] X. Qiu and L. Wu. Two-dimensional nearest neighbor discriminant analysis. *Neurocomputing*, 70(13-15):2572–2575, 2007.

[74] G. Rätsch, S. Mika, B. Schölkopf, and K. Müller. Constructing boosting algorithms from svms: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, 2002.

[75] S. Romdhani, B. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. In *IEEE International Conference on Computer Vision (ICCV'01)*, Vancouver, British Columbia, Canada, July 7-14, 2001.

[76] J. Rurainsky and P. Eisert. Eye center localization using adaptive templates. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, Washington, DC, USA, June 27 - July 2, 2004.

[77] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.R. Muller, G. Ratsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[78] P. Shih and C.J. Liu. Face detection using discriminating feature analysis and support vector machine. *Pattern Recognition*, 39:260–276, 2006.

[79] P. Shih and C. Liu. Comparative assessment of content-based face image retrieval in different color spaces. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(7):873–893, 2005.

[80] D.L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

[81] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, Anchorage, Alaska, USA, June 24-26, 2008.

[82] V. Vapnik. The nature of statistical learning theory, 1995. Springer-Verlag, New York, NY.

[83] V. Vapnik. Statistical learning theory, 1998. Wiley, New York, NY.

[84] V. N. Vapnik. The nature of statistical learning theory, 2000.

[85] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, HI, USA, December 8-14, 2001.

[86] A.J. Wan Mohd Khairosfaizal, W.M.K.and Nor'aini. Eye detection in facial images using circular hough transform. In *2009 IEEE Conference on Signal Processing and Its Application*, Singapore, May 15-17, 2009.

[87] J. Wang and H.L. Zhao. Eye detection based on multi-angle template matching. In *International Conference on Image Analysis and Signal Processing*, Taizhou, China, April 11-12, 2009.

[88] P. Wang and Q. Ji. Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding*, 105(2):99–111, 2007.

[89] X. Wang and X. Tang. Random sampling lda for face recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Washington DC, June 27 - July 2, 2004.

[90] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, 2004.

[91] J.X. Wu and Z.H. Zhou. Efficient face candidates selector for face detection. *Pattern Recognition*, 36(5):1175–1186, 2003.

[92] Z.H. Zhou and X. Geng. Projection function for eye detection. *Pattern Recognition*, 37(5):1049–1056, 2004.

[93] M. Zhu and A.M. Martínez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1274–1286, 2006.

[94] Z.W. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.

[95] Ripley dataset: http://www.stats.ox.ac.uk/pub/PRNN/.

[96] NTU dataset: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[97] UCI dataset: http://www.ics.uci.edu/~mlearn/MLRepository.html/.

[98] Verilook SDK: http://www.neurotechnology.com/verilook.html.