

# Robust object detection with efficient features and effective classifiers

**Author:**

Paisitkriangkrai, Sakrapee

**Publication Date:**

2011

**DOI:**

<https://doi.org/10.26190/unsworks/23730>

**License:**

<https://creativecommons.org/licenses/by-nc-nd/3.0/au/>

Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/50894> in <https://unsworks.unsw.edu.au> on 2024-04-23

# **Robust Object Detection with Efficient Features and Effective Classifiers**



**Sakrapee Paisitkriangkrai**

**School of Computer Science and Engineering**

**The University of New South Wales**

A thesis submitted in fulfilment  
of the requirements for the degree of

*Doctor of Philosophy*

August 2011

**PLEASE TYPE****THE UNIVERSITY OF NEW SOUTH WALES  
Thesis/Dissertation Sheet**Surname or Family name: **PAISITKRIANGKRAI**First name: **SAKRAPEE**

Other name/s:

Abbreviation for degree as given in the University calendar: **PhD**School: **COMPUTER SCIENCE AND ENGINEERING**Faculty: **FACULTY OF ENGINEERING**Title: **Robust Object Detection with Efficient Features and Effective Classifiers****Abstract 350 words maximum: (PLEASE TYPE)**

This thesis contains three main novel contributions that advance the state of the art in object detection. The first contribution focuses on a real-time pedestrian detector using a combination of Haar-like features and covariance features. Unlike the original work of Tuzel et al., where the feature selection and weak classifier training are performed on the Riemannian manifold, weak classifiers are trained in the Euclidean space for faster computation. To this end, a novel approach based on AdaBoost with weighted Fisher Linear Discriminant Analysis (FLDA) based weak classifiers is designed. To further accelerate the detection, a faster strategy, known as a multiple-layer boosting with heterogeneous features, is adopted to exploit the efficiency of Haar-like features and the discriminative power of covariance features. Experimental results show that by combining Haar-like and covariance features, the efficiency of final detectors improves by an order of magnitude with a slight drop in the detection performance.

The second contribution reveals the drawback of commonly used AdaBoost and a more effective approach, termed Boosted Greedy Sparse Linear Discriminant Analysis (BGS LDA), is proposed. BGS LDA exploits a class-separability criterion of LDA and a sample re-weighting property of boosting. Experimental results demonstrate an improvement in the detection performance compared to the original AdaBoost framework. This new finding provides a significant opportunity to argue that AdaBoost and its variants are not the only method that can achieve a high classification accuracy in a high dimensional problem, such as object detection.

The last contribution points out the drawback of offline object detection frameworks and an efficient online framework is proposed. Unlike many existing online boosting algorithms, which apply exponential or logistic loss, the proposed online algorithm makes use of LDA's learning criterion that not only aims to maximize the class-separation criterion but also incorporates the asymmetrical property of training data distributions. The new approach provides a better alternative to online boosting algorithms in the context of training a visual object detector. Experimental results on handwritten digits and face data sets show that object detection tasks benefit significantly when trained in an online manner.

Finally, this thesis concludes with a discussion and future works, which explore potential advances in the learning of feature descriptors, cascade classifiers as well as online object detectors.

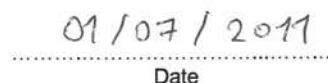
**Declaration relating to disposition of project thesis/dissertation**

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstracts International (this is applicable to doctoral theses only).

  
Signature

  
Witness

  
Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

**FOR OFFICE USE ONLY**

Date of completion of requirements for Award:

**THIS SHEET IS TO BE GLUED TO THE INSIDE FRONT COVER OF THE THESIS**

---

## **Originality Statement**

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgment is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the projects design and conception or in style, presentation and linguistic expression is acknowledged.

Sakrapee Paisitkriangkrai



## **Copyright Statement**

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or hereafter known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I also authorise University Microfilms to use the abstract of my thesis in Dissertations Abstract International (this is applicable to doctoral theses only). I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.

Sakrapee Paisitkriangkrai





## **Authenticity Statement**

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.

Sakrapee Paisitkriangkrai



## Abstract

This thesis contains three main novel contributions that advance the state of the art in object detection. The first contribution focuses on a real-time pedestrian detector using a combination of Haar-like features and covariance features. Unlike the original work of Tuzel *et al.*, where the feature selection and weak classifier training are performed on the Riemannian manifold, weak classifiers are trained in the Euclidean space for faster computation. To this end, a novel approach based on AdaBoost with weighted Fisher Linear Discriminant Analysis (FLDA) based weak classifiers is designed. To further accelerate the detection, a faster strategy, known as a multiple-layer boosting with heterogeneous features, is adopted to exploit the efficiency of Haar-like features and the discriminative power of covariance features. Experimental results show that by combining Haar-like and covariance features, the efficiency of final detectors improves by an order of magnitude with a slight drop in the detection performance.

The second contribution reveals the drawback of commonly used AdaBoost and a more effective approach, termed Boosted Greedy Sparse Linear Discriminant Analysis (BGSLDA), is proposed. BGSLDA exploits a class-separability criterion of LDA and a sample re-weighting property of boosting. Experimental results demonstrate an improvement in the detection performance compared to the original AdaBoost framework. This new finding provides a significant opportunity to argue that AdaBoost and its variants are not the only method that can achieve a high classification accuracy in a high dimensional problem, such as object detection.

The last contribution points out the drawback of offline object detection frameworks and an efficient online framework is proposed. Unlike many

existing online boosting algorithms, which apply exponential or logistic loss, the proposed online algorithm makes use of LDA's learning criterion that not only aims to maximize the class-separation criterion but also incorporates the asymmetrical property of training data distributions. The new approach provides a better alternative to online boosting algorithms in the context of training a visual object detector. Experimental results on handwritten digits and face data sets show that object detection tasks benefit significantly when trained in an online manner.

Finally, this thesis concludes with a discussion and future works, which explore potential advances in the learning of feature descriptors, cascade classifiers as well as online object detectors.

I would like to dedicate this thesis to my parents for their love, patience,  
understanding and never-ending support.



## **Acknowledgements**

First and foremost I want to express my deepest gratitude to my thesis supervisor, A/Prof. Jian Zhang, and co-supervisor, Dr. Chunhua Shen, for their devotion during my PhD study. This thesis would not have been completed without their continued inputs, helps, commitments and efforts.

Dr. Zhang has always been a great supervisor. Not only he introduced me to research but he also taught me how to see things in different perspectives. He imbibed in me skills that cannot be learned from a book, the determination and the need for perfection in both research and life. His encouragement and invaluable teaching will always be my source of inspiration. I thank him for offering me the opportunity to do a PhD on such interesting and challenging topics.

I am very grateful to have Dr. Shen as my co-supervisor. I have learned a great deal on how to conduct a world class research under his supervision. He always initiated and stimulated my research thoughts. Despite his workload, he would always have time for me and help revise my writings. I have learnt enormously from his invaluable comments and useful suggestions. His thoughtful guidance, enthusiasm and passion for research will always be my source of inspiration in the future.

I am also grateful to Dr. Tao Mei and Dr. Xian-Sheng Hua for giving me an opportunity to do my internship study at Microsoft Research Asia (MSRA) and their continued help and support during my visit. Working as an intern at MSRA had provided me with the greatest opportunity to meet many world class researchers and research students and an incredibly wonderful experience that I will always remember.

I would also like to thank Dr. Wei Wang, Dr. Peter Cai and Dr. Ying Zhang for their valuable comments and making time available for my PhD annual

reviews. Without their guidance, this thesis could have been headed in the wrong direction. I would also like to thank Ms. Alpana Lal, Ms. Chirsanthi Theodosakis and Ms. Fatima Portada for their helps with administrative works at NICTA and the university.

During my PhD study, I have received considerable helps from many friends and colleagues both at NICTA, UNSW and MSRA. I would like to thank them for making my PhD study fun and interesting, especially to Pranam Janney, Nobuyuki Morioka, Jie Xu, Gunawan Herman, Jun Yang for putting up with me during these past four years. We shared many interesting discussions and thoughts. I would also like to thank friends in Microsoft Research Asia (MSRA), Bo Gang, Ike Cheng, Kaiyuan, Li Chang for their help, insightful discussions and suggestions in various research topics during my internship study in Beijing. Furthermore, I would also like to thank Bang Zhang, Carlos Aydos, Chen Cai, Tue Hue Thi, Weihong Wang, Werayut Saesue, Worapan Kusakunniran and Zhidong Li for the lively working atmosphere. There are also many high school and university friends from the past and present, who have directly and indirectly contributed toward my understanding in both research and life. I would like to thank them from the bottom of my heart, for making me what I am today.

I am also very grateful for the financial support I received in the form of scholarships during my PhD study from the Australian Postgraduate Award (APA), National ICT Australia (NICTA) and Faculty of Engineering, The University of New South Wales.

Last, but not least, I would like to express my deepest gratitude and thanks to my parents, my sister and my brother for their unconditional love and support, and for putting up with me and always staying by my side.



## List of Publications

### Journals

- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Incremental training of a detector using online sparse eigen-decomposition. *IEEE Transactions on Image Processing (TIP)*, **20**(1):213-226, 2011.
- C. SHEN, S. PAISITKRIANGKRAI, AND J. ZHANG. Efficiently learning a detection cascade with sparse eigenvectors. *IEEE Transactions on Image Processing (TIP)*, **20**(1):22-35, 2011.
- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Transactions on Circuits System Video Technology (TCSVT)*, **18**(8):1140-1151, 2008.
- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Performance Evaluation of Local Features in Human Classification and Detection, *IET Computer Vision Journal*, **2**(4):236-246, 2008.

### Referred conference papers

- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Face Detection with Effective Feature Extraction, *In Proc. of The Tenth Asian Conference on Computer Vision (ACCV)*, Queenstown, New Zealand, November 2010.
- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Efficiently training a better visual detector with sparse eigenvectors. *In Proc. of IEEE Conference Computer Vision Pattern Recognition (CVPR)*, Miami, Florida, USA, June 2009.

- C. SHEN, S. PAISITKRIANGKRAI, AND J. ZHANG. Face detection from few training examples, *In Proc. of IEEE International Conference on Image Processing (ICIP)*, San Diego, California, USA, October 2008.
- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. An experimental study on pedestrian classification using local features, *In Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, Seattle, Washington, USA, May 2008.
- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. An experimental evaluation of local features for pedestrian classification, *In Proc. of International Conference on Digital Image Computing - Techniques and Applications (DICTA)*, Adelaide, Australia, December 2007

#### **Referred workshop papers**

- S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Real-time Pedestrian Detection Using a Boosted Multi-layer Classifier, *In Proc. of The Eighth International Workshop on Visual Surveillance (VS)*, in conjunction with ECCV, Marseille, France, October 2008.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Visual Object Detection . . . . .	3
1.1.1	Definition . . . . .	3
1.1.2	Motivation . . . . .	4
1.1.3	Applications . . . . .	4
1.2	Challenges . . . . .	5
1.3	General Background Knowledge . . . . .	7
1.3.1	Background Subtraction . . . . .	8
1.3.2	Feature Extraction . . . . .	9
1.3.2.1	Raw Pixel Intensity . . . . .	9
1.3.2.2	Brightness Histogram . . . . .	10
1.3.2.3	Colour . . . . .	10
1.3.2.4	Texture . . . . .	10
1.3.2.5	Edges . . . . .	10
1.3.2.6	Shape . . . . .	11
1.3.3	Global Appearance Versus Local Appearance . . . . .	11
1.3.3.1	Sparse Representation . . . . .	11
1.3.3.2	Dense Representation . . . . .	12
1.3.4	Classification . . . . .	12
1.3.4.1	Generative Models . . . . .	12
1.3.4.2	Discriminative Models . . . . .	12
1.4	Contributions . . . . .	13
1.5	Thesis Outline . . . . .	14

## CONTENTS

---

<b>2</b>	<b>Related Works</b>	<b>17</b>
2.1	Overview . . . . .	17
2.2	Boosted Cascade Classifiers on Haar-like Features . . . . .	27
2.2.1	Viola and Jones' Cascade Design . . . . .	28
2.2.2	Viola and Jones' Boosted Classifier . . . . .	30
2.2.3	Viola and Jones' Haar-like Features with Integral Image . . . . .	31
2.3	Improvements of Boosted Cascade Classifiers . . . . .	33
2.3.1	Shortcomings of Cascade Designs . . . . .	33
2.3.2	Shortcomings of AdaBoost and Boosted Classifiers . . . . .	39
2.3.3	Shortcomings of Haar-like Features . . . . .	46
2.3.4	Other Issues Related to Viola and Jones' Face Detector . . . . .	50
<b>3</b>	<b>Fast Pedestrian Detection using Boosted Covariance Features</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Boosted Covariance Features . . . . .	54
3.2.1	Weighted Fisher Linear Discriminant Analysis . . . . .	57
3.2.2	A Cascade of Covariance Descriptors . . . . .	59
3.3	Multiple-layer Boosting with Heterogeneous Features . . . . .	60
3.4	Experiments . . . . .	62
3.4.1	Experiments on Daimler-Chrysler Data Sets with Boosted Co- variance Features . . . . .	63
3.4.1.1	Experiment Setup . . . . .	63
3.4.1.2	Results based on Boosted Covariance Features . . . . .	64
3.4.2	Experiments on Daimler-Chrysler data sets with two-layer boost- ing . . . . .	67
3.4.2.1	Experiment setup . . . . .	67
3.4.2.2	Results based on Multi-layer Boosting . . . . .	67
3.4.3	Experiments on INRIA Human Data Sets with Boosted Co- variance Features . . . . .	68
3.4.3.1	Experiment Setup . . . . .	70
3.4.3.2	Results based on Boosted Covariance Features . . . . .	70
3.4.4	Experiments on INRIA Human Data Sets with Two-layer Boost- ing . . . . .	72

3.4.4.1	Experiment Setup . . . . .	72
3.4.4.2	Results based on Multi-layer Boosting . . . . .	72
3.4.5	Detection Performance versus Speed Trade-off for the Two-layer Boosting . . . . .	74
3.4.6	Discussion . . . . .	76
3.5	Conclusion . . . . .	76
<b>4</b>	<b>Efficiently Training a better Visual Detector with Sparse Eigenvectors</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Algorithms . . . . .	82
4.2.1	Greedy Sparse Linear Discriminant Analysis . . . . .	82
4.2.2	Linear Discriminant Analysis on Asymmetric Data . . . . .	84
4.2.3	Boosted Greedy Sparse Linear Discriminant Analysis . . . . .	87
4.2.4	Training Time Complexity of BGSLDA . . . . .	90
4.3	Experiments . . . . .	91
4.3.1	Face Detection with the GSLDA Classifier . . . . .	92
4.3.1.1	Performance on Single-node Classifiers . . . . .	92
4.3.1.2	Performance on Cascades of Strong Classifiers . . . . .	94
4.3.2	Face Detection with the BGSLDA Classifier . . . . .	99
4.3.2.1	Performance on Single-node Classifiers . . . . .	99
4.3.2.2	Performance on Cascades of Strong Classifiers . . . . .	99
4.3.3	Pedestrian Detection with GSLDA Classifiers . . . . .	103
4.3.3.1	Pedestrian Detection on Daimler-Chrysler Data Sets with Haar-like Features . . . . .	103
4.3.3.2	Pedestrian Detection on INRIA Data Sets with Covariance Features . . . . .	103
4.4	Conclusion . . . . .	105
<b>5</b>	<b>Incremental Training using Online Sparse Eigen-decomposition</b>	<b>107</b>
5.1	Introduction . . . . .	107
5.2	Online Learning of GSLDA Classifiers . . . . .	110
5.2.1	Incremental Update of Between-class and Within-class Matrices	113
5.2.1.1	Updating Between-class Scatter Matrix: . . . . .	113
5.2.1.2	Updating Within-class Scatter Matrix: . . . . .	114

## CONTENTS

---

5.2.1.3	Updating Inverse of Within-class Scatter Matrix: . .	115
5.2.2	Updating Weak Learners' Coefficients and Threshold . . . . .	115
5.2.2.1	Incremental Learning Computational Complexity .	119
5.3	Experiments . . . . .	121
5.3.1	USPS Digits Classification . . . . .	121
5.3.2	Frontal Face Detection . . . . .	124
5.3.2.1	Performance on Single-node Classifiers . . . . .	124
5.3.2.2	Performance on Cascades of Strong Classifiers . . .	129
5.4	Conclusion . . . . .	132
<b>6</b>	<b>Conclusions and Future works</b>	<b>135</b>
6.1	Summary . . . . .	135
6.2	Future Works . . . . .	137
6.2.1	Cascade Design . . . . .	137
6.2.2	Strong Classifier Learning . . . . .	138
6.2.3	Weak Classifier Learning . . . . .	138
6.2.4	Haar-like Features . . . . .	138
6.2.5	Massive Training Data . . . . .	140
6.2.6	Bootstrapping . . . . .	140
	<b>References</b>	<b>156</b>

## List of Figures

1.1	An example of pedestrians and vehicles in object detection problem. .	3
1.2	An example illustrating difficulties of visual object detectors. . . . .	7
2.1	An illustration of cascade classifiers. . . . .	29
2.2	An illustration of strong classifiers. . . . .	30
2.3	An illustration of integral images. . . . .	32
2.4	An example of Haar-like rectangle features. . . . .	33
2.5	An illustration of soft cascade and multi-exit cascade. . . . .	36
2.6	An illustration of matrix-structural learning. . . . .	38
2.7	A shortcoming of AdaBoost illustrated on toy data sets. . . . .	42
2.8	An illustration of variants of Haar-like features. . . . .	47
2.9	An illustration of Edgelet features. . . . .	48
2.10	An illustration of sparse granular features. . . . .	49
3.1	Detection examples on AVSS 2007 and CAVIAR data sets. . . . .	55
3.2	An architecture of the propose pedestrian detection system. . . . .	56
3.3	The first and second selected covariance regions. . . . .	59
3.4	The structure of the proposed two-layer pedestrian detector. . . . .	61
3.5	A performance comparison between covariance and Haar-like features. .	62
3.6	A performance comparison of the proposed boosted covariance features. .	64
3.7	A performance comparison on Daimler-Chrysler data sets. . . . .	65
3.8	Examples of mistakes made by our boosted covariance detector . . . .	66
3.9	A number of weak classifiers used and a performance comparison. . .	68

## LIST OF FIGURES

---

3.10	A performance comparison between the proposed approach and state-of-the-art approaches. . . . .	71
3.11	A performance comparison of two-layer boosting approaches based on ( <i>left</i> ) <b>classification</b> and ( <i>right</i> ) <b>detection</b> . . . . .	73
3.12	Human detection examples. . . . .	74
3.13	A performance comparison on human detection using $18 \times 36$ pixels training data. . . . .	75
3.14	Detection rate versus speed trade-off for different configurations of two-layer boosting. . . . .	77
3.15	Detection examples on images collected randomly from the internet. . . . .	78
4.1	An illustration of cascade classifiers . . . . .	80
4.2	AdaBoost versus GSLDA on toy data sets . . . . .	88
4.3	Random samples of face images used during training. . . . .	92
4.4	An error rate comparison of different approaches. . . . .	93
4.5	Comparison on MIT+CMU face test sets. . . . .	95
4.6	Comparison of different approaches on MIT+CMU test sets. . . . .	96
4.7	First seven Haar-like rectangle features selected using different classifiers. . . . .	97
4.8	Face detection examples using the BGSLDA detector. . . . .	100
4.9	A comparison of various approaches on MIT+CMU test sets. . . . .	101
4.10	A comparison of BGSLDA with a different value of $\gamma$ . . . . .	102
4.11	A performance comparison on pedestrian detection. . . . .	104
4.12	Pedestrian detection examples on INRIA test sets. . . . .	106
5.1	A comparison of various online classifiers on toy data sets . . . . .	118
5.2	A classification error rate between offline GSLDA and online GSLDA . . . . .	121
5.3	A comparison of classification error rate and computation cost between online GSLDA and offline GSLDA . . . . .	122
5.4	Detection examples of different face detectors . . . . .	125
5.5	A comparison between offline and online GSLDA . . . . .	127
5.6	Random samples of face images used during training. . . . .	127
5.7	A comparison of classification error rates between offline GSLDA and online GSLDA. . . . .	129



## LIST OF FIGURES

---

5.8	A performance comparison on MIT+CMU test sets. . . . .	130
5.9	A training time comparison between GSLDA and online GSLDA. . .	130

## **LIST OF FIGURES**

---

## List of Tables

2.1	A summary of existing object detection approaches. . . . .	19
2.2	An overview of recently proposed cascade structures. . . . .	34
2.3	An overview of recently classification approaches. . . . .	41
2.4	An overview of recently proposed features. . . . .	46
3.1	An average time required to evaluate covariance and Haar-like features.	62
3.2	An average evaluation time of different approaches on Daimler-Chrysler test sets. . . . .	66
3.3	An average evaluation time of two-layer approaches on Daimler-Chrysler test sets. . . . .	69
3.4	An average evaluation time of different approaches on INRIA test sets.	71
3.5	An average evaluation for two-layer boosting approaches on INRIA test sets. . . . .	74
4.1	The size of training and test sets used on a single node classifier. . . .	92
4.2	A summary of training time, the number of nodes and evaluation time of different classifiers. . . . .	98
4.3	A breakdown of CPU time of proposed approaches. . . . .	102
5.1	Notation . . . . .	111
5.2	A performance comparison of different face detectors . . . . .	126
5.3	The size of training and test sets used in the experiment. . . . .	128
6.1	A comparison between the total number of features of different ap- proaches. . . . .	139

**LIST OF TABLES**

---

# 1

## Introduction

As computer has become more and more powerful, it has turned out to be a vital part in our daily lives. Researchers have focused on how to extend their uses to perform a more intelligent task such as visual scene analysis. The study in this area has given rise to a new discipline in engineering and science known as computer vision. Computer vision involves three major disciplines:- Image Processing (Engineering and Physics), Artificial Intelligence (Computer Science) and Pattern Recognition (Mathematics). The objective of computer vision is to grant a machine the ability to see and think for itself as similar to us.

During its early stage, a large amount of work has been carried out for translating 2D low-level images into 3D high-level semantics. Many researchers had focused on real-world vision applications, *e.g.*, autonomous robots, vision based manufacturing inspection and content-based image retrieval. Since there is no single solution to solving computer vision tasks; there exist abundant methods in the literature. Some of these approaches turned out to be very task specific and can seldom be generalized over a wide range of applications.

## 1. INTRODUCTION

---

Due to the high complexity of 3D image representations and an increasing popularity of statistical methods and pattern recognition, numerous researchers have switched their focus to 2D image analysis [10, 116]. Statistical values are extracted from image regions to produce a set of meaningful features. These statistical representations are then analyzed to produce another output which corresponds to a given set of patterns. This approach, along with an advance in computer hardware, has lead to many major breakthroughs in computer vision areas such as real-time object segmentation, real-time object detection, real-time object classification and real-time object tracking.

One application, that statistics-based approach has gained a tremendous success, is an intelligent video surveillance system. Airports, police stations, office buildings, shopping centers, train stations, bus stops, *etc.* have numerous security cameras recording at all times covering numerous scenes. However, due to the vast amount of data being accumulated each day and the fact that most surveillance systems are being monitored by only a few operators, operators tend to miss many important events after a certain period. Strictly speaking, it is impossible for human being to monitor all surveillance cameras in a non-automatic fashion. Hence, an automated system is needed in order to detect and respond to an abnormal event in real-time.

In order to detect a predefined event, *e.g.*, loitering and trespassing, it is important that pre-specified objects can be first detected with high accuracy. Object detection is a computer vision task that identifies and determines locations and sizes of predefined objects in arbitrary images. The objective of object detection in surveillance video is to detect one or several objects in any scenes (high/low resolution) under varied condition, pose, appearance, illumination and background clutter with high accuracy and low false positives (Figure 1.1)

Although this task seems to be trivial and incredibly easy to humans, computers are currently far behind us in performing such analysis and inference. It remains a challenge to explain how humans perceive objects so quickly and accurately with very little effort. Thus, our primary goal is to grant computer the ability to see, analyze and identify the object of interest in arbitrary images using modern vision algorithms. In this chapter, we briefly introduce the problem of object detection, discuss challenges involved, describe general background knowledge and briefly present our contributions. Section 1.1 begins with an introduction to the problem, our motivation and key applications. Section 1.2 discusses difficulties of visual object detection. Section 1.3 gives



**Figure 1.1:** An example of pedestrians and vehicles in object detection problem. Note large variations in pose, appearance, illumination and background clutter. Courtesy of MIT CBCL, <http://cbcl.mit.edu/cbcl/software-datasets/index.html>

some background information on object detection and some perspectives on related works in this field. Section 1.4 summarizes our own approaches and contributions. Finally, we conclude with an outline of this thesis in Section 1.5.

## 1.1 Visual Object Detection

---

This section covers the definition of object detection, its motivation and key applications of real-time object detection.

### 1.1.1 Definition

Real-time visual object detection is a computer vision task that rapidly identifies and determines locations and sizes of visual objects in arbitrary images and videos. The ability to rapidly identify objects is important in many human-computer interaction and real-time monitoring applications. Here we distinguish the difference between other related vision tasks, *e.g.*, object recognition and object identification, and visual object detection. The goal of object recognition tasks is to recognize and tell differences between several pre-specified object classes; *e.g.*, given images of four-legged animals, the system should be able to tell differences between cats, dogs, horses, cows and elephants. On the other hand, the task of object identification is to recognize an individual

## 1. INTRODUCTION

---

instance of an object; *e.g.*, identification of a specific person's face or fingerprint. For our problem, we try to successfully locate all instances of pre-specified objects despite the presence of background clutters or partial occlusions; *e.g.*, all faces in a given image. Clearly, object detection is the first fundamental step to object recognition and object identification.

### 1.1.2 Motivation

Security and safety are probably two primary motivations for effective real-time object detection. As society continues to advance, public surveillance and public safety will become a more important aspect of our modern life. More cameras with intelligent surveillance capabilities will be equipped in public places to detect and prevent crimes and accidents. These surveillance data will be intelligently processed to help security operators locate emergency events and respond to them in real-time.

There are also other motivations for real-time object detection from a multimedia point of view. With the advance in a media compression technology, a broadband internet access and a popularity of media sharing web sites, more and more digital media are being uploaded to the internet at an exponential rate. However, the usability of these media collections is limited by a lack of effective retrieval methods. Currently, to find a specific image in such a collection, we have to manually and exhaustively search the entire database collection. This process is rather slow and tedious. In order to fully utilize these resources, a tool that can efficiently and effectively manage this massive video collection has become indispensable. Automatic object detection and recognition can be used to extract more information from these images and help automatically label and categorize them.

### 1.1.3 Applications

Object detection is a fascinating problem since it is the first fundamental step to many vision applications. Automated visual object detection has attracted a lot of research attentions in recent years. Effective real-time object detection has tremendous uses. Here we briefly discuss some of its applications.



- *Faces*. Detecting faces is the first vital step in vision-based human computer interaction systems, *e.g.*, face modeling, face recognition, face authentication, face tracking, face pose estimation, facial expression recognition and automatic tagging in social media and digital content management. It can also be applied to electronic equipments like digital camera, allowing the device to automatically focus and zoom on detected faces.
- *Humans*. Human detection could be applied to detect humans for various applications, *e.g.*, gait recognition, intrusion detection, video surveillance, border security and pedestrian accident prevention in smart vehicles, market analysis and survey.
- *Vehicles*. Vehicle detectors could be used for automatically monitoring traffic, traffic analysis at the intersection, road accident and video surveillance.
- *Hands*. Hand detectors are a necessary component in a gesture-based control. Electronic devices, *e.g.*, televisions, stereos, cameras, can be controlled without the need of remote controls.
- *Boats*. Boat detectors can be used to help border patrol officers in monitoring invasion zones. In intelligent video search would help reduce the burden of human operators.
- *Ground target*. Ground target detection can be used in search and rescue operations for many military applications.

## 1.2 Challenges

---

Automated visual object detection is a difficult task. While humans can do this task effortlessly, it turns out to be a very challenging task for machines. Little progress has been made over the past few decades in building a robust detector. The foremost difficulty lies in the amount of object variations (Figure 1.2). In this section, we break down these difficulties into following factors:-

## 1. INTRODUCTION

---

- *Large variations in appearance.* A small change in object's position or orientation with respect to the camera can change its appearance considerably on 2D images. As an example, the same human captured by two different cameras at different position and angle could look completely different on images. In addition, pixel's intensity on 2D images depends on many factors in the environment, *e.g.*, light sources, their colour and their intensity. The robust detector must be able to handle these viewpoint and scale changes issues.
- *Large within-class variations.* Most object classes have large within-class variations. For example, human appearance is strongly influenced by the clothing they wear, their pose and accessories during the time of capture; a vehicle appearance is varied due to colors, models and manufacturers. A robust detector must be able to detect these variations in human and vehicles, regardless of their poses, colors, manufacturers.
- *Image background.* Background clutter is common and varies from images to images. For example, images taken from outdoor scenes have different background than those taken from indoor environments. Outdoor scenes usually consist of a number of natural objects or large man made structures, *e.g.*, trees, roads, buildings, while indoor scenes usually consist of indoor furniture, *e.g.*, television, sofa, chairs, wall. These differences in background can cause object boundaries to be different. In addition, background structures can accidentally be similar to a person's shape and appearance, *e.g.*, street and light poles. The detector must be able to distinguish object classes from complex background regions.
- *Occlusion.* Occlusions create further difficulties because only a few parts of objects are visible for processing.
- *Lighting and weather condition.* Due to the movement of sun and cloud, a lighting condition can change rapidly in outdoor scenes. Depending on lighting conditions, average pixel values can be rather bright or dark. Since most cameras often fail to adjust their hardware to these changes, overexposure and underexposure are very common. The system must be invariance to these changes.



**Figure 1.2:** An example illustrating difficulties of visual object detectors on faces (*left*) and humans (*right*). Note the amount of variation in images due to pose (orientation of the object and the position of the camera), appearance, illumination, differences in human clothing and background clutter, *etc.* Courtesy of <http://www.creationscience.com> and INRIA person detection data sets [24].

In this thesis we focus our study on the detection of fully visible visual objects. In such poses, the object appearance is relatively constrained. One can thus learn relevant feature vectors or descriptors and build the robust detector.

### 1.3 General Background Knowledge

---

Due to its enormous vision applications, various approaches had been proposed for an automated object detection problem, ranging from simple intensity-based approaches to complex high-level approaches utilizing advanced learning methods. Although a number of approaches had been proposed, a robust real-time object detector was much too far to be practical. Here we categorized object detection into four categories [150].

- *Knowledge-based methods.* These rule-based methods encode human knowledge of what constitutes a visual object. Usually, these rules capture relationships between object parts. These approaches have been designed mainly for object localization [148].

## 1. INTRODUCTION

---

- *Feature invariant approaches.* These algorithms aim to find structural features that exist even when the pose, viewpoint or lighting conditions vary. These features are then used to locate objects. Several invariant features have been adopted, *e.g.*, grouping of edges on facial features [64], integration of multiple features [57].
- *Template matching methods.* Several object patterns are pre-selected and stored in the database. During evaluation, correlations between a test image and stored templates are computed. The correlation value will be highest at places where the image structure matches the template structure [44]. The technique is commonly used in manufacturing as a part of quality control [4].
- *Appearance-based methods.* These algorithms learn a model from a set of training images. By capturing the representative variability of object appearances, the learned model can be used for object detection.

In general, appearance-based methods using learning algorithms have shown excellent results in many vision applications and have been an active area of research. For the rest of this section, our primarily focus will be on appearance-based methods and their variations. Interested readers in other approaches should refer to recent literature surveys for more details [41, 42, 112, 150].

Based on appearance-based methods, a region of interest is selected based on prior scene knowledge [41] or low-level features, *e.g.*, background subtraction [46, 48, 59, 97], image difference [16] and scale-invariant key-points [72]. An object detector then operates on this selected region. The object detection algorithm can be further broken down into two components:- feature extraction and classification algorithm. Feature extraction involves a representation of image patches (regions) as discriminative feature vectors (also known as descriptors); while the classification algorithm makes decision based on these feature vectors. This section discusses each component in details.

### 1.3.1 Background Subtraction

Background subtraction is a commonly used technique to segment foreground objects from the background. The popularity of background subtraction is largely due to its

computational efficiency. Numerous algorithms based on background subtraction have been proposed. Some of popular and commonly applied approaches are mixture of Gaussians [59], Kernel density estimators [48], sequential kernel density approximation [46] and eigenbackgrounds [97]. Background subtraction works well when the camera is static and lighting condition do not change rapidly.

### 1.3.2 Feature Extraction

Feature extraction is often the first fundamental technique in any computer vision applications. The process not only reduces the amount of data one needs to calculate but also creates a new set of representations which is distinct and unique so that patches can be easily distinguished in the feature space. Mathematically, a feature is an  $N$ -dimensional vector which is extracted from image patches. Several techniques exist for generating image patches. The commonly applied technique is the sliding window technique, where a fixed sized window (based on prior scene knowledge) is shifted at various scales and locations over the image. For face detection, a fixed sized window is often a square of size  $19 \times 19$  pixels [104],  $20 \times 20$  pixels [67] or  $24 \times 24$  pixels [140]. For human detection, a window can be a rectangle of size  $64 \times 128$  pixels [24, 88, 104] or  $18 \times 36$  pixels [89]. Since one image can consist of hundred thousands of patches, several researchers have tried to restrict the search-space based on known camera geometry [51], prior information about the target object class [41], moving objects using low-level image features, *e.g.*, background subtraction [48, 155] or optical flow [33]. Patches can then be sampled from these regions.

Feature extraction typically captures intensity patterns, appearances, texture details, motions, shapes and contour information. Different cues have different characteristics. Choosing the right cue for the right object is an art in itself. Here cues commonly used in computer vision are briefly explained.

#### 1.3.2.1 Raw Pixel Intensity

This is the most basic feature. Unfortunately, due to its poor performance, its use is very limited. Part of the reasons is that raw pixel intensity does not encode any specific domain knowledge. Another reason is that images are often degraded by some random noise during image capturing, transmission or processing.

## 1. INTRODUCTION

---

### 1.3.2.2 Brightness Histogram

The histogram provides the frequency of the brightness in the image. Due to its simplicity, two different images with similar brightness could produce similar brightness histogram.

### 1.3.2.3 Colour

Colour is the most basic visual content and it is one of the most widely used features. Color is usually represented in RGB (red, green, blue) color space. Colour histogram is one of the best known color features [94]. It has been used to represent a distribution of colours in an image. The advantage of color histogram is that it is invariant to rotation and translation.

### 1.3.2.4 Texture

Texture is often used to describe characteristics of object surfaces. It measures the intensity variation of a surface. Compare to color features, texture is less sensitive to illumination changes. Well known and commonly used texture descriptors are co-occurrence matrices [47], edge frequency [26], primitive length [39], fractal texture descriptor [107], multi-scale texture descriptor, *e.g.*, Gabor transforms [25] and wavelet transforms [76]. Recently, Local Binary Pattern (LBP), which describes the local texture information around each pixel, has been proven effective in texture classification [96].

### 1.3.2.5 Edges

It is a well-known fact that a human visual system is sensitive to a local luminance contrast at edges. Edge detector can be used to identify these changes. It is one of the very first few features to be proposed in the vision community. Edges can be considered as a strong change in image intensity. An important property of edges is that they are less sensitive to illumination changes compared to color features. There exists a large number of edge detection algorithms in literature. Some well known techniques are Roberts, Sobel, Prewitt and Canny [19, 44]. An evaluation of various edge detection algorithms can be found in Bowyer *et al.* [13]. Over the last decade, researchers

have focused a great deal of attentions on corners and interest points, which can be considered as a subset of edges. Some of recently proposed interest point detectors are Harris corner points [49], Scale Invariant Feature Transform (SIFT) [72], Maximally Stable Extremal Regions (MSER) [79].

### 1.3.2.6 Shape

Shape contains important semantic information of objects. Shape descriptors generate a numeric feature vector which characterizes properties of described objects. The descriptors can be used for measuring a shape similarity. Numerous shape descriptors have been proposed, *e.g.*, chain codes [132], Fourier descriptors [106], B-spline representation [5], shape context [9], *etc.*

### 1.3.3 Global Appearance Versus Local Appearance

Some researchers represent a global appearance of the object as a template [90, 133], while others focus on regions which contain discriminative information [24, 140]. Global appearance methods capture the common appearance of the object from training images. Global appearance is fast but has several disadvantages. Global feature often emphasizes on coarse attributes of object appearance rather than discriminative object parts. Hence, it fails to extract a meaningful component if there is a large variation in the object's appearances and poses. Furthermore, a template matching on global appearance is sensitive to small differences in scale, position and orientation.

On the other hand, local appearance decomposes the object into smaller parts and represents the object based on these parts. Since features are extracted from object parts, it is less sensitive to above problems. The feature extraction on local appearance often involves sparse and dense representation.

#### 1.3.3.1 Sparse Representation

Features are extracted from a set of salient image regions. The motivation is that not all image regions contain useful information, *i.e.*, some are uniform and textureless. The intuition is based on a human eye-tracking, *i.e.*, a local spatial contrast is significantly higher at interest points than at random locations. Well known interest point

## 1. INTRODUCTION

---

detectors are Harris-Laplace [83], Difference of Gaussian (DoG) [77], Hessian-Affine [82], MSER [79], *etc.*

### 1.3.3.2 Dense Representation

Features are computed on every small image region [24, 140]. The intuition behind this approach is that all image regions are equally important. The later stage will then make a decision which regions are the most relevant.

### 1.3.4 Classification

Several classification models and techniques have been studied in the literature and were also reviewed by [137]. Here we roughly divide them into two categories:- generative and discriminative model [137]. Both generative and discriminative approaches can be used during the training stage. Typically generative approaches use Bayesian graphical models with Expectation-Maximization (EM) to characterize object parts and to model their co-occurrences. On the other hand, discriminative approaches use machine learning techniques to classify each feature vector as belonging to the object or not. The main difference between generative and discriminative models is how posterior probabilities are estimated for each class.

#### 1.3.4.1 Generative Models

Generative approaches model the appearance of object class in terms of its class-conditional density functions. Combining with class priors, posterior probability can be inferred using a Bayesian approach. Generative models are popular in object recognition, especially for matching similar object categories. The advantage of generative approaches is with its ability to learn the representative characteristic of seen objects and can be used to infer on unseen objects.

#### 1.3.4.2 Discriminative Models

Discriminative models approximate the Bayesian maximum-a-posteriori decision by learning parameters of discriminant functions (hyperplanes) between positive and neg-



ative class from training examples. Discriminative approach explicitly explores the distinction of objects from background, and uses that knowledge to learn the model.

## 1.4 Contributions

---

Since the objective of our thesis is real-time object detection, we propose our approaches based on the work of Viola and Jones [140]. Viola and Jones proposed the use of Haar-like features for a face detection task. However, Haar-like features fail to capture the shape of other objects, *e.g.*, pedestrian and human. Hence, there has been considerable interest in applying other features on pedestrian detection problems. Some of these features are local receptive field [89], covariance features [136] and histogram of oriented gradient [24]. In order to find the right feature, we first present a comprehensive experimental study on pedestrian detection using state-of-the-art locally extracted features. Building upon the finding of our experiments, we propose a new, simpler pedestrian detector using covariance features. Unlike the existing approach, where the feature selection and weak classifier training are performed on the Riemannian manifold [136], we select features and train weak classifiers in the Euclidean space for faster computation. To this end, AdaBoost with weighted Fisher linear discriminant analysis-based weak classifiers are proposed. To further accelerate the detection, we adopt a faster strategy — multiple layers boosting with heterogeneous features — to exploit the efficiency of Haar-like features and the discriminative power of covariance features.

Based on our observations, AdaBoost is sub-optimal for training a visual object detector since it operates under the assumption that the number of positive and negative samples are equal. In other words, it ignores the fact that the training data in object detection problem is often *imbalanced* and *highly skewed*. In order to further improve the performance, we introduce a new classifier, termed Greedy Sparse Linear Discriminant Analysis (GSLDA), for its conceptual simplicity and computational efficiency. Unlike Adaboost, GSLDA takes the number of training samples in each class into consideration when solving the optimization problem. This extra information helps minimize the effect of imbalanced data sets and improves the overall classification accuracy at the same runtime cost.

## 1. INTRODUCTION

---

One major drawback of GSLDA is that decision stumps' thresholds are fixed for the entire duration of classifier training, *i.e.*, once calculated, we do not re-train these weak classifiers. Having a fixed threshold value could result in a sub-optimal weak classifier. We propose a new technique, termed Boosted Greedy Sparse Linear Discriminant Analysis (BGS LDA), to efficiently train a weak classifier. BGS LDA exploits the sample re-weighting property of boosting to update weak classifiers' thresholds and the class-separability criterion of GSLDA to train a strong classifier.

Although offline object detectors have shown a tremendous success. One major drawback of offline techniques is that a complete set of training data has to be collected beforehand. In addition, once learned, an offline detector cannot make use of newly arriving data. In order to alleviate these shortcomings, online learning has been adopted with following objectives:- the technique should be computational and storage efficient; and the updated classifier must maintain its high classification accuracy.

Improving upon the GSLDA classifier, an effective and efficient framework for learning an online GSLDA model is proposed. Unlike existing online object detection algorithms, *e.g.*, Grabner and Bischof [45] or Pham and Cham [110], our online approach makes use of LDA's learning criterion which has been shown in our previous experiment to outperform the AdaBoost's learning criterion for the offline object detection task. Our updating algorithm is very efficient since we neither replace weak learners nor throw away any weak learners during an updating phase. Finally, we adopt a learning technique similar to a semi-supervised learning where the classifier makes use of the unlabeled data in conjunction with a small amount of labeled data.

### 1.5 Thesis Outline

---

Subsequent chapters of this thesis are organized as follows. Chapter 2 gives an overview of related works in object detection, discusses the work of Viola and Jones [140], lists some of their shortcomings and presents recently proposed approaches. In Chapter 3, a fast method to train human detection is proposed. The key idea of our detector is based on a combination of projected covariance and Haar-like features. In Chapter 4, the performance of AdaBoost on skewed data sets is analysed and its drawbacks are discussed. The LDA based classifier is also proposed. In Chapter 5, shortcomings of offline detectors are pointed out and the incremental classifier for object detection

problems are proposed. Finally, Chapter 6 concludes this thesis and discusses some future works.

The works described in Chapters 3, 4 and 5 have been presented in [100, 101, 102, 127].

## **1. INTRODUCTION**

---

# 2

## Related Works

Like in many fields of science and engineering, the field of computer vision is very diverse. A multitude of literatures exist for solving various computer vision tasks. This chapter discusses existing works related to object detection problems. Some of these works focus on specific objects, *e.g.*, faces, human, vehicles, while other can be applied to general objects. The chapter consists of three sections. Section 2.1 gives an overall overview of different visual object detection approaches. In the next section, we introduce the object detection framework based on Viola and Jones [140], which have been shown to give excellent results with real-time performance. Finally, in Section 2.3, we discuss some limitations and drawbacks of traditional object detectors and briefly present recently proposed approaches.

### 2.1 Overview

---

The primary goal of object detection is to successfully locate all instances of pre-specified objects despite the presence of background clutters or partial occlusions,

## 2. RELATED WORKS

---

*e.g.*, to detect all human faces in a given photo. There exists a number of related works which cover methods for general object detection to a more specific object, *e.g.*, face detection, human detection, body pose estimation and vehicle detection. The research in this field is rather broad and could be classified into many categories. Early research concentrated on easy-to-classified objects using global parameters. Due to recent advances in computer hardware, attention has been shifted towards local features, which have shown to be more robust to object variations and complex illuminations. We briefly tabulate several well known object detection methods in Table 2.1 (sorted by year of publications). In the following section, we outline these methods in more details.

Turk and Pentland [133] proposed the use of global features, known as eigenfaces. They projected face images onto a feature space that spanned the significant variations among known face images. To be more specific, they applied principal component analysis (PCA) to identify the most expressive feature. Unlike local feature approaches, where features correspond to eyes, ears or noses, eigenface features capture information from whole faces. Original faces can be reconstructed from a weighted sum of eigenface features.

The intuition behind their approach is that images of faces do not change radically when projected onto the face space, while the projection of non-face images appears differently. By calculating the distance in the face space at every location in the image, presence of faces can be detected. The technique works well on face images since human faces have a fixed structure, *e.g.*, two eyes, one nose, one mouth and human eyes are at the top and mouth is at the bottom. In addition, faces in an image are often not occluded. However, the drawback of their approach is that the system is not able to cope with large variations in object's appearance, occlusions, poses and illumination conditions. In these conditions, global features fail to extract meaningful object representations.

Active shape model has been one of the most widely used shape modeling techniques. In Baumberg [6], the shape model is generated using a set of foreground regions containing walking pedestrians. The estimated shape is scaled to an appropriate size. The position of pedestrians and their shape estimates are adjusted until the contour of the person is found. The strength of the system was the efficiency and robustness of the system. The system achieved the speed between 14.75 – 33 frames

**Table 2.1:** A summary of existing object detection approaches. *Extraction method* indicates how a set of local image patches is sampled, *e.g.*, densely, randomly, using a key-point detector, using background modeling, *etc.* *Feature descriptors* are sets of representations extracted from image patches. *Classification* is the machine learning algorithm used to exploit detection decision based on given feature descriptors.

	Year	Extraction Method	Feature Descriptors	Classification	Apps.
[133]	1991	Densely	PCA	Euclidean dist.	Faces
[6]	1995	Background	Shape	B-spline	Human
[120]	1998	Densely	LRFs	Neural Networks	Faces
[104]	2000	Densely	Haar Wavelet	SVM	General
[16]	2000	Image Diff.	Shape	Rule based	Human
[48]	2000	Background	Silhouette edges	SAD	Human
[1]	2002	Keypoint	Appearance	SNoW	Vehicles
[40]	2002	Edge Det.	Shape template	Chamfer dist.	Human
[141]	2003	Densely	Wavelet/Motion	AdaBoost (Stumps)	Human
[140]	2004	Densely	Wavelet template	AdaBoost (Stumps)	Faces
[129]	2004	Densely	Motion cue	SVM	Human
[84]	2004	Densely	Feature co-occurrence	AdaBoost (Likelihood ratio)	Human
[24]	2005	Densely	Gradients (HOG)	SVM	Human
[62]	2005	Keypoint	Appearance/Shape	ISM/Chamfer dist.	Human
[156]	2005	Temporal Diff.	Shape	Hist. Intersection	Human
[81]	2006	Keypoint	PCA-SIFT	Prob. model	General
[157]	2006	Densely	Gradients (HOG)	AdaBoost (SVM)	Human
[58]	2006	Densely	Gradients (HOG)	AdaBoost (LDA)	General
[136]	2008	Densely	Covariance	LogitBoost (Regression)	Human

## 2. RELATED WORKS

---

per second on SGI workstation with a 64-bit super pipelined RISC CPU when tested on gray-scale images at full PAL resolution. However, there existed a number of limitations. Firstly, the system was pre-trained to detect human with a specific shape. In other words, it fails to detect human engaging in other activities, *e.g.*, running, jumping or sitting down. Secondly, the system performs well only when there exist a high contrast between human and backgrounds. By relying only on edge information, the system fails when a large percentage of edge points are not detected, *e.g.*, a person moving in front of backgrounds similar to person's clothes.

Rowley *et al.* [120] proposed a multilayer perceptron neural network-based face detection system. The authors performed experiments with both single neural networks and modular systems consisting of several neural networks. Each neural network consisted of one layer of hidden units, where each hidden unit has a receptive field of either  $5 \times 5$ ,  $10 \times 10$  or  $20 \times 5$  pixels. The authors compared their approach with several other state-of-the-art face detection systems and showed that their approach achieved comparable performance in terms of detection and false-positive rates. The drawback of their approach is the high computation time. Given that there are three types of hidden units: four  $10 \times 10$  pixels subregions, sixteen  $5 \times 5$  pixels subregions and six  $20 \times 5$  pixels regions. The total number of CPU operations per window is over 2,000. In their implementation, it took approximately 383 seconds to evaluate an image of  $320 \times 240$  pixels on a 200 MHz R4400 SGI Indigo 2.

Papageorgiou and Poggio [104] proposed a general, trainable object detection system which is purely based on pattern classification. Object class is represented in terms of an over-complete dictionary of local, oriented, multi-scale intensity differences between adjacent regions, called *Haar wavelet transform*. Wavelet coefficients from two frequency bands are used as input to a quadratic classifier. The coefficients in the quadratic classifier are learned by Support Vector Machines (SVMs) from a large set of training samples. Their system is the first human detection system that does not rely on motion, tracking, background subtraction or any assumptions on the scene structure. To improve the detection speed of their framework, the authors proposed a scheme for feature selection based on feature variance. However, the speed improvement comes at a significant drop in detection rate at low false positive rates.

Broggi *et al.* [16] proposed the method to detect pedestrians from vehicle mounted camera. Their goal was to develop a safety system which can act as an automatic pilot



for standard road vehicles. The approach used morphological characteristics and the strong vertical symmetry of human shapes for pedestrian detection and recognition. The authors assumed three hypotheses. First, there exist vertical edges with a strong symmetry with respect to vertical axis. Secondly, size and aspect ratio of detection bounding boxes must satisfy specific constraints. Finally, the pedestrian must be in a specific region and the whole pedestrian must be present in the image. In their implementation, vertical edges are first extracted using Sobel operator [44]. After background has been removed, areas which present high vertical symmetry are considered. Remaining edge pixels are matched with pedestrian's head model. The bounding box is determined from the object's lateral, bottom boundaries and the pedestrian's head. The advantage of using morphological operation is the efficiency of the overall system. However, the system makes use of edges which might not be robust in scenarios where large percentage of edge points cannot be detected.

Haritaoglu *et al.* [48] proposed a real-time visual surveillance system, called  $W^4$ .  $W^4$  constructs dynamic models of people's movements to answer questions about what they are doing, where they are, who they are and when they act. The technique employs a combination of shape analysis and tracking to locate human and their parts. To be more specific, detection process consists of two steps:- background scene modeling and foreground region detection. For human classification,  $W^4$  generates a set of shape (local and global) and appearance features for each detected foreground object. The authors combined detection with tracking to enhance the robustness of their system. Due to its careful design and simplicity, the system achieves a real-time performance on off-the-shelf PCs. However, it remains a challenge whether such a system would be able to distinguish partial occlusion by relying on silhouette information alone. Also, the system relies on background subtraction to detect foreground objects. The disadvantage of background subtraction is that the camera needs to be static.

Agarwal and Roth [1] proposed an approach for learning to detect instances of objects based on sparse, part-based representation. A vocabulary of object parts is automatically constructed from a set of sample images. Original objects are then represented as binary feature vectors based on a presence of vocabulary on object parts, along with spatial relations observed between pairs of parts. Due to the feature property (sparse feature representations), the authors trained a classifier using the sparse

## 2. RELATED WORKS

---

network of winnows (SNoW) learning architecture [119]. SNoW learns a linear function over the feature space using a feature-efficient variation of the Winnow learning algorithm. Their experimental results show that the approach achieves high detection accuracy and are highly robust to partial occlusions and background variations. However, their learning approach relies on the repeated observation of co-occurrences between object parts. The only way to achieve this is to train a system with a large number of training samples.

Gavrila and Giebel [40] proposed a pedestrian detection approach, which can deal with a challenging scenario of moving cameras mounted on a vehicle. The authors represented human model by their shapes. For classification, shape-based template matching is performed based on the Chamfer distance. To allow for efficient matching, a hierarchical tree of templates is constructed from a set of templates. This hierarchy is constructed automatically using partition clustering. During shape matching, the process starts at the root and works its way towards the leaves to find the best matching template based on the chamfer distance. The method also includes a Kalman filter based tracker for taking advantage of the temporal information for filling in missed detections.

Viola and Jones [140] proposed the first robust **real-time** face detection framework. Their approach consists of three key contributions. The first contribution is the introduction of a new image representation called *integral image*, which allows rectangular features to be computed very quickly. Their second contribution is the use of AdaBoost learning algorithm to select a small subset of critical visual features for building a simple and efficient classifier. Their third contribution is a method for combining classifiers in a cascade structure which allows non-face images to be discarded quickly. Their system performs comparable to the best system previously reported while achieving real-time performance on off-the-shelf PCs. Nonetheless, being example-based learning approach, their framework requires a large set of training samples to achieve high detection rates.

Improving upon their previous work [140], Viola *et al.* [141] integrated image intensity information with motion information to detect a walking person. For efficient representation of motion, they extract motion information from the difference between shifted versions of the image in the current frame,  $I_t$ , with the image from the previous frame,  $I_{t-1}$ , where  $I$  represent an image frame and  $t$  is the time it was captured. In

their implementation, motion filters operate on five directions:- no shift in the image in the current frame, image in the current frame is shifted up by one pixel, down by one pixel, to the left by one pixel and to the right by one pixel. They also capture image intensity information from motion images. These features measure something similar to motion shear. For classification, the detector is trained using AdaBoost. Their approach achieves a frame rate of 4 frames/second with very low false positive rates. Furthermore, their system was able to operate on low resolution images. The drawback of their technique is that the approach requires a massive number of training data to achieve reasonable performance. The second drawback is related to the training data collection process. Since human motion has a large variation and AdaBoost is not a multi-class classification algorithm, it might be a challenge whether AdaBoost would be able to handle large variations of motions and noisy training data.

Sidenbladh [129] proposed a human detection framework using a motion cue. The authors believed that the appearance of human varies highly due to many uncontrollable factors like clothing, weather, illumination, *etc.* In contrast, human motion is a more discriminative cue than appearance. Dense optical flow was used as the motion cue. Due to the high dimensionality of the state-space and low number of training samples, the authors used SVMs to learn and classify human/non-human. The author used 443 human flow patterns and 11,688 non-human flow patterns to evaluate their approach. Their experimental results on a set of videos are encouraging. However, in their paper, human flow patterns were chosen from a very simple scene, *e.g.*, a very quiet street with few people walking, scenes with not many moving objects, *etc.* It would be challenging whether the system would perform well in busy scenes with a variety of object classes, *e.g.*, traffic intersection where there are lots of human's and vehicles' motion.

Mikolajczyk *et al.* [84] proposed an approach for detecting human in the presence of clutters and occlusions. The authors modeled humans as flexible assemblies of parts. Seven different body parts (frontal head, face, profile head, profile face, frontal upper body, profile upper body and frontal legs) are used. Body parts appearance is represented by orientation-based features. Orientation is either based on first derivatives (gradient orientation and gradient magnitude) or second derivatives (Laplacian magnitude and the orientation of the second derivative). After computing the dominant

## 2. RELATED WORKS

---

gradient orientation, features are grouped together. Hence, there will be four different feature group types:- horizontal and vertical groups for gradient orientations, and horizontal and vertical groups for the Laplacian. In order to increase the robustness of their features to small shifts in location, the authors quantized the location into a  $5 \times 5$  grids. Feature selection and part detectors are learned from training images using AdaBoost with a linear combination based on log likelihood ratio as weak classifiers. Detection proceeds in three stages. Individual features are first detected across the image at multiple scales. Individual parts are then detected based on these features. Finally, bodies are detected based on assemblies of these parts. The major advantage of using part based approaches is an improvement in system performance. However, this extra calculation has a high computation cost. In their experiments, the system took 10 seconds to evaluate an image of resolution  $640 \times 480$  pixels, which is far from real-time performance.

After Lowe [72] had proposed Scale Invariant Feature Transformation (SIFT) in 1999, numerous researchers have studied the use of orientation histograms in other vision areas. Dalal and Triggs [24] reviewed various existing edge and gradient based descriptors and concluded that grids of histograms of oriented gradients (HOG) outperformed existing feature sets for the task of human detection. HOG works on the assumption that the shape of objects can be represented by a distribution of local intensity gradients or edge directions. In their paper, this is achieved by dividing the image into smaller cells and finding the histogram of edge orientations over all pixels in the cell. To be more specific, a human input image of size  $64 \times 128$  pixels is divided into cells of size  $8 \times 8$  pixels. A group of  $2 \times 2$  cells is integrated into a block. Each cell consists of a 9-bin HOG and each block contains a concatenated vector of all its cells. A vector is then normalized to a unit length. For classification, feature vectors are extracted from human and non-human images, and a linear binary classifier is trained using SVM. This classifier can then be applied to a new input image at several scales to detect human at various sizes. The major disadvantage of their approaches is that a pre-defined single size cell ( $8 \times 8$  pixels) with a fixed size block ( $2 \times 2$  cells) is used. Hence, a HOG block would fail to capture body parts which do not have square shapes, *e.g.*, human limbs which have a rectangular shape. The second disadvantage is the high evaluation time. Since features in every block have to be calculated, the system spends a lot of time extracting both discriminant and non-discriminant features.

Leibe *et al.* [62] employed the appearance-based feature for pedestrian detection in crowded scenes with severe overlaps. Their approach combines local and global cues via probabilistic top-down segmentation. The feature extraction proceeds in two steps. First, a codebook is learned based on Implicit Shape Model (using DoG interest point detector [72] and agglomerative clustering scheme [61]). In the second step, they learn the spatial occurrence distribution of each codebook entry from all training images. During recognition, each patch is matched to the codebook and matching codebook entries cast votes for possible object positions and scales. The final segmentation is obtained from the likelihood ratio between figure and ground probabilities. The authors combined local appearance features with global cues from pedestrian silhouettes. The combination scheme exploits the similarity between the inferred segmentation and pedestrian silhouettes. The combined approach provides better results than either method alone. Experimental results show that their technique is able to detect walking pedestrians under crowded scene with few false positives. However, their approach is far from real-time performance. Combining both local and global features can further improve the performance at a cost of higher computation time.

Zhou and Hoang [156] proposed a real time human detection by applying temporal differencing to segment blob and use codebook to classify a human being from other detected objects. The advantage of using temporal differencing is that it is adaptive to dynamic environment and can be computed quickly. In order to identify human, the authors introduced a codebook to classify human from other objects. Details of their approach can be briefly summarized as follows. First, the object was normalized to a size of  $20 \times 40$  pixels. The shape of objects was then extracted as visual features. Next, codeword with smallest distortion to the feature vector of objects is matched. If the minimum distortion is less than a threshold, this object is classified as human. The algorithm is simple, fast and has proved to be robust to varying environments. However, temporal differencing assumes that the camera is static and the differences are caused only by pre-specified foreground objects, *i.e.*, if there is an overlap between two people, the system will fail to recognize the blob as having two people. This drawback has also been mentioned in their paper that the system fails to detect human when there is a partial occlusion or when two people walk close together. Furthermore, it remains a challenge how well this technique would perform in a busy environment, *e.g.*, traffic intersection, or with camera mounted in a moving vehicle.

## 2. RELATED WORKS

---

Mikolajczyk *et al.* [81] proposed an approach to locate multiple object classes using a generative model. The recognition method is based on a hierarchical codebook representation where appearance clusters, built from edge based features, are shared among several object classes. The structure is efficiently constructed during learning to allow for efficient object detection during evaluation. For classification, a probabilistic model is used for detecting various objects in the same image. The authors reported an excellent performance on several object categories over a wide range of scales, in-plane rotations, background clutter and partial occlusions. Their algorithm is comparable to those state-of-the-art approaches dedicated to a single object class recognition problem. Nonetheless, similar to other probabilistic model, the major drawback of their approach is the high computation cost. The system has to first extract local features, cluster these points, perform matching and compute the likelihood. Depending on the number of features in the given image, the process could take up to 10 seconds. In addition, the approach has a number of parameters involved. Finding the right parameter value for several objects can be rather tedious and requires a lot of preliminary experiments.

Zhu *et al.* [157] used similar features as in Dalal and Triggs [24]. However, they integrated the cascade approach [140] with HOG features to achieve a fast and accurate human detection system. Instead of using a fixed size block, they used HOG of variable-size blocks with AdaBoost for feature selection. Linear SVMs trained with a concatenated vector of 250 random blocks are used as AdaBoost weak learners. The authors reported a speed-up of 70 times at accuracy comparable to the original approach [24] (0.1 second versus 7 seconds). Although the authors had improved the speed of [24], we think that their performance is still sub-optimal. In other words, they randomly selected 250 blocks as their weak learner and there is no guarantee that the selected learner will give optimal performance.

Laptev [58] proposed a method for object detection based on AdaBoost learning with local histogram features. Unlike in Dalal and Triggs [24] and Zhu *et al.* [157], the position and shape of histogram features were chosen to minimize the training error. A complete set of rectangular regions in the object window were used to compute histogram of oriented gradients. AdaBoost procedure was used to select vector valued histogram features and to learn an object classifier. In their approach, a weak learner based on Weighted Fisher Linear Discriminant was adopted on Viola and Jones' object

## 2.2 Boosted Cascade Classifiers on Haar-like Features

---

detection framework [140]. The approach achieves comparable performance to other state-of-the-art methods submitted to the 2005 Pascal Visual Object classes challenge [34]. However, similar to other example-based learning approaches, their framework requires a large number of training samples in order to build a robust classifier.

Tuzel *et al.* [136] proposed a new algorithm to detect pedestrians in still images by utilizing covariance matrices as object descriptors. Due to a property of covariance matrices (symmetric positive definite matrices), distance between two covariance matrices do not lie on a Euclidean space. The authors represented covariance matrices in a Riemannian manifold. To calculate features, they first compute normalized covariance matrices for all rectangular regions. For each region, covariance mean (point which minimizes the sum of squared Riemannian distances) can then be computed. For classification, they use LogitBoost to learn a set of regression functions. For training weak learners, the authors learned regression functions on the tangent space at the weighted mean of covariance points. The authors tested their approach on INRIA human data sets [24] and reported that their approach outperformed all other methods significantly. The only drawback of their approach is the heavy calculation of eigenvalue decomposition, which require  $O(d^3)$  arithmetic operations (where  $d$  is the number of rows or columns of covariance matrices).

## 2.2 Boosted Cascade Classifiers on Haar-like Features

---

Despite a multitude of literature on object detection, a large number of proposed approaches were far from real time performance. This had limited their use in real-world applications. It was not until recently that object detection problem received considerable attention among researchers owing to the impressive performance of Viola and Jones' face detector [140].

The work of Viola and Jones was the first method that achieved **real-time detection speed** and high accuracy comparable to previous state-of-the-art methods. Parts of their approaches were based on Papageorgiou and Poggio [104]. However, instead of designing a single complex classifier like in Papageorgiou and Poggio, coarse-to-fine search, termed cascade classifier, is adopted for computational efficiency. The cascade



## 2. RELATED WORKS

---

structure reflects the fact that within any single image there are very few faces and a vast majority of sub-windows are negative. As such, the cascade attempts to reject as many non-face patches as possible at the early stage of the cascade. An input patch is classified as a face only if it passes tests in all nodes.

As previously discussed in the last section, their work consists of three contributions. The first contribution is a cascade of classifiers. The second contribution is the boosted classifier where a combination of linear classifiers is formed to achieve fast calculation time with high accuracy. The last contribution is a simple rectangular Haar-like feature which can be extracted and computed in fewer than ten Central Processing Unit (CPU) operations using integral image. The rest of this section discusses their contributions and analyzes the component that gives rise to a robust object detector, which yields high detection performance and extremely low false positives.

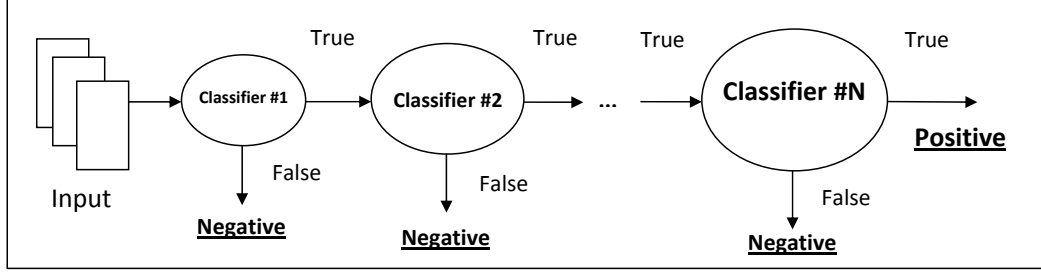
### 2.2.1 Viola and Jones' Cascade Design

Since face detection problem can be classified as a rare event detection task, *i.e.*, the problem typically consists of a very imbalanced ratio of positive and negative samples, a majority of execution time is spent in rejecting negative samples. A cascade is a sequence of classifiers arranged in a coarse-to-fine manner (from simple to complex). It can be viewed as a degenerate decision tree [113]. The key insight to achieving real-time detection speed of Viola and Jones' framework is the idea of designing a cascade of classifiers with increasing complexity, illustrated in Figure 2.1. The complexity of each node can be determined from the number of weak classifiers it contains. The cascade classifier operates as follows. A positive result from the first node triggers the evaluation of a second classifier node. A positive result from the second node triggers the evaluation of a third classifier node, and so on. A negative outcome at any node leads to the rejection of the sub-window. Using this classifier arrangement, a large number of negative patches will be rejected during early node classifiers, allowing the detector to achieve real-time performance.

In order to achieve a high detection rate on the final cascade classifier, each node needs to have a very high detection rate. The global detection rate,  $D$ , and false positive rate,  $F$ , are the product of the detection rate and false positive rate of each individual



## 2.2 Boosted Cascade Classifiers on Haar-like Features



**Figure 2.1:** An illustration of cascade classifiers. The oval size represents the complexity of classifiers. Note that the complexity of classifiers increases as we progress along the cascade.

node classifier. They can be calculated as,

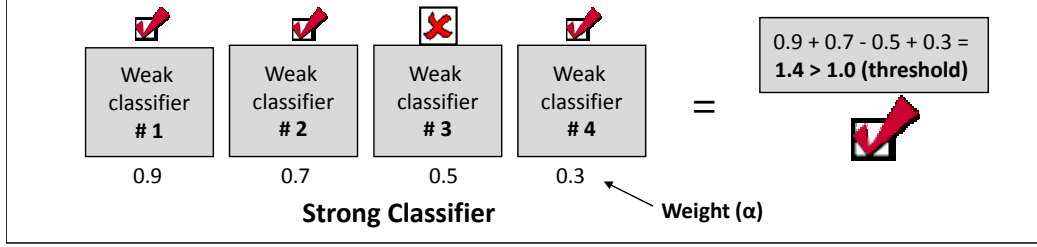
$$D = \prod_{i=1}^N d_i, \quad (2.1)$$

$$F = \prod_{i=1}^N f_i, \quad (2.2)$$

where  $d_i$  is a detection rate of the  $i^{th}$  node classifier and  $f_i$  is a false positive rate of the  $i^{th}$  node classifier. Based on (2.1) and (2.2), Viola and Jones chose all nodes' detection rate to be  $D^{1/N}$  and  $F^{1/N}$ , respectively. In other words, to achieve 95% detection rate with less than  $10^{-5}$  false positive rate in a 20-stage cascade architecture, each node classifier should achieve a minimal detection rate of 99% and false positive rate of around 50%.

Note that during cascade training, an approach known as bootstrapping is often used. Bootstrapping is a general machine learning technique that iteratively trains and evaluates a classifier in order to improve the overall performance. During training, the first classifier node is trained with random negative patches and positive patches. The second classifier node is then trained with false positives from the first node. The third node is then trained with false positives from the first and second node, and so on. By having a sequence of classifiers, efficient and robust object detectors can be generated. In Viola and Jones' cascade design, the same set of positive samples is used in all node classifiers.

## 2. RELATED WORKS



**Figure 2.2:** An illustration of strong classifiers. The final strong classifier takes the form of weighted combinations of weak classifiers. A tick indicates a positive response (+1) and a cross indicates a negative response (−1).

### 2.2.2 Viola and Jones’ Boosted Classifier

Boosting is a meta-algorithm that has been proposed to improve the classification performance of base classifiers (weak classifiers). Several algorithms have been used in conjunction with boosting, for example decision stumps and decision trees. Several boosting algorithms have been proposed. The most popular boosting algorithm is AdaBoost (Adaptive Boosting) by Schapire [123].

AdaBoost combines a collection of weak learners to form a stronger classifier. The algorithm can be interpreted as a greedy feature selection process. In each iteration, a weak learner is called to solve a learning problem. The weak learning algorithm selects the single feature which best separates the positive and negative examples. After the first weak learner is selected, examples are re-weighted in order to emphasize those which were incorrectly classified by the initial weak learner. The process continues until all samples are correctly classified or the maximum number of iterations is reached. The final strong classifier takes the form of a weighted combination of weak classifiers followed by a threshold, illustrated in Figure 2.2.

The strong boosted classifier is a linear combination of  $T$  weak classifiers, which can be defined as,

$$H(\mathbf{x}) = \text{sign}(F(\mathbf{x})) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right), \quad (2.3)$$

where  $\alpha_t$  is the coefficient associated with weak classifiers. A new weak classifier  $h_t(\mathbf{x})$  is learned by minimizing an exponential upperbound of the classification error

## 2.2 Boosted Cascade Classifiers on Haar-like Features

---

**Algorithm 1** AdaBoost Training algorithm.

---

**Input:**

- Training set with their labels;  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$  where  $\mathbf{x} \in R^k$  and  $y \in \{-1, 1\}$ ;
- The number of boosting iterations,  $N_w$ .

```

1 Initialize:  $t = 0; s(i) = \frac{1}{N}, i = 1, \dots, N; F(\cdot) = \phi$ 
2 while  $t < N_w$  do
3   1.  $t = t + 1$ ;
4   2. Train base learner using sample weights,  $s$ , on the training data;
5   3. Find the best weak classifier,  $h_t(\cdot)$ , with minimal weighted
6     misclassification error,  $e_t = \sum_{i=1}^N s(i) \mathbf{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$ ;
7   4. Compute  $\alpha_t = \frac{1}{2} \log(\frac{1-e_t}{e_t})$ ;
8   5. Update sample weights,  $s(i) = s(i) \exp(-y_i \alpha_t h_t(\mathbf{x}_i)), i = 1, \dots, N$ ;
9   6. Update  $F(\cdot) = \sum_{t=1}^T \alpha_t h_t(\cdot)$ .
```

**Output:**

- Final classifier  $H(x) = \text{sign}(F(x)) = \text{sign}\left(\sum_{t=1}^{N_w} \alpha_t h_t(x)\right)$ .
- 

of boosted classifiers.

$$\underset{\alpha_t, h_t(\mathbf{x})}{\text{minimize}} \quad \mathbf{E} \left[ \exp \left( \sum_{i=1}^{t-1} \alpha_i h_i(\mathbf{x}) + \alpha_t h_t(\mathbf{x}) \right) \right], \quad (2.4)$$

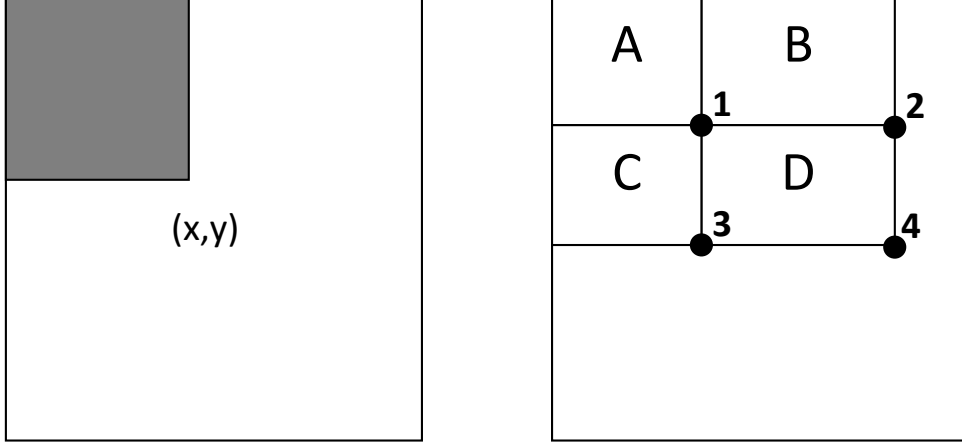
where  $\mathbf{E}$  denotes the expectation with respect to the empirical distribution. The new weak learner is selected from a set of feature classifiers. An algorithmic overview of AdaBoost is shown in Algorithm 1.

### 2.2.3 Viola and Jones' Haar-like Features with Integral Image

The authors proposed a very efficient way to compute Haar-like features based on a new image representation called *integral image*. A sum of pixels in any rectangles can be computed very rapidly using the integral image. The integral image at location

## 2. RELATED WORKS

---



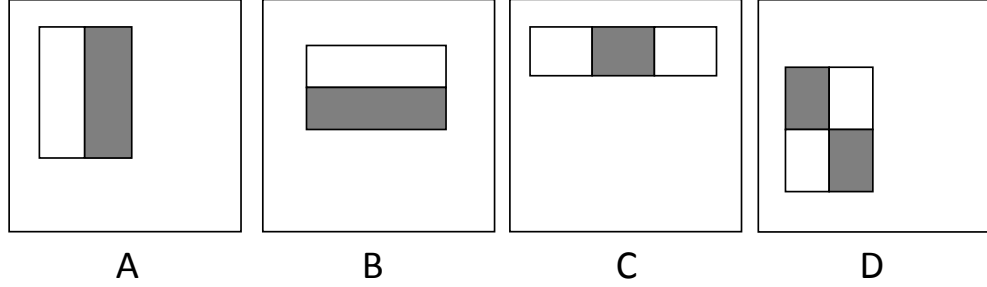
**Figure 2.3:** *Left:* The value of the integral image at point  $(x, y)$ ,  $J(x, y)$ , is the sum of all pixels above and to the left. *Right:* The sum of pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of pixels in rectangle A. The value at location 2 is  $A + B$ , at location 3 is  $A + C$ , and at location 4 is  $A + B + C + D$ . The sum within D can be computed as  $4 + 1 - (2 + 3)$ . Courtesy of [140].

$(x, y)$  contains the sum of pixels above and to the left of  $(x, y)$ :

$$J(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y'), \quad (2.5)$$

where  $J(x, y)$  is the integral image and  $I(x, y)$  is the original image. By using the integral image, the sum of pixel intensities of a rectangle can be computed using at most four references to the integral image, independent of its location or size. Figure 2.3 gives an overview of integral image and operations required to compute the rectangular sum. Once integral image is computed, Haar-like features can be calculated in constant time. Three different kinds of Haar-like features at various scales and locations were used in their paper (Figure 2.4).

Despite the success of object detector proposed by Viola and Jones, there exists a number of challenging learning issues. Numerous researchers in recent year have focused their attention to various aspects of these issues, *e.g.*, improving the cascade classifier, learning an alternative classifier instead of AdaBoost, designing better features, improving the complexity of weak learners' training time. In the next section,



**Figure 2.4:** An example of Haar-like rectangle features. The sum of pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles: (A,B) Two-rectangle features (C) three-rectangle features and (D) four-rectangle feature. The value of a two-rectangle feature (A,B) is the difference between the sum of pixels within two rectangular regions. A three-rectangle feature (C) computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally, a four-rectangle feature (D) computes the difference between diagonal pairs of rectangles. Courtesy of [140].

we categorize these improvements into four categories and discuss each of them in details.

## 2.3 Improvements of Boosted Cascade Classifiers

---

In this section, we discuss issues related to the traditional boosted cascade classifier and review recently proposed methods to overcome these drawbacks.

### 2.3.1 Shortcomings of Cascade Designs

The drawback of Viola and Jones' cascade is that it is not known beforehand how many boosted classifiers are needed or which combination of Receiver Operating Characteristics (ROC) curves produces an optimal cascade. In the original design, these parameters are obtained mainly by trial and error. For simplicity, Viola and Jones used the same detection rate and false positive rate for all nodes in the cascade structure. In this section, we discuss some of recently proposed techniques that deal with these drawbacks. Table 2.2 briefly summarizes issues related to the traditional cascade classifier and recently proposed approaches

## 2. RELATED WORKS

---

**Table 2.2:** An overview of recently proposed approaches over cascade structure

	Drawbacks of original cascade	Proposed approaches
[131]	By fixing a detection rate, <i>e.g.</i> , 99%, and a false positive rate, <i>e.g.</i> , 50%, at each node classifier, the cascade classifier is sub-optimal.	The authors proposed a cascade indifference curve framework for automatic cascade learning.
[12]	Information learned from earlier nodes are discarded.	The authors trained one large strong classifier and injected decision threshold at every weak classifier.
[130]	Existing approach used ad hoc parameter setting.	The authors formulated this trade-off as Wald's sequential probability ratio test and build a single boosted ensemble.
[73]	Node thresholds designed using Viola and Jones' cascade classifier is not optimal.	The authors proposed to globally allocate optimal trade-offs across all node classifiers in the cascade.
[146]	Improve upon [12]	The authors proposed to train one large strong classifier, similar to [12], but incorporated several improvements.
[147]	Only negative samples are bootstrapped. Information of weak classifiers from previous nodes is ignored.	The authors proposed an efficient learning algorithm that bootstraps both positive and negative samples. A new cascade structure was also introduced.
[31]	The cascade classifier does not generalize well on noisy data.	The authors proposed the joint optimization of cascade classifiers.
[17]	The cascade classifier is sub-optimal	The authors introduced a fully-automatic framework for training a cascade classifier based on a probabilistic prediction.
[111]	Tradeoff between speed and accuracy wasn't well studied in [12]	The authors proposed multi-exit classifier, where previous scores are propagated from one boosted classifier to the next.

## 2.3 Improvements of Boosted Cascade Classifiers

---

Some progress towards automatic training of cascade classifiers has been made. Sun *et al.* [131] proposed a cascade indifference curve framework for automatic cascade learning algorithm. The approach connects the learning objective for an individual node to the overall cascade performance. In their paper, a new cost function based on a cascade risk is derived for learning a node classifier. The authors adjusted the learning goal according to the difficulty of node learning problems and demonstrated that this new cost function yields an optimal learning goal for each node.

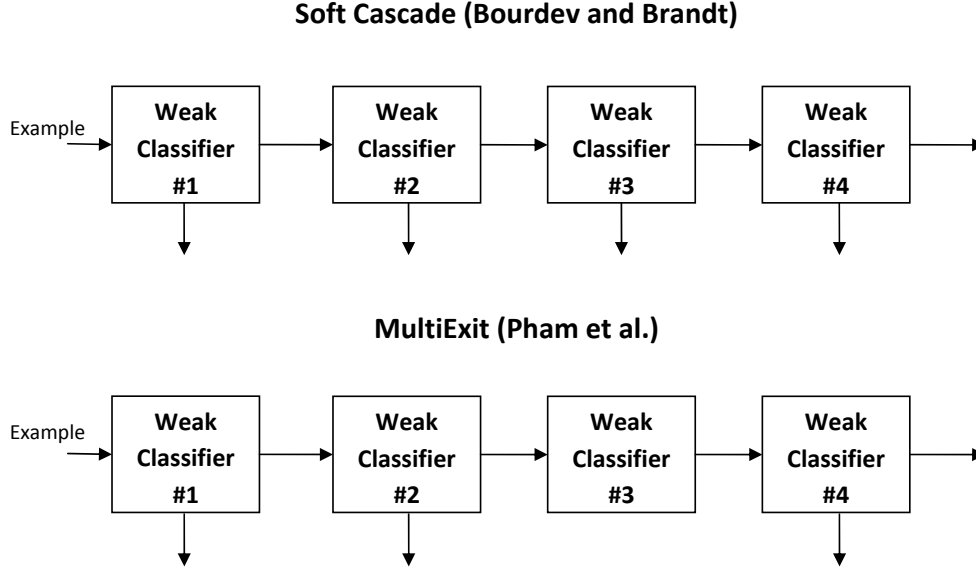
Based on their paper, the approach assumes that subsequent nodes perform similar to the previous node. However, achieving the same node objective (e.g. a detection rate of 99.5% and a false positive rate of 50%) in every node might be almost impossible, *i.e.*, negative samples in subsequent nodes are often harder to classify. Similar to Viola and Jones, the authors introduced an additional parameter, termed maximum number of weak classifiers per node, to terminate node learning when an additional computation cannot decrease cascade risk cost function. In summary, the approach consists of three unknown parameters, *i.e.*,  $[f_{min}, f_{max}]$ , which indicates a range of acceptable false positive rates in each stage, a trade-off parameter,  $\lambda$ , which balances speed and the predicted detection rate. Finding optimal values among these parameters often require extensive experiments.

Bourdev and Brandt [12] suggested that the traditional cascade classifier had a large number of weak classifiers because information learned from weak classifiers in early nodes were discarded. They generalized the cascade structure by training a single monolithic boosted classifier using AdaBoost and called it the *Soft Cascade*. The authors proposed a calibration algorithm that breaks the large boosted classifier into a cascade by augmenting a rejection threshold function into every weak classifier (Figure 2.5). The authors demonstrated that the approach achieved a high detection rate using fewer features compared to state-of-the-art detectors. The advantage of their methods is that the accuracy/speed trade-off can be systematically explored through the ROC surface. Although the authors showed a performance improvement, it is arguable whether their structure would lead to an optimal cascade. Since thresholds were obtained after all weak classifiers had been trained, their final classifier could be sub-optimal.

Sochman and Matas [130] formulated a classification problem in the framework of sequential decision-makings called *WaldBoost*. The decision threshold is chosen based

## 2. RELATED WORKS

---



**Figure 2.5:** An illustration of classifiers with soft cascade [12] (*top*) and multiple exit nodes [111] (*bottom*).

on the class-conditional response of sequence of strong classifiers. During evaluation, the detector decides whether to accept an instance, reject an instance or continue the evaluation to the next weak classifier. WaldBoost can be considered as a theoretically justifiable boosted cascade classifier proposed by Viola and Jones. The authors evaluated their approach on face detection problem and showed that their results are superior to the state-of-the-art method. Since the approach has a decision threshold at every weak classifier, bootstrapping is required after each weak classifier training. Hence, their approach has a high computational cost during training.

Luo [73] proposed an optimization algorithm for designing a cascade classifier. The approach attempts to jointly optimize thresholding parameters of all node classifiers after the full cascade has been trained. Their performance improvement clearly signifies the importance of node thresholds in cascade classifier. In their paper, it was suggested that the approach could serve as a useful post-processing process for cascaded design. However, two important things seems not to be addressed in their paper. The first one is related to node decision thresholds and bootstrapped data. In traditional cascade classifier, modifying decision threshold value could result in a com-



## 2.3 Improvements of Boosted Cascade Classifiers

---

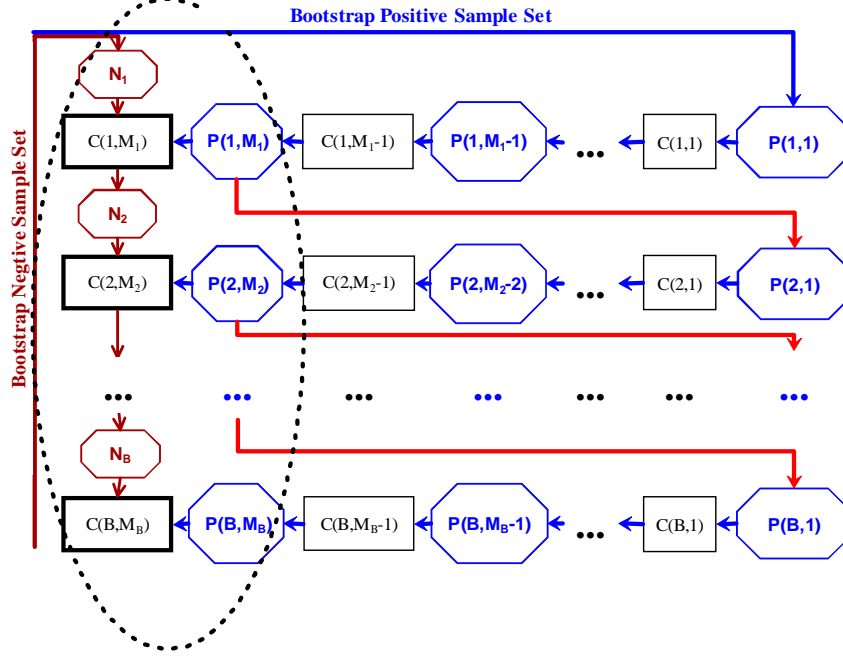
pletely different bootstrapped data. In their approach, it remains a challenge whether weak classifiers in later nodes should be modified based on their new threshold value. The other is whether threshold should be chosen during cascade training instead of after full cascade has been trained.

Xiao *et al.* [146] proposed a novel cascade structure called *Dynamic Cascade*, for training an efficient face detector on massive data sets. Unlike the Soft Cascade, which calculates thresholds after all weak classifiers have been trained, Dynamic Cascade calculates rejection thresholds and updates training sets before training each weak classifier. To address the challenge of massive training sets, the authors made use of a weight trimming technique [38]. Experimental results show that their approach effectively improves the detection performance. However, similar to the Soft Cascade, it remains a challenge whether it is beneficial to have a decision threshold at every weak classifier. Also, their approach has a high computation cost during training since bootstrapping is required after each weak classifier training.

Yan *et al.* [147] proposed a novel matrix-structural learning (MSL) method which overcomes the limitation of cascade classifiers (Figure 2.6). Unlike in the original Viola and Jones' cascade, which bootstraps only negative samples (Section 2.2), the proposed structure bootstraps both positive and negative samples. The authors also proposed an accumulative technique to inherit features learned previously. The authors used 230,000 face training samples to learn a classifier and their detector achieves a performance better than the state-of-the-art on CMU+MIT frontal face test sets.

Since increasing the size of training set can further improve the classification performance, it remains a challenge whether a performance gain in their experiment is the result of using a larger training set or the result of adopting their MSL method. Based on their experimental results, their performance performs similar to Bourdev and Brandt [12], which also reuses some of previously learned classifiers (*i.e.*, they inject the decision threshold at every weak classifier). Hence, it remains a challenge whether the performance gain is a result of MSL or their feature-inheriting technique. In my view, bootstrapping is a process of collecting small and representative training sets. If all positive training samples can be fit into memory, their technique would turn out to be obsolete. However, if one is fortunate to have a massive training set of positive samples, bootstrapping positive samples might be more convincing.

## 2. RELATED WORKS



**Figure 2.6:** An illustration of matrix-structural learning.  $C(i, j)$  indicates classifier built by  $i^{th}$  stage learning from  $j^{th}$  positive training sample set.  $P(i, j)$  indicates positive training set for  $C(i, j)$ .  $N_i$  indicates bootstrapped negative training set for  $C(i, .)$ .  $B$  is the total number of stages.  $M_i$  is the iteration number of positive bootstrap of  $i^{th}$  stage. Each row is an iteration of positive samples bootstrap while the negative bootstrap is conducted similar to Viola and Jones' framework. Courtesy of [147].

Dundar and Bi [31] proposed a different training architecture known as AND-OR learning. Instead of training each node classifiers independently, all nodes were trained in a joint fashion. Unlike, traditional approach, which used greedy algorithm to train node classifiers sequentially, their approach optimizes all node classifiers in parallel based on mutual feedback between node classifiers. The basic intuition behind their approach is based on the fact that an example is classified as positive if it is labeled as positive by all nodes and negative if it is rejected at any node. The algorithm assigns different loss functions to positive and negative samples. The approach iteratively optimizes the overall cascade performance by adjusting a single node's parameters while fixing all other nodes' parameters. The authors applied the approach to the problem of automatically detecting polyps from multi-slice CT images and showed a

## 2.3 Improvements of Boosted Cascade Classifiers

---

significant speed-up while achieving comparable performance to the current state-of-the-art.

Brubaker *et al.* [17] proposed another fully-automatic framework for training a cascade classifier based on a probabilistic prediction. They used a probabilistic framework based on validation time. Based on this probability, a cost function is defined. The cost model allows one to decide the minimal amount of required weak hypotheses and offers a better exploitation of false alarms versus correct detection rate trade-off. Additionally, existing node classifiers can be split into smaller classifiers in order to improve efficiency. By splitting the node classifier, negative samples can be discarded with fewer evaluations.

Pham *et al.* [111] proposed a boosted classifier with multiple exit nodes. The authors combined the idea of propagating scores across boosted classifiers with the use of asymmetric goals. The intuition behind their approach is that the classification problem becomes harder in later cascade stages since most easy-to-classify negative samples have already been removed. Having a decision made at every weak classifier, like in the Soft Cascade [12], effectively discards important information that may have been exploited if decisions are postponed until further downstream (Figure 2.5). In their approach, the classification score obtained from previous boosted classifier is propagated to the next classifier. Experimental results shows a significant reduction in training time and number of weak classifiers, as well as better accuracy, compared to conventional cascades and multi-exit boosted classifiers.

### 2.3.2 Shortcomings of AdaBoost and Boosted Classifiers

Since the objective of AdaBoost is to minimize misclassification error, Viola and Jones introduced a new decision threshold to each boosted classifier to guarantee a high detection rate with moderate false positives. However, in object detection problem, the probability of observing a positive sample is much lower than the probability of observing a negative sample. Their simple modification turned out to be sub-optimal and had raised a number of questions. In recent year, a number of researchers, including Viola and Jones themselves, have later addressed this problem by introducing new learning objective function that penalize a false negative much more than a false positive (asymmetric objective). Furthermore, several researchers raised a number of questions

## 2. RELATED WORKS

---

related to AdaBoost coefficients and introduced alternative classifiers, which are more suited to object detection problem. Table 2.3 briefly summarizes issues related to the traditional boosted classifier and recently proposed approaches.

Fan *et al.* [35] proposed AdaCost where a misclassification cost adjustment function is introduced into the weight updating rule. The cost function increases weights of costly wrong classifications more aggressively, but decreases weights of costly correct classifications more conservatively. In brief, weights for expensive examples are higher and weights for inexpensive examples are comparatively lower. As a result, the final ensemble will correctly predict more costly instances. The authors evaluated their algorithms on seven data sets using Cohen’s RIPPER [21] as a weak learner. They observed that AdaCost shows a consistent and significant reduction in misclassification cost over AdaBoost.

Nonetheless, the technique has some drawbacks when applying to object detection problems. Firstly, there is no fixed rule to estimate a trade-off parameter between false positive and false negative. Unlike in fraud detection applications, where prior experience can be used to estimate average financial cost of false positive and false negative, in face detection, one has to search for the cost factor that achieves the pre-defined node learning goal. This often needs extensive trials for best performance. Choosing the wrong parameter value could result in a complex classifier which barely rejects any negative samples and cannot be run in real-time.

Secondly, using AdaCost threshold does not always guarantee the cascade objective (high detection rate with moderate false positives). Adjusting this parameter might again result in sub-optimal performance. Finally, to find boosting coefficients, the authors suggested that the estimation method can be used to find a candidate [37] and numerical methods can be applied to fine-tune this estimate [124]. Nonetheless, their technique of estimating boosting coefficients are not provably optimal and the algorithm can perform worse than simply using traditional AdaBoost.

Viola and Jones [139] later addressed the problem of highly skewed example distributions in cascade classifiers by introducing a new learning objective function that penalizes a false negative much more than a false positive. In the new objective function, they introduce an additional term called asymmetric loss where a false negative costs  $k$  times more than a false positive. Here  $k$  is an asymmetric cost parameter. In terms of algorithm, the only difference between Asymmetric AdaBoost and AdaBoost

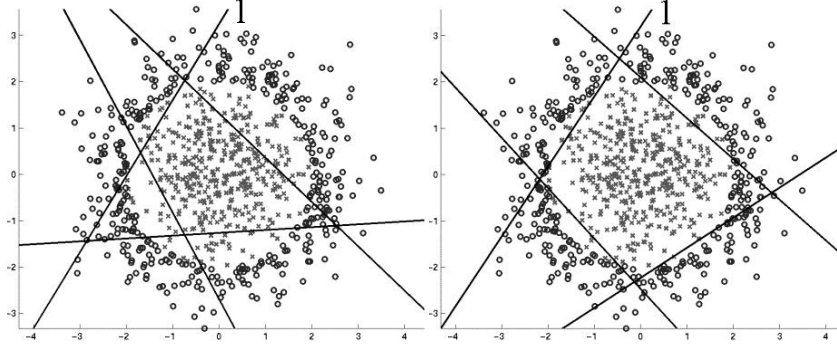
## 2.3 Improvements of Boosted Cascade Classifiers

**Table 2.3:** An overview of recently proposed approaches over AdaBoost based classifiers.

	Drawbacks of boosted classifiers	Proposed approaches
[35]	A false negative is penalized the same as a false positive.	The authors proposed AdaCost that penalizes a false negative and a false positive differently.
[139]	A false negative is penalized the same as a false positive.	The authors proposed a new learning objective function that penalizes a false negative more than a false positive (Asymmetric AdaBoost).
[68]	AdaBoost is not robust to outliers.	The authors considered different loss functions, <i>e.g.</i> , Gentle AdaBoost which has been shown to be more resistant to noisy data.
[74]	A false negative is penalized the same as a false positive.	The authors proposed a cost-sensitive AdaBoost algorithm, which has (1) unequal initial sample weights and (2) different weight updating rules for positive and negative samples.
[69]	AdaBoost and weak learners used in Viola and Jones are sub-optimal.	The authors introduced KL feature, which is based on Kullback-Leibler divergence of two-class histograms, to a boosting framework.
[67]	AdaBoost is a stagewise greedy optimization, which can be characterized as being short-sighted and non-recoverable.	The authors introduced a backtrack mechanism to AdaBoost learning.
[52]	Improving upon [74, 139], where fixed asymmetric factors were used for all subsequent weak learners.	The authors proposed Asymmetric AdaBoost with variable asymmetric factor.
[78]	A false negative is penalized the same as a false positive.	The authors proposed another alternative cost-sensitive boosting algorithm.
[110]	Improve upon [139]	The authors defined a rule so asymmetric goal can be chosen prior to training.

## 2. RELATED WORKS

---



**Figure 2.7:** A shortcoming of AdaBoost illustrated on toy data sets.  $\times$ 's and  $\circ$ 's represent positive and negative samples, respectively. Weak classifiers are linear separators. The first feature selected is labeled '1'. On the left is the result of symmetric AdaBoost. Subsequent features attempt to balance positive and negative errors (note the large number of false negatives). On the right is the result of asymmetric AdaBoost. The final strong classifier yields very high detection rates and moderate false positive rates. Courtesy of [139].

is in the distribution of sample weights. For Asymmetric AdaBoost, one needs to multiply additional asymmetric parameter,  $\exp(\frac{1}{N}y_i \log \sqrt{k})$ , to sample weights before each round of boosting. Intuitively, by increasing sample weights of misclassified positive samples, the subsequent weak learner is forced to focus asymmetrically on these positive examples. The authors showed that their new asymmetric loss yields significant improvements in performance over conventional AdaBoost on both toy data sets (Figure 2.7) and face data sets.

The major drawback of their approach is in choosing the optimal asymmetric parameter,  $k$ . The second drawback of their approach is that the asymmetric parameter is always fixed for all subsequent weak learners. Although this parameter can be calculated by cross-validation, there is no guarantee that the selected parameter will yield higher performance results than using the original AdaBoost classifier. Furthermore, it remains a challenge whether it would be beneficial to train classifier using Asymmetric AdaBoost in later cascade stages when negative patches look visually similar to positive patches.

Lienhart *et al.* [68] proposed to apply different boosting algorithms, namely Real

## 2.3 Improvements of Boosted Cascade Classifiers

---

and Gentle AdaBoost, together with various weak learners, *e.g.*, decision stumps and decision trees for training face detector. The only difference between each boosting algorithm is in their learning. Real AdaBoost is a more generalized version of the original AdaBoost [124] and Gentle AdaBoost is a more robust implementation of Real AdaBoost [38]. Real AdaBoost is based on confidence-rated predictions while Gentle AdaBoost is based on weighted least square. Due to weighted least square, Gentle AdaBoost has been shown to be more resistant to outliers and perform considerably better than AdaBoost on noisy data. Based on their experimental results, Gentle AdaBoost with CART trees is the most successful learning procedure tested for face detection.

Ma and Ding [74] proposed a method of detecting faces based on Cost-Sensitive AdaBoost (CS-AdaBoost) algorithm. Two main differences between CS-AdaBoost and the original AdaBoost are (1) unequal initial weights according to its misclassification cost and (2) sample weights are updated separately for positive and negative samples at each boosting step. Compared to the original face detector [140] which has 38 layers and 6,000 features, their detector only has 20 layers and 3,000 features. Nonetheless, a few important information is missing in their paper, *e.g.*, the value of cost parameter used in their cascade training, how they derived this parameter and it remains a challenge whether this parameter should be adjusted in each cascade layer. Currently, it seems this parameter is very application dependent and extensive experiments are needed to achieve the best performance.

Liu and Shum [69] introduced a Kullback-Leibler boosting (KLBoosting) to derive weak learners by maximizing projected KL distances. KLBoosting computes weak learners by maximizing the relative entropy between two 1-D projected distributions of face and non-face samples. Unlike conventional AdaBoost, KLBoosting learns the coefficients by minimizing the recognition error each time a new feature is added to the classifier. The drawback of KLBoosting is that data weights are updated according to authors' heuristic formulas.

Li and Zhang [67] proposed an extension to AdaBoost, called *FloatBoost*. FloatBoost applies a backtrack mechanism after each iteration of AdaBoost learning to minimize the error rate. The backtrack mechanism deletes those weak classifiers, which do not help in terms of the error rate, from the set of learned weak classifiers. Since deletions in backtrack are performed according to the error rate, a lower error rate and



## 2. RELATED WORKS

---

reduced feature set are guaranteed. The authors showed that by incorporating the idea of backward elimination into AdaBoost, the final detector achieves a lower error rate with the same number of weak classifiers. However, one major drawback of FloatBoost is that the algorithm takes a long time to train, especially for those classifiers in later cascade stages. In my opinion, in later cascade stages, negative patches can look visually similar to positive patches. The system continues removing a slightly worse weak classifier (backward elimination) and adding a slightly better weak classifier (forward selection) to the set of selected weak classifiers. Although the training error continues to decrease slowly, it is arguable whether this would lower the generalization error.

Improving upon the work of Fan *et al.* [35], Masnadi-Shirazi and Vasconcelos proposed Asymmetric Boosting, which exploited the statistical interpretation of boosting by replacing symmetric loss with asymmetric loss. The cost-sensitive extension minimizes this asymmetric loss by gradient descent on the functional space of convex combinations of weak learners. Using gradient descent, optimal coefficients can be computed on an average of 6 iterations of bisection search. The authors evaluated their algorithm on a face database of 9832 positive and 9832 negative examples. 6,000 samples were used for training and the remaining is used for testing. Weak learners and visual features used in their experiment are the same as in Viola and Jones. Asymmetric boosting was shown to consistently outperform all other methods, achieving the smallest misclassification cost at all cost factors evaluated. Although the authors have shown accuracy improvement compared to [35], it remains a challenge how asymmetric loss can be applied to cascade classifier. In the early stage of cascades, the cost factor between positive and negative samples might be large. However, in later stages when negative samples are harder to separate, this cost factor might need to be changed accordingly. However, this has not been pointed out in the paper.

The drawback of Asymmetric AdaBoost, proposed by [74, 78, 139], is that the asymmetric factor was fixed for all subsequent weak learners, *i.e.*, in Viola and Jones [139], the authors distributed the asymmetric weights among all weak classifiers equally. As a result, subsequent weak classifiers often fail to balance the overall node objective (requirement for high detection rate and moderate false acceptance rate).

Hou *et al.* [52] proposed the Asymmetric AdaBoost with variable asymmetric factor. They assigned different values for positive and negative samples (they use the real number,  $a$ , for positive samples instead of  $+1$  and a real number,  $-b$ , for negative



## 2.3 Improvements of Boosted Cascade Classifiers

---

samples instead of  $-1$ ). For each weak learner, the authors calculated the maximal margin by fine-tuning the parameter  $a$  and  $b$ . The weak learner with maximal margin is selected. AdaBoost coefficients and sample weights are updated based on the AdaBoost algorithm. By automatically selecting the most appropriate asymmetric factor for each weak learner, the algorithm is shown to perform slightly better than other AdaBoost-based methods. Although the asymmetric factor can now be computed, the authors have introduced another parameter,  $a$  and  $b$ . It remains a challenge whether this parameter should be fixed in each cascade layer or adjusted accordingly. Another drawback of their technique is the computation cost in training weak learners. Tuning both parameters for hundred thousand features can be very computationally expensive.

Instead of distributing asymmetric weights equally among all weak classifiers, Pham and Cham [110] distributed asymmetric weights based on equal label skewness. In other words, asymmetric factor is chosen to ensure equal label skewness presented to weak classifiers. By balancing the skewness of labels, the authors reported a performance gain compared to naively assigning equal asymmetric weight [139]. Although the approach sounds convincing, there are a number of issues involved. Firstly, similar to [139], the approach has two unknown parameters; namely asymmetric factor and the total number of weak learners. Since the number of weak learners in each cascade stage is often not known beforehand, one has to estimate the total number of weak learners before one can estimate the asymmetric factor. Without any prior knowledge, it is quite difficult to estimate this value. Secondly, the authors proposed an approach to distribute asymmetric factor among all weak learners but failed to mention whether this asymmetric factor should be distributed among nodes in the cascade classifier.

Recently, more and more boosting techniques have been proposed [29, 36, 38]. Some examples are LogitBoost and BrownBoost. LogitBoost was first formulated by Friedman *et al.* [38] as a boosting algorithm which applies logistic regression as the cost function. Unlike AdaBoost, the algorithm minimizes the logistic loss instead of exponential loss.

BrownBoost was first introduced by Freund [36]. Unlike AdaBoost, which focuses on repeated misclassified samples, BrownBoost ignores these samples which are repeatedly misclassified. In other words, samples are divided into two classes, noisy samples and non-noisy samples. The final classifier is learned only from those non-noisy samples. BrownBoost uses a non-convex loss function. It solves a system of

## 2. RELATED WORKS

**Table 2.4:** An overview of recently proposed approaches over Haar-like features.

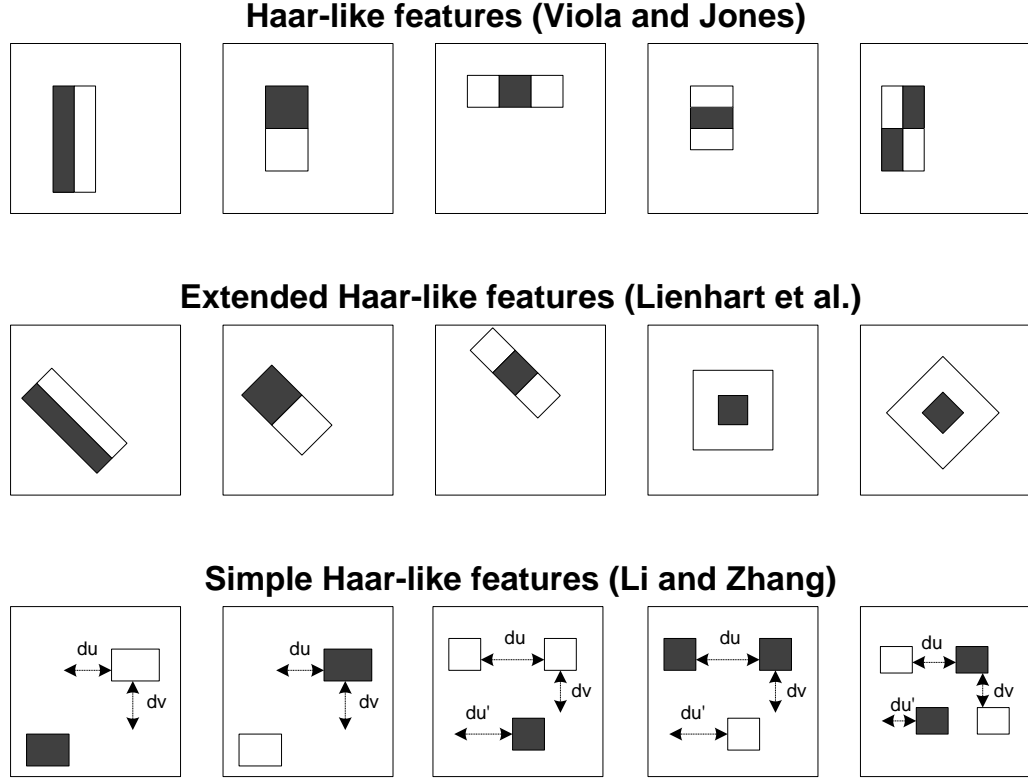
	Drawbacks of Haar-like features	Proposed approaches
[68]	Haar-like features fail to capture diagonal edges.	The authors proposed efficient rotated Haar-like features.
[67]	The features work well only frontal faces.	The authors separated the Haar-like wavelet boxes for multi-view face detection.
[65]	Haar-like features require a huge training database for good performance.	The authors proposed EOHs for both frontal and profile view faces.
[144]	Haar-like features do not work well on human.	The authors proposed Edgelet features for body part detection.
[53]	Haar-like features are limited to frontal face detection due to their rigorous structural constraints.	The authors proposed sparse granular features, which represent a sum of pixel intensities in a square.
[109]	Haar-like features proposed only represent edge, line and diagonal line.	The authors included corner and center-surrounded features. To improve weak classifier training time, the authors used statistical based features.

two equations and two unknowns (hypothesis coefficient and amount of time) using standard numerical methods.

### 2.3.3 Shortcomings of Haar-like Features

The simplicity of Haar-like features is the key to a success of Viola and Jones' frontal face detector. However, the features are not discriminative enough to distinguish more complex objects, *e.g.*, profile view of faces, pedestrian and vehicles. Table 2.4 briefly summarizes issues related to traditional Haar-like wavelet features and recently proposed approaches.

Numerous researchers have introduced more types of wavelets to extend Haar-like's discriminative power. Lienhart *et al.* [68] proposed to generalize Haar-like features by in-plane rotating Haar-like features by 45 degrees (Figure 2.8). For fast feature



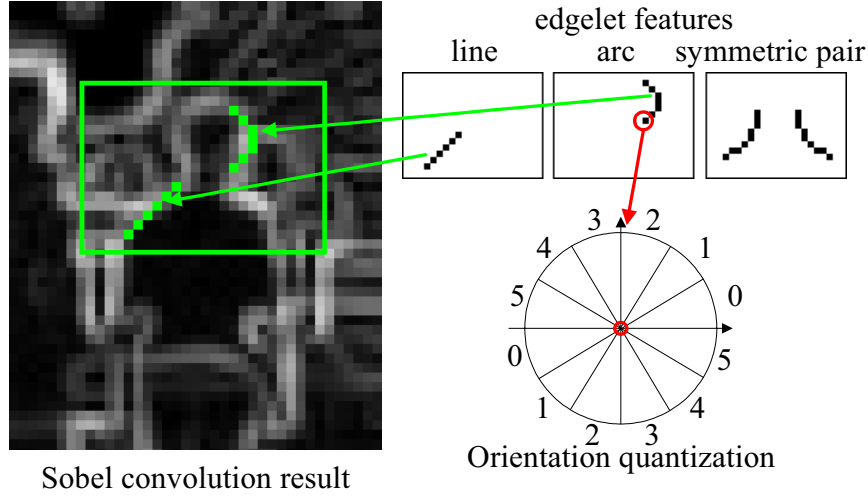
**Figure 2.8:** An illustration of variants of Haar-like features [140] (*top*), Extended Haar-like features [68] (*middle*) and Simple Haar-like features [67] (*bottom*). Courtesy of [67, 68, 140].

extraction, they introduced a 45-degree rotated integral image. Interestingly, the total number of proposed feature sets is only 30% larger than the total number of Haar-like features (117, 941 versus 91, 536). Based on their experimental result, their new rotated features yield an average of 10% lower false alarm rate at the same hit rate compared to Haar-like features. This finding indicates that features are one of the most important factors required to achieve a robust face detector and that there is still room for further improvement.

Li and Zhang [67] proposed a *simple Haar wavelet*, which separates Haar-like rectangles at some distance apart (Figure 2.8). The authors tested their proposed features on multi-view faces and demonstrated excellent performance. However, it takes a very long time to train their face detector since their total number of features is a few orders

## 2. RELATED WORKS

---

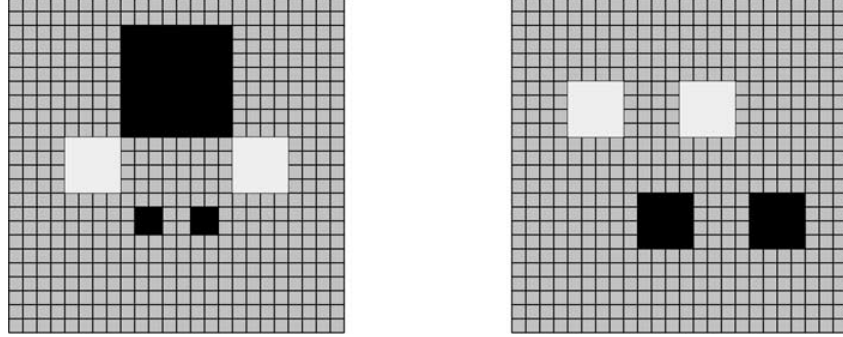


**Figure 2.9:** An illustration of Edgelet features. Courtesy of [144].

of magnitude larger than Haar-like features.

Levi and Weiss [65] proposed *local edge orientation histograms* (EOHs), which divide edges into a number of bins. Three set of features are used to describe an image region:- a ratio between each orientation, a ratio between a single orientation and the difference between two symmetric orientations. For frontal face detection, EOHs achieve state-of-the-art performance while using only a few hundred training images. For profile view faces, EOHs outperform the state-of-the-art in real-time systems even with a small number of training examples. However, compared to Haar-like features, their features have higher computation time and memory storage.

Wu and Nevatia [144] improved the work of Viola and Jones by modeling human as an assembly of natural body parts. The authors introduced a new type of silhouette oriented features, called *Edgelet features* (Figure 2.9). Part detectors are learned by a boosting method and responses are combine to form a joint likelihood model that includes cases of multiple, possibly inter-occluded humans. The detection method results in better performance for individual human detection and furthermore can deal with crowded scenes. Based on their implementation, the detector performed at about 1 frame per second on an image with a resolution of  $384 \times 288$  pixels on a 2.8GHz CPU. However, detecting faces is often simpler than detecting human and it is computationally expensive to compute the likelihood for multiple faces when overlapping



**Figure 2.10:** An illustration of sparse granular features. White and black blocks are positive and negative granules, respectively. Each granule can be pre-computed and calculated in one lookup table (access memory only once). Courtesy of [53].

between two faces rarely occurs.

Huang *et al.* [53] further extended Haar-like features in a slightly different way. Instead of using rectangles, they proposed *sparse granular features*, which represent a sum of pixel intensities in a square (Figure 2.10). An efficient weak learning algorithm is introduced which adopts heuristic search method in pursuit of discriminative sparse granular features. Since sparse granular features have a smaller rectangular region than Haar-like features; it has a better discriminative power for multi-view faces due to their less within-class variance. The advantage of using sparse granular features is that features can be calculated in one lookup table, compared to Haar-like features, which often require 6 - 9 memory operations. However, the major disadvantage of sparse granular features is that the approach is heuristic and has a large number of parameters. The performance of the system relies heavily on sparse granular features which are calculated heuristically.

Pham and Cham [109] proposed to use only statistics of the weighted input data to train a weak classifier. The advantage of using statistical based features is that it has a smaller training time complexity,  $O(Nd^2 + M)$ , compared to the traditional technique,  $O(NM \log N)$ , where  $N$  is the number of samples (approximately 10,000),  $d$  is the number of pixels in the training sample (usually less than 500) and  $M$  is the number of features (approximately 100,000). Since the training time complexity is no longer a product of the number of samples and the number of features, the authors can

## 2. RELATED WORKS

---

afford to introduce more mother wavelets to the feature set. In addition to edge, line and diagonal line features, the authors included corner and centre-surrounded features. Experimental results revealed a significant reduction in weak classifiers' training time to the order of seconds. However, the drawback of statistical based features is that it is not robust to noisy data and outliers.

Recently, more invariance local features have been applied to object detection tasks. Some examples are SIFT-based descriptors [72] and local binary pattern (LBP) texture operator [96]. The SIFT descriptor is a 3-D histogram of gradient locations and orientations. A block is quantized into a number of  $n \times n$  cells and the gradient angle in each cell is quantized into  $k$  orientations. In the original paper, Lowe set  $n$  to be equal to 4 and  $k$  to be equal to 8, which result in 128-dimensional descriptors. To reduce the effect of illumination changes, the descriptor is normalized to a unit length. The influence of large gradient magnitudes is minimized by thresholding the descriptor in each histogram bin to a maximum value of 0.2.

LBP is a simple and powerful texture operator. It labels pixels by thresholding the neighbourhood of each pixel with the value of the centre pixel and considers the result as a binary number. Due to its computational simplicity and high discriminative power, LBP has become a popular method in various applications, *e.g.*, face recognition [3], moving object detection [50], facial expressions [153], *etc.* LBP has several properties that favour its usage as texture descriptor. The features are robust against monotonic gray-scale changes, fast to compute, do not require many parameters to be set and have high discriminative power.

### 2.3.4 Other Issues Related to Viola and Jones' Face Detector

Since the choice of good training samples plays an important role in the generalization ability of cascade classifiers, Chen *et al.* [20] presented an approach to optimize the training data. The authors proposed a genetic algorithm and manifold-based method to resample a given training set for more robust face detection. An initial training set is first expanded by a genetic algorithm (crossover operations) and undergone re-lighting (mutations) to generate new samples. After each round of genetic algorithms, generated samples are tested against a face classifier and those which have too large variations are discarded. Once a sufficient amount of samples have been generated,

a manifold space is created by means of the Isomap algorithm to represent the local distances of the face space in a lower-dimensional space. Finally, the set was resampled to a sufficiently large set of samples that evenly covered the original face space. In their paper, a one-class SVM was used as an additional last step to reject further false positives, thereby lowering the false alarm rate compared to cascade classifiers. Their approach achieves 90.73% accuracy with no false alarm on MIT+CMU frontal face test set. This finding emphasizes the important of having well-represented training samples.

Although a cascade of boosted classifiers exhibit real-time run-time performance, training time usually takes from days to weeks. There have been a number of attempts to shorten the training time while maintaining run-time accuracy. Based on the training time complexity, factors that affect the training time are the amount of training samples and feature-set size [145]. The traditional training approach has a run-time of  $O(MTN\log(N))$  where  $N$  represents the number of samples,  $M$  is the number of features and  $T$  is the number of boosting iterations. Since reducing the number of training samples often deteriorate generalization ability, a lot of work have focused on reducing the size of feature sets by removing less discriminative features.

Bourdev and Brandt [12] focused on shortening the training time by reducing the number of trained weak classifiers. Wu *et al.* [145] proposed a forward feature selection (FFS) algorithm in order to avoid re-training AdaBoost weak learners which is time consuming. Wu *et al.* first trained weak classifiers using uniform sample weights. A subset of pre-trained weak classifiers are sequentially added to an ensemble classifier until the classifier's learning goal is met. The overall time complexity of the FFS is  $O(NMT + NM\log(N))$ , which is only  $1/T$  of the AdaBoost training time. Pham and Cham [109] proposed to model a response of Haar-like features with Gaussian distribution. The approach avoids the recalculation of feature values by using Gaussian to model the distribution of feature values. Therefore, the training time can be reduced to  $O(TNd^2 + TM)$ , where  $d$  is the amount of pixels within the probed feature region.

The most successful and straight-forward approach to reduce the training time is probably the use of memory caching via lookup table as suggested by Wu *et al.* [145]. In AdaBoost, each weak classifier has to be trained based on the current weight distribution. For Haar-like features, the classification is a simple threshold decision on the scalar difference of rectangular areas. In each boosting iteration, sample weights

## 2. RELATED WORKS

---

change, but not the computed rectangular areas. Precalculating and caching the order of these rectangular sums for all samples is the fastest way to train a weak learner. Their technique was able to compute the best threshold for each feature in time  $O(N)$  regardless of sample weights. Using their strategy, it is possible to reduce the training time from  $O(NMT \log N)$  to  $O(NMT)$ . In their paper, the authors reported a significant speed-up in training time compared to the traditional approach. The drawback of their technique is that it requires a large amount of memory to store the sorted order of features. As an example, to train 40,000 samples on  $24 \times 24$  pixels faces, which have at least 100,000 Haar-like features, requires at least 10 GB of RAM (Random Access Memory).



# 3

## Fast Pedestrian Detection using Boosted Covariance Features

### 3.1 Introduction

---

Efficiently and accurately detecting pedestrians is of fundamental importance for many applications in computer vision, *e.g.*, smart vehicles, surveillance systems with intelligent query capabilities, sports video content analysis. In particular, there is growing effort in the development of intelligent video surveillance systems. An automated method for finding human in a scene serves as the first important preprocessing step in understanding human activity. Despite the multitude of literature on this subject, the problem of automated object detection is far to be solved (*e.g.*, [41, 62, 104, 125, 140, 141, 144]). Pedestrian detection in still images is one of the most difficult examples of generic object detection. Most challenges are due to a wide range of poses that human can adopt, large variations in clothing, as well as cluttered backgrounds and environmental conditions.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---

In this chapter, a new simpler pedestrian detection technique using covariance features is proposed. The first contribution of this chapter is an approach to integrate multi-dimensional covariance features with weighted linear discriminant analysis for the AdaBoost classifier. To be more specific, the AdaBoost framework is adapted to vector-valued covariance features and a weak classifier is designed according to the weighted linear discriminant analysis. This technique is not only accurate but also faster. In order to support our claim, the proposed approach is compared against the state-of-the-art pedestrian detection technique evaluated in Munder and Gavrila [89].

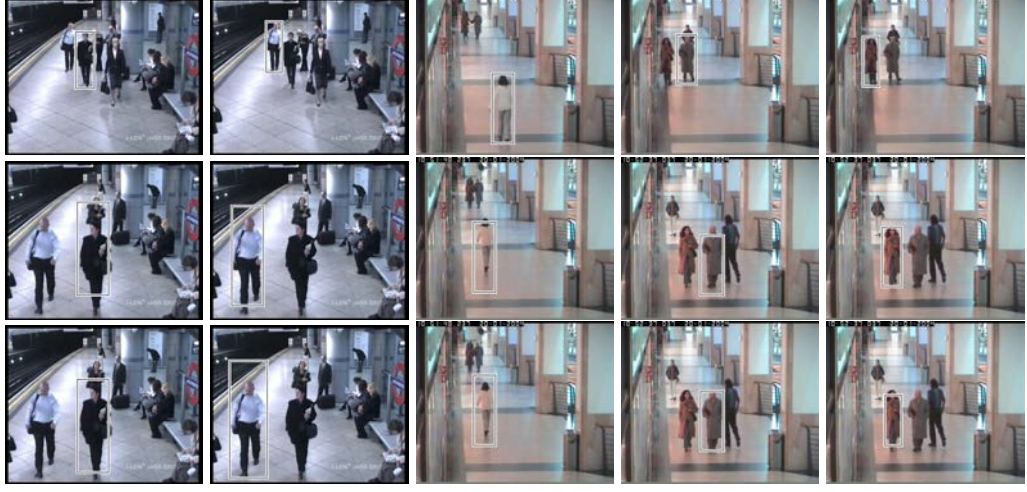
The proposed boosted covariance detector achieves about four times faster detection speed than the method proposed in Tuzel *et al.* [136]. Unfortunately, it is still not fast enough for real-time applications. On one hand, Haar-like features can be computed rapidly due to its simplicity [140] but is less powerful for classification [65]. On the other hand, covariance features are more powerful in capturing human body parts but not fast enough for a real-time performance. In order to further accelerate the proposed detector, a novel strategy known as two-layer boosting with heterogeneous features, is adopted to exploit the efficiency of Haar-like features and the discriminative power of covariance features in a single framework. Due to the flexibility of cascaded classifiers, Haar-like features based classifiers are employed at the beginning of the cascade; and covariance features are applied at latter stages. Experiments show that the proposed approach performs at an order of magnitude faster than the conventional covariance detector [136] while achieving a comparable detection performance. On a  $360 \times 288$  pixels image, the proposed system can process at around 4 frames per second with an unoptimized code. To our knowledge, this is the first real-time covariance features based pedestrian detector.

The chapter is organized as follows. Section 3.2 gives a detailed description of the proposed approach. The experimental setup and experimental results are presented in Section 3.4. The chapter concludes in Section 3.5.

## 3.2 Boosted Covariance Features

---

It is well known that the choice of weak classifiers is vital to the classification accuracy of boosting techniques. Although effective weak classifiers increase the performance



**Figure 3.1:** Detection examples on AVSS 2007 and CAVIAR data sets. *Top:* Input region. *Middle:* Best matching region found using covariance features based on distance in the Riemannian manifold [134]. *Bottom:* Best matching region found using covariance features based on distance in the Euclidean space.

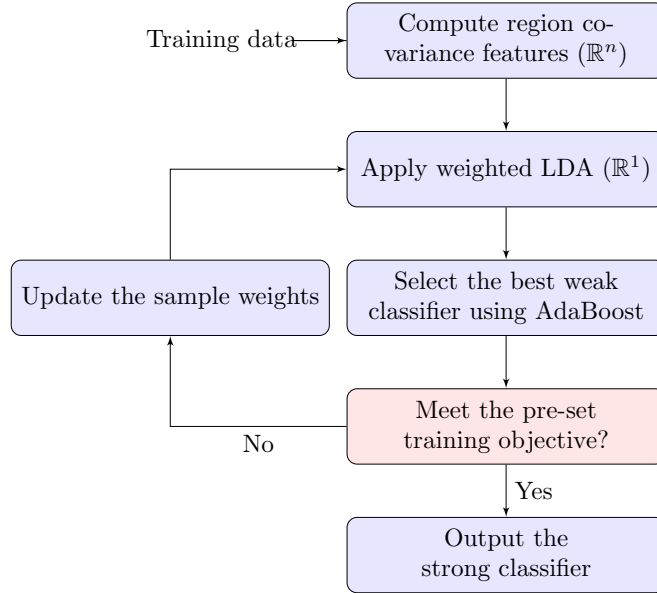
of the final strong classifiers, the large amount of potential features make the computation prohibitively heavy with the use of complex classifiers such as SVMs. For scalar features such as Haar-like features [140, 141] or LBP feature [95], a very efficient stump can be used. Unfortunately, for vector-valued features, such as HOG or covariance features, seeking an optimal linear discriminant would require much longer time. In this chapter, a more efficient approach is adopted. The multi-dimensional feature is projected onto a 1D line using weighted Fisher linear discriminant analysis (WLDA). WLDA finds a linear projection function which guarantees optimal classification of normally distributed samples of two classes.

Note that the approach is different from [134, 136], where the covariance matrix is directly used as the feature and the distance between features is calculated in the Riemannian manifold<sup>1</sup>. However, eigen-decomposition is involved for calculating the distance in the Riemannian manifold. Eigen-decomposition is computationally expensive ( $O(d^3)$  arithmetic operations). Instead, correlation coefficient is vectorized and the distance is measured in the Euclidean space, which is faster. The extracted covariance descriptor assumes that the image statistics follow a single Gaussian distribution.

<sup>1</sup>Covariance matrices are symmetric and positive semi-definite, hence they reside in the Riemannian manifold.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---



**Figure 3.2:** An architecture of the proposed pedestrian detection system using boosted covariance features.

Although this assumption may look overly simple, experiments prove the covariance features' efficacy, *e.g.*, Jin *et al.* [55] have used an identical idea for network intrusion detection.

A preliminary experiment, similar to the one described in Tuzel *et al.* [134], was conducted on covariance descriptor in Euclidean space and Riemannian space. The experiment compares two different distance measures:- distance based on two normalized covariance matrices in the Euclidean space and distance based on two normalized covariance matrices in the Riemannian manifold. Figure 3.1 shows some of experimental results. From the figure, it can be concluded that both distance metrics yield reasonable matching on pedestrian patches and are comparable.

Figure 3.2 shows the structure of the proposed approach. This section begins with a short explanation of Fisher linear discriminant analysis (LDA) and weighted LDA. Next, the technique used to train multi-dimensional covariance features on a cascade of AdaBoost classifiers is described. Finally, a new two-layer pedestrian detector, which utilizes the efficiency of Haar-like features and the discriminative power of covariance features, is introduced.

### 3.2.1 Weighted Fisher Linear Discriminant Analysis

Let us assume that we have a set of training patterns  $\mathbf{x} = [x_1, x_2, \dots, x_M]^\top$  where each of which is assigned to one of two classes,  $C_1$  and  $C_2$ . One can find a weight vector  $\mathbf{w} = [w_1, w_2, \dots, w_M]^\top$  and a threshold  $w_0$  such that,

$$\begin{aligned} \mathbf{w}^\top \mathbf{x} + w_0 &> 0 \quad (\mathbf{x} \in C_1), \\ \mathbf{w}^\top \mathbf{x} + w_0 &< 0 \quad (\mathbf{x} \in C_2). \end{aligned} \quad (3.1)$$

In general, one seeks the vector  $[w_0, w_1, w_2, \dots, w_M]$  that best satisfies (3.1). The data are said to be linearly separable if for all  $\mathbf{x}$ , (3.1) is satisfied.

The objective of the Fisher's criteria is to find a linear combination of variables that can separate two classes as much as possible. The computed linear combination reduces the number of our data dimensions to one dimension. The criterion proposed by Fisher is the ratio of between-class to within-class variances which can be written as,

$$\begin{aligned} J &= \frac{[\mathbf{w}^\top (\mathbf{m}_{C_1} - \mathbf{m}_{C_2})]^2}{\sum_{c \in \{C_1, C_2\}} \sum_{\mathbf{x} \in c} (\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \mathbf{m}_c)^2} \\ &= \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}, \end{aligned} \quad (3.2)$$

$$S_b = (\mathbf{m}_{C_1} - \mathbf{m}_{C_2})(\mathbf{m}_{C_1} - \mathbf{m}_{C_2})^\top, \quad (3.3)$$

$$S_w = \sum_{c \in \{C_1, C_2\}} \sum_{\mathbf{x} \in c} (\mathbf{x} - \mathbf{m}_c)(\mathbf{x} - \mathbf{m}_c)^\top. \quad (3.4)$$

Here  $\mathbf{m}_c$  is the mean of class  $c$ ,  $\bar{\mathbf{m}}$  is the global mean,  $S_b$  and  $S_w$  are the so-called between-class and within-class scatter matrices. The numerator of (3.2) denotes the distance between the projected means and the denominator denotes the variance of the pooled data. We want to find linear projections  $\mathbf{w}$  that maximizes  $J$ , the distance between the means of the two classes while minimizing the variance within each class. The solution can be obtained by the generalized eigen-decomposition. The optimal solution  $\mathbf{w}$  is the eigenvector corresponding to the maximal eigenvalue and can be expressed as [30],

$$\mathbf{w} \propto S_w^{-1}(\mathbf{m}_{C_2} - \mathbf{m}_{C_1}). \quad (3.5)$$

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---

However, the criterion proposed by Fisher assumes uniform weighted training samples. In AdaBoost training, each data point is associated with a weight which measures how difficult to correctly classify them. Therefore, we need to apply a weighted Fisher linear discriminant analysis (WLDA). Similar to LDA, WLDA finds a linear combination of the variables that can separate two classes as much as possible with emphasis on training samples with high weights.

Let us suppose that each sample is assigned with AdaBoost weights,  $s_i$ , where  $\sum_{i=1}^{N_s} s_i = 1$  and  $N_s$  is the total number of instances. Using the previous notation, the between-class variance and within-class variance of the weighted data can be written as,

$$S'_b \propto (m'_{C_1} - m'_{C_2}) (m'_{C_1} - m'_{C_2})^\top, \quad (3.6)$$

$$S'_w = \sum_{c \in \{C_1, C_2\}} \sum_{\mathbf{x}_i \in c} s_i (\mathbf{x}_i - \mathbf{m}'_c) (\mathbf{x}_i - \mathbf{m}'_c)^\top. \quad (3.7)$$

Here  $\mathbf{m}'_c$  is the weighted mean of class  $c$  which can be expressed as  $\frac{\sum_{\mathbf{x} \in c} s_i \mathbf{x}_i}{\sum_{\mathbf{x} \in c} s_i}$ . Hence, the optimal solution  $\mathbf{w}$  for WLDA can be expressed as,

$$\mathbf{w}' \propto S'^{-1}_w (\mathbf{m}'_{C_2} - \mathbf{m}'_{C_1}). \quad (3.8)$$

Each weak learner can then be defined as,

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}'^\top \mathbf{x} > w_0; \\ -1 & \text{otherwise,} \end{cases} \quad (3.9)$$

where  $h(\cdot)$  defines a weak learner,  $\mathbf{x}$  is a vector of calculated covariance features and  $w_0$  is an optimal threshold such that the minimum number of examples are misclassified.

Covariance features efficiently capture the relationship between different image statistics. Combining with WLDA, this information can be used to represent a distinct part of the human body. At each AdaBoost iteration, a simple classifier is trained from the collection of region covariance features. The experimental results show that the covariance region selected by AdaBoost are physically meaningful and can be easily interpreted as shown in Figure 3.3. The first selected feature focuses on the bottom part of the human body while the second selected feature focuses on the top part of the body. It turns out that covariance features are well adapted to capture patterns that are invariant to illumination changes and human poses/appearance changes.



**Figure 3.3:** The first and second covariance regions selected by AdaBoost. The first two covariance regions overlaid on human training samples are shown in the first column. The second column displays human body parts selected by AdaBoost. The first covariance feature represents human legs (two parallel vertical bars) while the second covariance feature captures the information of the head and the human body.

### 3.2.2 A Cascade of Covariance Descriptors

In order to reduce computation time during human detection phase, a cascade of classifiers is built [140]. The key insight is that efficient boosted classifiers, which can reject many of simple non-pedestrian patches while detecting almost all pedestrian patches, are constructed and placed at early stages of the cascade. Time-consuming and complex boosted classifiers, which can remove difficult non-pedestrian samples, are placed in later stages of the cascades. By constructing classifiers in this way, the system can quickly discard simple background regions of the image, *e.g.*, sky, building, road, *etc.* while spending more time on pedestrian-like regions. Only samples that can pass through all stages of the cascade are classified as pedestrians. The proposed boosted covariance features based detection framework is summarized in Algorithm 2.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---

**Algorithm 2** The training algorithm for building a cascade of boosted covariance detector

---

**Input:**

- $\{(\mathbf{x}_n, y_n)_{n=1}^{N_s}\}$ : Training set;
- $D_{\min}$ : minimum acceptable detection rate per cascade level;
- $F_{\max}$ : maximum acceptable false positive rate per cascade level;
- $F_{\text{target}}$ : target overall false positive rate.

```

1 Initialize:  $i = 0$ ;  $D_i = 1$ ;  $F_i = 1$ ;
2 while  $F_{\text{target}} < F_i$  do
3    $i = i + 1$ ;  $f_i = 1$ ;
4   while  $f_i > F_{\max}$  do
5     1. Normalize sample weights,  $s_i, i = 1, \dots, N_s$  such that  $\sum_{i=1}^{N_s} s_i = 1$ ;
6     2. Calculate projection vector,  $\mathbf{w}'$  (3.8), and project covariance features to 1D;
7     3. Train decision stumps using the training set (3.9);
8     4. Add the best decision stump classifier into the strong classifier;
9     5. Update sample weights,  $s_i, i = 1, \dots, N_s$ , in the AdaBoost manner;
10    6. Lower AdaBoost threshold such that  $D_{\min}$  holds;
11    7. Update  $f_i$  using this threshold.
12   $D_{i+1} = D_i \times D_{\min}$ ;  $F_{i+1} = F_i \times f_i$ ; and remove correctly classified negative samples
    from training sets;
13  if  $F_{\text{target}} < F_i$  then
14    Evaluate the current cascade classifier on negative images and add false positive
    samples to negative training sets.

```

**Output:** A cascade of boosted covariance classifiers for each cascade level  $i = 1, \dots$ .

---

### 3.3 Multiple-layer Boosting with Heterogeneous Features

---

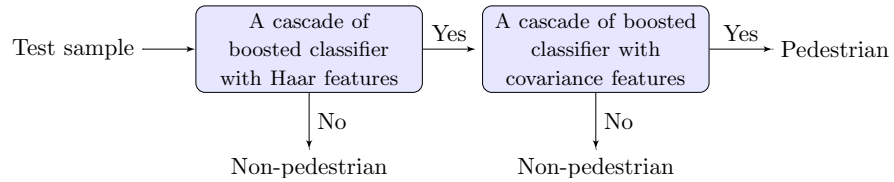
In order to further accelerate the proposed detector, an approach which consists of a multiple-layer cascade of classifiers is built [80]. The intuition is to achieve high detection speed while maintaining comparable accuracy. The idea is to place simple and fast-to-compute features in the first layer while putting a more accurate but slow-to-compute features in the second layer of the cascade. The simple feature filters out most simple non-pedestrian patterns in the early stage of the cascade.

Haar-like wavelet features have proved to be extremely fast and robust in the appli-



### 3.3 Multiple-layer Boosting with Heterogeneous Features

cation of face detections [140]. However, it performs poorly in the context of human detection as reported in Viola *et al.* [141]. In order to improve the overall accuracy, the boosted covariance detector is applied in the second layer. In other words, the first layer consists of a number of Haar-like based AdaBoost classifiers while the second layer consists of a number of Boosted covariance classifiers. This way, the efficiency of Haar-like features and the discriminative power of covariance features can be utilized in a single framework. Figure 3.4 illustrates an architecture of the two-layer approach.



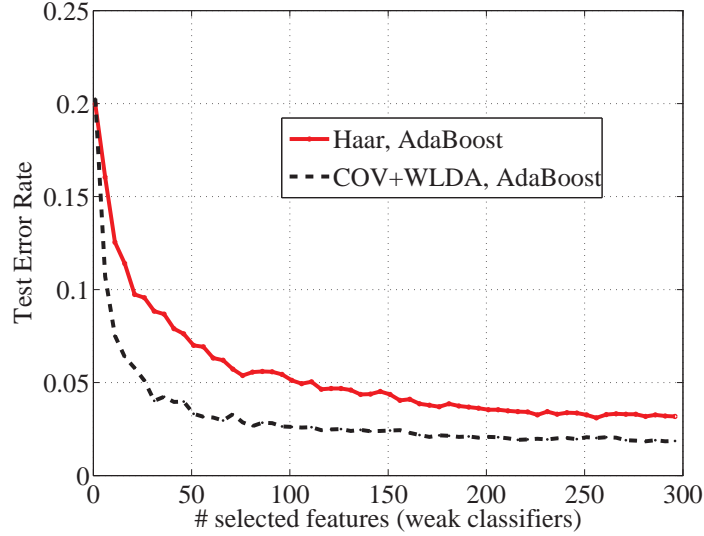
**Figure 3.4:** The structure of the proposed two-layer pedestrian detector.

Covariance and Haar-like features are experimentally evaluated on the same training set using AdaBoost. The positive training set is extracted from INRIA data sets [24], which consist of 2,416 human samples (mirrored). The negative training set comes from random patches extracted from negative images. The classifiers are evaluated on the INRIA test set. Figure 3.5 gives a comparison of performances of different feature types. The following observation can be made from the figure. The test error decreases quickly with the number of AdaBoost iterations for all features. The test error of covariance features run into saturation after about 100 iterations while the test error rate of Haar-like features continues to decrease slowly. The results can also be interpreted in terms of the number of selected features and test error rate. For example, it is possible to achieve a 5% test error rate using either 25 covariance features or 100 Haar-like features. Table 3.1 shows the computation time for different feature types (including computation overhead of integral images). The computation of Haar-like features are much faster than the computation of covariance features.

Due to the flexibility of the cascaded structure, it is easy to integrate multiple heterogeneous features. Although Haar-like and covariance features are used here, different combination of other features may lead to a better performance. It remains a future study topic on how to find the best combination.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---



**Figure 3.5:** A performance comparison between covariance and Haar-like features on INRIA test sets [24].

## 3.4 Experiments

---

The proposed approach is evaluated on two publicly available data sets:- Daimler-Chrysler pedestrian data sets [89] and INRIA human test sets [24]. The first data sets contain a set of extracted pedestrian and non-pedestrian samples, scaled to a resolution of  $18 \times 36$  pixels. Three experiments are conducted using covariance features trained with SVM and AdaBoost. The second data sets contains 1, 176 human samples from 288 images. Two experiments are conducted using covariance features trained

**Table 3.1:** An average time required to evaluate covariance and Haar-like features.

# features	COV ( $\mu$ -seconds)	Haar-like ( $\mu$ -seconds)
20	71	5
50	137	7
100	250	11
200	490	20
300	715	29

with AdaBoost. To our knowledge, Dalal and Triggs [24] and Tuzel *et al.* [136] are current state-of-the-art human detection approaches in literature. Hence, the proposed approach is compared with these two techniques.

The experimental section is organized as follows. First, the experimental setup and parameter values chosen are described. Experimental results are then presented with an in-depth analysis. In all experiments, associated parameters are optimized via cross-validation.

### 3.4.1 Experiments on Daimler-Chrysler Data Sets with Boosted Covariance Features

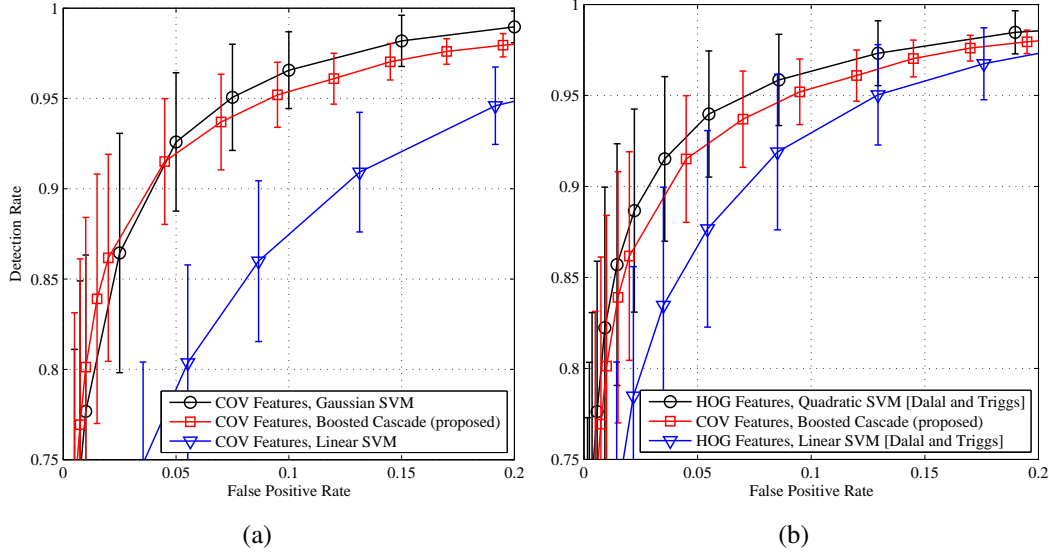
#### 3.4.1.1 Experiment Setup

For boosted cascade of covariance features, a full set of overcomplete rectangular covariance filters is generated and the overcomplete set is sub-sampled in order to keep a manageable set during the training phase. The set contains approximately 1,120 covariance filters. Each filter (weak classifier) consists of four parameters *e.g.*,  $x$ -coordinate,  $y$ -coordinate, width and height. A strong classifier consisting of several weak classifiers is built in each stage of the cascade. At each stage, weak classifiers are added until the predefined objective is met. In this experiment, the minimum detection rate is set to 99.5% and the maximum false positive rate is set to 50% in each stage. Negative samples used in each stage of the cascade are collected from false positives of previous stages of the cascade.

Since the resolution of the test samples is quite small, the border of each test sample is extended by one pixel. The extra margin helps shifting the pedestrian in the test sample to the center. Doing so increases a flexibility of boosted classifiers. During classification, the number of positively classified sub-windows is counted. The total number of sub-windows is then used to determine whether the test sample is pedestrian or non-pedestrian.

As the baseline in our comparisons, we train two state-of-the-art features, namely Histogram of Oriented Gradients (HOG) [24] and covariance features [136], with various SVM (Support Vector Machine) classifiers. The reasons these two features are selected along with SVM are: (1) It has been shown that these two local features are best

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES



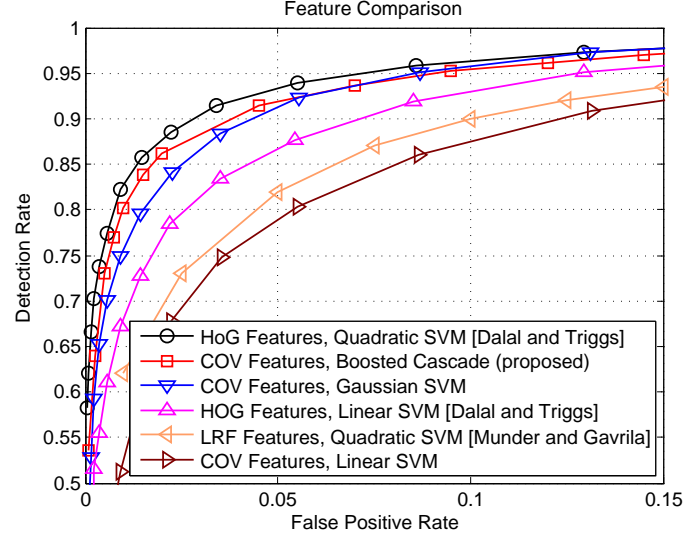
**Figure 3.6:** A performance comparison of the proposed cascade of boosted covariance features with (a) covariance features trained using SVM and (b) histogram of oriented gradients (HOG) features trained using SVM.

candidates for pedestrian detection tasks; (2) SVM is one of the most advanced classifiers. It is easy to train and, unlike neural networks, the global optimum is guaranteed. Hence the variance caused by suboptimal training is avoided for fair comparisons. In the following experiment, features are concatenated and train with SVM classifiers of two different functions: linear and RBF kernels (Gaussian).

#### 3.4.1.2 Results based on Boosted Covariance Features

Figure 3.6 shows detection results of proposed covariance features. The proposed approach performs best when compared with a linear SVM. When compared with a non-linear SVM, the proposed approach performs very similar to covariance features + Gaussian SVM and slightly worse than HOG features + Gaussian SVM. Note that the non-linear SVM has a much higher computation cost, during both training and evaluation, than the proposed approach.

It might not be fair to compare three detectors directly since the boosted cascade is trained with more non-pedestrian samples, *i.e.*, by making use of cascade structure,



**Figure 3.7:** A performance comparison of different feature types on Daimler-Chrysler data sets.

the negative training size has been manually increased. In order to compare the performance of three detectors, a bootstrapping technique is applied to HOG [24] and covariance features. Bootstrapping is applied iteratively, generating 10,000 new non-pedestrian samples at each iteration. It was observed that collecting the first 10,000 new non-pedestrian samples did not take long but the second iteration took a long time. This is what one would expect since the new classifier has better accuracy than the previous classifier. Hence, it takes longer time to collect more new false positive patches. We observe that the improvement of training HOG feature using bootstrapping technique over initial classifier is up to 7% increase in detection rate at 2.5% false positives rate while the improvement is slightly lower in covariance features (about 3% increases at 2.5% false positives rate). However, this performance gain comes at a higher computation cost for training.

The best performing result of different feature types are compared in Figure 3.7. The following observations can be made. Out of three features, both HOG and covariance features perform much better than LRF. One can observe that gradient information is very helpful in human detection tasks. In all experiments, nonlinear SVMs (quadratic or Gaussian RBF SVM) improve performance significantly over linear ones.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---



**Figure 3.8:** Examples of mistakes made by our boosted covariance detector on the Daimler-Chrysler data sets. The first row shows false negative examples and the last row shows false positive examples.

However, this comes at the cost of a much higher computation time, *i.e.*, in this experiment, building a non-linear SVM model is approximately 50 times slower than building a linear SVM model.

Figure 3.8 presents a qualitative assessment of the errors made by our detector, showing some false negative (non-pedestrian-like pedestrians) and false positive (pedestrian-like non-pedestrian) examples from our detectors point of view. It can be seen that most of false negatives are due to the subject's pose deformation, occlusions, or the very difficult illumination environments. False positives usually contain gradient information which looks like human body boundaries. It is interesting to see that many false positives are road signs. In contrast, most false negatives are due to the subject's pose deformation, occlusions, or the very difficult illumination environment. False positives usually contain gradient information which looks like human body boundaries.

The advantages of the proposed method over features trained using SVM are ease of parameter tuning and much faster detection speed. SVM has more parameters com-

**Table 3.2:** An average time required to evaluate 10 frames of half resolution videos ( $384 \times 288$  pixels) of various approaches. Each image consists of 17,280 windows (scale factor of 0.8 and step-size of 4 pixels).

	windows per sec	seconds per frame
HOG, Quadratic SVM	25	714
HOG, Linear SVM	4800	3.6
Proposed COV approach	6000	2.9

pared to the boosted cascade, *e.g.*, trade-off between training error and margin, parameters of the nonlinear kernel, *etc.* These parameters need to be manually optimized for the specific classification task using cross-validation.

In the next experiment, the processing speed of two best classifiers, HOG with quadratic SVM and 20 stages of boosted covariance features, is compared. Both classifiers are evaluated on a sequence of 10 images with a resolution of  $384 \times 288$  pixels in width and height. Table 3.2 shows the average detection speed for both classifiers. As expected, the detection speed of boosted covariance features is much faster than the detection speed of the non-linear SVM classifier.

### 3.4.2 Experiments on Daimler-Chrysler data sets with two-layer boosting

#### 3.4.2.1 Experiment setup

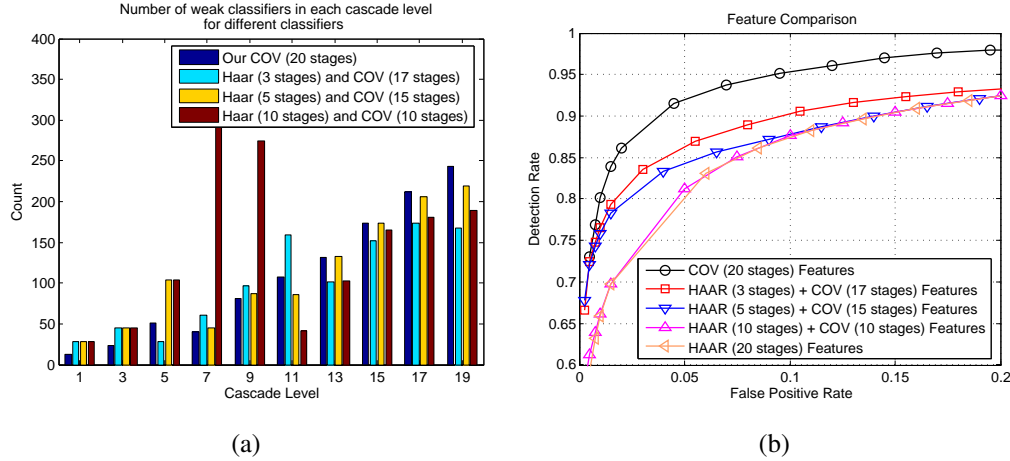
A set of overcomplete Haar-like wavelet filters is generated and the overcomplete set is sub-sampled. The set of Haar-like features that is used to train the cascade contained 20,547 filters: 5,540 vertical two-rectangle features, 5,395 horizontal two-rectangle features, 3,592 vertical three-rectangle features, 3,396 horizontal three-rectangle features and 2,624 four-rectangle features. From the preliminary experiments on signed and unsigned wavelets, it was observed that the performance of signed wavelets outperform unsigned wavelets. Hence, the sign of intensity gradients is preserved in this experiment. For covariance features, a set of rectangular covariance features from previous section is used.

In this experiment, the minimum detection rate and maximum false positive rate for both cascade (a layer of boosted Haar-like features and a layer of boosted covariance features) is set to 99.5% and 50%. Figure 3.9(a) gives some details about the proposed two-layer boosting cascade.

#### 3.4.2.2 Results based on Multi-layer Boosting

Table 3.3 shows the evaluation time in windows per second for different hybrid configurations. Adding more stages of Haar-like wavelet features as a preprocessing step increases the detection speed approximately *exponentially*. Figure 3.9(b) shows the

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES



**Figure 3.9:** (a) A number of weak classifiers in different cascade levels on Daimler-Chrysler data sets. Note that adding Haar-like features as a preprocessing step hardly affect the number of covariance features in later stages. (b) A performance comparison of the two-layer boosting approach and a cascade of boosted covariance features on Daimler-Chrysler data sets. The two-layer boosting approach performs comparable to cascade of boosted covariance features at low false positive rate ( $< 0.01$ ), which is the range of interest.

performance of two-layer boostings. The curve of the proposed method is generated by adding one cascade level at a time. Boosted covariance features outperform all other approaches. The performance of hybrid classifiers is quite poor at high false positive rate due to haar-like features in the initial stages of the cascade. Nonetheless, the performance improves as more covariance features have been added to later stages of cascades.

#### 3.4.3 Experiments on INRIA Human Data Sets with Boosted Covariance Features

INRIA data sets consist of one training set and one test set. The training set contains 1, 208 pedestrian samples (2, 416 mirrored samples) and 1, 200 non-pedestrian images. The pedestrian samples were obtained from manually labeling images taken from a digital camera at various time of days and locations. The pedestrian samples are mostly



in standing position. A border of 8 pixels is added to the sample in order to preserve contour information. All samples are scaled to size  $64 \times 128$  pixels. The test set contains 1,176 pedestrian samples (mirrored) extracted from 288 images.

The proposed approach is evaluated on given test sets using both classification and detection methods. For human classification, cropped human samples taken from test images are used. During classification, the number of positively classified windows is used to determine if the test sample is human or non-human. For human detection, a fixed size window is used to scan test images with a scale factor of 0.95 and a step size of 4 pixels. As in Tuzel *et al.* [136], mean shift clustering [22] is used to cluster multiple overlapping detection windows. Simple rules as in Viola and Jones [140] are also applied on the clustering results to merge those close detection windows.

The criteria similar to the one used in PASCAL VOC Challenge [142] is adopted here. Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the area of overlap between the predicted bounding box,  $B_p$ , and ground truth bounding box,  $B_{gt}$ , must exceed 40% by,

$$a_0 = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} > 40\%.$$

Multiple detections of the same object in an image are considered false detections. For quantitative analysis, miss rate versus false positive rate curves are plotted on a log-log scale. The experiments are conducted using a standard desktop with 2.8 GHz Intel Pentium-D CPU and 2 GB RAM.

**Table 3.3:** An average evaluation time in windows per second for different parameters of two-layer boosting approaches.

	windows per sec
Proposed COV (20 stages)	6,000
Haar-like (3 stages) and COV (17 stages)	30,000
Haar-like (5 stages) and COV (15 stages)	50,000
Haar-like (10 stages) and COV (10 stages)	100,000
Haar-like (20 stages)	200,000

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

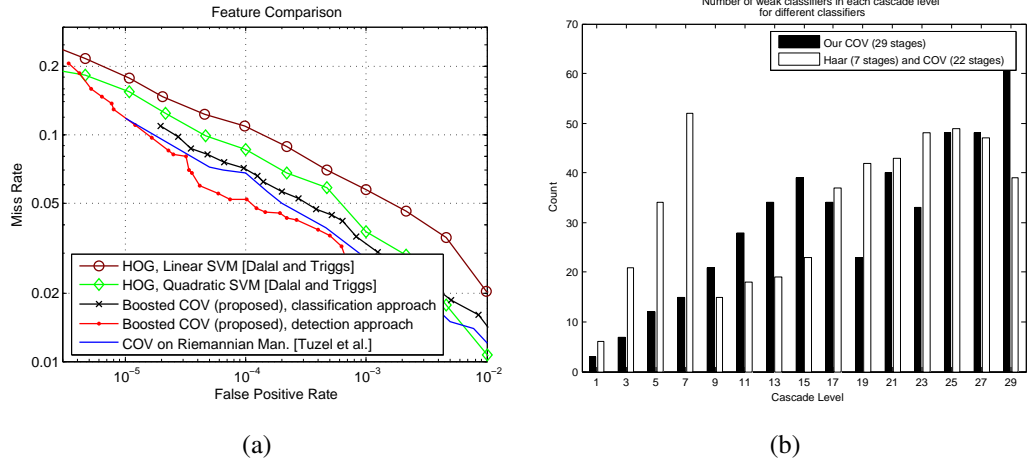
---

#### 3.4.3.1 Experiment Setup

Similar to previous experiments, a set of overcomplete rectangular covariance filters is generated and the overcomplete set is subsampled. The set contains approximately 15, 225 covariance filters. In each stage, weak classifiers are added until the predefined objective is met. In this experiment, for each stage, the minimum detection rate is set to 99.5% and the maximum false positive rate is set to 50%. Each stage is trained with 2, 416 human samples and 5, 000 non-human samples. Negative samples used in each stage are collected from false positives of previous stages of cascades. The final cascade consists of 29 stages.

#### 3.4.3.2 Results based on Boosted Covariance Features

Figure 3.10(a) shows a comparison of proposed approach with different algorithms. The curve of the proposed approach is generated by adding one cascade level at a time. Based on the figure, the proposed system's performance is much better than HOG with linear SVM [24] while achieving a comparable detection rate to the technique described in Tuzel *et al.* [136]. Tuzel *et al.* calculates distance between covariance matrix on the Riemannian manifold. An eigen-decomposition is required which slows down the computation speed. In contrast, the proposed approach avoids the eigen-decomposition and therefore it is much faster. It is also easier to implement. The figure also shows the performance of the proposed system on human detection problem. In order to achieve the results at low false positive rate, *i.e.*,  $< 10^{-5}$ , the minimum neighbor threshold (a number of merged detections) is manually adjusted. From Figure 3.10(a), covariance technique with detection approach outperforms the same technique with classification approach. The reason is due to the clustering and merging techniques used. By clustering and merging multiple overlapping detection windows, one is able to further reduce the number of false detections. As a result, the curve is slightly shifted to the left. As for the processing time, an unoptimized implementation of the proposed approach in C++ can search about 12, 000 detection windows per second. Due to the cascade structure, the search time is faster when human is against plain backgrounds and slower when human is against more complex backgrounds. Table 3.4 shows the average detection speed for three different classifiers. Compared to Dalal and Triggs [24] and Tuzel *et al.* [136], the proposed approach



**Figure 3.10:** (a) A performance comparison between boosted covariance features, HOG + linear SVM [24] and covariance on Riemannian manifold [136]. (b) The number of weak classifiers in different cascade levels on INRIA test data sets [24].

has a smaller evaluation time than both techniques (2.2 times faster than [24]<sup>1</sup> and 4 times faster than [136]<sup>2</sup>). Note that the system in [136] is implemented in C++ on a Pentium-D 2.8 GHz processor with 2GB RAM, which is the same as the system used in this experiment.

The next experiment demonstrates how adding a cascade of Haar-like wavelet fea-

<sup>1</sup>Based on the experiment.

<sup>2</sup>Personal communication with the authors of [136].

**Table 3.4:** An average evaluation time on  $240 \times 320$  pixels images (12,800 windows per image) of different approaches.

	windows per sec
HOG, Quadratic SVM [24]	60
COV, Riemannian Manifold [136]	3,000
HOG, Linear SVM [24]	5,500
Proposed COV approach	12,000

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---

tures to a cascade of boosted covariance features could help improve the detection speed while maintaining a high detection rate.

#### 3.4.4 Experiments on INRIA Human Data Sets with Two-layer Boosting

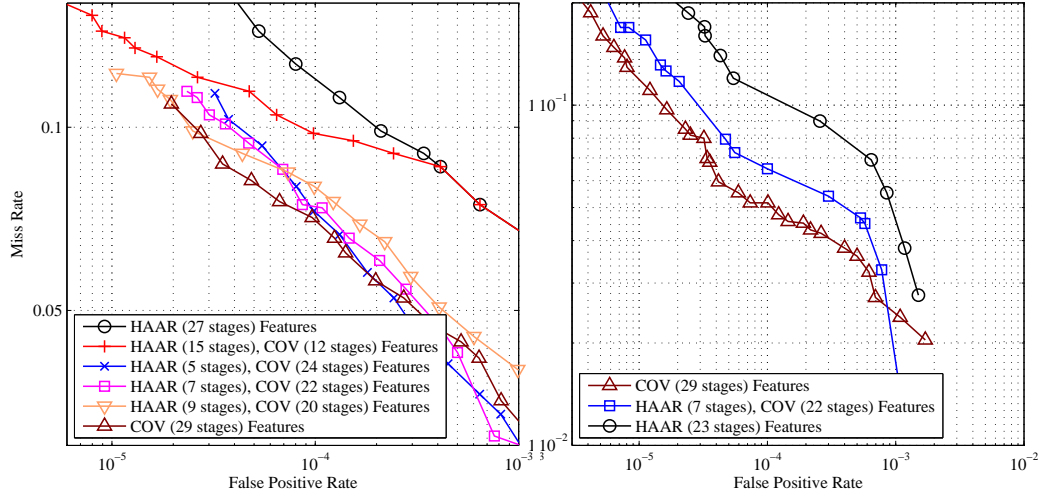
##### 3.4.4.1 Experiment Setup

Similar to experiments on data sets of [89], we subsample the overcomplete set of Haar-like features to 54,779 filters: 11,446 vertical two-rectangle features, 14,094 horizontal two-rectangle features, 8,088 vertical three-rectangle features, 10,400 horizontal three-rectangle features and 10,751 four-rectangle features. Unlike in previous experiment, the performance of unsigned wavelets seems to outperform the performance of signed wavelets. We think that when the human resolution is large, clothing and background details can be easily observed and intensity gradient sign becomes irrelevant. In other words, a wide range of clothing and background colors make the gradient sign uninformative, *e.g.*, a person with a black shirt in front of a white background should have the same information as a person with a white shirt in front of a dark background. Hence, absolute values of the wavelet responses are used in this experiment. For covariance features, a set of rectangular covariance features from previous section is used. Figure 3.10(b) gives some details about the two-layer boosting cascade.

##### 3.4.4.2 Results based on Multi-layer Boosting

The evaluation time in windows per second for different hybrid configurations is shown in Table 3.5. Similar to previous results, adding Haar-like wavelet features as a pre-processing step increases the detection speed significantly. Compared with the original covariance detector in [136], the two-layer boosting approach is 10 times faster (Table 3.5).

Figure 3.11 shows the performance of two-layer boosting approach using the *classification* and *detection* approaches. For the classification approach, the overall performance of different hybrid configurations is very similar to the performance of a cascade of boosted covariance features. A hybrid classifier with 15 levels of Haar-like features



**Figure 3.11:** A performance comparison between different configurations of the two-layer boosting approach based on (*left*) **classification** and (*right*) **detection** on INRIA test sets. Overlapping amongst ROC curves indicates the performance similarity.

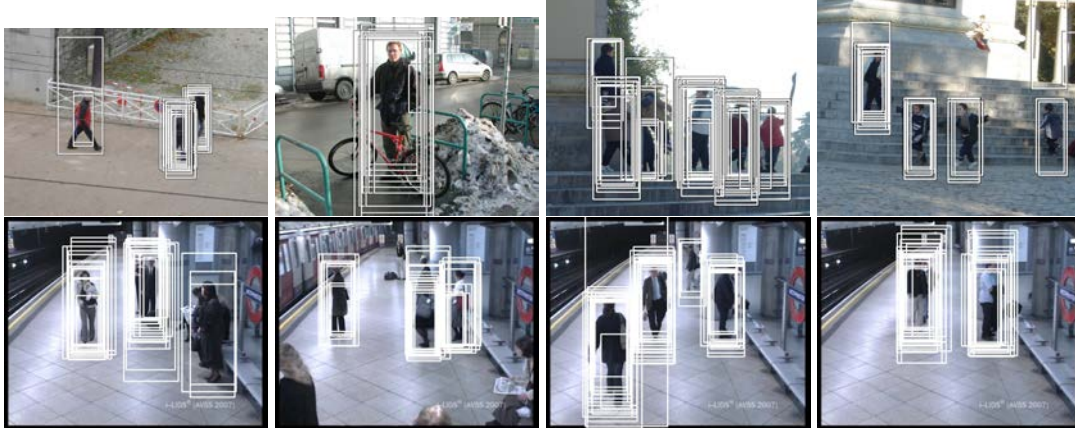
and 12 levels of covariance features might seem to perform poorly at high false positive rate. However, at a low false positive rate, *i.e.*,  $2 \times 10^{-5}$ , both approaches perform similarly. For the detection approach, the two-layer boosting approach performs slightly inferior to the cascade of boosted covariance features. This is not surprising since INRIA human data sets contain humans with various poses which Haar-like features are less capable to discriminate. Nonetheless, applying boosted covariance features in the second layer improves the overall accuracy of Haar-like features significantly. Figure 3.12 demonstrates some detection examples using the proposed hybrid detector on INRIA test data set and Advanced Video and Signal based Surveillance (AVSS) 2007 data sets <sup>1</sup>.

The two-layer boosting approach and HOG features are also compared on INRIA data sets with  $18 \times 36$  pixels training data (instead of  $64 \times 128$  used in previous experiments). The experimental setup used in this experiment is similar to the one used in previous experiments (Sections 3.4.1 and 3.4.2). Figure 3.13 shows experimental results of different approaches. Results obtained are slightly different from results in Section 3.4.2 due to the different resolution used. However, the overall results seem to

<sup>1</sup>[http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007\\_d.html](http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html)

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---



**Figure 3.12:** Human detection examples. Boxes show detection results of the proposed hybrid classifier (9 levels of Haar-like features and 22 levels of covariance features). *Top:* INRIA test sets. *Bottom:* AVSS 2007 data sets. Note that no post-processing has been applied to detection results.

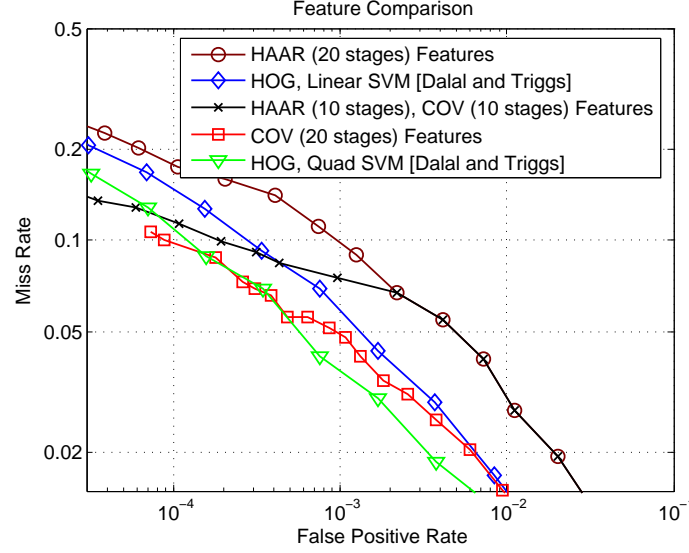
be consistent with results shown in Figures 3.7 and 3.9(b).

#### 3.4.5 Detection Performance versus Speed Trade-off for the Two-layer Boosting

From previous experiments, results show that the speed of Haar-like features based classifiers is much faster than the speed of covariance features based classifier. Therefore, it's best to place as many stages of Haar-like features in the first layer of the

**Table 3.5:** An average evaluation time (in windows per second) for different parameters of two-layer boosting approaches.

	windows per sec
Proposed COV (29 stages)	12,000
Haar-like (7 stages) and COV (22 stages)	35,000
Haar-like (9 stages) and COV (20 stages)	40,000
Haar-like (15 stages) and COV (12 stages)	52,000
Haar-like (27 stages)	200,000



**Figure 3.13:** A performance comparison on human detection between the proposed approach (Haar-like features plus covariance features) and HOG features on INRIA test sets with training data of resolution  $18 \times 36$  pixels.

classifier. However, having too many stages of Haar-like features will degrade the overall performance. The objective of this section is to find the best combination that will give best overall results.

To study the trade-off between the detection performance and detection speed, a test is performed on different false positive rates. For example, to achieve a  $5 \times 10^{-4}$  false positive rate for boosted covariance classifier on INRIA data sets, only first 19 stages of covariance features (instead of all 29 stages) are used. The average computation time is calculated by evaluating the 19 stages classifier on a test sequence of images. Figure 3.14 shows the detection rate and computation time for different configurations of multiple-layer boosting on the data sets of and INRIA data sets. From the figure, it can be concluded that there is a trade-off between the detection performance and speed. In order to achieve a high detection rate, only a small number of Haar-like based AdaBoost classifiers should be placed in the first layer of cascades. For small resolution data sets ( $18 \times 36$  pixels), a configuration of 5 stages Haar-like/15 stages covariance cascade classifier seems to perform best at a reasonable computation time. For larger resolution data sets ( $64 \times 128$  pixels), a configuration of 7 stages

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---

Haar-like/22 stages covariance cascade classifier seems to perform best.

#### 3.4.6 Discussion

This section tests previously trained classifier (classifier trained on INRIA data sets) on random internet images with pedestrians having variable illumination, appearance, pose and occlusion. Some results are shown in Figure 3.15. The top row shows raw detection results. The bottom row shows merged detection results using mean shift clustering. Based on experimental results, the system works well on images where there is a small gap between pedestrians, *i.e.*, no occlusion between pedestrians. When humans stand in a group or occlude one another, the human contour is quite complex and different from what the classifier was trained. In addition, there exist a lot of multiple overlapping detection windows when human occludes one another. Mean shift clustering fails to merge the detection windows correctly when there is a lot of overlapping windows. As a result, the system fails to detect most of pedestrians. Note that a lot of false detections came from various human body parts, *e.g.*, human limbs and body. This is not surprising since negative training samples used do not contain any of these body parts.

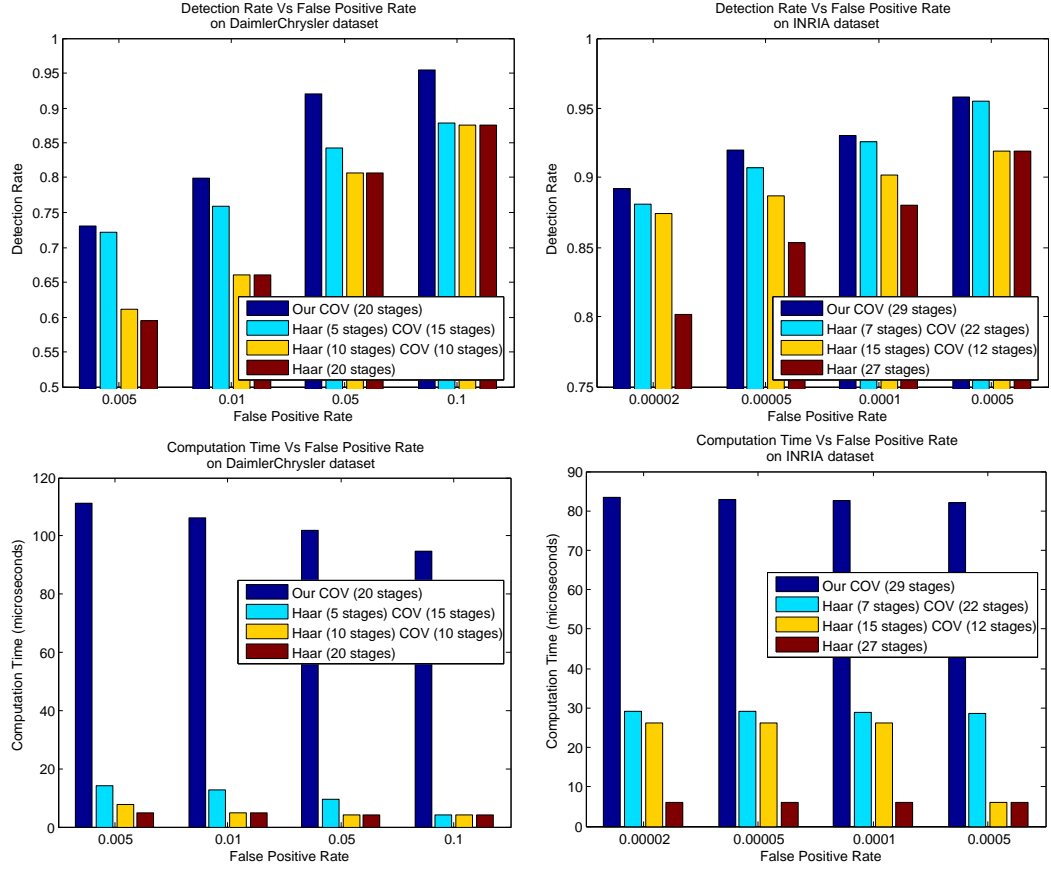
### 3.5 Conclusion

---

This chapter has presented a fast and robust pedestrian detection technique. We use weighted Fisher linear discriminant analysis as the weak classifier for AdaBoost training. In order to speed up the computation time, a cascaded classifier architecture is adopted [140]. From the experimental results on Daimler-Chrysler data sets [89], the proposed system has shown to give high detection performance at a low false positive rate. Comparing with techniques using linear SVM classifier, the proposed system outperforms all systems evaluated. When compared with non-linear SVM systems, the system is shown to perform very similar to covariance features with Gaussian SVM and slightly inferior compared to HOG with quadratic SVM. Nonetheless, the computation time of HOG with quadratic SVM is much higher than the proposed approach.

The performance of the proposed approach is also evaluated on INRIA human data sets [24]. The performance of the proposed approach is comparable to the state-of-

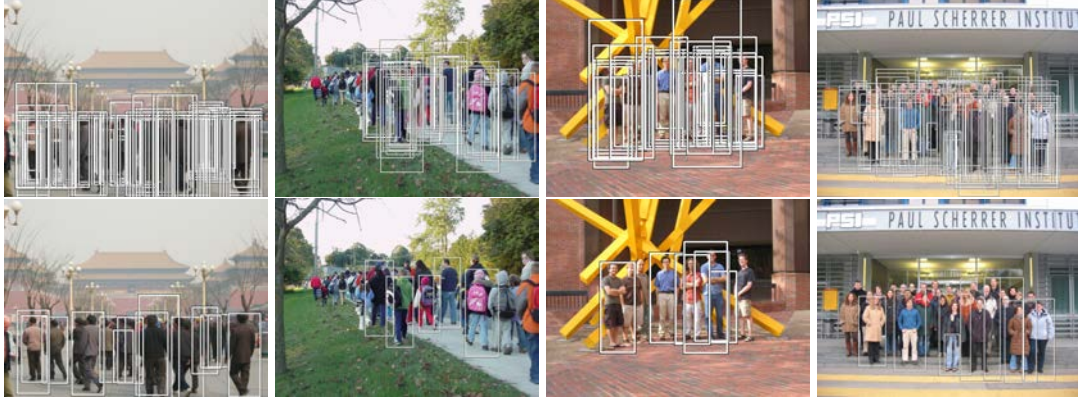




**Figure 3.14:** Detection rate versus speed trade-off for different configurations of two-layer boosting. First two figures show Detection Rate vs. False Positive Rate on Daimler-Chrysler data sets and INRIA data sets, respectively. The last two figures show Computation Time vs. False Positive Rate. Clearly, covariance features have the highest detection rate across all false positive rates while Haar-like features have the lowest detection rate. On the other hand, Haar-like features are the fastest to compute while covariance features are the slowest.

### 3. FAST PEDESTRIAN DETECTION USING BOOSTED COVARIANCE FEATURES

---



**Figure 3.15:** Detection examples on images collected randomly from the internet. *Top:* raw detection results. *Bottom:* merged detection results using a mean shift clustering technique.

the-art [136] while is almost 4 times faster during evaluation due to its new design. To further accelerate the detection speed, a faster strategy—two-layer boosting with heterogeneous features—is introduced. The approach exploits the efficiency of Haar-like features and the discriminative power of covariance features. This way the proposed detector runs 10 times faster than the original covariance feature detector [136]<sup>1</sup>.

One major drawback of Boosting based classifiers is that the algorithm ignores the imbalanced property of training data, *i.e.*, a typical natural image often contains many more negative background patterns than object patterns. In the next chapter, a new learning criterion, which considers this highly skewed data distribution, is introduced. Experiments demonstrate that classifiers trained with the new criterion outperforms AdaBoost and its variants. This finding provides a significant opportunity to argue that AdaBoost is not the only method that can achieve high classification results for high dimensional data in object detection.

---

<sup>1</sup> Note that this speedup factor would have been lower if the two-layer approach is also applied to [136].

# 4

## Efficiently Training a better Visual Detector with Sparse Eigenvectors

### 4.1 Introduction

---

Real-time object detection, objection detection such as face detection, has numerous computer vision applications, *e.g.*, intelligent video surveillance, vision based teleconference systems and human motion analysis [91, 92]. Various detectors have been proposed in the literature [100, 117, 121, 126, 140]. Object detection is challenging due to large variations of the visual appearances, poses and illumination conditions. Furthermore, object detection is a *highly-imbalanced* classification task. A typical natural image contains many more negative background patterns than object patterns. The number of background patterns can be 100,000 times larger than the number of object patterns. That means, if one wants to achieve a high detection rate, together with a low false detection rate, one needs to design a specific and sensitive classifier that takes the imbalanced data distribution into consideration [139].

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---



**Figure 4.1:** An illustration of cascade classifiers. Here a circle represents a node classifier. An input patch is classified as a target only when it passes tests at all node classifiers.

Viola and Jones [140] proposed the first real-time AdaBoost based face detector. They introduced a framework for selecting discriminative features and training classifiers in a cascaded manner as shown in Figure 4.1. The cascade framework allows most non-face patches to be rejected quickly before reaching the final node, resulting in fast performances. A test image patch is reported as a face only if it passes tests in all nodes. This way, most non-face patches are rejected by these early nodes. Cascade detectors have led to very fast detection speed and high detection rates. Due to their tremendous success, numerous further work have been proposed. Most of them focused on improving the underlying boosting method or accelerating the training process. For example, AsymBoost was introduced in Viola and Jones [139] to alleviate the limitation of AdaBoost in the context of highly skewed example distribution. Li *et al.* [67] proposed FloatBoost for a better detection accuracy by introducing a backward feature elimination step into the AdaBoost training procedure. Wu *et al.* [145] used forward feature selection for fast training by ignoring the re-weighting scheme in AdaBoost. Another technique based on the statistics of the weighted input data was used in Pham and Cham [109] for even faster training. KLBoost was proposed in Liu and Shum [70] to train a strong classifier. The weak classifiers of KLBoost are based on histogram divergence of linear features. Notice that in KLBoost, the classifier design is separated from feature selection process. Bourdev and Brandt [12] developed the Soft Cascade to reduce the complexity of cascade design and training. The idea was further improved in multi exit boosted classifiers [111]. Cascade classifiers were applied not only to boosting based classifiers, but also to Support Vector Machines (SVMs) [117]. In this chapter, an improved learning algorithm for object detection, known as Boosted Greedy Sparse Linear Discriminant Analysis (BGSLDA), is proposed.

One issue that contributes to the efficacy of the system comes from the use of Ad-

aBoost for training cascade nodes (Chapter 2). Chapter 2 introduces the AdaBoost classifier. AdaBoost combines an ensemble of weak classifiers to produce a final strong classifier with high classification accuracy. AdaBoost chooses a small subset of weak classifiers and assigns them with proper coefficients. The linear combination of weak classifiers can be interpreted as a decision hyper-plane in the weak classifier space. The proposed BGSLDA differs from the original AdaBoost in the following aspects. Instead of selecting decision stumps with minimal weighted error as in AdaBoost, the proposed BGSLDA algorithm finds a new weak learner that maximizes the class-separability criterion. As a result, the coefficients of selected weak classifiers are updated repetitively during the learning process according to this criterion.

The proposed technique differs from Wu *et al.* [145] in the following aspects. Wu *et al.* proposed the concept of Linear Asymmetric Classifier (LAC) by addressing the asymmetries and *asymmetric node learning* goal in the cascade framework. Unlike the proposed work where features are selected based on the Linear Discriminant Analysis (LDA) criterion, Wu *et al.* selects features using AdaBoost/AsymBoost algorithms. Given the selected features, Wu *et al.* then build an optimal linear classifier for the node learning goal using LAC or LDA. Note that similar techniques have also been applied in neural network. In Webb and Lowe [143], a nonlinear adaptive feed-forward layered network with linear output units has been introduced. The input data is nonlinearly transformed into a space in which classes can be separated more easily. Since LDA considers the number of training samples of each class, applying LDA at the output of neural network hidden units has been shown to increase the classification accuracy of two-class problem with unequal class membership. As experimental results show, in terms of feature selection, the proposed BGSLDA method is better than AdaBoost and AsymBoost for object detection.

Viola and Jones pointed out the limitation of AdaBoost in the context of highly skewed example distribution [139]. Since AdaBoost minimizes weighted exponential loss function, it does not minimize the number of false negatives. As a result, the selected features are no longer optimal for the task of rejecting negative examples. The authors proposed a new variant of AdaBoost called AsymBoost which is experimentally shown to give a significant performance improvement over conventional boosting. In brief, the sample weights were updated before each round of boosting with the ex-

## 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

tra exponential term which causes the algorithm to gradually pay more attention to positive samples in each round of boosting.

The key contributions of this chapter are as follows.

- Firstly, GSLDA is introduced as an alternative approach for training face detectors. Similar results are obtained compared with Viola and Jones' approach.
- Secondly, a new algorithm, known as BGSLDA, is proposed. The approach combines the sample re-weighting schemes typically used in boosting into GSLDA. Experiments show that BGSLDA can achieve better detection performances.
- Thirdly, it is shown that feature selection and classifier training techniques can have different objective functions (in other words, the two processes can be separated) in the context of training a visual detector. This offers more flexibility and even better performance. Note that previous boosting based approaches select features and train a classifier simultaneously.
- Finally, experimental results confirm that it is beneficial to consider the highly skewed data distribution when training a detector. LDA's learning criterion has already incorporated this imbalanced data information. Hence it is better than standard AdaBoost's exponential loss for training an object detector.

## 4.2 Algorithms

---

In this section, an alternative approach to AdaBoost for object detection is presented. The section begins with a brief explanation of the concept of GSLDA [86]. It then compares LDA and AdaBoost on asymmetric toy data sets. Next, a new algorithm, that makes use of sample re-weighting scheme commonly used in AdaBoost to select a subset of relevant features for training the GSLDA classifier, is proposed. Finally, the training time complexity of proposed methods is analyzed.

### 4.2.1 Greedy Sparse Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) can be cast as a generalized eigenvalue decomposition. Given a pair of symmetric matrices corresponding to the between-class ( $S_b$ )

and within-class covariance matrices ( $S_w$ ), one maximizes a class-separability criterion defined by the generalized Rayleigh quotient:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}. \quad (4.1)$$

The optimal solution of a generalized Rayleigh quotient is the eigenvector corresponding to the maximal eigenvalue. The sparse version of LDA is to solve (4.1) with an additional sparsity constraint:

$$\text{Card}(\mathbf{w}) = k, \quad (4.2)$$

where  $\text{Card}(\cdot)$  counts the number of nonzero components, also known as the  $\ell_0$  norm.  $k \in \mathbb{Z}$  is an integer set by a user. Due to this sparsity constraint, solving (4.1) with an additional constraint of (4.2) becomes non-convex and NP-hard. Moghaddam *et al.* presented a technique to compute optimal sparse linear discriminants using branch and bound approach [86]. Nevertheless, finding the exact global optimal solution for high dimensional data is infeasible. The algorithm was later improved by the same authors with new sparsity bounds and efficient matrix inverse technique. Their new algorithm shows 1,000 fold speedup relative to branch and bound approach [87]. The technique works by sequentially adding the new variable which yields the maximum eigenvalue (greedy forward selection) until a maximum number of elements are selected or some predefined condition is met. As shown in Moghaddam *et al.* [87], for two-class problem, the computation can be made very efficient as the only finite eigenvalue  $\lambda_{\max}(S_b, S_w)$  can be computed in closed-form<sup>1</sup> as  $\mathbf{b}^\top S_w^{-1} \mathbf{b}$  with  $S_b = \mathbf{b} \mathbf{b}^\top$  because in this case  $S_b$  is a rank-one matrix and  $\mathbf{b}$  is a column vector. Therefore, the computation is mainly determined by the inverse of  $S_w$ . When a greedy approach is adopted to sequentially find the suboptimal  $\mathbf{w}$ , a simple rank-one update for computing  $S_w^{-1}$  significantly reduces the computation complexity [87]. In this thesis, forward greedy search is applied due to its simplicity. For forward greedy search, if  $l$  is the current subset of  $k$  indices and  $m = l \cup i$  for candidate  $i$  which is not in  $l$ . The new augmented inverse  $(S_w^m)^{-1}$  can be calculated in a fast way by recycling the last step's result  $(S_w^l)^{-1}$ :

$$(S_w^m)^{-1} = \begin{bmatrix} (S_w^l)^{-1} + a_i \mathbf{u}_i \mathbf{u}_i^\top & -a_i \mathbf{u}_i \\ -a_i \mathbf{u}_i & a_i \end{bmatrix}, \quad (4.3)$$

---

<sup>1</sup>Note that the optimal solution to (4.1) can be computed in closed-form.

## 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

where  $\mathbf{u}_i = (S_w^l)^{-1} S_{w,li}$  with  $(li)$  indexing the  $l$ -th row and  $i$ -th column of  $S_w$  and  $a_i = 1/(S_{w,ii} - S_{w,li}^\top \mathbf{u}_i)$  [43, 87].

Note that other sparse linear regression and classification algorithms, *e.g.*,  $\ell_1$ -norm linear support vector machines,  $\ell_1$ -norm regularized log-linear models, *etc.*, have also been experimented. However, the major drawback of these techniques is that they do not have an *explicit* parameter that controls the number of features to be selected. The trade-off parameter (regularization parameter) only controls the degree of sparseness. One has to tune this parameter using cross-validation. Also  $\ell_1$  penalty methods often lead to sub-optimal sparsity [152]. Hence, GSLDA, which makes use of greedy feature selection and the number of features can be predefined, is applied. It would be of interest to compare the proposed method with  $\ell_1$ -norm induced sparse models [28].

The following paragraph explains how the GSLDA classifier is applied [87] as an alternative feature selection method to classical Viola and Jones' framework [140]. Here, an explanation of cascade classifiers is omitted. Interested readers should refer to Chapter 2 for details. The proposed GSLDA based detection framework can be summarized in Algorithm 3. The algorithm operates as follows. The set of selected features is initialized to an empty set. The first step (lines 4 – 5 in Algorithm 3) is to train weak classifiers, for example, decision stumps on Haar features.<sup>1</sup> For each Haar-like rectangle feature, the threshold that gives the minimal classification error is pre-computed and stored in memory. In order to achieve maximum class separation, the output of each decision stump is examined and the decision stump whose output yields the maximum eigenvalue is sequentially added to the list (line 7, step (1)). The process continues until the predefined condition is met (line 6).

### 4.2.2 Linear Discriminant Analysis on Asymmetric Data

In cascade classifiers, one would prefer to have a classifier that yields high detection rates without introducing many false positives. In the Bayes sense, linear discriminant classifiers are optimum for normal distributions with equal covariance matrices. However, due to its simplicity and robustness, linear discriminant classifiers have shown to

---

<sup>1</sup> Note that any weak classifiers can be applied here. For the time being, decision stumps on Haar-like features are shown as examples. Details of other weak classifiers, *e.g.*, covariance features, will be shown later.



---

**Algorithm 3** The training procedure for building a cascade of GSLDA object detector.

---

**Input:**

- A positive training set and a negative training set;
- A set of Haar-like rectangle features;
- $DR_{\min}$ : minimum acceptable detection rate per cascade level;
- $FP_{\max}$ : maximum acceptable false positive rate per cascade level;
- $FP_{\text{target}}$ : target overall false positive rate;

```

1 Initialize:  $i = 0$ ;  $DR_i = 1$ ;  $FP_i = 1$ ;
2 while  $FP_{\text{target}} < FP_i$  do
3    $i = i + 1$ ;  $f_i = 1$ ;
4   foreach rectangle feature do
5     Train a weak classifier,  $h_1, h_2, \dots$ , with the smallest error on the training set;
6   while  $f_i > FP_{\max}$  do
7     1. Add the best weak classifier that yields the maximum class separation;
8     2. Lower classifier threshold,  $\theta$  in (4.4), such that  $DR_{\min}$  holds;
9     3. Update classifier's false positive rate,  $f_i$ , using this classifier threshold;
10   $DR_{i+1} = DR_i \times DR_{\min}$ ;  $FP_{i+1} = FP_i \times f_i$ ; and remove correctly classified negative
    samples from the training set;
11  if  $FP_{\text{target}} < FP_i$  then
12    Evaluate the current cascaded classifier on the negative images and add misclassified
    samples into the negative training set;

```

**Output:** A cascade of classifiers for each cascade level  $i = 1, \dots$ ;

---

perform well not only for normal distributions with unequal covariance matrices but also non-normal distributions. A linear discriminant classifier can be written as

$$F(\mathbf{x}) = \begin{cases} +1 & \text{if } \sum_{t=1}^n w_t h_t(\mathbf{x}) + \theta \geq 0; \\ -1 & \text{otherwise,} \end{cases} \quad (4.4)$$

where  $h(\cdot)$  defines a function which returns binary outcome,  $\mathbf{x}$  is the input image features and  $\theta$  is an optimal threshold such that the minimum number of examples are misclassified.

The asymmetric goal for training cascade classifiers can be written as a trade-off between false acceptance rate  $\varepsilon_1$  and false rejection rate  $\varepsilon_2$  as

$$r = \varepsilon_1 + \mu \varepsilon_2, \quad (4.5)$$

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

where  $\mu$  is a trade-off parameter, representing an acceptable false rejection rate at the cost of higher false acceptance rate. Various approaches have been proposed to determine this trade-off [12, 131, 140, 145]. The objective of LDA is to maximize the projected between-class covariance matrix (the distance between the mean of two classes) and minimize the within-class covariance matrix. The choice of weight coefficients,  $w$ , which satisfies the LDA objective is guaranteed to achieve this goal. Having large projected mean difference and small projected class variance indicates that the data can be separated more easily and, hence, the asymmetric goal can also be achieved more easily. On the other hand, AdaBoost minimizes symmetric exponential loss function that does not guarantee high detection rates with few false positives [139]. The selected features are therefore no longer optimal for the task of rejecting negative samples.

Another way to think of this is that AdaBoost sets initial positive and negative sample weights to  $0.5/N_p$  and  $0.5/N_n$  ( $N_p$  and  $N_n$  is the number of positive samples and negative samples). The prior information about the number of samples in each class is encoded only in the initial distribution of sample weights. The information gradually *phased out* during subsequent weak learners' training. In contrast, LDA takes the number of samples in each class into consideration when solving the optimization problem, *i.e.*, the number of samples is used in calculating the between-class covariance matrix ( $S_b$ ). Hence,  $S_b$  is the weighted difference between class mean and sample mean, which writes

$$S_b = \sum_{c_i} N_{c_i} (\mu_{c_i} - \bar{x})(\mu_{c_i} - \bar{x})^\top, \quad (4.6)$$

where  $\mu_{c_i} = N_{c_i}^{-1} \sum_{j \in c_i} x_j$ ;  $\bar{x} = N^{-1} \sum_j x_j$ ;  $N_{c_i}$  is the number of samples in class  $c_i$  and  $N$  is the total number of samples. Taking into consideration the number of samples in each class,  $N_{c_i}$ , minimizes the effect of imbalanced data sets.

In order to demonstrate this, an artificial data set similar to one used in Viola and Jones [139] is generated. A strong classifier consisting of 4 linear classifiers is learned and the results are shown in Figure 4.2. From the figure, it can be observed that the first weak classifier (#1) selected by both algorithms are the same since it is the only linear classifier with minimal error. AdaBoost then re-weights the samples and selects the next classifier (#2) which has the smallest weighted error. From the figure, the second weak classifier (#2) introduces more false positives to the final classifier. Since

most positive samples are correctly classified, the positive samples' weights are close to zero. AdaBoost selects the next classifier (#3) which classifies all samples as negative. Therefore it is clear that all but the first weak classifier learned by AdaBoost are poor because it tries to balance positive and negative errors. The final combination of these classifiers are not able to produce high detection rates without introducing many false positives. In contrast to AdaBoost, GSLDA selects the second and third weak classifier (#2, #3) based on the maximum class separation criterion. Only the linear classifier whose outputs yields the maximum distance between two classes is selected. As a result, the selected linear classifiers introduce much less number of false positives (Figure 4.2).

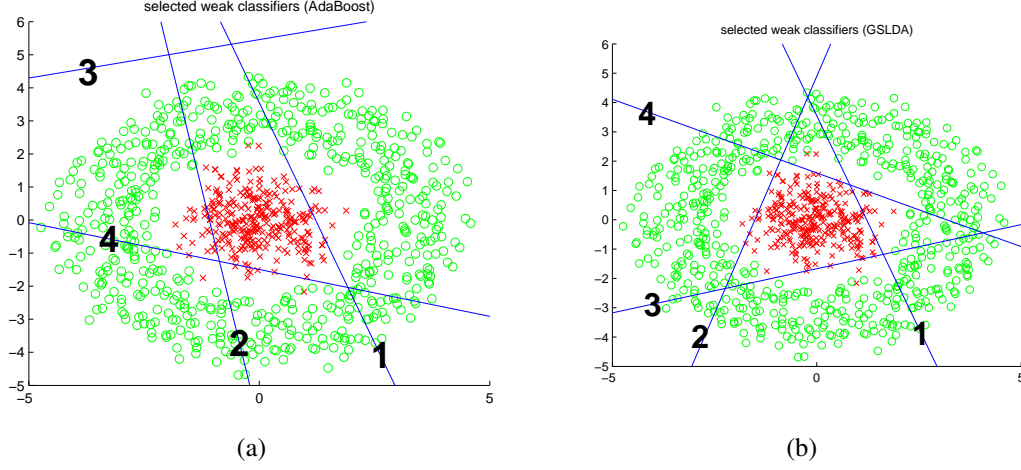
Viola and Jones [139] pointed out the limitation of AdaBoost in the context of highly skewed data distribution and proposed a new variant of AdaBoost called AsymBoost which is experimentally shown to give a performance improvement over conventional boosting. In brief, the sample weights are updated before each round of boosting with the extra exponential term which causes the algorithm to gradually pay more attention to positive samples in each round of boosting. The proposed scheme based on LDA's class-separability can be considered as an alternative classifier to AsymBoost that also takes asymmetry information into consideration.

### 4.2.3 Boosted Greedy Sparse Linear Discriminant Analysis

Before introducing the concept of BGSLDA, an explanation of boosting algorithms is briefly discussed. Boosting is one of the most popular learning algorithms. It was originally designed for classification problems. It combines the output of many weak classifiers to produce a single strong learner. A weak classifier is defined as a classifier with classification accuracy on training sets greater than random guessing. There exist many variants of boosting algorithms, *e.g.*, AdaBoost (minimizing the exponential loss), GentleBoost (fitting regression function by weighted least square methods), LogitBoost (minimizing the logistic regression cost function) [38], LPBoost (minimizing the hinge loss) [27, 63], *etc.* All of them rely on sample re-weighting and weighted majority voting. One of widely used boosting algorithms is AdaBoost [123]. AdaBoost is a greedy algorithm that constructs an additive combination of weak classifiers such that the exponential loss  $L(y, F(\mathbf{x})) = \exp(-yF(\mathbf{x}))$  is minimized. Here  $\mathbf{x}$  is the

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---



**Figure 4.2:** Two examples on toy data sets: (a) the AdaBoost classifier; (b) the GSLDA classifier (forward pass).  $\times$ 's and  $\circ$ 's represent positive and negative samples, respectively. Weak classifiers are plotted as lines. The number on the line indicates the order in which weak classifiers are selected. AdaBoost selects weak classifiers for attempting to balance weighted positive and negative error. Notice that AdaBoost's third weak classifier classifies all samples as negative due to the very small positive sample weights. In contrast, GSLDA selects weak classifiers based on the maximum class separation criterion. It can be seen that four weak classifiers of GSLDA model the positives well and most of the negative are rejected.

labeled training examples and  $y$  is its label  $y \in \{-1, +1\}$ ;  $F(\mathbf{x})$  is the final decision function where its sign predicts the class label. Each training sample receives a weight  $u_i$  that determines its significance for training the next weak classifier. In each boosting iteration, the value of weak classifier's weight coefficients,  $\alpha_t$ , is computed and the sample weights are updated according to the exponential rule (4.7). AdaBoost then selects a new hypothesis,  $h(\cdot)$ , that best classifies updated training samples with minimal weighted classification error  $e$ . The final decision rule  $H(\cdot)$  is a sign of the linear combination of the selected weak classifiers weighted by their coefficients  $\alpha_t$ . The classifier decision is given by

$$H(\mathbf{x}) = \text{sign}\left(F(\mathbf{x})\right) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right),$$

where  $\alpha_t$  is a weight coefficient;  $h_t(\cdot)$  is a weak learner and  $T$  is the number of weak classifiers.

In the previous section, the concept of GSLDA was introduced in the domain of object detection. However, decision stumps used in GSLDA are learned *only once* to save computation time. In other words, once learned, an optimal threshold, which gives smallest classification error on training sets, remains unchanged during the GSLDA training. This speeds up the training process as also shown in forward feature selection of [145]. However, it limits the number of decision stumps available for the GSLDA classifier to choose from. As a result, GSLDA fails to perform at its best. In order to achieve the best performance from the GSLDA classifier, we propose to extend decision stumps used in GSLDA training with sample re-weighting techniques used in boosting methods. In other words, each training sample receives a weight and the new set of decision stumps are trained according to these sample weights. The new classifier is termed Boosted GSLDA (in short, BGS LDA). The BGS LDA cascade learning algorithm is shown in Algorithm 4. Since the BGS LDA based object detection framework has the same input/output as GSLDA based detection framework (Algorithm 3), lines 2 – 10 in Algorithm 3 are replaced with with Algorithm 4.

In Algorithm 4, the criterion used to select the best decision stump is similar to the one applied in step (1) in Algorithm 3. Step (3) in Algorithm 4 is introduced in order to speed up the GSLDA training process. By saying that, decision stumps with a weighted error larger than  $e_{min} + \varepsilon$  are removed. Here  $e_{min}$  (smallest weighted error from all selected hypotheses) =  $\frac{1}{2} - \frac{1}{2}\beta_{max}$ ,  $\varepsilon$  is an arbitrarily small constant,  $\beta_{max}$  (largest edge of all selected hypotheses) =  $\max_{j=1,2,\dots,t} (\sum_{i=1}^N u_i^{(t)} y_i h_j(x_i))$ ,  $N$  is the number of samples,  $u_i^{(t)}$  is the current weight of sample  $x_i$ ,  $y_i$  is the class label of sample  $x_i$  and  $h_j(x_i)$  is the prediction of the training data  $x_i$  using weak classifier  $h_j$ .

Given the set of decision stumps, GSLDA selects the stump that results in maximum class separation (step (4)). The sample weights can be updated using different boosting algorithms (step (5)). In the experiment, AdaBoost re-weighting scheme (BGS LDA - scheme 1) is used:

$$u_i^{(t+1)} = \frac{u_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z^{(t+1)}}, \quad (4.7)$$

## 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

with

$$Z^{(t+1)} = \sum_i u_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i)).$$

Here  $\alpha_t = 0.5 \log((1 - e_t)/(e_t))$ ,  $e_t$  is the weighted error,  $Z^{(t+1)}$  is a normalization factor chosen such that  $u_i^{(t+1)}$  will be a probability distribution. We also use AsymBoost [139] re-weighting scheme (BGS LDA - scheme 2).

$$u_i^{(t+1)} = \frac{u_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i)) \exp(y_i \log \sqrt{k})}{Z^{(t+1)}}, \quad (4.8)$$

with

$$Z^{(t+1)} = \sum_i u_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i)) \exp(y_i \log \sqrt{k}).$$

Since BGS LDA based object detection framework has the same input/output as GS LDA based detection framework, lines 2 – 10 in Algorithm 3 are replaced with Algorithm 4.

### 4.2.4 Training Time Complexity of BGS LDA

In order to analyze the complexity of the proposed system, one need to analyze the complexity of boosting and GS LDA training. Let the number of training samples in each cascade layer be  $N$ . For boosting, finding the optimal threshold of each feature needs  $O(N \log N)$ . Assume that the size of the feature set is  $M$  and the number of weak classifiers to be selected is  $T$ . The time complexity for training boosting classifier is  $O(MTN \log N)$ . The time complexity for GS LDA forward pass is  $O(NMT + MT^3)$ .  $O(N)$  is the time complexity for finding mean and variance of each features.  $O(T^2)$  is the time complexity for calculating correlation for each feature. Since there are  $M$  features and the number of weak classifiers to be selected is  $T$ , the total time complexity for GS LDA is  $O(NMT + MT^3)$ . Hence, the total time complexity is  $O(\underbrace{MTN \log N}_{\text{weak classifier}} + \underbrace{NMT + MT^3}_{\text{GS LDA}})$ . When  $N \gg T$ , most of the computation is spent on training weak classifiers. On the other hand, when  $T$  is large, most of the computation time is spent on GS LDA calculation (finding the feature that maximizes class-separability criterion). For cascaded structure, The value of  $T$  can be set to be small, *i.e.*, the maximum number of weak classifiers in each cascade node. For face detection using Haar-like features with cascade classifiers [140],  $N$  is 4,916,  $M$  is 160,000 ( $24 \times 24$  pixels patch) and  $T$  is usually less than 200.

---

**Algorithm 4** The training algorithm for building a cascade of BGSLDA object detector.

---

```

1 while  $FP_{\text{target}} < FP_i$  do
2    $i = i + 1$ ;
3    $f_i = 1$ ;
4   while  $f_i > FP_{\text{max}}$  do
5     1. Normalize sample weights  $\mathbf{u}$ ;
6     2. Train weak classifiers  $h(\cdot)$  (e.g., decision stumps by finding an optimal
7       threshold  $\theta$ ) using the training set and sample weights;
8     3. Remove those weak classifiers with weighted error larger than  $e_{\text{max}} + \varepsilon$ 
9       (section 4.2.3);
10    4. Add the weak classifier whose output yields the maximum class separation;
11    5. Update sample weights  $\mathbf{u}$  in the AdaBoost manner (Eq. (4.7)) or AsymBoost
12       manner (Eq. (4.8));
13    6. Lower threshold such that  $DR_{\text{min}}$  holds;
14    7. Update  $f_i$  using this threshold;
15    $DR_{i+1} = DR_i \times DR_{\text{min}}$ ;
16    $FP_{i+1} = FP_i \times f_i$ ; and remove those correctly classified negative samples from
17     the training set;
18   if  $FP_{\text{target}} < F_i$  then
19     Evaluate the current cascaded classifier on the negative images and add
20     misclassified samples into the negative training set;

```

---

For fast AdaBoost training of Haar-like rectangle features, the pre-computing technique similar to [145] is applied here.

## 4.3 Experiments

---

This section is organized as follows. Data sets used in this experiment, including how the performance is analyzed, are described. Experiments and the parameters used are then discussed. Finally, experimental results and analysis of different techniques are presented.

## 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---



**Figure 4.3:** Random samples of face images used during training.

### 4.3.1 Face Detection with the GSLDA Classifier

Due to its efficiency, Haar-like rectangle features [140] have become a popular choice as image features in the context of face detection. Similar to the work in Viola and Jones [140], the weak learning algorithm known as decision stump and Haar-like rectangle features are used here due to their simplicity and efficiency. The following experiments compare AdaBoost, FloatBoost (AdaBoost with backtrack mechanism) [67] and GSLDA learning algorithms in their performances in the domain of face detection.

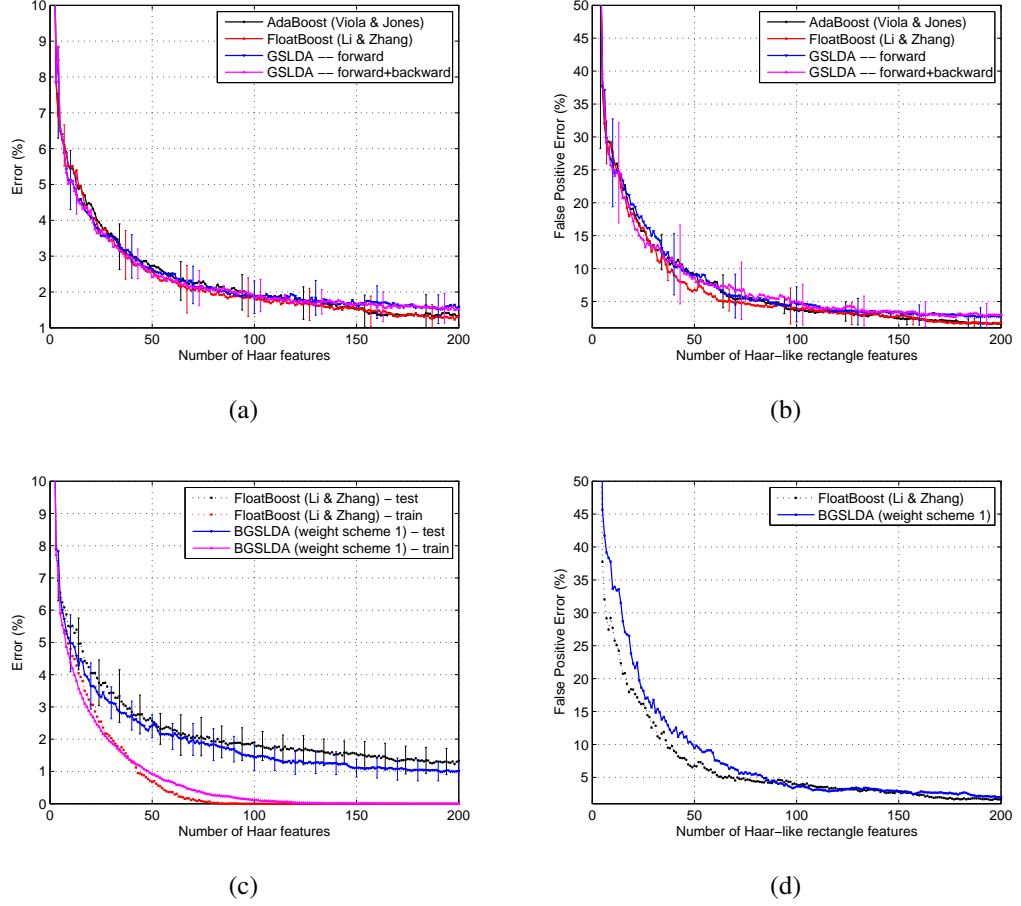
#### 4.3.1.1 Performance on Single-node Classifiers

This experiment compares single strong classifier learned using AdaBoost, FloatBoost and GSLDA algorithms in their classification performance. Data sets consist of three training sets and two test sets. Each training set contains 2,000 face examples and 2,000 non-face examples (Table 4.1). Data sets consist of 10,000 mirrored faces. The faces were cropped and rescaled to images of size  $24 \times 24$  pixels. For non-face examples, 10,000 random non-face patches are selected from non-face images obtained from the internet. Figure 4.3 shows a random sample of face training images.

**Table 4.1:** The size of training and test sets used on a single node classifier.

#	data splits	faces/split	non-faces/split
Train	3	2000	2000
Test	2	2000	2000





**Figure 4.4:** See text for details (best viewed in color). (a) A comparison of test error rates between GSLDA and AdaBoost. (b) A comparison of false alarm rates on test sets between GSLDA and AdaBoost. The detection rate on validated face sets is fixed at 99%. (c) A comparison of train and test error rates between BGSLDA (scheme 1) and AdaBoost. (d) A comparison of false alarm rates on test sets between BGSLDA (scheme 1) and AdaBoost.

For each experiment, three different classifiers are generated, each by selecting two out of the three training sets and the remaining training set for validation. The performance is measured by two different curves:- the test error rate and the classifier learning goal (the false alarm error rate on test sets given that the detection rate on validation sets is fixed at 99%). A 95% confidence interval of the true mean error rate

## 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

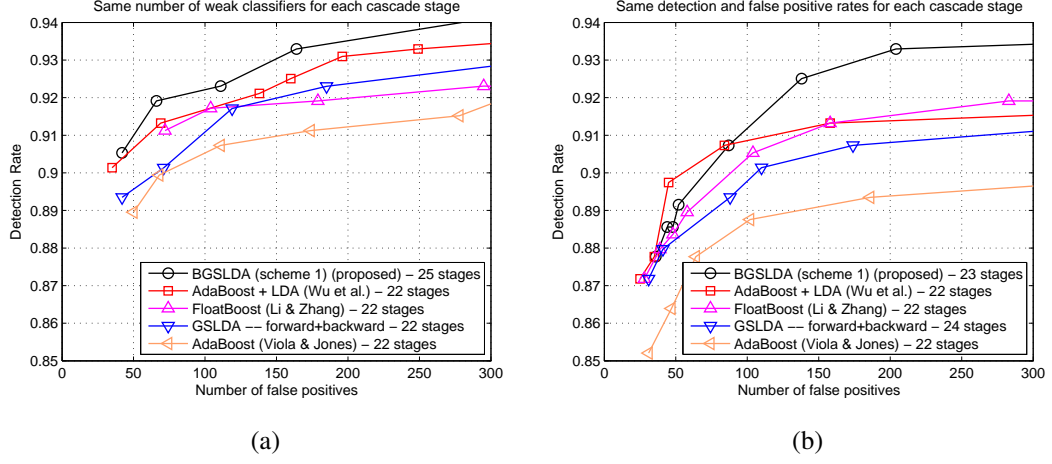
is given by the t-distribution. In this experiment, two different approaches of GSLDA: forward-pass GSLDA and dual-pass (forward+backward) GSLDA<sup>1</sup> are tested. The results are shown in Figure 4.4. The following observations can be made from these curves. Having the same number of learned Haar-like rectangle features, GSLDA achieves a comparable error rate to AdaBoost/FloatBoost on test sets (Figure 4.4(a)). GSLDA seems to perform slightly better with less number of Haar-like features ( $< 100$ ) while AdaBoost and FloatBoost seem to perform slightly better with more Haar-like features ( $> 100$ ). However, both classifiers perform almost similarly within 95% confidence interval of the true error rate. This indicates that features selected using the GSLDA classifier are as meaningful as features selected using AdaBoost/FloatBoost classifiers. From the curve, GSLDA with bi-directional search yields better results than GSLDA with forward search only. Figure 4.4(b) shows the false positive error rate on test sets. From the figure, GSLDA, AdaBoost and FloatBoost achieve a comparable false positive error rate on test sets. Similar to [67], FloatBoost has a slightly lower error rate than AdaBoost.

### 4.3.1.2 Performance on Cascades of Strong Classifiers

In this experiment, 5,000 mirrored faces from previous experiments are used. Non-face samples used in each cascade layer are collected from false positives of the previous stages of the cascade (bootstrapping). The cascade training algorithm terminates when there are not enough negative samples to bootstrap. For fair evaluation, both techniques are trained with the same number of weak classifiers in each cascade. Note that since dual pass GSLDA (forward+backward search) yields better solutions than the forward search in the previous experiment, dual pass GSLDA classifier is used to train a cascade of face detectors. The proposed face detector is tested on the low resolution face database, MIT+CMU test sets. The database contains 130 images with 507 frontal faces. In this experiment, the scaling factor is set to 1.2 and window shifting step to 1 pixel. The technique used for merging overlapping windows is similar to [140]. Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, there must be

---

<sup>1</sup> Dual-pass GSLDA performs a backward elimination after the latest weak classifier is added by forward-pass GSLDA. The process removes those previously added weak classifiers which have little help in separating positive class from negative class.



**Figure 4.5:** Comparison on MIT+CMU face test sets (a) with the same number of weak classifiers in each cascade stage of AdaBoost and its variants. (b) with 99.5% detection rate and 50% false positive rate in each cascade stage of AdaBoost and its variants. BGSLDA (scheme 1) corresponds to the GSLDA classifier with decision stumps being re-weighted using an AdaBoost scheme.

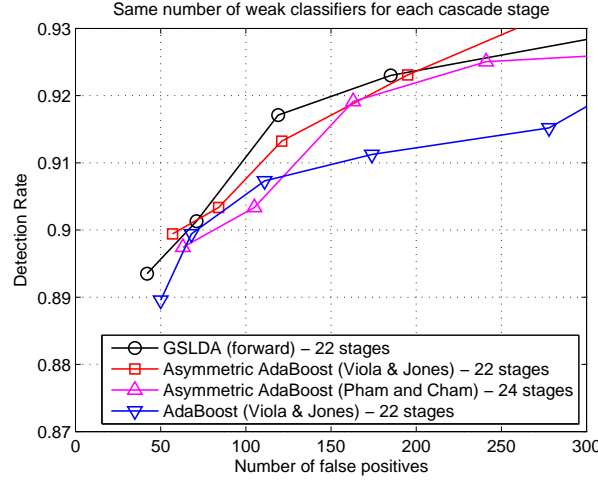
at least a 50% overlap between the predicted bounding box and ground truth bounding box. Multiple detections of the same face in an image are considered false detections.

Figures 4.5(a) and 4.5(b) show a comparison between the Receiver Operating Characteristic (ROC) curves produced by the GSLDA classifier, AdaBoost and FloatBoost. In Figure 4.5(a), the number of weak classifiers in each cascade stage is predetermined while in Figure 4.5(b), weak classifiers are added to the cascade until the predefined objective is met. The ROC curves show that GSLDA outperforms AdaBoost at all false positive rates. The observation is that by lowering the AdaBoost threshold (in order to achieve high detection rates with moderate false positive rates), the classification performance of AdaBoost is no longer optimal. Findings in this chapter are consistent with the experimental results presented in [67, 139, 145].

Wu *et al.* [145] used LDA weights instead of weak classifiers' weights provided by the AdaBoost algorithm. [67] introduced a backtrack mechanism to remove unfavorable weak classifiers (FloatBoost learning). The performance of AdaBoost+LDA, FloatBoost and GSLDA is observed to be similar. Since Haar-like features and the cascade structure similar to Viola and Jones [140] are used, it can be concluded that the

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

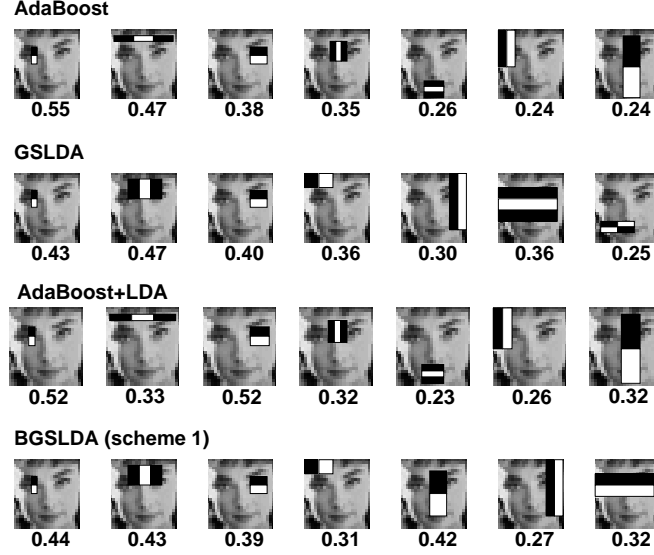


**Figure 4.6:** Comparison of different approaches on MIT+CMU face test sets using Haar-like features.

evaluation time of GSLDA face detectors is similar to that of AdaBoost face detectors.

Figure 4.6 compares the performance of LDA-based classifiers against asymmetric boosting-based classifiers [110, 139]. For [139], the asymmetric parameter is set to 1.1 using cross-validation. For [110], the classifier is trained in an offline mode and the asymmetric parameter is set to 2 using cross-validation. The number of weak classifiers in each node was set to be the same in all classifiers. Based on experimental results, the detection performance of GSLDA is similar to asymmetric AdaBoost. Since each cascade node is trained with equal number of faces and non-faces, the performance of [110] is very similar to [139]. Training each node with a different ratio of training faces and training non-faces might produce different performance results between the two versions of asymmetric boosting of [110] and [139].

Note that GSLDA not only performs better than AdaBoost but it is also much simpler. The weak classifier learning (decision stumps) is performed only once for the given set of samples (unlike AdaBoost or FloatBoost where weak classifiers have to be re-trained in each boosting iteration). GSLDA sequentially selects decision stump whose output yields the maximum eigenvalue. The process continues until the stopping criteria are met. Note that given the decision stumps selected by GSLDA, any linear classifiers can be used to calculate the weight coefficients. Based on preliminary



**Figure 4.7:** First seven Haar-like rectangle features selected from the first layer of different classifiers. The value below each Haar-like rectangle features indicates the normalized feature weight. AdaBoost and FloatBoost have the same Haar-like rectangle features in the first layer. For AdaBoost and FloatBoost, the value corresponds to the normalized  $\alpha$  where  $\alpha$  is computed from  $\log((1 - e_t)/e_t)$  and  $e_t$  is the weighted error. For LDA, the value corresponds to the normalized  $w$  such that for input vector  $x$  and a class label  $y$ ,  $w^\top x$  leads to maximum separation between two classes.

experiments, using linear SVM (maximizing the minimum margin) instead of LDA also gives a very similar result to GSLDA detector. The authors believe that using one objective criterion for feature selection and another criterion for classifier construction would provide a classifier with more flexibility than using the same criterion to select features and train weight coefficients. This finding was originally advocated in Wu *et al.* [145]. Experimental results in this chapter are consistent with experimental results reported in Wu *et al.* [145]. This finding opens up many more possibilities in combining various feature selection techniques with many existing classification techniques. We believe that a better and faster object detector can be built with careful design and experiments.

Haar-like rectangle features selected in the first cascade layer of different classifiers

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

**Table 4.2:** A summary of training time, the number of nodes and evaluation time of different classifiers. The number of cascade nodes and total weak classifiers were obtained from the classifier trained to achieve a detection rate of 99.5% and a maximum false positive rate of 50% in each cascade node. The average number of Haar-like rectangles evaluated was obtained from evaluating the trained classifier on MIT+CMU test sets. Dual-pass classifiers (forward+backward), *e.g.*, FloatBoost, GSLDA (dual-pass) take longer time to train than one-pass classifiers.

Method	Training time	# stages	# weak classifiers	Avg. # Haar features eval.
AdaBoost [140]	3 hours	22	1771	23.9
FloatBoost [67]	3+ hours	22	1532	23.3
AdaBoost+LDA [145]	3 hours	22	1436	22.3
GSLDA	16+ hours	24	2985	36.0
BGSLDA (scheme 1)	16+ hours	23	1696	24.2
AsymBoost [139]	3 hours	22	1650	22.6
AsymBoost+LDA [145]	3 hours	22	1542	21.5
BGSLDA (scheme 2)	16+ hours	23	1621	24.9

are shown in Figure 4.7. Note that all classifiers select Haar-like features that cover the area around the eyes and forehead. Table 4.2 compares the two cascade classifiers in terms of the number of weak classifiers and the average number of Haar-like rectangle features evaluated per detection window. Comparing GSLDA with AdaBoost, GSLDA has more weak classifiers and takes longer time to evaluate than AdaBoost. Unlike in AdaBoost, where training samples are reweighed in each boosting iteration, GSLDA does not update sample weights. In the proposed algorithm, weak classifiers (*e.g.*, decision stumps) are learned only once, *i.e.*, for decision stump, the threshold is trained once in the beginning. Once learned, the threshold parameter remains unchanged. This is different from AdaBoost where the threshold parameter is re-learned so that the weak classifier would yield minimal weighted misclassification error. Hence, the number of decision stumps available for training GSLDA is much smaller than the number of decision stumps used in training AdaBoost classifiers. In other words, AdaBoost can

choose a more powerful/meaningful decision stump during each boosting iteration. Interestingly, GSLDA outperforms AdaBoost. This indicates that the classifier trained to maximize class separation might be more suitable in the domain where the distribution of positive and negative samples is highly skewed. In the next section, experiments are conducted on BGSLDA.

### 4.3.2 Face Detection with the BGSLDA Classifier

The following experiment compares BGSLDA and different boosting learning algorithms in their performances for face detection. BGSLDA (weight scheme 1) corresponds to GSLDA with decision stumps being re-weighted using the AdaBoost scheme while BGSLDA (weight scheme 2) corresponds to GSLDA with decision stumps being re-weighted using the AsymBoost scheme (for highly skewed sample distributions). AsymBoost used in this experiment is from [139]. However, any asymmetric boosting approach can be applied here, *e.g.*, [35, 63].

#### 4.3.2.1 Performance on Single-node Classifiers

The experimental setup is similar to the one described in previous section. Results are shown in Figure 4.4. The following conclusions can be made from Figure 4.4(c). Given the same number of weak classifiers, BGSLDA always achieves a lower generalization error rate than FloatBoost. However, in terms of training error, FloatBoost achieves a lower training error rate than BGSLDA. This may be explained as FloatBoost has a faster convergence rate than BGSLDA. From the figure, FloatBoost only achieves lower training error rate than BGSLDA when the number of Haar-like rectangle features is larger than 50. Figure 4.4(d) shows the false alarm error rate. The false positive The false positive error rates of both classifiers are very similar.

#### 4.3.2.2 Performance on Cascades of Strong Classifiers

The experimental setup and evaluation techniques used here are similar to the one described in Section 4.3.1.1. The results are shown in Figure 4.5. Figure 4.5(a) shows a comparison between the ROC curves produced by BGSLDA (scheme 1) and FloatBoost trained with the same number of weak classifiers in each cascade. Both ROC curves show that the BGSLDA classifier outperforms both AdaBoost, FloatBoost [67]

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---

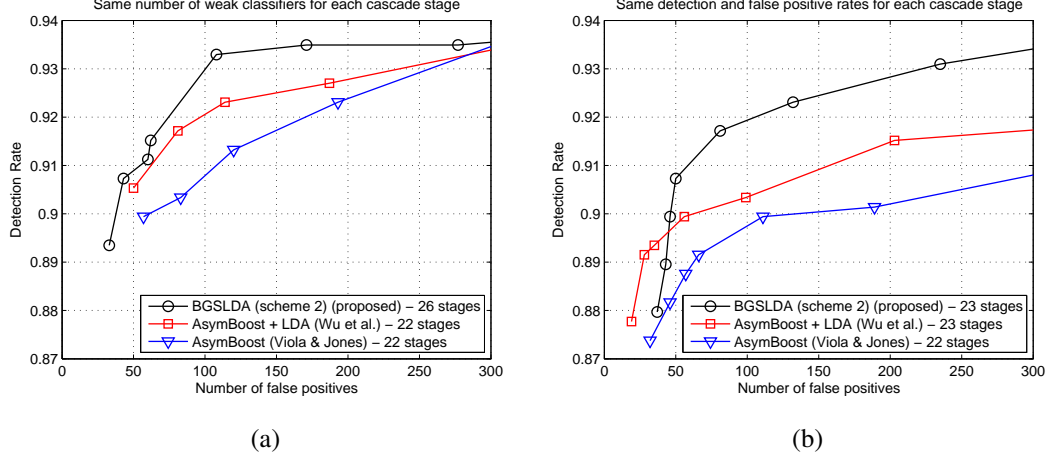


**Figure 4.8:** Face detection examples using the BGSLDA (scheme 1) detector on MIT+CMU test sets. The scaling factor is set to 1.2 and a window shifting step is set to 1 pixel. The technique used for merging overlapping windows is similar to [140].

and AdaBoost+LDA [145]. Figure 4.5(b) shows a comparison between the ROC curves of different classifiers when the number of weak classifiers in each cascade stage is no longer predetermined. At each stage, weak classifiers are added until the predefined objective is met. Again, BGSLDA significantly outperforms other evaluated classifiers. Figure 4.8 demonstrates some face detection results on the proposed BGSLDA (scheme 1) detector.

In the next experiment, the performance of BGSLDA (scheme 2) is compared with other classifiers using the asymmetric weight updating rule [139]. In other words, the asymmetric multiplier  $\exp(\frac{1}{N}y_i \log \sqrt{k})$  is applied to every sample before each round of weak classifier training. The results are shown in Figure 4.9. Figure 4.9(a) shows a comparison between the ROC curves trained with the same number of weak classifiers in each cascade stage. Figure 4.9(b) shows the ROC curves trained with 99.5% detection rate and 50% false positive rate criteria. From both figures, BGSLDA (scheme 2) outperforms other classifiers evaluated. BGSLDA (scheme 2) also outperforms BGSLDA (scheme 1). This indicates that asymmetric loss might be more suitable in domains where the distribution of positive examples and negative examples is highly imbalanced. Note that the performance gain between BGSLDA (scheme 1)





**Figure 4.9:** A comparison of various approaches on MIT+CMU test sets (a) with the same number of weak classifiers in each cascade stage on AsymBoost and its variants. (b) with 99.5% detection rate and 50% false positive rate in each cascade stage on AsymBoost and its variants. BGSLDA (scheme 2) corresponds to the GSLDA classifier with decision stumps being re-weighted using an AsymBoost scheme.

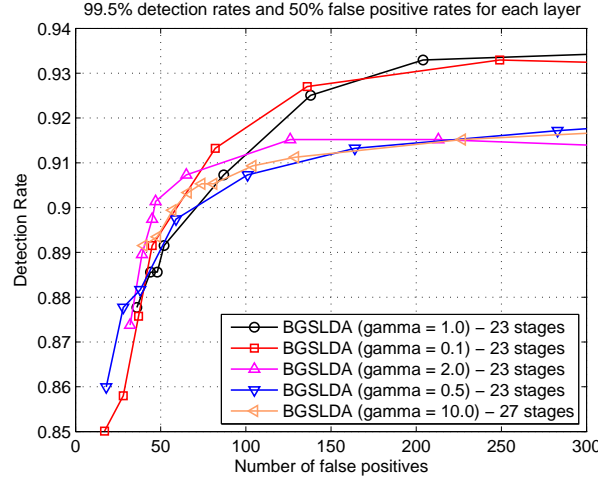
and BGSLDA (scheme 2) is quite small compared with the performance gain between AdaBoost and AsymBoost. Since LDA takes the number of samples of each class into consideration when solving the optimization problem, it is possible that this reduces the performance gap between BGSLDA (scheme 1) and BGSLDA (scheme 2).

Table 4.2 indicates that the proposed BGSLDA (scheme 1) performs at a speed comparable to AdaBoost, *i.e.*, AdaBoost requires 23.9 Haar-like features on average during evaluation while the proposed approach requires 24.2 Haar-like features on average. However, compared with AdaBoost+LDA, the performance gain of BGSLDA comes at a slightly higher cost during evaluation time (9% increase from AdaBoost+LDA classifier to BGSLDA (scheme 1) classifier). In terms of cascade training time, on a desktop with an Intel Core™ 2 Duo CPU T7300 with 4GB RAM, the total training time is less than one day. A breakdown of GSLDA training time is given in Table 4.3.

As mentioned in Cooke and Peake [23], a more general technique for generating discriminating hyper-planes is to define the total within-class covariance matrix as

$$S_w = \sum_{x_i \in C_1} (x_i - \mu_1)(x_i - \mu_1)^\top + \gamma \sum_{x_i \in C_2} (x_i - \mu_2)(x_i - \mu_2)^\top, \quad (4.9)$$

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS



**Figure 4.10:** A comparison of BGS LDA with a different value of  $\gamma$  in (4.9).

where  $\mu_1$  is the mean of class 1 and  $\mu_2$  is the mean of class 2. The weighting parameter  $\gamma$  controls the weighted classification error. Experiments were conducted on BGS LDA (scheme 1) with a different value of  $\gamma$ , namely  $\gamma \in \{0.1, 0.5, 1.0, 2.0, 10.0\}$ . All the other experiment settings remain the same as described in the previous section. The results are shown in Figure 4.10. Based on ROC curves, it can be seen that *all configurations of BGS LDA classifiers outperform the AdaBoost classifier at all false positive rates*. Setting  $\gamma = 1$  gives the highest detection rates when the number of false positives is larger than 200. Setting  $\gamma = 0.5$  performs best when the number of false positives is very small.

**Table 4.3:** A breakdown of CPU time of proposed approaches.

Process	Time
Weak classifier training	1h 20m
GSLDA feature selection	12h 40m
Bootstrapping	1h 50m

### 4.3.3 Pedestrian Detection with GSLDA Classifiers

In this section, the proposed algorithm is applied to pedestrian detection, which is considered a more difficult problem than face detection.

#### 4.3.3.1 Pedestrian Detection on Daimler-Chrysler Data Sets with Haar-like Features

In this experiment, the proposed approach is evaluated on Daimler-Chrysler pedestrian data sets [89]. Data sets contain a set of extracted pedestrian and non-pedestrian samples which are scaled to size  $18 \times 36$  pixels. Data sets consist of three training sets and two test sets. Each training set contains 4,800 pedestrian examples and 5,000 non-pedestrian examples. Performance on test sets is analyzed similarly to the techniques described in Munder and Gavrilu [89]. For each experiment, three different classifiers are generated. Testing all three classifiers on two test sets yields six different ROC curves. A 95% confidence interval of the true mean detection rate is given by the t-distribution. Two experiments are conducted using Haar-like features trained with two different classifiers: AdaBoost and GSLDA. The experimental setup is similar to the previous experiments.

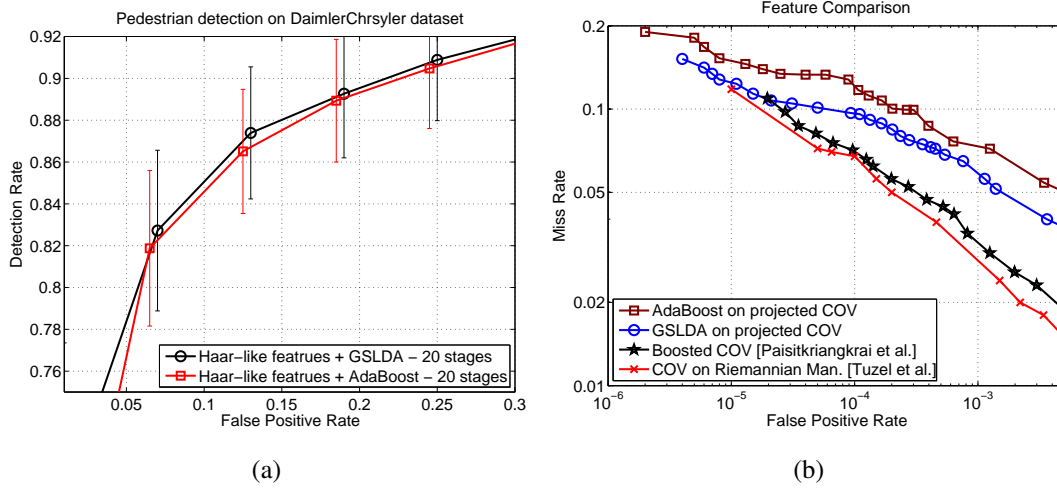
Figure 4.11(a) shows detection results of different classifiers. Again, the ROC curves show that LDA classifier outperforms the AdaBoost classifier at all false positive rates. Clearly these curves are consistent with those on face data sets.

#### 4.3.3.2 Pedestrian Detection on INRIA Data Sets with Covariance Features

Next, GSLDA is evaluated on INRIA pedestrian data sets. INRIA data sets [24] consists of one training set and one test set. The training set contains 2,416 mirrored pedestrian examples and 1,200 non-pedestrian images. The pedestrian samples were obtained from manually labeling images taken at various time of the days and various locations. The pedestrian samples are mostly in standing position. A border of 8 pixels is added to the sample in order to preserve contour information. All samples are scaled to size  $64 \times 128$  pixels. The test set contains 1,176 mirrored pedestrian examples extracted from 288 images and 453 non-pedestrian test images.

Since Haar-like features perform poorly on these data sets, covariance features are applied instead of Haar-like features [100, 135]. However, decision stump can not be

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS



**Figure 4.11:** A performance comparison on pedestrian detection on (a) Daimler-Chrysler pedestrian data sets [89] and (b) INRIA data sets [24].

directly applied since the algorithm is not applicable to multi-dimensional data. To overcome this problem, an approach similar to the one proposed in previous chapter is applied here. A multi-dimensional data is projected onto a 1D space using LDA. Decision stumps are then applied as weak classifiers. Note that this training technique is different from the one proposed in Chapter 3. Chapter 3 applied AdaBoost with weighted linear discriminant analysis (WLDA) as weak classifiers. The major drawback of the technique proposed in Chapter 3 is a slow training time. Since each training sample is assigned a weight, weak classifiers (WLDA) need to be trained  $T$  times, where  $T$  is the number of boosting iterations. In this experiment, weak classifiers (LDA) are trained only once and their projected results are stored into a memory. Because most of the training time in Chapter 3 is used to train WLDA, the new technique requires only  $\frac{1}{T}$  training time. After, multi-dimensional covariance features are projected onto a 1D space using LDA, decision stumps are trained on these 1D features. In other words, line 4 and 5 in Algorithm 3 are replaced with Algorithm 5.

In this experiment, a set of over-complete rectangular covariance filters is generated. We subsample the over-complete set in order to keep a manageable set for the training phase. The set contains approximately 45,675 covariance filters. In each stage, weak classifiers are added until the predefined objective is met. The minimum

---

**Algorithm 5** The algorithm for training multi-dimensional features.

---

```
1 foreach multi-dimensional feature do
2   1. Calculate the projection vector with LDA and project the multi-dimensional
   feature to 1D space;
3   2. Train decision stump classifiers to find an optimal threshold  $\theta$  using positive and
   negative training set;
```

---

detection rate is set to 99.5% and the maximum false positive rate is set to 35% in each stage. The cascade threshold value is then adjusted such that the cascade rejects 50% negative samples on training sets. Each stage is trained with 2,416 pedestrian samples and 2,500 non-pedestrian samples. The negative samples used in each stage of the cascades are collected from false positives of the previous stages of the cascades.

Figure 4.11(b) shows a comparison of covariance based human detectors using AdaBoost and GSLDA. The ROC curve is generated by adding one cascade level at a time. From the curve, the GSLDA classifier outperforms the AdaBoost classifier at all false positive rates. The results seem to be consistent with our results reported earlier on face detection. On a closer observation, the simplified technique performs very similar to existing covariance techniques [100, 135] at low false positive rates (lower than  $10^{-5}$ ). This method, however, seems to perform poorly at high false positive rates. Nonetheless, most real-world applications often focus on low false detections. Compared to boosted covariance features, the training time of cascade classifiers is reduced from *weeks to days* on a standard PC.

Some detection results on INRIA test sets are shown in Figure 4.12. Note that multiple scanning windows are merged using the simple technique similar to [100].

## 4.4 Conclusion

---

In this chapter, an alternative approach for visual object detection is proposed. The core of the new framework is greedy sparse linear discriminant analysis (GSLDA) [87], which aims to maximize the class-separation criterion. On various data sets for face detection and pedestrian detection, GSLDA outperforms AdaBoost when the distribution of positive and negative samples is highly skewed. To further improve the

#### 4. EFFICIENTLY TRAINING A BETTER VISUAL DETECTOR WITH SPARSE EIGENVECTORS

---



**Figure 4.12:** Pedestrian detection examples on INRIA test sets. The classifier is trained on INRIA training sets.

detection result, a boosted version GSLDA (BGSLDA) is proposed. BGSLDA combines boosting re-weighting scheme with GSLDA algorithm. Extensive experimental results reveal that the performance of BGSLDA is better than that of AdaBoost at a similar computation cost.

Although offline object detectors have performed remarkably well. One major drawback of offline techniques is that a complete set of training data has to be collected beforehand. In addition, once learned, an offline detector can not make use of newly arriving data. In the next chapter, an effective and efficient framework for learning an adaptive GSLDA model is proposed. The proposed approach could provide a better alternative to online boosting in the context of visual object detection.

# 5

## Incremental Training using Online Sparse Eigen-decomposition

### 5.1 Introduction

---

Object detection problems can be formulated as a classification task where a sliding window technique is used to scan the entire image and locate interested objects [24, 104, 140]. Viola and Jones [140] proposed an efficient detection algorithm based on the AdaBoost cascade. Their detector is the first highly-accurate real-time face detector. The authors trained classifier on data sets with a few thousand faces and a large number of negative non-faces. During the training procedure, negative samples are gradually bootstrapped and added to the training set of the boosting classifiers in the next stage. This method yields a very low false alarm rate. A large number of faces are used to cover different face appearances and poses. As a result, the computation cost and memory requirements of training the AdaBoost detector are unacceptably high. The authors spent weeks to train a model with 6,060 features (weak learners) on a face

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

training set of 4,916.

To speed up the training time bottleneck, several approaches have been proposed. Pham and Cham [109] reduced the training time of weak learners by approximating the decision stumps with class-conditional Gaussian distributions. Wu *et al.* [145] introduced a fast implementation of the AdaBoost method and proposed forward feature selection for fast training. Xiao *et al.* [146] applied distributed learning to learn their proposed dynamic cascade framework. They used over 30 desktop computers for parallel training. The authors managed to train a face detector on the training set with 500,000 positive samples and 10 billion negative samples in under 7 hours. However, these techniques are not applicable to some real-world applications where a complete set of training samples is often not given in advance. Re-training the model each time new data arrive would increase the time complexity by the factor of  $N$ , where  $N$  is the number of newly arrived samples. Hence, developing an efficient adaptive object detector has become an urgent issue for many applications of object detection in diverse and changing environments. To alleviate this problem, a few online incremental learning algorithms have been proposed for this purpose.

Online learning was first introduced in computational learning community. Since boosting is one of classifiers that have been successfully applied to many machine learning tasks, there has been considerable interest in applying boosting techniques on problems that require online learning. An online version of the boosting classifiers was proposed in [99]. The algorithm works by minimizing the classification error while updating the weak classifiers online. Grabner and Bischof [45] later applied online boosting to object detection and visual tracking. They proposed an online feature selection method, where a group of selectors is initialized randomly, each with its own feature pool. By interchanging weak learners based on lowest classification error, the algorithm would be able to capture the change in pattern induced by new samples. Huang *et al.* [54] proposed an incremental learning algorithm that adjusts a boosted classifier with domain-partitioning weak hypotheses to online samples. They showed that by incremental learning with few difficult unseen faces (*e.g.*, faces with sun glasses or extreme illumination), the performance of the online detector is now significantly improved. Parag *et al.* [105] proposed an online boosting algorithm where the parameters of the weak classifiers are updated using weighted linear regressor to minimize the weighted least square error. Liu and Yu [71] proposed a gradient-based



feature selection approach where the parameters of the weak classifiers are updated using gradient descent to minimize weighted least square error. Nonetheless, most of these proposed techniques concentrated on the application of visual tracking or object classification with small training sets and few online data sets. Hence, to date, it remains unknown whether there is any improvement in object detection by continuously updating existing models with a sufficiently large training sample set. This challenge will be revealed in Section 5.3.2.2.

Recently, Moghaddam *et al.* [87] presented a technique that combines the greedy approach with the efficient block matrix inverse formula. The proposed technique, termed greedy sparse linear discriminant analysis (GSLDA), speeds up the calculation time by  $1000\times$  compared with globally optimal solutions found by branch-and-bound search. GSLDA was applied to object detection task in previous chapter and showed very convincing results. GSLDA face detector outperforms AdaBoost based face detector due to the nature of the training data (the distribution of face and non-face samples is highly imbalanced). The objective of this chapter is to design an efficient incremental greedy sparse LDA algorithm that can accommodate new data efficiently while preserving a promising classification performance.

Unlike classical LDA where a lot of online learning techniques have been designed and proposed [103, 151, 154], there are very few works on online learning algorithm for sparse LDA. One of the difficulties is due to the fact that the sparse LDA problem is non-convex and NP-hard. It is not straightforward to design an incremental solution for sparse LDA. In this chapter, an algorithm, that efficiently learns and updates the sparse LDA classifier, is designed. The proposed online sparse LDA classifier not only incorporates new data efficiency but also yields an improvement in classification accuracy as new data become available. In brief, the approach proposed in Chapter 4 has been extended with an efficient online update schemes. The proposed method modifies weights of linear discriminant functions to adapt to new data sets. This update process generalizes the weights of linear discriminant functions and result in accuracy improvements on test set.

The key contributions of this chapter can be summarized as follows.

- An efficient incremental greedy sparse LDA classifier for training an object detector in an incremental fashion is proposed. The online algorithm integrates

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

the GSLDA based feature selection with our adaptation schemes for updating weights of linear discriminant functions and the linear classifier threshold. The proposed updating algorithm is very efficient since weak learners are neither replaced nor discarded in updating phase.

- A learning technique similar to semi-supervised learning, where the classifier makes use of the unlabeled data in conjunction with a small amount of labeled data, is adopted. As demonstrated in experiment section, the proposed online detector is able to adapt to changes in pose, view angle and illumination not captured by the set of initial training data. Compared to the initial classifier, the updated classifier shows a significant improvement in classification performance.
- Finally, extensive experiments have been conducted on several data sets that have been used in the literature. The experimental results confirm that incremental learning with online samples is beneficial to the initial classifier. The proposed algorithm can efficiently update the classifier when the new instance is inserted while achieving comparable classification accuracy to the batch algorithm<sup>1</sup>. These findings indicate that online learning plays a crucial role in object detection, especially when the initial number of training samples is small. Note that when trained with few positive samples, the detector often *under-performs* since it fails to capture the appearance variations of the target objects. By applying the proposed online technique, the classification performance can be further improved at the cost of a minor increase in training time.

The rest of the chapter is organized as follows. Section 5.2 proposes the new online GSLDA object detector. The results of numerous experiments are presented in Section 5.3. The chapter is concluded in Section 5.4.

### 5.2 Online Learning of GSLDA Classifiers

---

For ease of exposition, the symbols and their denotations used in this chapter are summarized in table 5.1. An introduction to Linear discriminant analysis (LDA) and

---

<sup>1</sup>We use the terms “batch learning” and “offline (batch) learning” interchangeably in this chapter.

**Table 5.1:** Notation

Notation	Description
$C_1, C_2$	Class 1 (positive class), class 2 (negative class)
$N$	Number of training samples in each classifier (cascade layer)
$N_1, N_2$	The number of training samples in first and second class, respectively
$M$	The size of the feature sets (for decision stumps, this is also equal to the number of weak learners)
$T$	The number of features to be selected
$\mathbf{X}$	Data matrix
$\mathbf{x}$	The new instance being inserted
$\bar{\mathbf{m}}$	The global mean of the training samples
$\mathbf{m}_1, \mathbf{m}_2$	The mean (centroid) of the first and second class, respectively
$\Sigma_1, \Sigma_2$	The covariance of the first and second class
$\mu_1, \mu_2$	The projected mean of the first and second class
$\sigma_1, \sigma_2$	The projected covariance of the first and second class
$S_b, \tilde{S}_b$	Between-class scatter matrix and its updated value after the new instance $\mathbf{x}$ has been inserted
$S_w, \tilde{S}_w$	Within-class scatter matrix and its updated value
$\mathbf{w}$	Weights of linear discriminant functions (also referred to as weak learners' coefficients in the context)
$w_0$	The linear classifier threshold

greedy sparse linear discriminant analysis (GSLDA) can be found in Chapters 3 and 4.

The major challenge of GSLDA object detectors in real-world applications is that a complete set of training samples is often not given in advance. As new data arrive, the between-class and within-class scatter matrices,  $S_b$  and  $S_w$ , will change accordingly. In offline GSLDA, the value of both matrices would have to be recomputed from scratch. However, this approach is unacceptable due to its heavy computation

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

and storage requirements. First, the cost of computing both matrices grows with the number of training samples. As a result, the algorithm will run slower and slower as time progresses. Second, the offline (batch) approach uses the entire set of training data for each update. In other words, the previous training data needs to be stored for the retraining purpose.

In order to overcome these drawbacks, an online learning algorithm, termed online greedy sparse LDA (OGSLDA), is proposed. The OGSLDA algorithm consists of two phases: the initial offline learning phase and the incremental learning phase. The training procedure in the initial phase is similar to the offline GSLDA algorithm outlined in Chapter 4. Here it is assumed that the number of training samples available initially is adequate and well represents the true density. In the second phase, the learned covariance matrices are updated in an incremental manner.

It is important to point out that many incremental LDA-like approximated algorithms have been proposed in [56, 103, 151]. Ye *et al.* proposed an efficient LDA-based incremental dimension reduction algorithm which applied matrix decomposition and matrix updating techniques for memory and computation efficiency [151]. Kim *et al.* proposed an incremental LDA by applying the concept of the sufficient spanning set approximation in each update step [56]. However, the authors did not find any of the existing LDA-like algorithms appropriate to our problems. Based on preliminary experiments, the projection matrix determined in subspace often gives worse discriminant power than that from full space. This might be due to their dimension reduction algorithms which reduced between-class and within-class scatter matrices to a much smaller size. The proposed online GSLDA guarantees to build the same between-class and within-class scatter matrices as offline (batch) GSLDA given the same training data. The reason why one needs not worry about large dimensions is because applying sparse LDA in the initial phase already reduces the number of dimensions one has to deal with. Hence, given the same set of features, the accuracy of the proposed online GSLDA is better than the existing incremental LDA-like approximated algorithms. The only expensive computation left in the proposed algorithm is eigen-analysis. In order to avoid the high computation complexity of continuously solving generalized eigen-decomposition, the efficient matrix inversion updating techniques based on inverse Sherman-Morrison formula is applied. As a result, the proposed incremental algorithm is very robust and efficient.

This section begins by proposing an efficient approach to incrementally update both within-class and between-class scatter matrices as new observations arrive. Then, an approach used to update the classifier threshold is described. Finally, the storage and training time complexity of the proposed method are analyzed.

### 5.2.1 Incremental Update of Between-class and Within-class Matrices

Since GSLDA assumes Gaussian distribution, the incremental update of class mean and class covariance can be computed very quickly. The techniques used to update both matrices can be easily derived. The procedure proceeds in three steps:

1. Updating between-class scatter matrix,  $S_b$ ;
2. Updating within-class scatter matrix,  $S_w$ ;
3. Updating inverse of within-class scatter matrix,  $S_w^{-1}$ .

#### 5.2.1.1 Updating Between-class Scatter Matrix:

For 2 classes, ( $C_1$  and  $C_2$ ),  $S_b$  can be written as,

$$S_b = \frac{N_1 N_2}{N} (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top. \quad (5.1)$$

The expression can be interpreted as the scatter of class 1 with respect to the scatter of class 2. Let  $\mathbf{x}$  be a new instance being inserted. The updated  $\tilde{\mathbf{m}}_1$  and  $\tilde{\mathbf{m}}_2$  can be calculated from

$$\begin{aligned} \tilde{\mathbf{m}}_1 &= \begin{cases} \mathbf{m}_1 + \frac{\mathbf{x} - \mathbf{m}_1}{N_1 + 1} & \text{if } \mathbf{x} \in C_1; \\ \mathbf{m}_1 & \text{otherwise,} \end{cases} \\ \tilde{\mathbf{m}}_2 &= \begin{cases} \mathbf{m}_2 & \text{if } \mathbf{x} \in C_1; \\ \mathbf{m}_2 + \frac{\mathbf{x} - \mathbf{m}_2}{N_2 + 1} & \text{otherwise.} \end{cases} \end{aligned} \quad (5.2)$$

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

### 5.2.1.2 Updating Within-class Scatter Matrix:

The covariance of a random vector  $\mathbf{X}$  is a square matrix  $\Sigma$  where  $\Sigma = \mathbf{E}[(\mathbf{X} - \mathbf{E}[\mathbf{X}])(\mathbf{X} - \mathbf{E}[\mathbf{X}])^\top]$ . Given the new instance  $\mathbf{x}$ , the updated covariance matrix is given by

$$\tilde{\Sigma} = ([\mathbf{X}, \mathbf{x}] - \tilde{\mathbf{m}}\mathbf{1}^\top)([\mathbf{X}, \mathbf{x}] - \tilde{\mathbf{m}}\mathbf{1}^\top)^\top. \quad (5.3)$$

Here  $\tilde{\mathbf{m}}$  is an updated mean after new instance has been inserted and  $\mathbf{1}$  is a column vector with each entry being 1. Its dimensionality should be clear from the context. Note that in (5.3), the constant term is left out since it makes no difference to the final solution.

$$\begin{aligned} [\mathbf{X}, \mathbf{x}] - \tilde{\mathbf{m}}\mathbf{1}^\top &= [\mathbf{X}, \mathbf{x}] - \mathbf{m}\mathbf{1}^\top + \mathbf{m}\mathbf{1}^\top - \tilde{\mathbf{m}}\mathbf{1}^\top \\ &= [\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{x} - \mathbf{m}] - (\tilde{\mathbf{m}} - \mathbf{m})\mathbf{1}^\top. \end{aligned}$$

Substitute the above expression into (5.3) and let  $\mathbf{u} = \mathbf{x} - \mathbf{m}$  and  $\mathbf{v} = \tilde{\mathbf{m}} - \mathbf{m}$ ,

$$\begin{aligned} \tilde{\Sigma} &= ([\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}] - \mathbf{v}\mathbf{1}^\top)([\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}] - \mathbf{v}\mathbf{1}^\top)^\top \\ &= ([\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}][\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}]^\top - [\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}](\mathbf{v}\mathbf{1}^\top)^\top \\ &\quad - (\mathbf{v}\mathbf{1}^\top)[\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}]^\top + (\mathbf{v}\mathbf{1}^\top)(\mathbf{v}\mathbf{1}^\top)^\top \\ &= \Sigma + \mathbf{u}\mathbf{u}^\top - [\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}]\mathbf{v}^\top \\ &\quad - \mathbf{v}([\mathbf{X} - \mathbf{m}\mathbf{1}^\top, \mathbf{u}]\mathbf{1})^\top + (N+1)\mathbf{v}\mathbf{v}^\top \\ &= \Sigma + \mathbf{u}\mathbf{u}^\top - (N\mathbf{m} - N\mathbf{m} + \mathbf{u})\mathbf{v}^\top \\ &\quad - \mathbf{v}(N\mathbf{m} - N\mathbf{m} + \mathbf{u})^\top + (N+1)\mathbf{v}\mathbf{v}^\top \\ &= \Sigma + \mathbf{u}\mathbf{u}^\top - \mathbf{u}\mathbf{v}^\top - \mathbf{v}\mathbf{u}^\top + (N+1)\mathbf{v}\mathbf{v}^\top \\ &= \Sigma + (\mathbf{u} - \mathbf{v})(\mathbf{u} - \mathbf{v})^\top + N\mathbf{v}\mathbf{v}^\top \\ &= \Sigma + (\mathbf{x} - \tilde{\mathbf{m}})(\mathbf{x} - \tilde{\mathbf{m}})^\top + N(\tilde{\mathbf{m}} - \mathbf{m})(\tilde{\mathbf{m}} - \mathbf{m})^\top. \end{aligned} \quad (5.4)$$

Note that  $\mathbf{X}\mathbf{1} = \mathbf{m}\mathbf{1}^\top\mathbf{1} = N\mathbf{m}$ . Next, we consider updating within-class scatter matrix. Let  $\mathbf{x}$  be a new instance being inserted. The updated matrix,  $\tilde{S}_w$ , can be calculated from

$$\tilde{S}_w = \begin{cases} \tilde{\Sigma}_1 + \Sigma_2 & \text{if } \mathbf{x} \in C_1; \\ \Sigma_1 + \tilde{\Sigma}_2 & \text{otherwise.} \end{cases} \quad (5.5)$$

### 5.2.1.3 Updating Inverse of Within-class Scatter Matrix:

As mentioned in Moghaddam *et al.* that the computational complexity of 2-class GSLDA relies heavily on the calculating of within-class scatter matrix inversion [87]. In order to update the matrix inversion efficiently, the technique known as inverse Sherman-Morrison decomposition, proposed by Sherman and Morrison [128], can be applied here. Let  $\Sigma$  be the square matrix of size  $M \times M$  which can be written as

$$\Sigma = \Sigma_0 + \mathbf{p}_1 \mathbf{q}_1^\top + \mathbf{p}_2 \mathbf{q}_2^\top. \quad (5.6)$$

Here  $\Sigma_0$  is assumed to be nonsingular and  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^M$ . The inverse of  $\Sigma$  is given by

$$\Sigma^{-1} = \Sigma_0^{-1} - \Sigma_0^{-1} U D^{-1} V^\top \Sigma_0^{-1} \quad (5.7)$$

where  $D^{-1} = \begin{bmatrix} r_1^{-1} & 0 \\ 0 & r_2^{-1} \end{bmatrix}$ ,  $U = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 - \frac{\mathbf{q}_1 \Sigma_0^{-1} \mathbf{p}_2}{r_1} \mathbf{p}_1 \end{bmatrix}$ ,  $V = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 - \frac{\mathbf{q}_2^\top \Sigma_0^{-1} \mathbf{p}_1}{r_1} \mathbf{q}_1 \end{bmatrix}$ ,  
 $r_1 = 1 + \mathbf{q}_1^\top \Sigma_0^{-1} \mathbf{p}_1$ ,

$$r_2 = 1 + \left( \mathbf{q}_2 - \frac{\mathbf{q}_2^\top \Sigma_0^{-1} \mathbf{p}_1}{r_1} \mathbf{q}_1 \right)^\top \Sigma_0^{-1} \mathbf{p}_2.$$

The updated inverse of within-class scatter matrix can be written as

$$\tilde{S}_w^{-1} = S_w^{-1} - S_w^{-1} U D^{-1} V S_w^{-1} \quad (5.8)$$

where  $\mathbf{p}_1 = \mathbf{q}_1 = \mathbf{x} - \tilde{\mathbf{m}}$ ,  $\mathbf{p}_2 = N_c(\tilde{\mathbf{m}} - \mathbf{m})$  and  $\mathbf{q}_2 = \tilde{\mathbf{m}} - \mathbf{m}$  (from (5.4) and (5.6)).

## 5.2.2 Updating Weak Learners' Coefficients and Threshold

Given the updated within-class matrix,  $\tilde{S}_w^{-1}$ , and between-class matrix,  $\tilde{S}_b$ , the updated weights of linear discriminant functions can now be calculated from matrix-vector multiplication using (3.5). To complete the linear classifier, the threshold  $w_0$  has to be obtained. Three criteria can be adopted.

The first criterion is to apply the optimal Bayesian classifier in the projected space. In other words, the selected threshold should be the value in which the one-dimensional distribution functions in the projected lines are equal. The mean and variance in the transformed space can be calculated as

$$\mu_c = \mathbf{w}^\top \mathbf{m}_c, \quad \sigma_c = \mathbf{w}^\top \Sigma_c \mathbf{w}. \quad (5.9)$$

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

Let  $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$  and  $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ , the optimal threshold is calculated as the point in which the one-dimensional density function of two classes are equal. Let  $\log \Pr(x_1) = \log \Pr(x_2)$ . After some algebraic expansions and simplifications, the expression can be written in the second-order polynomial,

$$ax^2 + bx + c = 0$$

where  $a = -\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2}$ ,  $b = \frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}$  and  $c = \frac{\mu_2^2}{2\sigma_2^2} + \log(\sigma_2) - \frac{\mu_1^2}{2\sigma_1^2} + \log(\sigma_1)$ . The quadratics have two roots,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

In the implementation, the threshold,  $w_0$ , is chosen to be the value between the two class means,

$$w_0 = x \text{ where } \mu_1 < x < \mu_2. \quad (5.10)$$

The second criterion is to choose the threshold which yields high detection rate with moderate false alarm rate. This asymmetric criterion is often adopted in cascade framework [140]. Let  $\phi(Z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^Z \exp(-\frac{1}{2}u^2)du$  be the cumulative distribution function (CDF) of the standard normal random variable  $Z$ . If  $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ , the CDF of  $X$  is  $\phi(Z)$  where  $Z = \frac{X - \mu_1}{\sigma_1}$ . Let the miss rate by  $p$ , the threshold which yields  $1 - p$  detection rate can be calculated as

$$w_0 = \mu_1 + Z\sigma_1 = \mu_1 + \phi^{-1}(p)\sigma_1. \quad (5.11)$$

The last criterion is to set the threshold to be the projected mean of the negative classes. This threshold helps us ensure the target asymmetric learning goal (moderate (50%) false positive rate with high detection rate). The threshold for the last criterion is

$$w_0 = \mu_2. \quad (5.12)$$

The above three threshold updating rules might look oversimple. However, Rueda [122] performed a few numerical simulations on multi-dimensional normally distributed classes and real-life data taken from UCI machine learning repository. It was reported



that selecting threshold using the simple approach as (5.10) often leads to smaller classification error than the traditional Fisher’s approach.

Unlike many online boosting algorithms which modify the parameters of the weak learners to adapt to new data sets. For example, in Grabner and Bischof [45], the parameters of the weak learners are updated using Kalman filtering; Parag *et al.* [105] updated the parameters using linear regression; Liu and Yu [71] updated the parameter using gradient descent, *etc.* The authors have found that extreme care has to be taken when one considers updating weak learners’ parameters for application of object detection. Artificial asymmetric data is generated to demonstrate this. The authors train two different incremental linear weak classifiers with different parameter updating schemes:

1. Incrementally update the model based on Gaussian distribution similar to Grabner and Bischof [45].
2. Incrementally update linear coefficients and intercept to minimize least square error (LSE) using linear regression similar to Parag *et al.* [105] (uniform sample weights are assumed here).

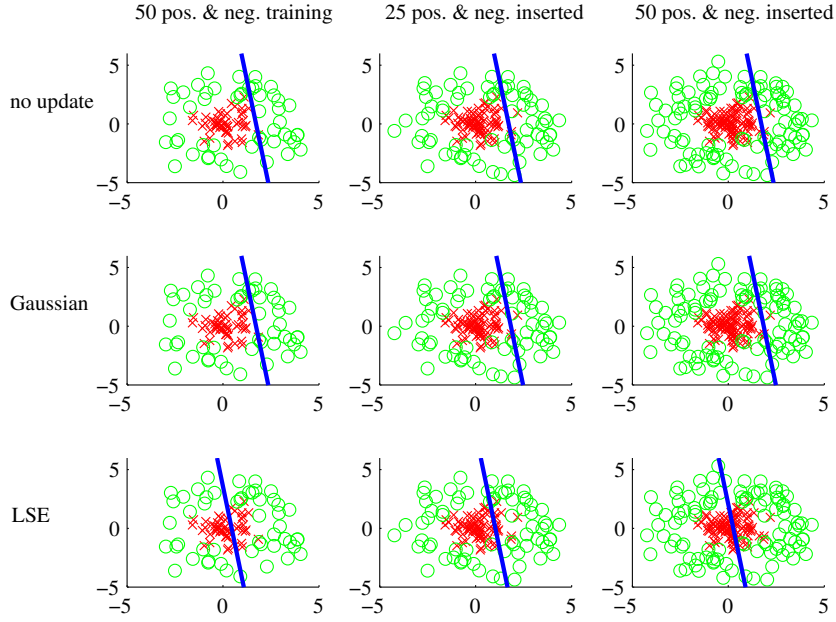
In this experiment, each weak learner represents a linear function with different coefficients (slopes). Each weak learner has one updatable parameter, *i.e.*, linear classifier threshold (intercept). The authors apply the GSLDA algorithm and select the weak learner with minimal classification error. Based on the selected weak learner, new samples are continuously inserted and the linear classifier threshold is updated. Figure 5.1 plots 9 different linear classifier thresholds. Top row shows the linear classifier with no parameter updating. Middle row shows the linear classifier with Gaussian updating rule. Bottom row shows the linear classifier using the linear regression algorithm. The first column shows the classifier thresholds on the initial training set. The middle and last columns show the classifier thresholds with new data being inserted. It is obvious that the top two classifier thresholds (no update and Gaussian) perform very similarly. LSE seems to perform worse when more new data are inserted. The reason may be attributed to the asymmetry of the data <sup>1</sup>. Based on these observations, parameter updating algorithms play a crucial role in the overall accuracy. The performance

---

<sup>1</sup>The regressor works very well when the data are linearly separable.

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---



**Figure 5.1:** Toy data sets.  $\times$ 's and  $\circ$ 's represent positive and negative samples, respectively. **Top row:** No update. The parameters of weak learners do not get updated. **Middle row:** Gaussian model. Linear classifier threshold is calculated from updated mean and variance (using (5.10)). **Bottom row:** Least square error. Linear classifier threshold is updated using linear regression. The leftmost column shows the classifier thresholds on the initial training set (50 positive and 50 negative training points). The middle column shows the classifier thresholds with 25 new positive and 25 new negative points inserted. The rightmost column shows the thresholds with 50 new positive and 50 new negative inserted. Due to the asymmetry of the data distributions, updating the parameters of the weak learners could result in performance deterioration.

of weak learners can be significantly weakened if parameters are not updated properly. In this chapter, only the GSLDA model is updated while weak learners' parameters are remained fixed.

The online GSLDA framework is summarized in Algorithm 6. Note that only a forward search of GSLDA algorithms is adopted here. In Chapter 4, it was shown that forward selection plus backward elimination improve the detection performance *slightly* but with extra computation.

## 5.2 Online Learning of GSLDA Classifiers

---



---

### Algorithm 6 Online GSLDA Algorithm

---

**Given:**

- The initial set of weak learners  $\{h_i; i \in [1, T]\}$  trained using offline GSLDA on small initial data;

**Input:**

- New training datum  $I$  and its corresponding class label  $y \in \{1, 2\}$ ;
- The current between-class covariance matrix,  $S_b$ ;
- The inverse of within-class covariance matrix,  $S_w^{-1}$ ;

- 1 Classify the new datum  $I$  using the given weak learners,  $\mathbf{x} = [h_1(I), h_2(I), \dots, h_T(I)]$ ;
- 2 Update  $S_b$  with  $\mathbf{x}$  using (5.1) and (5.2);
- 3 Update  $S_w^{-1}$  using (5.8);
- 4 Recalculate weak learners' coefficients,  $\mathbf{w}$ , using (3.5);
- 5 Update classifier threshold,  $w_0$ , based on node learning goal ((5.10) for minimal classification error,  $\min((5.11), (5.12))$  for asymmetric node learning goal (see Section 5.2.2);

**Output:**

- The updated between-class covariance matrix,  $\tilde{S}_b$ ;
  - The updated inverse of within-class covariance matrix,  $\tilde{S}_w^{-1}$ ;
  - The updated weak learners' coefficients,  $\tilde{\mathbf{w}}$ ;
  - The classifier threshold,  $\tilde{w}_0$
- 

#### 5.2.2.1 Incremental Learning Computational Complexity

Since the initial training of online GSLDA is the same as offline GSLDA, the time complexity of offline GSLDA is briefly explained here. Let us assume decision stumps are chosen as our weak learners. Let the number of training samples be  $N$ . Finding an optimal threshold of each feature needs  $O(N \log N)$ . Assume that the size of the feature set is  $M$ . The time complexity for training weak learner is  $O(MN \log N)$ . During GSLDA learning, one needs to find mean  $O(N)$ , variance  $O(N)$  and correlation  $O(T^2)$  for each feature. Since there are  $M$  features and the number of weak learners to be selected is  $T$ , the total time complexity for offline GSLDA is  $O(MN \log N + MNT + MT^3)$ .

Given the selected set of weak learners, the time complexity of online GSLDA when new instance is inserted can be calculated as follows. Since the number of weak

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

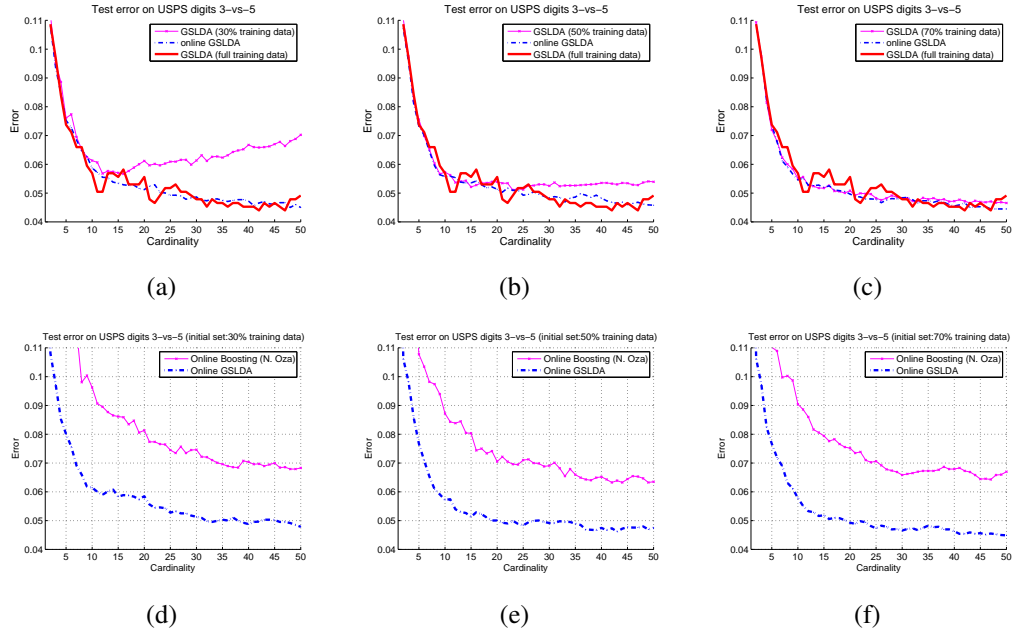
learners is  $T$ , the total time complexity to calculate  $\mathbf{x}$  in Step 1 is  $O(T)$ . It also takes  $O(T)$  to update the class mean in Step 2. In step 3, calculating  $U$ ,  $V$ ,  $r1$ ,  $r2$  take  $O(T^2)$ . In this step, the order in which one calculates the matrix-matrix multiplication affects the overall efficiency. Since the problem only involves with a small matrix chain multiplication, it is possible to go through each possible order and pick the most efficient one. For (5.7), matrix-matrix multiplication is performed in the following order  $(((\Sigma_0^{-1}U)D^{-1})(V^T \Sigma_0^{-1}))$ . The number of operations required to compute  $(\Sigma_0^{-1}U)$  is  $O(T \times T \times 2)$ ,  $((\Sigma_0^{-1}U)D^{-1})$  is  $O(T \times 2 \times 2)$ ,  $(V^T \Sigma_0^{-1})$  is  $O(2 \times T \times T)$  and  $(((\Sigma_0^{-1}U)D^{-1})(V^T \Sigma_0^{-1}))$  is  $O(T \times 2 \times T)$ . Hence, the complexity of updating matrix inversion is still in the order of  $O(T^2)$ . Since the size of within-class matrix is  $T \times T$ , the matrix-vector multiplication in Step 4 takes  $O(T^2)$ . Updating classifier threshold in Step 5 takes  $O(T^2)$  for the first criterion (First, the projected mean and covariance are computed:  $O(T)$  and  $O(T^2 + T)$ , respectively. Then, the closed-form second-degree polynomial is calculated). The second criterion in Step 5 takes  $O(T^2)$  (Again, the time complexity of projected mean and covariance is  $O(T)$  and  $O(T^2 + T)$ ). The third criterion in Step 5 takes  $O(T)$  (Here only the dot product of two vectors has to be calculated.). Hence, the time complexity of Step 5 is at most  $O(T^2)$ . Therefore, the total time complexity for online GSLDA with the insertion of a new instance is at most  $O(\underbrace{N_0 M \log N_0 + N_0 M T + M T^3}_{\text{Offline}} + \underbrace{T^2}_{\text{Online}})$ . Here  $N_0$  is the number of initial training samples which assumed to be small. Note that the speed-up of online GSLDA over offline (batch) GSLDA is noticeable, *i.e.*,  $\underbrace{O(NT^2)}_{\text{Online}} \ll \underbrace{O(N^2 \log N)}_{\text{Batch}}$ , when more instances are inserted into the training set ( $N \gg N_0$ ).

In terms of memory usage, between-class scatter matrix takes up  $O(2T)$ . The inverse of within-class scatter matrix occupies  $O(T^2)$ . For the first and second criteria in Step 5, the covariance matrices of  $\Sigma_1$  and  $\Sigma_2$  take up  $O(2T^2)$ . Hence, the extra memory requirements for online GSLDA are at most  $O(3T^2 + 2T)$ . Given that the selected number of weak classifiers in each cascade layer is often small ( $T < 200$ ), the time and memory complexity of online GSLDA is almost negligible.

## 5.3 Experiments

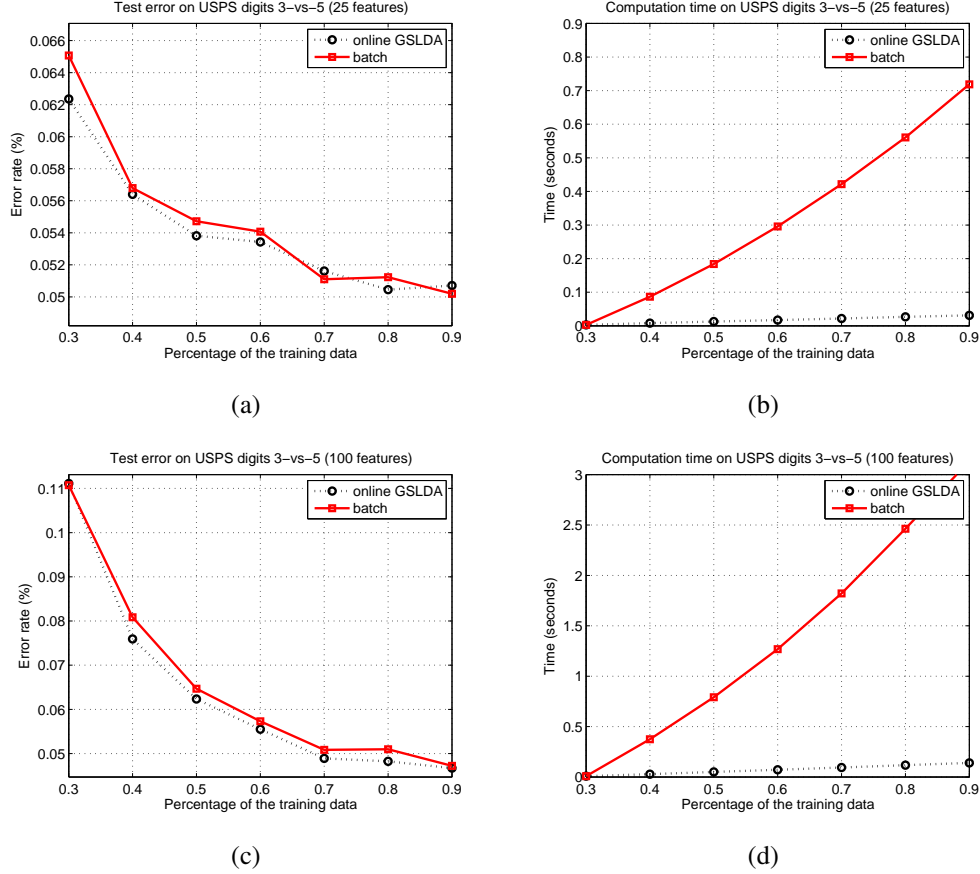
This section is organized as follows. The data sets used in this experiment, including how the performance is analyzed, are described. Experiments and the parameters used are then discussed. Finally, experimental results and analysis of different techniques are presented.

### 5.3.1 USPS Digits Classification



**Figure 5.2: Top:** A classification error rate between offline GSLDA and online GSLDA on  $16 \times 16$  pixels USPS digits data sets [115]. The number of initial training data for online GSLDA is (a) 30%, (b) 50%, (c) 70% of the available training data. All experiments, except offline (batch) GSLDA (trained with full training sets), are run 10 times. The mean of the errors are plotted. **Bottom:** A classification error rate between online GSLDA and online boosting [99]. The number of initial training data is (d) 30%, (e) 50%, (f) 70% of the available training data. All experiments are run 10 times.

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION



**Figure 5.3:** A comparison of classification error rate and computation cost between online GSLDA and offline (batch) GSLDA on  $16 \times 16$  pixels USPS digits data sets [115]. The number of nonzero components of the feature coefficients ( $\ell_0$  norm) is set to 25 (a,b) and 100 (c,d).

Online GSLDA is compared against offline (batch) GSLDA for classification of  $16 \times 16$  pixels USPS digits ‘3’ and ‘5’. Data sets consist of 406 training instances and 418 test instances for the digit ‘3’, 361 training instances and 355 test instances for digit ‘5’ [115]. The raw intensity value is used as the features. Hence, the total number of features is 256. For batch learning, greedy approach is applied to sequentially select feature which yields maximal class separation (forward search). The performance of the classifier on the given test set is evaluated and the error rate is measured [87]. For online learning, the authors randomly select 30/50/70 percent training samples as the

training set. Incremental updating is performed with the remaining training instances being inserted one at a time. Decision stumps are used as the weak learners for both classifiers. All experiments, except offline (batch) GSLDA (trained with full training sets), are run 10 times. The mean of the classification errors are plotted.

Figures 5.2(a), 5.2(b) and 5.2(c) show the achieved classification error rates by offline (batch) GSLDA and online GSLDA. In the figures, the horizontal axis shows the  $\ell_0$  norm of the feature coefficients, *i.e.*, the number of weak classifiers, and the vertical axis indicates the classification error rate on test data. Based on our observations, the error rate decreases when more training instances are used. It is important to point out that in this experiment the error rate of online GSLDA is quite close to that by offline (batch) GSLDA. The authors also train the offline GSLDA classifier with 30%, 50% and 70% training data. It was observed that error rates of GSLDA (30% training data) increase when the number of dimensions increases. This is not surprising since it is quite common for a classifier to overfit with large dimensions and small sample size. The performance of online GSLDA is compared with online boosting proposed in Oza and Russell [99]. For each weak classifier, a model is built by estimating the univariate normal distribution with weighted mean and variance for digits ‘3’ and ‘5’. The authors update the weak classifier by incrementally updating the mean and variance using weighted version of (5.2) and (5.3). The results of online boosting are shown in Figures 5.2(d), 5.2(e) and 5.2(f). The test error of online boosting decreases as the initial number of training samples increases. It can be observed that the performance of online boosting is remarkably worse than the performance of online GSLDA.

Figures 5.3(a) and 5.3(c) shows the achieved classification error rates by offline (batch) GSLDA and online GSLDA with 25 and 100 dimensions (features). In the figure, the horizontal axis shows the portion of training data instances and the vertical axis indicates the classification error rate. It can be observed that the error rate decreases when more and more training data instances are involved. Online GSLDA not only performs well on these data sets but it is also very efficient. Figures 5.3(b) and 5.3(d) show a comparison of the computation cost between offline (batch) GSLDA and online GSLDA. As can be seen, the execution time of online GSLDA is significantly smaller than that of offline (batch) GSLDA as the number of training samples grows.

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

### 5.3.2 Frontal Face Detection

Due to its efficiency, Haar-like rectangle features [140] have become a popular choice as image features in the context of face detection. Similar to the work in Viola and Jones [140], the weak learning algorithm known as decision stumps and Haar-like rectangle features are used here due to their simplicity and efficiency. The following experiments compare offline GSLDA and online GSLDA learning algorithm.

#### 5.3.2.1 Performance on Single-node Classifiers

Two experiments are conducted in this section. The first experiment compares single strong classifier learned using AdaBoost [140], AsymBoost [139], offline GSLDA [101] and the proposed online GSLDA. The data sets consist of 1,000 mirrored face examples (Figure 5.6) and 10,000 bootstrapped non-face examples. The face were cropped and rescaled to images of size  $24 \times 24$  pixels. For non-face examples, 1,000 random non-face patches are initially selected from non-face images. The other 9,000 non-face patches are added to the initial pool of training data by bootstrapping<sup>1</sup>.

Three offline face detectors are trained using AdaBoost, AsymBoost and GSLDA. Each classifier consists of 200 weak classifiers. The classifiers are tested on a challenged face videos, David Ross indoor data sets and trellis data sets<sup>2</sup>, which are publicly available on the internet. Both videos contain large lighting variation, cast shadows, unknown camera motion, and tilted face with in-plane and out-of-plane rotation. The first video contains 761 frames of a person moving from a dark to a bright area. Since the first few video frames has very low contrast (almost impossible to see faces), the first 100 frames are ignored. The second video contains 501 frames of a person moving underneath a trellis with large illumination change and cast shadows.

In this experiment, the scanning window technique is used to locate faces. The scaling factor is set to 1.2 and window shifting step is set to 1. The patch with highest classification score is classified as faces. In other words, there is only one selected face in each frame. The criteria similar to the one used in PASCAL VOC Challenge [142] is adopted here. Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the

---

<sup>1</sup>We incrementally construct new non-face samples using a trained classifier of [140].

<sup>2</sup><http://www.cs.toronto.edu/~dross/ivt/>





**Figure 5.4:** Detection examples of offline AdaBoost based frontal face detector [140] (**Top row**), AsymBoost based face detector [139] (**Second row**), GSLDA based face detector [101] (**Third row**) and our proposed OGSLDA face detector (**Last row**). All detectors are trained initially with 1,000 faces and 10,000 non-faces. Online GSLDA is incrementally updated with patches classified as faces from the previous video frames. The first video (*David indoor*) contains 761 frames of a person moving from a dark to a bright area undergoing large lighting and pose changes (frames 150, 250, 350, 409, 450, 494 and 592). The second video (*trellis*) contains 501 frames of a person moving underneath a trellis with large illumination change (frames 50, 85, 182, 231, 287, 386 and 457).

area of overlap between the predicted bounding box,  $B_p$ , and ground truth bounding

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

**Table 5.2:** A Performance comparison of four different frontal face detectors on David indoor and trellis test videos

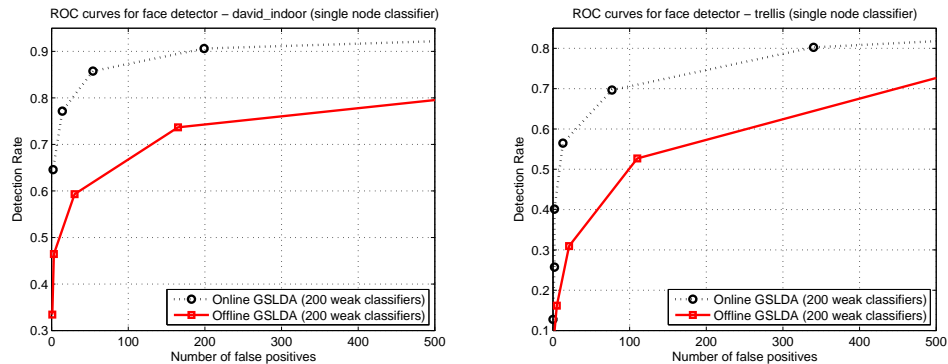
	detection rate	
	indoor sequence	trellis sequence
AdaBoost [140]	57.8%	35.3%
AsymBoost [139]	68.7%	37.5%
GSLDA [101]	70.3%	48.5%
The proposed OGSLDA	83.1%	62.1%

box,  $B_{gt}$ , must exceed 50% by the formula:

$$\frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} > 50\%.$$

For online GSLDA, predicted faces in previous frames are used to update the GSLDA model. Note that the updated sample could contain both true positives (faces) and false positives (misclassified non-faces). After the update process, the classifier predicts a single patch with highest classification score in the next frame as the face patch. This learning technique is similar to semi-supervised learning where the classifier makes use of the unlabeled data in conjunction with a small amount of labeled data. Note that unlike the work in Grabner and Bischof [45], where both positive and negative patches are used to incrementally update the model, only positive patches are used here.

Table 5.2 compares four face detectors in terms of their performance. From the table, the performance of AdaBoost face detector is the worst. This is not surprising since the distributions of training data are highly skewed (1,000 faces and 10,000 non-faces). Viola and Jones also pointed out this limitation [139]. Face detectors trained using AsymBoost and GSLDA perform quite similar on the first video. The results are consistent with those reported in Chapter 4. Experimental results show that online GSLDA performs best. Based on the observation, incrementally updating GSLDA model improves the detection results significantly at small increase in computation time. Figure 5.4 compares the empirical results between offline GSLDA and the proposed online GSLDA.



**Figure 5.5:** A comparison of ROC curves between offline and online GSLDA on David Ross indoor data sets (*left*) and trellis data sets (*right*). Note that online GSLDA outperforms offline GSLDA since the online GSLDA model is updated with predicted faces from previous frames.



**Figure 5.6:** Random samples of face images used during training.

Finally, the Receiver Operating Characteristic (ROC) curves between the offline GSLDA model (1,000 faces and 10,000 non-faces) and the online GSLDA model (initially trained with 1,000 faces and 10,000 non-faces + updated with 661 patches classified as faces) are compared. In this experiment, the scaling factor is set to 1.2 and window stepping size is set to 1. The techniques used for merging overlapping windows are similar to Viola and Jones [140]. Detections are considered true or false positives based on the area overlap with ground truth bounding boxes. The classifier threshold is adjusted and the ROC curves are plotted in Figure 5.5. Clearly, updating the trained model with relevant training data increases the overall performance of the classifiers.

In the next experiment, the performance of single strong classifiers, learned us-

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

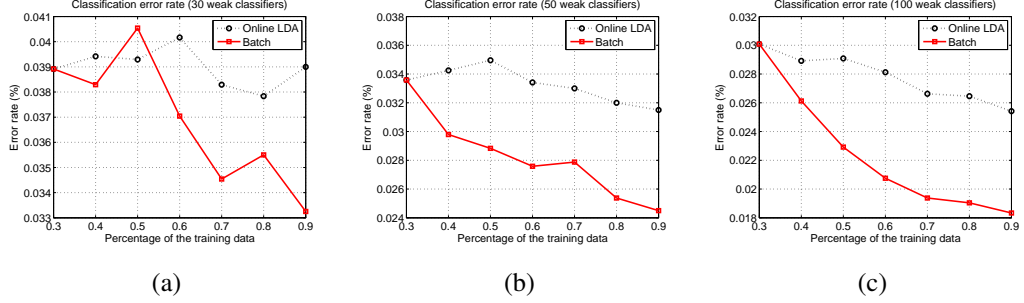
---

**Table 5.3:** The size of training and test sets used in the experiment.

#	data splits	faces/split	non-faces/split
Train	3	2000	2000
Test	2	2000	2000

ing offline GSLDA and online GSLDA on frontal faces database, is compared. The database consists of 10,000 mirrored faces. The faces were cropped and rescaled to images of size  $24 \times 24$  pixels. For non-face examples, 10,000 random non-face patches are selected from non-face images obtained from the internet. The collected patches are split into three training sets and two test sets. Each set contains 2,000 face examples and 2,000 non-face examples (Table 5.3). For each experiment, three different classifiers are generated, each by selecting two out of three training sets and the remaining training set for validation.

In this experiment, 30, 50 and 100 weak learners of Haar-like features are trained. The performance is measured by the test error rate. The results are shown in Figure 5.7. The following observations can be made from these curves. The error of both classifiers drops as the number of training samples increases. The error rate of offline (batch) GSLDA drops at a slightly faster rate than online GSLDA. This is not surprising. For offline (batch) learning, the previous set of training samples along with a new sample are used to update the decision stumps every time a new sample is inserted. For each update, GSLDA throws away previously selected weak classifiers and reselects the new 30, 50 and 100 weak classifiers. As a result, the training process is time consuming and requires a large amount of storage. In contrast, online GSLDA relies on the initial trained decision stumps. The new instance does not update the trained decision stumps but the between-class and within-class scatter matrices. The process is suboptimal compared to offline (batch) GSLDA. However, the slight increase in performance of offline (batch) GSLDA over online GSLDA (0.7% drop in test error rate for 100 weak classifiers) comes at a much higher storage cost and significantly higher computation time.



**Figure 5.7:** A comparison of classification error rates between offline (batch) GSLDA and on-line GSLDA. The number of weak learners (decision stumps on Haar-like features) in each experiment is (a) 30, (b) 50, (c) 100. The error of both classifiers drops as the number of training samples increases.

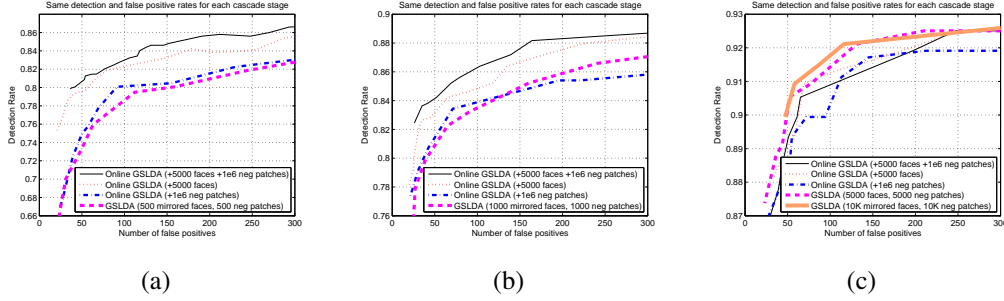
### 5.3.2.2 Performance on Cascades of Strong Classifiers

In this experiment, mirrored faces from previous experiment are used for batch learning and online learning. The number of initial positive samples used in each experiment is varied. 500 faces, 1,000 faces and 5,000 faces are used to initially train a face detector. In each experiment, four different cascaded detectors are trained. The first cascaded detector is the same as in Viola and Jones [140] *i.e.*, face data sets used in each cascade stage are the same while the non-face samples used in each cascade layer are collected from false positives of the previous stages of the cascade (bootstrapping). The cascade training algorithm terminates when there are not enough negative samples to bootstrap.

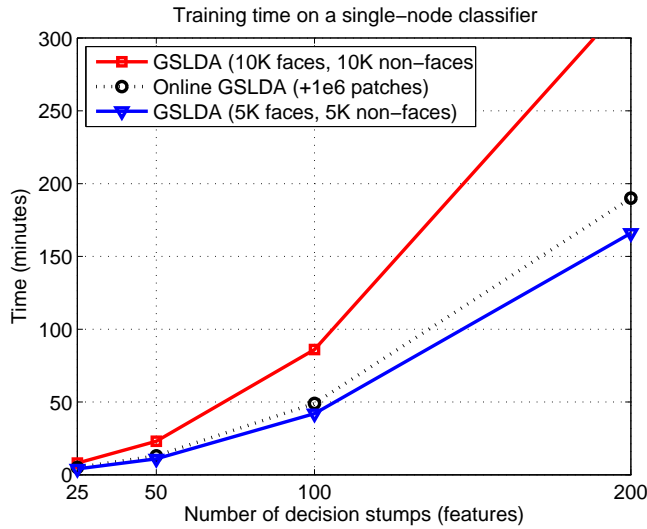
The second, third and forth face detectors are trained initially with the technique similar to the first cascaded detector. However, the second cascaded face detector is incrementally updated with new negative examples collected from false positives of the previous stages of cascade. The third cascaded face detector is incrementally updated with 5,000 unseen faces. The final face detector is incrementally updated with both false positives from previous stages and unseen faces. For each face detector, weak classifiers are added to the cascade until the predefined objective is met. In this experiment, the minimum detection rate in each cascade stage is set to 99% and the maximum false positive rate is set to 50%.

The performance of face detectors is evaluated on MIT+CMU frontal face test sets. The complete set contains 130 images with 507 frontal faces. In this experiment, the

## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION



**Figure 5.8:** A performance comparison on MIT+CMU face test sets. The four detectors are trained using (a) 500 faces, (b) 1,000 faces and (c) 5,000 and 10,000 mirrored faces. Note that online GSLDA outperforms offline GSLDA since the online GSLDA model is updated with an additional 5,000 faces.



**Figure 5.9:** A training time comparison between GSLDA and online GSLDA. The first and second GSLDA detectors are trained with 5,000 faces and 5,000 non-faces, and 10,000 faces and 10,000 non-faces, respectively. Online GSLDA is initially trained with 5,000 faces and 5,000 non-faces and updated with one million new patches. Notice that there is a slight increase in training time even though  $200\times$  more training samples have been inserted.

scaling factor is set to 1.2 and window shifting step is set to 1 pixel. The techniques

used for merging overlapping windows is similar to [140]. Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the area of overlap between the predicted bounding box and ground truth bounding box must exceed 50%. Multiple detections of the same face in an image are considered false detections.

Figure 5.8 shows ROC curves of the proposed approach with a different number of initial training data (500, 1, 000 and 5, 000 faces). To train online GSLDA, we first train offline GSLDA and update the offline detector using an additional 5, 000 faces or one million negative patches. We use offline GSLDA as the baseline in our comparisons. Figure 5.8(a) shows that online GSLDA outperforms GSLDA at all false positive rates when initially trained with 500 faces. Incrementally updating the GSLDA model with unseen faces (+5000 faces) yields a better result than updating the model with new false positives from previous stages of the cascade (+ $10^6$  negative patches). The online classifier performs best when updated with both new positive and negative patches. Figure 5.8(b) shows a comparison when the number of initial training samples have been increased to 1000 faces. The performance gap between GSLDA and online GSLDA is now smaller. The performance of both GSLDA and online GSLDA (+ $10^6$  negative patches) is observed to be very similar. This indicates that the cascade learning framework proposed by Viola and Jones might have already incorporated the benefit of massive negative patches. Incremental learning with new negative instances do not seem to improve the performance of cascaded detectors any further. Another way to explain experimental results is to use the concept of linear asymmetric classifier (LAC) proposed in Wu *et al.* [145]. In Wu *et al.*, the asymmetric node learning goal is expressed as

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} && \Pr_{\mathbf{x} \sim (\mathbf{m}_1, \Sigma_1)} \{ \mathbf{w}^\top \mathbf{x} \geq w_0 \}, \\ & \text{subject to} && \Pr_{\mathbf{y} \sim (\mathbf{m}_2, \Sigma_2)} \{ \mathbf{w}^\top \mathbf{y} \leq w_0 \} = \beta. \end{aligned} \quad (5.13)$$

Since the problem has no closed-form solution, the authors developed an approximate solution when  $\beta = 0.5$ . To find a closed-form solution, the authors assumed that  $\mathbf{w}^\top \mathbf{x}$  is Gaussian for any  $\mathbf{w}$ , class  $C_2$  distribution is symmetric and the median value of the class  $C_2$  distribution is close to its mean. The direction  $\mathbf{w}$  can then be approximated



## 5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION

---

by

$$\underset{w \neq 0}{\text{maximize}} \quad \frac{w^\top (m_1 - m_2)}{\sqrt{w^\top \Sigma_1 w}}. \quad (5.14)$$

From their objective functions, the only difference between FDA (3.5) and LAC (5.14) is that the pooled covariance matrix of FDA,  $\Sigma_1 + \Sigma_2$ , is replaced by the covariance matrix of class  $C_1$ ,  $\Sigma_1$ . In other words, when train the classifier with the asymmetric node learning goal for the cascade learning framework, the variance of negative classes becomes less relevant. In contrast, new instances of positive classes affect both the numerator and denominator in (5.14). Hence, it is easier to notice the performance improvement when new positive instances are inserted. Experimental results are consistent with their derivations.

The number of initial training faces is further increased to 5,000. All face detectors now seem to perform very similar to each other. The authors conjecture that this is the best performance that the proposed cascaded detector with the provided training set can achieve on MIT+CMU test sets. Experimental results of face detectors trained with 10,000 faces and 10,000 non-faces seem to support this assumption (Figure 5.8(c)). To further improve the performance, different cascade algorithms, *e.g.*, the Soft Cascade [12], WaldBoost [130], multi exit classifiers [111], *etc.* and a combination with other types of features, *e.g.*, edge orientation histograms (EOHs) [65], covariance features [126], *etc.*, can also be experimented. Figure 5.9 shows a comparison of the computation cost between offline (batch) GSLDA and online GSLDA. The horizontal axis shows the number of weak learners (decision stumps) and the vertical axis indicates the training time in minutes. From the figure, online learning is much faster than training the batch GSLDA classifier as the number of weak learners grows. On average, the proposed online classifier takes less than 1.5 millisecond to update a strong classifier of 200 weak learners on standard off-the-shelf PC with the use of GNU scientific library (GSL)<sup>1</sup>.

### 5.4 Conclusion

---

In this chapter, an efficient online object detection algorithm is proposed. Unlike many existing algorithms which applied boosting approach, the proposed framework makes

---

<sup>1</sup><http://www.gnu.org/software/gsl/>



use of greedy sparse linear discriminant analysis (GSLDA) based feature selection which aims to maximize the class-separation criterion. Experimental results demonstrate that the proposed incremental algorithm does not only perform comparable to the offline (batch) GSLDA algorithm but is also much more efficient. On USPS digits data sets, the proposed online algorithm with decision stumps weak learners outperforms online boosting with class-conditional Gaussian distributions. Extensive experiments on face detections reveal that it is always beneficial to incrementally train the detector with online samples.

## **5. INCREMENTAL TRAINING USING ONLINE SPARSE EIGEN-DECOMPOSITION**

---

# 6

## Conclusions and Future works

### 6.1 Summary

---

Several factors have contributed to the effectiveness and efficiency of our proposed approaches. We briefly summarize each one of them in this section.

- *Discriminative feature descriptors.* In our preliminary experiments, we have observed that feature descriptors play a vital role in the overall performance of object detectors. For example, Haar-like features perform well on frontal faces but perform poorly for the task of human detection. We redesign our feature representations using covariance features. With this new design, our visual descriptors yield improved detection performance.
- *Coarse-to-fine heuristics (Two-layer).* By applying two-layer with heterogeneous features, we have demonstrated that we can arrange features in a computationally feasible combination without compromising accuracy.
- *Training a classifier with an objective based on imbalanced data sets.* A lot of research had overlooked the importance of imbalanced data sets. In our experiments, we have found performance improvement when we take this factor into consideration through the use of LDA criterion.

## 6. CONCLUSIONS AND FUTURE WORKS

---

- *Large objects' variations and online learning.* In object detection, there are usually large variations due to visual appearances, object poses, illuminations and camera motions. AdaBoost often overfit and gives poor performance when there exists a large overlapping between object classes, *e.g.*, a mix of frontal, tilted and profile faces. By making use of unlabeled data in conjunction with a small amount of labeled data, our online detector is able to adapt to these changes.

In particular, we have advanced the state-of-the-art real-time object detection in following ways. Chapter 3 proposed two fast and robust pedestrian detection approaches. We integrated multi-dimensional covariance features with weighted Fisher Linear Discriminant Analysis for AdaBoost training. In order to speed up the computation time, a cascaded architecture was adopted [140]. All experiments were conducted on INRIA [24] and Daimler-Chrysler [89] benchmark data sets to allow direct comparison with previous works. Based on experimental results, our first approach had shown to give high detection performance at a low false positive rate. Comparing with techniques using linear SVM classifier, the proposed approach outperformed all systems evaluated. When compared with non-linear SVM systems, the system was shown to perform very similar to covariance features with Gaussian SVM and slightly inferior compared to HOG with quadratic SVM. However, the computation time of HOG with quadratic SVM was much higher than our proposed technique.

The second approach attempted to combine the efficiency of Haar-like features with the discriminative power of covariance features to further accelerate the speed of pedestrian detectors. Experiments showed that by combining Haar-like and covariance features, we speeded up the conventional covariance detector [136] by an order of magnitude in detection time without compromising the detection performance.

Prior to our work, only boosting based approaches had demonstrated robust real-time object detection. Chapter 4 proposed an alternative approach to train visual object detector. The core of the proposed approach is Greedy Sparse Linear Discriminant Analysis (GSLDA) [87], which aims to maximize the class-separation criterion. On various data sets for face detection and pedestrian detection, we had shown that the proposed approach outperformed AdaBoost when the distribution of positive and negative samples was highly skewed. One limitation of GSLDA is that a weak classifier

is not updated during feature selection. To overcome this drawback and further improve the detection result, we proposed a BGSLDA, which sequentially updates weak classifiers using boosting re-weighting scheme. Our extensive experimental results showed that the performance of BGSLDA is better than that of AdaBoost at a similar computation cost.

Chapter 5 proposed an efficient online object detection algorithm. Unlike many existing algorithms which applied boosting approach, our framework makes use of GSLDA based feature selection which aims to maximize the class-separation criterion. Our experimental results showed that our incremental algorithm did not only perform comparable to batch GSLDA algorithm but was also much more efficient. On USPS handwritten digit data sets, our online algorithm, with decision stumps as weak learners, outperformed online boosting with class-conditional Gaussian distributions. Our extensive experiments on face detections revealed that it is always beneficial to incrementally train the detector with online samples.

## 6.2 Future Works

---

There are several research areas that can be a continuation of this work. Some potential future works include:

### 6.2.1 Cascade Design

An approach that can optimally build a cascade classifier may be a future topic. In Chapter 2, we have discussed several approaches that can be applied to improve the performance of original cascade classifiers. In many works, finding better choices of threshold values and propagating scores from previous classifiers often lead to a substantial performance improvement [12, 111, 146]. We think that an approach based on these existing works would be a prime candidate for designing a better cascade structure.

Another interesting research direction is to design a cascade classifier which can detect multiple-view objects or classes, *e.g.*, multi-view face detector and multi-view human detector.

## 6. CONCLUSIONS AND FUTURE WORKS

---

### 6.2.2 Strong Classifier Learning

In this thesis, we have shown that an alternative cost function, *e.g.*, least square loss, is less prone to imbalanced training data. However, other cost function may also have a positive impact on the performance. A new cost function should be carefully designed such that it is less sensitive to imbalanced data sets. Also, the new cost function should introduce minimal number of parameters.

The new approach should also be computationally feasible for large data sets. One simple approach to handling large data sets is an approach similar to FilterBoost [15]. Unlike traditional boosting, FilterBoost avoids maintaining a distribution over a training set. The algorithm uses a rejection sampling mechanism to deal with large data sets. Data is drawn from an infinitely large source called an oracle. The filter receives a sample from the oracle and accepts it with some probability. Sampling continues until small data sets are constructed, at which time the best weak learner can be selected.

The new algorithm should also be robust to noise and overfitting. The idea similar to BrownBoost [36], where the classifier will give up on examples that are repeatedly misclassified, could also be applied. Note that traditional AdaBoost focuses on examples that are repeatedly misclassified. Hence, it does not perform well on noisy data sets.

### 6.2.3 Weak Classifier Learning

Decision stumps with Haar-like features have shown to work extremely well for frontal faces. However, having a single feature in a weak classifier might not be discriminative enough to separate difficult objects. Joint features, similar to Mita *et al.* [85], can be applied to further improve the performance of object detections. Another possibility is to make use of shared features between multiple classes in a weak classifier.

### 6.2.4 Haar-like Features

Learning an over-complete dictionary for sparse representation has proven to be very effective for signal reconstruction and classification. Numerous algorithms have been proposed for the design of dictionaries, including predefined and adaptive ones. Predefined dictionaries include an over-complete wavelet [76], Discrete Cosine Transform

[114], Fourier Transform [14], *etc.* For frontal face detection, an over-complete Haar-like wavelet has proven to be very effective and efficient. Viola and Jones selected the best set of features from a pool of over-complete Haar-like wavelets. However, predefined dictionaries have several disadvantages. First, the over-complete set grows exponentially with the resolution of object patches. Table 6.1 illustrates the number of different visual features versus the patch resolution. Searching for the best feature (finding the best weak classifier) can be problematic and time consuming due to a large set of possible features. Secondly, selected features, *e.g.*, Haar-like wavelet, may not be optimal for detecting some objects, *e.g.*, human or vehicles. An exhaustive search over all possible features, *e.g.*, HoG [24], EOHs [65], Edgelets [144], LBP [96], Co-variance [136], is extremely expensive and almost infeasible. These issues raise several questions whether one can generalize optimal features without exhaustively searching the entire set of features, *e.g.*, an automatic approach to train features that best separate objects from non-objects.

Recently, adaptive dictionary learning algorithms have been shown to achieve superior performance in several image processing applications, including de-noising [32], image compression [18] and super resolution [149]. There is a large body of literature related to sparse dictionary [2, 60, 66, 75, 98, 108]. By building adaptive features, one can explore a trade-off between sparsity and feature extraction time. It would also be interesting to apply this idea to integral image of faces (instead of raw face images) and verify excellent results of Haar-like features on frontal face detection.

Other direction of ongoing works may also include the search for new heuristic features for general object detection.

**Table 6.1:** A comparison between the number of rectangle features (ratio of 1:1, 1:2 and 2:1) [157], the number of Haar-like features (5 basic types) [140] and the number of rotated Haar-like features [68] with a given patch size (pixels  $\times$  pixels).

Patch size	# Rectangles	# Haar-like features	# Rotated Haar-like
16 $\times$ 16	3, 264	28, 288	42, 436
20 $\times$ 20	6, 300	68, 460	106, 790
24 $\times$ 24	10, 800	141, 600	207, 930
32 $\times$ 32	25, 344	444, 576	688, 064

## 6. CONCLUSIONS AND FUTURE WORKS

---

### 6.2.5 Massive Training Data

Since the choice of good training samples plays an important role in the generalization ability of cascade classifiers, the classifier should be designed to handle massive training data sets without compromising training time. One possible extension is to sample the representative training set from a large set of data and use them to train a classifier, *e.g.* weight trimming [38] and FilterBoost [15]. Another extension can be in the area of semi-supervised learning, where unlabeled data is incorporated into the learning process. Semi-supervised learning provides a principled way of incorporating prior knowledge and has been shown to outperform supervised learning model. Some of the well known approaches include EM with generative mixture models [93], self-training [118], co-training [11], transductive support vector machines [138], and graph-based methods [7, 8].

### 6.2.6 Bootstrapping

As more negative samples are rejected, it becomes harder to generate false positives for training subsequent boosted classifiers. At the moment, negative samples are first generated randomly from a large set of non-object images. The trained cascade is then used to filter all true negatives, leaving only false positives to train the next boosted classifier. The problem is, as more classifiers are trained, it becomes harder to generate a new set of false positive and the bootstrap time can be very large. In other words, the training time itself for feature selection and classifier construction becomes negligible compared to the bootstrapping time. Presumably, high bootstrapping time could be due to the fact that (1) training background images do not contain enough difficult image patches (2) training background images do not well represent background patches of test images. In this situation, it is unclear whether one should stop the training process or continue training with more complex background images. In the future, a different strategy to generate false positives may be invented to avoid this problem.



## References

- [1] S. AGARWAL AND D. ROTH. Learning a sparse representation for object detection. In *Proc. Eur. Conf. Comp. Vis.*, pages 97–101, 2002. [19](#), [21](#)
- [2] M. AHARON, M. ELAD, AND A. BRUCKSTEIN. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, **54**(11):4311, 2006. [139](#)
- [3] T. AHONEN, A. HADID, AND M. PIETIKINEN. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(12):2037–2041, 2006. [50](#)
- [4] M. S. AKSOY, O. TORKUL, AND I. H. CEDIMOGLU. An industrial visual inspection system that uses inductive learning. *J. of Intell. Manufacturing*, **15**(4):569–574, 2004. [8](#)
- [5] D. H. BALLARD AND C. M. BROWN. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ., 1982. [11](#)
- [6] A. M. BAUMBERG. *Learning deformable models for tracking human motion*. PhD thesis, The University of Leeds, 1995. [18](#), [19](#)
- [7] M. BELKIN, I. MATVEEVA, AND P. NIYOGI. Regularization and semi-supervised learning on large graphs. In *Proc. Annual Conf. Learn. Theory*, pages 624—638, 2004. [140](#)
- [8] M. BELKIN, P. NIYOGI, AND V. SINDHWANI. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, **7**:2399–2434, 2006. [140](#)

## REFERENCES

---

- [9] S. BELONGIE, J. MALIK, AND J. PUZICHA. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(4):509–522, 2002. [11](#)
- [10] C. BISHOP. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995. [2](#)
- [11] A. BLUM AND T. MITCHELL. Combining labeled and unlabeled data with co-training. In *Proc. Annual Conf. Learn. Theory*, page 100, 1998. [140](#)
- [12] L. BOURDEV AND J. BRANDT. Robust object detection via soft cascade. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, pages 236–243, 2005. [34](#), [35](#), [36](#), [37](#), [39](#), [51](#), [80](#), [86](#), [132](#), [137](#)
- [13] K. BOWYER, C. KRANENBURG, AND S. DOUGHERTY. Edge detector evaluation using empirical roc curves. *Comp. Vis. Image Understanding*, **84**(1):77–103, 2001. [10](#)
- [14] R. N. BRACEWELL. *The Fourier Transform and Its Applications*. McGraw-Hill, Boston, 2000. [139](#)
- [15] J. K. BRADLEY AND R. SCHAPIRE. Filterboost: Regression and classification on large datasets. In *Proc. Adv. Neural Inf. Process. Syst.*, **20**, pages 185–192, 2008. [138](#), [140](#)
- [16] A. BROGGI, M. BERTOZZI, A. FASCIOLI, AND M. SECHI. Shape-based pedestrian detection. In *IEEE Intel. Vehicles Symp.*, pages 215–200, 2000. [8](#), [19](#), [20](#)
- [17] S. C. BRUBAKER, J. WU, J. SUN, M. D. MULLIN, AND J. M. REHG. On the design of cascades of boosted ensembles for face detection. *Int. J. Comp. Vis.*, **77**(1):65–86, 2008. [34](#), [39](#)
- [18] O. BRYT AND M. ELAD. Compression of facial images using the k-svd algorithm. *J. of Vis. Comm. and Image Rep.*, **19**(4):270–282, 2008. [139](#)
- [19] J. CANNY. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6):679698, 1986. [10](#)

## REFERENCES

---

- [20] J. CHEN, X. CHEN, J. YANG, S. SHAN, R. WANG, AND W. GAO. Optimization of a training set for more robust face detection. *Pattern Recogn.*, **42**(11):2828–2840, 2009. [50](#)
- [21] W.W. COHEN. Fast effective rule induction. In *Proc. Int. Conf. Mach. Learn.*, pages 115–123, 1995. [40](#)
- [22] D. COMANICIU AND P. MEER. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(5):603–619, 2002. [69](#)
- [23] T. COOKE AND M. PEAKE. The optimal classification using a linear discriminant for two point classes having known mean and covariance. *J. of Multivariate Anal.*, **82**:379–394, 2002. [101](#)
- [24] N. DALAL AND B. TRIGGS. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 886–893, San Diego, CA, 2005. [7](#), [9](#), [11](#), [12](#), [13](#), [19](#), [24](#), [26](#), [27](#), [61](#), [62](#), [63](#), [65](#), [70](#), [71](#), [76](#), [103](#), [104](#), [107](#), [136](#), [139](#)
- [25] JOHN DAUGMAN. Complete discrete 2-d gabor transofrms by neural networks for image analysis and compressions. *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, **36**(7):1169–1179, 1988. [10](#)
- [26] L. S. DAVIS AND A. MITICHE. Edge detection in textures—maxima selection. *Computer Graph. and Image Proc.*, **16**(2):158–165, 1981. [10](#)
- [27] A. DEMIRIZ, K. P. BENNETT, AND J. SHAWE-TAYLOR. Linear programming boosting via column generation. *Mach. Learn.*, **46**(1-3):225–254, 2002. [87](#)
- [28] A. DESTRERO, C. DE MOL, F. ODONE, AND A. VERRI. A sparsity-enforcing method for learning face features. *IEEE Trans. Image Process.*, **18**(1):188–201, 2009. [84](#)
- [29] T. G. DIETTERICH. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.*, **40**(2):139–158, 2000. [45](#)

## REFERENCES

---

- [30] R. DUDA, P. HART, AND D. STORK. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2001. [57](#)
- [31] M. DUNDAR AND M. J. BI. Joint optimization of cascaded classifiers for computer aided detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, 2007. [34](#), [38](#)
- [32] M. ELAD AND M. AHARON. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, **15**(12):3736–3745, 2006. [139](#)
- [33] M. ENZWEILER, P. KANTER, AND D. M. GAVRILA. Monocular pedestrian recognition using motion parallax. In *Intell. Vehicles Symp.*, pages 792–797, 2008. [9](#)
- [34] M. EVERINGHAM, A. ZISSEMAN, C. WILLIAMS, L. VAN GOOL, M. ALLAN, C. BISHOP, O. CHAPELLE, N. DALAL, T. DESELAERS, G. DORKO, S. DUFFNER, J. EICHHORN, J. FARQUHAR, M. FRITZ, C. GARCIA, T. GRIFFITHS, F. JURIE, D. KEYSERS, M. KOSKELA, J. LAAKSONEN, D. LARLUS, B. LEIBE, H. MENG, H. NEY, B. SCHIELE, C. SCHMID, E. SEEMANN, J. SHAWE-TAYLOR, A. STORKEY, S. SZEDMAK, B. TRIGGS, I. ULUSOY, V. VIITANIEMI, AND J. ZHANG. The 2005 pascal visual object classes challenges. In *Selected Proc. of the First PASCAL Challenges Workshop*, 2006. [27](#)
- [35] W. FAN, S. J. STOLFO, J. ZHANG, AND P. K. CHAN. Adacost: Misclassification cost-sensitive boosting. In *Proc. Int. Conf. Mach. Learn.*, pages 97–105, San Francisco, CA, USA, 1999. [40](#), [41](#), [44](#), [99](#)
- [36] Y. FREUND. An adaptive version of the boost by majority algorithm. *Mach. Learn.*, **43**(3):293–318, 2001. [45](#), [138](#)
- [37] Y. FREUND AND R. SCHAPIRE. A decision-theoretic generalization of on-line learning and an application to boosting. *Comp. Learn. Theory*, **904**:23–37, 1995. [40](#)

## REFERENCES

---

- [38] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI. Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, **28**(2):337–407, 2000. [37](#), [43](#), [45](#), [87](#), [140](#)
- [39] M. M. GALLOWAY. Texture classification using gray level run lengths. *Computer Graph. and Image Proc.*, **4**:172–179, 1975. [10](#)
- [40] D. M. GAVRILA, J. GIEBEL, M. PERCEPTION, D. C. RES, AND G. ULM. Shape-based pedestrian detection and tracking. In *IEEE Intel. Vehicle Symp.*, pages 8–14, 2002. [19](#), [22](#)
- [41] D. M. GAVRILA AND S. MUNDER. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comp. Vis.*, **73**(1):41–59, 2007. [8](#), [9](#), [53](#)
- [42] D.M. GAVRILA. The visual analysis of human movement: A survey. *Comp. Vis. Image Understanding*, **73**(1):82–98, 1999. [8](#)
- [43] G. H. GOLUB AND C. VAN LOAN. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996. [84](#)
- [44] R. GONZALEZ AND R. WOODS. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., 2nd edition, 2002. [8](#), [10](#), [21](#)
- [45] H. GRABNER AND H. BISCHOF. On-line boosting and vision. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 260–267, Washington, DC, USA, 2006. IEEE Computer Society. [14](#), [108](#), [117](#), [126](#)
- [46] B. HAN, D. COMANICIU, Y. ZHU, AND L. S. DAVIS. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(7):1186–1197, 2008. [8](#), [9](#)
- [47] R. M. HARALICK, K. SHANMUGAM, AND I. H. DINSTEIN. Textural features for image classification. *IEEE Trans. Syst., Man, Cybern.*, **3**(6):610–621, 1973. [10](#)
- [48] I. HARITAOGLU, D. HARWOOD, AND L. S. DAVIS. W4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(8):809–830, 2000. [8](#), [9](#), [19](#), [21](#)

## REFERENCES

---

- [49] C. HARRIS AND M. STEPHENS. A combined corner and edge detector. In *4th Alvey Vis. Conf.*, pages 147–151, 1988. [11](#)
- [50] M. HEIKKILA AND M. PIETIKAINEN. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(4):657–662, 2006. [50](#)
- [51] D. HOIEM, A. A. EFROS, AND M. HEBERT. Putting objects in perspective. *Int. J. Comp. Vis.*, **80**(1):3–15, 2008. [9](#)
- [52] X. HOU, C. L. LIU, AND T. TAN. Learning boosted asymmetric classifiers for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 330–338, 2006. [41](#), [44](#)
- [53] C. HUANG, H. AI, Y. LI, AND S. LAO. High-performance rotation invariant multiview face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(4):671–686, 2007. [46](#), [49](#)
- [54] C. HUANG, H. AI, T. YAMASHITA, S. LAO, AND M. KAWADE. Incremental learning of boosted face detector. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1–8, Rio de Janeiro, 2007. [108](#)
- [55] S. JIN, D. S. YEUNG, AND X. WANG. Network intrusion detection in covariance feature space. *Pattern Recogn.*, **40**:2185–2197, 2007. [56](#)
- [56] T.-K. KIM, S.-F. WONG, B. STENGER, J. KITTLER, AND R. CIPOLLA. Incremental linear discriminant analysis using sufficient spanning set approximations. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, Minneapolis, 2007. [112](#)
- [57] R. KJELDSEN AND J. KENDER. Finding skin in color images. In *Int’l Conf. Automatic Face and Gesture Recog.*, pages 312–317, 1996. [8](#)
- [58] I. LAPTEV. Improvements of object detection using boosted histograms. In *British Mach. Vis. Conf.*, pages 949–958, Edinburgh, UK, 2006. [19](#), [26](#)
- [59] D. LEE. Effective gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(5):827–832, 2005. [8](#), [9](#)

## REFERENCES

---

- [60] H. LEE, A. BATTLE, R. RAINA, AND A. Y. NG. Efficient sparse coding algorithms. In *Proc. Adv. Neural Inf. Process. Syst.*, **19**, page 801, 2006. [139](#)
- [61] B. LEIBE AND B. SCHIELE. Scale-invariant object categorization using a scale-adaptive mean-shift search. *Pattern Recogn.*, **3175**:145–153, 2004. [25](#)
- [62] B. LEIBE, E. SEEMANN, , AND B. SCHIELE. Pedestrian detection in crowded scenes. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 878–885, San Diego, USA, 2005. [19](#), [25](#), [53](#)
- [63] J. LESKOVEC. Linear programming boosting for uneven datasets. In *Proc. Int. Conf. Mach. Learn.*, pages 456–463. AAI Press, 2003. [87](#), [99](#)
- [64] T. K. LEUNG, M. C. BURL, AND P. PERONA. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 637–644, 1995. [8](#)
- [65] K. LEVI AND Y. WEISS. Learning object detection from a small number of examples: The importance of good features. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, Washington, DC, 2004. [46](#), [48](#), [54](#), [132](#), [139](#)
- [66] M. S. LEWICKI AND T. J. SEJNOWSKI. Learning overcomplete representations. *Neural Computation*, **12**(2):337–365, 2000. [139](#)
- [67] S. Z. LI AND Z. ZHANG. Floatboost learning and statistical face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(9):1112–1123, 2004. [9](#), [41](#), [43](#), [46](#), [47](#), [80](#), [92](#), [94](#), [95](#), [98](#), [99](#)
- [68] R. LIENHART, A. KURANOV, AND V. PISAREVSKY. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Pattern Recogn.*, **2781**, pages 297–304, 2003. [41](#), [42](#), [46](#), [47](#), [139](#)
- [69] C. LIU AND H. Y. SHUM. Kullback-leibler boosting. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 587–594, 2003. [41](#), [43](#)
- [70] C. LIU AND H.-Y. SHUM. Kullback-Leibler boosting. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 587–594, Madison, June 2003. Wisconsin. [80](#)

## REFERENCES

---

- [71] X. LIU AND T. YU. Gradient feature selection for online boosting. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1–8, Rio de Janeiro, 2007. [108](#), [117](#)
- [72] D. G. LOWE. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, **60**(2):91–110, 2004. [8](#), [11](#), [24](#), [25](#), [50](#)
- [73] H. LUO. Optimization design of cascaded classifiers. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **1**, pages 480–485, 2005. [34](#), [36](#)
- [74] Y. MA AND X. DING. Robust real-time face detection based on cost-sensitive adaboost method. In *Proc. of Int. Conf. on Multimedia and Expo*, pages 465–468, 2003. [41](#), [43](#), [44](#)
- [75] J. MAIRAL, F. BACH, J. PONCE, G. SAPIRO, AND A. ZISSERMAN. Discriminative learned dictionaries for local image analysis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, 2008. [139](#)
- [76] S. G. MALLAT. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **11**(7):674–693, 1989. [10](#), [138](#)
- [77] D. MARR AND E. HILDRETH. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sci.*, **207**(1167):187–217, 1980. [12](#)
- [78] H. MASNADI-SHIRAZI AND N. VASCONCELOS. Asymmetric boosting. In *Proc. Int. Conf. Mach. Learn.*, pages 609–619, 2007. [41](#), [44](#)
- [79] J. MATAS, O. CHUM, M. URBAN, AND T. PAJDLA. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comp.*, **22**(10):761–767, 2004. [11](#), [12](#)
- [80] J. MEYNET, V. POPOVICI, AND J.-P. THIRAN. Face detection with boosted Gaussian features. *Pattern Recogn.*, **40**(8):2283–2291, 2007. [60](#)
- [81] K. MIKOLAJCZYK, B. LEIBE, AND B. SCHIELE. Multiple object class detection with a generative model. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 36–43, 2006. [19](#), [26](#)



- 
- [82] K. MIKOLAJCZYK AND C. SCHMID. An affine invariant interest point detector. In *Proc. Eur. Conf. Comp. Vis.*, pages 128–142, 2002. [12](#)
- [83] K. MIKOLAJCZYK AND C. SCHMID. Scale & affine invariant interest point detectors. *Int. J. Comp. Vis.*, **60**(1):63–86, 2004. [12](#)
- [84] K. MIKOLAJCZYK, C. SCHMID, AND A. ZISSERMAN. Human detection based on a probabilistic assembly of robust part detectors. In *Proc. Eur. Conf. Comp. Vis.*, **1**, pages 69–81, Prague, Czech Republic, May 2004. [19](#), [23](#)
- [85] T. MITA, T. KANEKO, B. STENGER, AND O. HORI. Discriminative feature co-occurrence selection for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(7):1257–1269, 2008. [138](#)
- [86] B. MOGHADDAM, Y. WEISS, AND S. AVIDAN. Generalized spectral bounds for sparse lda. In *Proc. Int. Conf. Mach. Learn.*, pages 641–648, New York, NY, USA, 2006. ACM. [82](#), [83](#)
- [87] B. MOGHADDAM, Y. WEISS, AND S. AVIDAN. Fast pixel/part selection with sparse eigenvectors. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1–8, 2007. [83](#), [84](#), [105](#), [109](#), [115](#), [122](#), [136](#)
- [88] A. MOHAN, C. PAPAGEORGIOU, T. POGGIO, K. COMMUN, AND R. CITY. Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**(4):249–361, 2001. [9](#)
- [89] S. MUNDER AND D. M. GAVRILA. An experimental study on pedestrian classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(11):1863–1868, 2006. [9](#), [13](#), [54](#), [62](#), [72](#), [76](#), [103](#), [104](#), [136](#)
- [90] H. MURASE AND S. K. NAYAR. Visual learning and recognition of 3-d objects from appearance. *Int. J. Comp. Vis.*, **14**(1):5–24, 1995. [11](#)
- [91] S. NADIMI AND B. BHANU. Physical models for moving shadow and object detection in video. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(8):1079–1087, 2004. [79](#)

## REFERENCES

---

- [92] B. NI, A. A. KASSIM, AND S. WINKLER. A hybrid framework for 3D human motion tracking. *IEEE Trans. Circuits Syst. Video Technol.*, **18**(8):1075–1084, 2008. [79](#)
- [93] K. NIGAM, A. K. MCCALLUM, S. THRUN, AND T. MITCHELL. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, **39**(2):103–134, 2000. [140](#)
- [94] C. L. NOVAK AND S. A. SHAFER. Anatomy of a color histogram. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 599–605, 1992. [10](#)
- [95] T. OJALA, M. PIETIKAINEN, AND T. MAENPAA. Gray scale and rotation invariant texture classification with local binary patterns. In *Proc. Eur. Conf. Comp. Vis.*, pages 404–420, 2000. [55](#)
- [96] T. OJALA, M. PIETIKINEN, AND T. MENP. Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(7):971–987, 2002. [10](#), [50](#), [139](#)
- [97] N. M. OLIVER, B. ROSARIO, AND A. P. PENTLAND. A bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(8):831–843, 2000. [8](#), [9](#)
- [98] B. A. OLSHAUSEN AND D. J. FIELD. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vis. Res.*, **37**(23):3311–3325, 1997. [139](#)
- [99] N. C. OZA AND S. RUSSELL. Online bagging and boosting. In *Proc. Artificial Intell. & Statistics*, pages 105–112. Morgan Kaufmann, 2001. [108](#), [121](#), [123](#)
- [100] S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. Circuits Syst. Video Technol.*, **18**(8):1140–1151, 2008. [15](#), [79](#), [103](#), [105](#)
- [101] S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Efficiently training a better visual detector with sparse eigenvectors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, Miami, Florida, June 2009. [15](#), [124](#), [125](#), [126](#)

## REFERENCES

---

- [102] S. PAISITKRIANGKRAI, C. SHEN, AND J. ZHANG. Incremental training of a detector using online sparse eigen-decomposition. *IEEE Trans. Image Process.*, **20**(1):213–226, 2011. [15](#)
- [103] S. PANG, S. OZAWA, AND N. KASABOV. Incremental linear discriminant analysis for classification of data streams. *IEEE Trans. Syst., Man, Cybern. B*, **35**(5):905–914, 2005. [109](#), [112](#)
- [104] C. PAPAGEORGIOU AND T. POGGIO. A trainable system for object detection. *Int. J. Comp. Vis.*, **38**(1):15–33, 2000. [9](#), [19](#), [20](#), [27](#), [53](#), [107](#)
- [105] T. PARAG, F. PORIKLI, AND A. ELGAMMAL. Boosting adaptive linear weak classifiers for online learning and tracking. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, Anchorage, 2008. [108](#), [117](#)
- [106] T. PAVLIDIS. *Structural Pattern Recognition*. Springer Verlag, 1977. [11](#)
- [107] A. PENTLAND. Fractal-based description of natural scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, **6**(6):661–674, 1984. [10](#)
- [108] D. S. PHAM AND S. VENKATESH. Joint learning and dictionary construction for pattern recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, 2008. [139](#)
- [109] M. T. PHAM AND T. J. CHAM. Fast training and selection of haar features using statistics in boosting-based face detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1–7, 2007. [46](#), [49](#), [51](#), [80](#), [108](#)
- [110] M.-T. PHAM AND T.-J. CHAM. Online learning asymmetric boosted classifiers for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **0**, pages 1–8, 2007. [14](#), [41](#), [45](#), [96](#)
- [111] M. T. PHAM, V. D. D. HOANG, AND T. J. CHAM. Detection with multi-exit asymmetric boosting. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, Alaska, US, 2008. [34](#), [36](#), [39](#), [80](#), [132](#), [137](#)
- [112] R. POPPE. Vision-based human motion analysis: An overview. *Comp. Vis. Image Understanding*, **108**(1–2):4–18, 2007. [8](#)

## REFERENCES

---

- [113] J. R. QUINLAN. Induction of decision trees. *Mach. Learn.*, **1**(1):81–106, 1986. [28](#)
- [114] K. R. RAO AND P. YIP. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, 1990. [139](#)
- [115] C. E. RASMUSSEN AND C. K. I. WILLIAMS. *Gaussian Processes for Machine Learning*. MIT Press, 2006. [121](#), [122](#)
- [116] B. RIPLEY. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996. [2](#)
- [117] S. ROMDHANI, P. TORR, B. SCHÖLKOPF, AND A. BLAKE. Computationally efficient face detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, **2**, pages 695–700, Vancouver, 2001. [79](#), [80](#)
- [118] C. ROSENBERG, M. HEBERT, AND H. SCHNEIDERMAN. Semi-supervised self-training of object detection models. In *IEEE Workshop on App. of Computer Vision*, **1**, pages 29–36, 2005. [140](#)
- [119] D. ROTH. The snow learning architecture. Technical report, University of Illinois at Urbana-Champaign, 1999. [22](#)
- [120] H. A. ROWLEY, S. BALUJA, AND T. KANADE. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(1):23–38, 1998. [19](#), [20](#)
- [121] H. A. ROWLEY, S. BALUJA, AND T. KANADE. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(1):23–38, 1998. [79](#)
- [122] L. G. RUEDA. An efficient approach to compute the threshold for multi-dimensional linear classifiers. *Pattern Recogn.*, **37**(4):811–826, 2004. [116](#)
- [123] R. E. SCHAPIRE. Theoretical views of boosting and applications. In *Proc. Int. Conf. Algorithmic Learn. Theory*, pages 13–25, London, UK, 1999. [30](#), [87](#)
- [124] R.E. SCHAPIRE AND Y. SINGER. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, **37**(3):297–336, 1999. [40](#), [43](#)

## REFERENCES

---

- [125] V. SHARMA AND J. DAVIS. Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians. In *Proc. IEEE Int. Conf. Comp. Vis.*, Rio de Janeiro, Brazil, 2007. [53](#)
- [126] C. SHEN, S. PAISITKRIANGKRAI, AND J. ZHANG. Face detection from few training examples. In *Proc. IEEE Int. Conf. Image Process.*, pages 2764–2767, 2008. [79](#), [132](#)
- [127] C. SHEN, S. PAISITKRIANGKRAI, AND J. ZHANG. Efficiently learning a detection cascade with sparse eigenvectors. *IEEE Trans. Image Process.*, **20**(1):22–35, 2010. [15](#)
- [128] J. SHERMAN AND W. J. MORRISON. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist.*, **21**(1):124–127, 1950. [115](#)
- [129] H. SIDENBLADH. Detecting human motion with support vector machines. In *Proc. IEEE Int. Conf. Patt. Recogn.*, 2004. [19](#), [23](#)
- [130] J. SOCHMAN AND J. MATAS. Waldboost - learning for time constrained sequential detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, pages 150–156, 2005. [34](#), [35](#), [132](#)
- [131] J. SUN, J. M. REHG, AND A. BOBICK. Automatic cascade training with perturbation bias. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, pages 276–283, Washington, DC, USA, 2004. [34](#), [35](#), [86](#)
- [132] W. H. TSAI AND S. S. YU. Attributed string matching with merging for shape recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **7**(4):453–462, 1985. [11](#)
- [133] M. TURK AND A. PENTLAND. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, **3**(1):71–86, 1991. [11](#), [18](#), [19](#)
- [134] O. TUZEL, F. PORIKLI, AND P. MEER. Region covariance: A fast descriptor for detection and classification. In *Proc. Eur. Conf. Comp. Vis.*, **2**, pages 589–600, Graz, Austria, May 2006. [55](#), [56](#)

## REFERENCES

---

- [135] O. TUZEL, F. PORIKLI, AND P. MEER. Human detection via classification on Riemannian manifolds. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, Minneapolis, MN, 2007. 103, 105
- [136] O. TUZEL, F. PORIKLI, AND P. MEER. Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(10):1713–1727, 2008. 13, 19, 27, 54, 55, 63, 69, 70, 71, 72, 78, 136, 139
- [137] I. ULUSOY AND C. M. BISHOP. Generative versus discriminative methods for object recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, pages 258–265, 2005. 12
- [138] V. VAPNIK. *The nature of statistical learning theory*. Statistics for Engineering and Information Science. Springer Verlag, Berlin, 2000. 140
- [139] P. VIOLA AND M. J. JONES. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1311–1318. MIT Press, 2002. 40, 41, 42, 44, 45, 79, 80, 81, 86, 87, 90, 95, 96, 98, 99, 100, 124, 125, 126
- [140] P. VIOLA AND M. J. JONES. Robust real-time face detection. *Int. J. Comp. Vis.*, **57**(2):137–154, 2004. 9, 11, 12, 13, 14, 17, 19, 22, 26, 27, 32, 33, 43, 47, 53, 54, 55, 59, 61, 69, 76, 79, 80, 84, 86, 90, 92, 94, 95, 98, 100, 107, 116, 124, 125, 126, 127, 129, 131, 136, 139
- [141] P. VIOLA, M. J. JONES, AND D. SNOW. Detecting pedestrians using patterns of motion and appearance. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2003. 19, 22, 53, 55, 61
- [142] PASCAL VOC. The PASCAL visual object classes challenge (VOC 2007). <http://www.pascal-network.org/challenges/VOC/voc2007/index.html>. 69, 124
- [143] A. R. WEBB AND D. LOWE. The optimised internal representation of multi-layer classifier networks performs nonlinear discriminant analysis. *IEEE Trans. Neural Netw.*, **3**(4):367–375, 1990. 81

## REFERENCES

---

- [144] B. WU AND R. NEVATIA. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *Proc. IEEE Int. Conf. Comp. Vis.*, **1**, pages 90–97, Beijing, China, 2005. [46](#), [48](#), [53](#), [139](#)
- [145] J. WU, S. C. BRUBAKER, M. D. MULLIN, AND J. M. REHG. Fast asymmetric learning for cascade face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(3):369–382, 2008. [51](#), [80](#), [81](#), [86](#), [89](#), [91](#), [95](#), [97](#), [98](#), [100](#), [108](#), [131](#)
- [146] R. XIAO, H. ZHU, H. SUN, AND X. TANG. Dynamic cascades for face detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, Rio de Janeiro, 2007. [34](#), [37](#), [108](#), [137](#)
- [147] S. YAN, S. SHAN, X. CHEN, W. GAO, AND J. CHEN. Matrix-structural learning (msl) of cascaded classifier from enormous training set. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–7, 2007. [34](#), [37](#), [38](#)
- [148] G. YANG AND T. S. HUANG. Human face detection in complex background. *Pattern Recogn.*, **27**(1):53–63, 1994. [7](#)
- [149] J. YANG, J. WRIGHT, T. HUANG, AND Y. MA. Image super-resolution as sparse representation of raw image patches. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, 2008. [139](#)
- [150] M-H. YANG, D. J. KRIEGMAN, AND N. AHUJA. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(1):34–58, 2002. [7](#), [8](#)
- [151] J. YE, Q. LI, H. XIONG, H. PARK, R. JANARDAN, AND V. KUMAR. Idr/qr: An incremental dimension reduction algorithm via qr decomposition. *IEEE Trans. Knowl. Data Eng.*, **17**(9):1208–1222, 2005. [109](#), [112](#)
- [152] T. ZHANG. Multi-stage convex relaxation for learning with sparse regularization. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1929–1936, 2008. [84](#)
- [153] G. ZHAO AND M. PIETIKAINEN. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(6):915–928, 2007. [50](#)

## REFERENCES

---

- [154] H. ZHAO AND P. C. YUEN. Incremental linear discriminant analysis for face recognition. *IEEE Trans. Syst., Man, Cybern. B*, **38**(1):210–221, 2008. [109](#)
- [155] T. ZHAO AND R. NEVATIA. Tracking multiple humans in complex situations. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(9):1208–1221, 2004. [9](#)
- [156] J. ZHOU AND J. HOANG. Real time robust human detection and tracking system. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 149–149, 2005. [19](#), [25](#)
- [157] Q. ZHU, S. AVIDAN, M. YEH, AND K.-T. CHENG. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, **2**, pages 1491–1498, New York, 2006. [19](#), [26](#), [139](#)