Feature and Classifier Development for Human Detection

Amit Satpathy

School of Electrical & Electronic Engineering

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

2014
I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.................. ....................
Date Amit Satpathy
To my family and friends for their encouragements and love.
Acknowledgements

I would like to express my deepest gratitude to my supervisors, Assoc Prof Xudong Jiang and Dr How-Lung Eng, for their many suggestions and constant support during this research. Their suggestions and insights helped to shape the research and results presented in this thesis.

I would also like to thank my scholarship sponsor, Agency for Science, Technology and Research (A*STAR), for providing me with the financial support for my PhD studies at Nanyang Technological University for the first 4 years of my graduate research. They have, at every opportunity, taken the pains to help me with my studies in any way possible. For this, I am extremely grateful.

I would also like to express my heartfelt thanks to the various professors who have taught me in the courses throughout my undergraduate and graduate years. Their knowledge passed on to me has helped to grow and mould my research in various ways.

During the years of my studies, I have been in the company of wonderful people in my laboratories in Nanyang Technological University and Institute for Infocomm Research. I have enjoyed spending my time with Miao Zhenwei, Chen Changsheng, Li Sheng, Liu Siyuan, Cao Hong, Ren Jianfeng, Geng Cong and Tan Guoxian. Their comments and insights for my research were very much welcomed and appreciated. In particular, I am indebted to Stewart Chu who has been extremely helpful with pro-
Acknowledgements

Providing me with various resources for my research. In Institute for Infocomm Research, I am thankful for all the technical help I received from my colleagues, namely Dr Lim Joo Hwee, Dr Panachit, Chew Boon Fong, Myo Thida and Chris Wirianto.

I would also like to thank Dr Piotr Dollar from Microsoft Research and Dr Xiaoyu Wang from NEC Labs America for their assistance in understanding their work and publications which has been critical for my research.

Last but not least, I am grateful to my parents, brother, and friends for their patience, support and love.
Abstract

Computer vision has been gaining increasing popularity in this age of automation and advancing technology. The desire to automate a number of labour-intensive tasks and reduce dependence on humans has contributed to this. In particular, human detection has been looked into quite thoroughly in the recent years. Human detection is important in areas of surveillance, pedestrian detection, human-computer interaction etc where humans need to be detected very accurately. With accurate detection, post-processing of human actions can then be done effectively. For accurate detection, it is necessary that the features and classification framework adopted are robust and effective.

In this thesis, we propose several features for human detection that make use of gradient and/or texture information present in still images for representation and a non-linear classification framework for human detection. We present comprehensive results to validate our proposed features and classification framework.

We first investigate the limitations of Histogram of Oriented Gradients and Histogram of Gradients which are popular features used in human detection. Histogram of Gradients distinguishes a dark human from a bright background and vice versa. This increases the intra-class variations of humans. For the purpose of detection, this variation is insignificant and undesirable. Histogram of Oriented Gradients solves this issue of Histogram of Gradients by considering unsigned gradients which treats all
Abstract

gradients of opposite directions as gradients of a same orientation. However, for the same cell, Histogram of Oriented Gradients maps gradients of opposite directions to the same bin in a histogram. This causes some different structures to have the same feature representation. Analyzing these limitations, we propose Extended Histogram of Gradients.

Extended Histogram of Gradients is a concatenation of 2 histograms derived from Histogram of Gradients. The first histogram is Histogram of Oriented Gradients which, in our work, is observed to be the sum of 2 halves of Histogram of Gradients. The second histogram is the Difference of Histogram of Gradient which is obtained by considering the absolute difference between the bins and their corresponding opposite direction bins of Histogram of Gradients. The concatenation of the 2 histograms produces the proposed Extended Histogram of Gradients. In addition to the proposed feature, we also propose an alternative normalization scheme for Histogram of Oriented Gradients and Extended Histogram of Gradients. In our work, we find that the default normalization scheme for Histogram of Oriented Gradients which uses clipped L2 normalization in the last step causes some different patterns to be similarly represented after clipping. This reduces the discriminative capabilities of the Histogram of Oriented Gradients. Furthermore, we also find that using the default normalization scheme does not effectively suppress any noisy gradient pixels with large magnitudes or abrupt intensity changes in the image for Extended Histogram of Gradients as typically, only the noise in the Histogram of Oriented Gradients is suppressed while the noise remains unsuppressed in the Difference of Histogram of Gradients. In this dissertation, we propose an alternative normalization scheme where clipped L2 normalization is first performed on Histogram of Gradients. The Histogram of Oriented Gradients and Extended Histogram of Gradients features are then computed from the normalized Histogram of Gradients.
Local Binary and Ternary Patterns are another set of features that are highly used in texture classification and face detection. However, their applications in human detection are limited as they also differentiate a bright human against a dark background and vice versa which increases the intra-class variations of humans. Different objects have different shapes and textures. It is therefore desirable to represent objects using both texture and edge information. In order to be robust to illumination and contrast variations, Local Binary and Ternary Patterns do not differentiate between a weak contrast local pattern and a similar strong one. They only capture the object texture information. Object contours, which also contain discriminatory information, tend to be situated in strong contrast regions. Therefore, by totally discarding contrast information, the object contour may not be effectively represented by these descriptors.

In this thesis, we address these issues of Local Binary and Ternary Patterns in human detection and propose new features, Discriminative Robust Local Binary and Ternary Patterns. The proposed features are a concatenation of 2 histograms - Robust Local Binary/Ternary Patterns and Difference of Local Binary/Ternary Patterns. The first histogram is obtained by summing the 2 halves of Local Binary/Ternary Pattern histograms. This histogram alleviates the intensity reversal problem of object and background. However, by doing so, there are some patterns that are misrepresented as the codes and their complements are merged in the same block to the same histogram bin. Hence, the second histogram, Difference of Local Binary/Ternary Patterns, is proposed which takes the absolute difference between the bins and their corresponding complement bins. By concatenating the 2 histograms, the misrepresentation of patterns is resolved. In addition, the proposed features do not completely ignore the contrast information of image patterns as the histograms are weighted by the gradient magnitudes at the corresponding pixels. The proposed features contain both edge and texture information which is desirable for object recognition.
Existing classification frameworks use Support Vector Machines or boosting-based classifiers for classification. We examine limitations of such frameworks in this dissertation and also analyze the distribution of features in the feature space to determine an appropriate boundary for classification. As such, it is discovered that a hyper-quadratic boundary is appropriate for classification. Therefore, we propose a classification framework that includes a modified Minimum Mahalanobis Distance classifier and Asymmetric Principal Component Analysis for dimensionality reduction. For high-dimensional features, the estimated eigenvalues in some feature dimensions deviate greatly from that of the data population which results in overfitting of the Minimum Mahalanobis Distance classifier. Hence, there is a need to reduce feature dimensions to minimize the overfitting problem. Furthermore, training sets, usually, contain much fewer positive samples than the negative ones which results in the negative covariance matrix being more reliable than the positive covariance matrix. Using Principal Component Analysis is inefficient as the unreliable dimensions from the less reliable covariance matrix are not effectively removed. To tackle the problems of dimensionality reduction and the asymmetry issue of human training sets, we propose using Asymmetric Principal Component Analysis for dimension reduction. As a result, the projected features allow for a more robust classifier to be trained that less overfits the training data.

We extend the use of our features to general visual object detection and image classification to demonstrate that the application is not only limited to human detection. Furthermore, using Extended Histogram of Gradients, we demonstrate that parts-based modeling improves the detection performance.
Contents

Acknowledgements i

Abstract iii

List of Figures x

List of Tables xiii

List of Acronyms xiv

1 Introduction 1
   1.1 Motivation ......................................................... 1
   1.2 Objectives .......................................................... 3
   1.3 Major Contributions of the Thesis .............................. 5
   1.4 Thesis Organization ................................................ 6

2 Literature Review 8
   2.1 Feature Extraction .................................................. 11
      2.1.1 Sparse Feature Representation ............................. 11
      2.1.2 Dense Feature Representation ............................. 13
      2.1.3 Other Feature Representations ............................. 21
   2.2 Classification Framework ......................................... 23
   2.3 Discussion .......................................................... 26

3 Extended Histogram Of Gradients 30
   3.1 Histogram of Gradients and Histogram of Oriented Gradients 30
      3.1.1 Descriptor formation ....................................... 30
3.1.2 Limitations of Histogram of Gradients and Histogram of Oriented Gradients .............................. 33
3.2 Extended Histogram of Gradients ............................................. 37
3.3 Overall Results ................................................................. 44
  3.3.1 Training of classifiers ..................................................... 45
  3.3.2 Performance comparison of ExHoG against Hog and Hg ........... 45
  3.3.3 Results on INRIA Data Set ............................................. 46
  3.3.4 Results on Caltech Pedestrian Detection Benchmark Data Set .. 47
3.4 Discussion ................................................................. 52

4 Discriminative Robust Local Binary and Ternary Patterns 54
  4.1 Local Binary Pattern, Local Ternary Pattern and Robust Local Binary Pattern ........................................ 55
     4.1.1 Descriptor formation .................................................. 55
     4.1.2 Limitations of LBP, LTP and RLBP ................................ 58
  4.2 Discriminative Robust Local Binary and Ternary Patterns .......... 60
     4.2.1 The Proposed Discriminative Robust Local Binary Pattern .... 60
     4.2.2 The Proposed Discriminative Robust Local Ternary Pattern ... 64
       4.2.2.1 Robust Local Ternary Pattern .................................. 64
       4.2.2.2 Discriminative Robust Local Ternary Patterns ............... 67
     4.2.3 Efficient computation of DRLTP using ULBP and LLBP .......... 70
  4.3 Overall Results ............................................................. 76
     4.3.1 Performance comparison of DRLBP and DRLTP against LBP, LTP and RLBP ............................................. 77
     4.3.2 Results on INRIA Data Set .......................................... 78
     4.3.3 Results on Caltech Data Set ........................................ 79
  4.4 Discussion ................................................................. 85

5 Quadratic Classification Framework for Human Detection in Subspace 86
  5.1 Popular Classifiers ......................................................... 87
    5.1.1 Support Vector Machines ............................................. 87
    5.1.2 Boosting-based Classifiers ......................................... 89
  5.2 Proposed Classification Framework ..................................... 90
    5.2.1 Dimensionality Reduction ........................................... 92
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Shapes of humans</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Intra-class variations exhibited by humans and varying external conditions</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>A typical human detection system</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Articulated body model with its 14 joints and 15 body parts in [93]</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Examples of rectangle filters used by Viola et al.</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Histogram of Oriented Gradients by Dalal et al.</td>
<td>16</td>
</tr>
<tr>
<td>2.5</td>
<td>Pose invariant method by Lin et al.</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Feature pyramid by Felzenszwalb et al.</td>
<td>19</td>
</tr>
<tr>
<td>2.7</td>
<td>Hierarchical holistic shape templates of humans used in [32]</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>Histogram of Gradients Computation</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Flowchart of Hg computation procedure, training and testing.</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Problem of Histogram of Gradients (Hg) and its solution by Histogram of Oriented Gradients (Hog)</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Problem of Histogram of Oriented Gradients (Hog)</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Example of 2 structures that have similar Histogram of Oriented Gradients (Hog)</td>
<td>39</td>
</tr>
<tr>
<td>3.6</td>
<td>Problem of normalization directly on Extended Histogram of Gradients (ExHoG)</td>
<td>40</td>
</tr>
<tr>
<td>3.7</td>
<td>Effect of the proposed normalization scheme on the 2 structures in Fig. 3.5</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Proposed normalization scheme for ExHoG</td>
<td>42</td>
</tr>
<tr>
<td>3.9</td>
<td>ExHoG representations of local structures in Fig. 3.4</td>
<td>43</td>
</tr>
<tr>
<td>3.10</td>
<td>Same ExHoGs are produced for patterns in Fig. 3.4</td>
<td>44</td>
</tr>
<tr>
<td>3.11</td>
<td>Performance of ExHoG against Hog using different normalization schemes</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Figures

3.12 Performance of ExHoG against Hog and Hg ......................... 48
3.13 Performance of ExHoG against existing approaches on INRIA .......... 49
3.14 Performance under different conditions on test set of the Caltech Pedestrian Data Set ......................................................... 50
3.15 Detection results of ExHoG on various data sets used in the experiments 53

4.1 Problem of Local Binary Pattern (Lbp) ........................................ 59
4.2 Problem of Robust Lbp (RLbp) .................................................. 60
4.3 DRLbp representations of local structures in Fig. 4.2 .................... 63
4.4 Same DRLbps are produced for patterns in Fig. 4.1 ....................... 64
4.5 Illustration of “upper” and “lower” Lbp codes of Ltp for 2 situations .. 65
4.6 Problem of Local Ternary Pattern (Ltp) ....................................... 68
4.7 Problem of Robust Ltp (RLtp) .................................................. 69
4.8 DRLtp representations of local structures in Fig. 4.7 .................... 71
4.9 Same DRLtts are produced for patterns in Fig. 4.6 ....................... 72
4.10 Performance of DRLbp and DRLtp with/without weighting against Lbp, Ltp and RLbp ................................................................. 79
4.11 Performance of DRLbp and DRLtp against existing approaches on INRIA 80
4.12 Performance under different conditions on test set of the Caltech Pedestrian Data Set ......................................................... 81
4.13 Detection results of DRLbp on various data sets used in the experiments 83
4.14 Detection results of DRLtp on various data sets used in the experiments 84

5.1 Distribution of ExHoG features in the APCA subspace .................. 91
5.2 Cross-validation results of different parameter for APCA ................. 97
5.3 Performance of ExHoG APCA against existing approaches on INRIA .. 100
5.4 Performance under different conditions on test set of the Caltech Pedestrian Data Set ......................................................... 101
5.5 Detection results of ExHoG APCA against ExHoG on INRIA .......... 104
5.6 Detection results of ExHoG APCA against ExHoG on Caltech .......... 105

6.1 Detection results of DRLtp, DRLbp and ExHoG on various data sets used in the experiments .......................................................... 112
6.2 Examples of some classes in Caltech 101 for category classification ... 113
6.3 Examples of some classes in Caltech 256 for category classification ... 115
6.4 PCA of ExHoG features ............................................................ 117
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>One-component model with 8 parts learned using INRIA</td>
<td>120</td>
</tr>
<tr>
<td>6.6</td>
<td>Performance of ExHoG LatSvm-V2 on INRIA</td>
<td>121</td>
</tr>
<tr>
<td>6.7</td>
<td>Detection results of ExHoG LatSvm-V2 on INRIA</td>
<td>123</td>
</tr>
<tr>
<td>6.8</td>
<td>Three-component models with 8 parts learned</td>
<td>126</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Comparison of Most Common Human Detectors ....................... 27
4.1 Number of hard negatives and feature dimensionality of each feature. 76
5.1 Performance of ExHoG against Hog and/or Hg on INRIA. ............. 98
5.2 Summary of Runtime Performance of ExHoG (INRIA). .................. 103
6.1 Recognition Accuracy at Equal Error Rate on UIUC Cars data set .... 110
6.2 Performance on Caltech 101 data set. ................................. 111
6.3 Performance on Caltech 256 data set. ................................. 114
6.4 Summary of PASCAL VOC 2007 Results. .............................. 124
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APCA</td>
<td>Asymmetric Principal Component Analysis</td>
</tr>
<tr>
<td>DA</td>
<td>Discriminant Analysis</td>
</tr>
<tr>
<td>DHG</td>
<td>Difference of Histogram of Gradients</td>
</tr>
<tr>
<td>DLBP</td>
<td>Difference of Local Binary Pattern</td>
</tr>
<tr>
<td>DLTP</td>
<td>Difference of Local Ternary Pattern</td>
</tr>
<tr>
<td>DRLBP</td>
<td>Discriminative Robust Local Binary Pattern</td>
</tr>
<tr>
<td>DRLTP</td>
<td>Discriminative Robust Local Ternary Pattern</td>
</tr>
<tr>
<td>ExHoG</td>
<td>Extended Histogram of Gradients</td>
</tr>
<tr>
<td>EER</td>
<td>Equal Error Rate</td>
</tr>
<tr>
<td>FPPI</td>
<td>False Positive Per Image</td>
</tr>
<tr>
<td>FPPW</td>
<td>False Positive Per Window</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Oriented Gradients</td>
</tr>
<tr>
<td>HG</td>
<td>Histogram of Gradients</td>
</tr>
<tr>
<td>LAMR</td>
<td>Log-Average Miss Rate</td>
</tr>
<tr>
<td>LBP</td>
<td>Local Binary Pattern</td>
</tr>
<tr>
<td>LTP</td>
<td>Local Ternary Pattern</td>
</tr>
<tr>
<td>MMDC</td>
<td>Minimum Mahalanobis Distance classifier</td>
</tr>
<tr>
<td>MR</td>
<td>Miss Rate</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>RLBp</td>
<td>Robust Local Binary Pattern</td>
</tr>
<tr>
<td>RLTP</td>
<td>Robust Local Ternary Pattern</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Computer vision and machine intelligence has been a major area of focus for researchers and engineers alike ever since the invention of computers to aid humans in their daily lives. Over recent years, researchers have been working on substituting humans with computers to take over the labour-intensive and time-consuming tasks that humans perform from day-to-day. A key area in this work has been finding ways for computers to analyze and make decisions on the observations made. This is something that humans do everyday with little difficulty. The human visual system is able to easily discriminate objects with large intra-class variations and make decisions. However, computers are still far behind humans with respect to this ability.

One of the key fields in the work is the ability to detect various different objects and to classify them from images and videos. Particularly, the detection of humans has picked up intensively over the last 10 years. This is partly due to the September 11, 2001 incidents of the World Trade Center and the Pentagon. In both incidents, there were lapses where the responsible individuals escaped being noticed by the relevant law enforcement agencies due to human errors. Hence, in addition to face recogni-
Chapter 1. Introduction

Figure 1.1: Shapes of humans which enable discrimination of humans from other objects. The distinctive “Ω”-shape formed by the head and shoulders is easily discernible from the background. Note, in the columns on the right, the shapes of humans, even in an edge (binary) representation, enables discrimination of humans from the non-humans.

Law enforcement agencies have been looking into automated video surveillance systems where they can monitor suspicious human behaviour and raise alarms if necessary to prevent such incidents from happening again. Furthermore, the rise in vehicular accidents involving pedestrians has also been pushing the research in this area. Pedestrian detection systems in vehicles are being looked into so as to minimize such accidents from happening. Several other areas of applications have also spurred the interest of human detection and are listed as follows:

- Human-Computer Interaction for video games, interactive media et cetera
1.2 Objectives

- Robotics
- Analysis of digital media content for forgery detection

In order to describe humans so that computers can differentiate them from non-humans, feature-based approaches are normally adopted that describe the shape of humans. Humans have a characteristic shape that differentiates them from most other objects (Fig. 1.1). For example, the head and shoulders of the humans form an “Ω”-shape feature which is discriminative. Together with other regions of humans, the contour describing humans contains descriptive information which feature-based approaches try to represent. The computer is trained to recognize patterns characterized by the features and to perform classification based on the representation of these features.

Yet, human detection is an intrinsic and challenging problem due to the huge intra-class variation exhibited by humans due to colour, different clothing, pose and appearance. Furthermore, external conditions such as partial occlusions, illumination or background clutter further compound the problem (Fig. 1.2).

Such problems have driven and motivated researchers to investigate and develop various features and classifiers to tackle the problem of human detection.

1.2 Objectives

In this dissertation, the work focuses on the problem of human detection in images. Human detection consist of 2 blocks: a feature extraction component that uses information present in image regions to create feature vectors, and a detector that uses the created features to provide human/non-human decisions. In particular, we focus on the feature extraction component as existing features have their limitations. In our
Chapter 1. Introduction

Figure 1.2: Intra-class variations exhibited by humans and varying external conditions that compounds the human detection and classification problem. In the images, it can be observed how the detection of humans can be affected by the pose variations, clothings, illuminations, occlusions et cetera.

work, we develop better and more robust features for the detection of humans from the perspective of computer vision where detectors scan images or videos for instances of humans and localizes them. We analyze existing state-of-the-art features and their limitations. Our proposed new features alleviates these limitations and improves human detection accuracy in existing benchmark data sets.

Furthermore, we analyze existing classification frameworks. Existing human de-
1.3 Major Contributions of the Thesis

The major contributions of this thesis are:

- A new gradient-based feature, Extended Histogram of Gradients (ExHoG). ExHoG addresses the issues identified in Histogram of Oriented Gradients [9] and Histogram of Gradients [69]. This proposed feature outperforms current state-of-the-art single type of features on existing human benchmark data sets.

- A new set of edge-texture based features, Discriminative Robust Local Binary Pattern (DRLbp) and Discriminative Robust Local Ternary Pattern (DRLtp). These features address the issues observed in Local Binary Pattern (Lbp) and Local Ternary Pattern (Ltp) descriptors along with some of their recent derivatives. These set of features outperform current state-of-the-art single type of features on existing human benchmark data sets.

- A full quadratic classification framework, using ExHoG as example, which includes a modified Minimum Mahalanobis Distance classifier and Asymmetric
Principal Component Analysis [51, 52]. Here, we address the asymmetry issue in human training sets and the linear versus non-linear classification of humans. Our results show that by considering these issues, we can obtain a better detection accuracy in existing benchmark data sets.

- Applications of the proposed features to general object detection applications and parts-based modeling using ExHoG for enhanced detection.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 presents a discussion of existing state-of-the-art human detection methods and discussions.

- Chapter 3 describes the issues with the gradient-based Histogram of Oriented Gradients and Histogram of Gradients and presents the proposed feature, Extended Histogram of Gradients.

- Chapter 4 discusses limitations of texture-based Local Binary Patterns and Local Ternary Patterns as well as some of their derivatives. 2 sets of new features, Discriminative Robust Local Binary Pattern and Discriminative Robust Local Ternary Pattern, are described.

- Chapter 5 presents the full quadratic classification framework using the proposed features from Chapter 3. Here, the asymmetry of human training sets and the need for non-linear classification are discussed.

- Chapter 6 discusses the application of the proposed features to general object detection problems using several object detection data sets. Parts-based represen-
1.4. Thesis Organization

tation of Extended Histogram of Gradients is also discussed for enhanced robust detection.

• Chapter 7 summarizes the discussions and research contributions of this thesis and proposes some future work.
Chapter 2

Literature Review

In this chapter we provide an overview of existing work on human detection. A typical human detection system is shown in Fig. 2.1. Humans are detected in still images or videos. In videos, the detection system typically differs a lot more than still images. Hence, we limit the discussion to detection of humans in still images as this is one of the foci of the dissertation. The steps of a typical human detection system for still images are described as follows.

(a) The images are typically preprocessed before any feature extraction can be done. The type of preprocessing usually depends on the type of features being extracted. For instance, if gradient features are desired, edge filters are used on the original images to obtain gradient and orientation images from which the features can then be extracted. In some other cases, other preprocessing steps may be done such as histogram equalization or gamma correction to adjust the contrast of the images before feature extraction. Some scaling of the images may also occur in this step to detect humans that are larger or smaller than the detector window size. Typically, once the images have been preprocessed, the image is then split into grids of blocks from which features are then extracted.

(b) Based on the desired cues, relevant features can then be extracted from the pro-
Figure 2.1: A typical human detection system. The images are usually preprocessed before feature extraction. The type of preprocessing is dependant on the type of features being extracted. After which, based on the desired cues, the relevant features are then extracted from the images for representation. The feature vectors are then grouped into their corresponding classes and passed to the classifiers for training. The resultant classifiers are used during testing to decide if a human is present in the image or not.

(c) The features are passed to a trained classifier which evaluates each feature according to a decision function to determine if it represents a human or not. The trained classifiers are obtained from a training phase where there are labeled examples of humans and non-humans. Depending on the application needs, an appropriate classifier is usually trained. For instance, for fast detection, linear SVM or boosting-based classifiers are usually employed.

For still images, typically, the image is scanned densely using a window at different scales and positions. The feature extracted from each window is passed to the classifier for decision. Since there can be multiple positive detections in the neighbourhood of a true positive, non-maximum suppression [9, 14, 15] is performed as a post-processing step to merge neighbouring windows and to discard those that do not meet the crite-
Scanning images using windows becomes increasingly intensive with larger image resolutions. There have been some works that propose more efficient methods to scan images. In [55], the authors proposed a method called Efficient Subwindow Search to localize objects using a branch-and-bound search scheme. In their work, the image was divided into sub-windows. A bound for the highest score that a quality function could take on any of the sub-windows was computed for each sub-window. The algorithm terminated when it identified a sub-window with a quality score that was at least as good as the upper bound of all remaining candidate regions. This guaranteed that a global maximum had been found. Their experiment results showed that they could achieve a scanning speed that was 2 orders of magnitude faster than the standard scanning window speeds without losing too much accuracy. Harzallah et al. [42] proposed a 2-stage approach where a linear Support Vector Machine was first used to obtain an initial set of regions with high scores. A non-linear Support Vector Machine was then employed to refine the region sets. Pedersoli et al. [90] proposed a recursive coarse-to-fine localization refinement for a detector built on a multiresolution pyramid of dense features with spatial constraints. Wojek et al. [119] used parallel architectures that exist in current computers to speed-up the scanning of the images. Their work showed that they could achieve up to 30 times the normal scanning speed of images.

The work on human detection can be categorized into 2 groups as follows:

- Feature extraction
- Classification framework

In feature extraction, features are created based on the most informative visual content to describe humans as accurately as possible. In the classification framework, the obtained features are utilized to train classifiers using specific procedures. Sometimes, in this step, classifiers are specifically created for the purpose of human detection.
2.1 Feature Extraction

Feature extraction is a significant step in human detection. Instead of using the raw pixel values of the image directly as features, feature extraction methods for human detection use some form of understanding of context and transformation to obtain descriptors that characterize the human appropriately. Raw pixel values are not usually used as features as they do not contain characteristics/information/patterns that have been decorrelated. This leads to suboptimal classifier performance. Furthermore, using raw values directly increases the computational complexity of the detection. Feature extraction methods are mainly divided into two categories based on the representation: sparse and dense.

2.1.1 Sparse Feature Representation

In sparse feature representation, local features are obtained using interest point algorithms and are used to represent humans holistically (whole) or by parts. The main advantage of using interest point detectors are that they pre-sample possible locations and therefore provide a sparser set for learning and recognition. In [50, 93, 75], models for detecting humans by parts were proposed whereby features were extracted for each part and based on geometric constraints, assembled to form descriptors. An example of such a model is shown in Fig. 2.2. However, these methods produced feature vectors of large dimensions which made them computationally intensive. Furthermore, the geometric constraints varied according to window size and scale which was tedious as they needed to be pre-calculated.

In [74], edge points of interest were described using Scale Invariant Feature Transformation (SIFT) features [69] and shared among several object classes. A probabilistic model was used to detect various objects in an image. In [97, 59, 58], Implicit Shape
Chapter 2. Literature Review

702 R. Ronfard, C. Schmid, and B. Triggs

parts: the complete body, the head, the torso, and the left and right upper arms, forearms, hands, thighs, calves and feet, numbered from 1 to 15 as in Figure 1. Each body part $P_i$ is a rectangle parametrized in image coordinates by its centre $[x_i, y_i]$, its length or size $s_i$ and its orientation $\theta_i$. A coarse resolution whole-body image is also included in case 'the whole is greater than the sum of the parts'. During training and detection, we discretize the admissible range of sizes and orientations. As discussed later, we use a range of 8 scales, and 36 orientations equally spaced every 10 degrees. 14 body joints connect the parts: the plexus between body and torso, the neck between head and torso, the hips between torso and thighs, the knees between thighs and calves, the ankles between calves and feet, the shoulders between torso and upper arms, the elbows between upper arms and forearms and the wrists between forearms and hands. Figure 1 shows the body model in average position, using a single aspect ratio of 16:9 for all body parts.

Figure 2.2: Articulated body model with its 14 joints and 15 body parts in [93] (figure from [93]). As seen, each part is manually labeled and its constraint to neighbouring parts is clearly stated. However, the huge number of parts results in larger feature dimensions and the geometric constraints need to be pre-calculated.
Models were used to obtain interest points and using shape templates of humans as global cues, the local and global features were combined using a probabilistic top-down segmentation and stored in a codebook representation. Gao et al. [30] proposed using Adaptive Contour Feature, created from a chain of granules in Oriented Granular Space that is learnt using the AdaBoost algorithm, for human detection. Based on constraints, the features were linked to one another to describe the human contour. However, their method relies on the proper initialization of the feature parameters based on the data sets. Nguyen et al. [82] proposed using a variant of the Local Binary Pattern (LBP) [83] to describe regions around local interest points obtained using a SIFT detector for human detection. These features were stored in a codebook for matching.

2.1.2 Dense Feature Representation

In dense feature representation, features are extracted densely over a detection window and concatenated into a high-dimensional descriptor. Evaluation is then done as to whether a human exists in the window or not. In [88], a polynomial Support Vector Machine (SVM) was trained using Haar wavelets. This work was further extended in [77] to a parts-based approach to handle occlusions. In [111], the authors proposed using Haar-like (Fig. 2.3) and motion features with a boosting framework in video/image sequences. Shashua et al. [99] used 13 overlapping parts with each part described by a SIFT descriptor. A classifier was trained for each part. The training set was divided into clusters to deal with the high intra-class variability of humans and for each cluster, 13 classifiers were trained. The outputs of the classifiers were passed to an AdaBoost classifier for final decision.

Dalal et al. [9] introduced a popular dense descriptor that was a variant of SIFT [69]
Figure 2.3: Examples of rectangle filters used by Viola et al. (figure from [111]). The difference between the sum of the pixels in the lighter rectangles and the sum of the pixels in the darker rectangles constitute the Haar-like features. For filters containing 3 rectangles, the sum of the pixels in the darker rectangle is multiplied by 2 before the difference is taken.

known as the Histogram of Oriented Gradients (Hog) (Fig. 2.4). The image window was divided into grids of cells. A group of 4 adjoining cells was considered to be a block. The block was weighted using a Gaussian mask to weigh the centre pixels more heavily. For each cell, a histogram of gradients based on 9 orientations was computed. The votes into the bins were the gradient magnitudes of the pixels of the cell. The votes were tri-linearly interpolated between cells. The resulting 4 histograms were concatenated and normalized to form the Hog descriptor for the block. There are many works that exploit and propose improvements to Hog. In [25], Principal Component Analysis was performed on cell Hog features to identify the top eigenvectors and patterns that it represented. Based on the analysis, the authors came up with an analytic reduction
2.1. Feature Extraction

technique for reducing the dimensionality of Hog features without sacrificing accuracy. Wang et al. [92] considered rotation-invariant Hog where dominant orientations are found for each block and Hog feature with respect to the dominant orientations are computed. They further used Adaboost to select a small set of Hog features from a large pool of variable-sized block Hog features. Ge et al [34] proposed using templates of different sizes to obtain co-occurrences of Hog features. Watanabe et al. [117] proposed a second-order gradient feature known as the Co-occurrence Histograms of Oriented Gradients that built co-occurrence matrices based on pairs of gradient orientations. Their feature was extremely high-dimensional but could describe complex shapes that Hog could not. However, the problem with second-order features such as those by [34] or [117] was that the feature dimensionality was extremely high and it was computationally intensive to create such features.

Tuzel et al. [108, 109] proposed using holistic covariance descriptors made up of gradient, orientation and location features for human detection. Tosato et al. [106] extended this to a parts-based method in their work to handle occlusions. However, it was more computationally intensive than Hog. Wu et al. [121] proposed silhouette oriented features known as Edgelets for parts-based human detection in crowded static images. In [126], they extended the parts-based detection to a hierarchical structure with more regions for accurate detections. However, the problem with Edgelets was that it was too simple in representation i.e. “weak” feature. Without the classification framework proposed by Wu et al., the performance was poor. In [63, 66], the authors suggested using hierarchical part-template tree models using shape templates in conjunction with Hog to create pose-invariant descriptors (Fig. 2.5) for human detection and segmentation in images and videos.

Sabzmeydani et al. [94] used AdaBoost to select a subset of local average gradient responses in different directions within a sub-window of a detection window. This
For the person class, the HOG classifiers cue mainly on silhouette contours, especially the head, shoulders and feet. More precisely the chosen cells are ones on the contour, normalised using blocks centred on the image background just outside the contour.

(a) The average gradient image over the training examples. (b) Each “pixel” shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A sketch portraying the most relevant blocks – those lying just outside the contour. (e) A test image. (f) Its computed R-HOG descriptor. (g,h) The R-HOG descriptor weighted respectively by the positive and negative SVM weights. Only the dominant orientation is shown for each cell.

The key HOG parameters for several other object classes. We optimised all of the key parameters for each object class in the Pascal VOC 2006 challenge targets image classification and localisation for 10 different classes: bicycle, bus, car, cat, cow, dog, horse, motorbike, person and sheep. Table 4.2 summarises the main changes that occurred. The overall conclusion is that most of the parameters are very similar to those for person class, and those that do vary can be easily grouped and structured. This can help us by providing quick first guess of the HOG parameters for any given new object class. The VOC object classes can be broadly divided into two groups: natural objects such as horses, cows and sheep, and man made objects such as cars, motorbikes and buses. We treat the person class as an exception and place it in a separate category: even though people are natural objects whose articulations in result in characteristics similar to the natural object category and their clothing results in appearance features similar to the man made object category. We now comment on how performance varies

subset of features which consisted of a combination of gradients with different orientations and magnitudes at different locations was termed as Shapelets in their work. However, in comparison to other features, the performance was quite poor [14, 15].

Figure 2.4: Histogram of Oriented Gradients by Dalal et al. (figure from [9]). (a) Average gradient image over all training images (b) Maximum positive SVM weights in the blocks centered on the pixels (c) Maximum negative SVM weights in the blocks centered on the pixels (d) The most relevant blocks those lying just outside the contour and on the head-shoulder (e) Sample Image (f) Hog descriptors (g) Hog descriptors with positive SVM weights (h) Hog descriptors with negative SVM weights
2.1. Feature Extraction

3 Pose-Invariant Descriptors

3.1 Low-Level Feature Representation

For pedestrian detection, histograms of oriented gradients (HOG) [4] exhibited superior performance in separating image patches into human/non-human. These descriptors ignore spatial information locally, hence are very robust to small alignment errors. We use a very similar representation as our low-level feature description, i.e. (gradient magnitude-weighted) edge orientation histograms. Given an input image $I$, we calculate gradient magnitudes $|\nabla I|$ and edge orientations $O_I$ using simple difference operators ($-1, 0, 1$) and ($-1, 0, 1$) in horizontal-$x$ and vertical-$y$ directions, respectively. We quantize the image region into local $8 \times 8$ non-overlapping cells, each represented by a histogram of (unsigned) edge orientations (each surrounding pixel contributes a gradient magnitude-weighted vote to the histogram bins). Edge orientations are quantized into $N_b = 9$ orientation bins $[k\pi/N_b, (k+1)\pi/N_b)$, where $k = 0, 1, ..., N_b - 1$. For reducing aliasing and discontinuity effects, we also use trilinear interpolation as in [4] to vote for the gradient magnitudes in both spatial and orientation dimensions. Additionally, each set of neighboring $2 \times 2$ cells form a block. This results in overlapping blocks where each cell is contained in multiple blocks. For reducing illumination sensitivity, we normalize the group of histograms in each block using $L_2$ normalization with a small regularization constant $\epsilon$ to avoid dividing-by-zero. Figure 2 shows example visualizations of our low-level HOG descriptors.

The above computation results in our low-level feature representation consisting of a set of raw (cell) histograms (gradient magnitude-weighted) and a set of normalized block descriptors indexed by image locations. As will be explained in the following, both unnormalized cell histograms and block descriptors are used for inferring poses and computing final features for detection.

3.2 Part-Template Tree Model

For highly articulated objects like humans, part-based detection approaches (e.g. [7, 12]) have been shown to be capable of handling partial object/pose variations of humans [68], the authors introduced a family of descriptors known as Granularity-tunable Gradients Partition for human detection with versatile representations by defining spatial and angular uncertainty of line segments in Hough space and backprojecting these uncertainties into the image space by orientation-space partitioning. These descriptors can represent specific geometry information of humans like Edgelets or statistical representation of humans like HOG. Mu et al. [78] proposed variants of LBP for human detection as the original LBP descriptors required large storage and suffered from small rotational effects. However, the performance of the descriptors were not better than HOG or covariance descriptors due to some inherent weaknesses of LBP (See Chapter 4 for more details). Tang et al. [102] introduced templates that contained a pixel and its 2 neighbouring pixels and based on whether they satisfy a pre-defined function, histograms of pixels satisfying various templates are created for a set of functions. In [103], their work was extended to multi-scale representation. Though their approach was simple, their method was highly dependent on the number and type of templates that the user had to define.

There have also been some works that use deformable models to handle pose variations of humans. In [24], the authors proposed collecting parts arranged in a de-
formable configuration to represent objects. The appearance of each part was modeled separately and the deformable configuration was described by spring-like connections between pairs of connected parts. Felzenszwalb et al. [23] suggested a deformable model defined by a coarse template, a number of high resolution part templates and a spatial model for each part location (Fig. 2.6). Hog features from different image scales were used in their work. They further extended their work in [25] by considering mixture models and using modified Hog features which were lower dimensional. Their approach produced very good results for object detection. However, due to the complexity of their approach, the detection is quite slow. Ying et al. [127] suggested using locally deformable Markov models for human detection and tracking in videos. Their method captured local nonrigidity of humans quite effectively. Hussein et al. [48] proposed using deformable features that can translate in a small neighbourhood around a central location to locate the physical structure that they represent within the neighbourhood.

Lately, there has been immense interest in combining features for detecting humans. Dollar et al. [12] proposed feature mining strategies to automate feature design. In their work, they used Haar wavelet features computed over different image channels such as gradient magnitude, Gabor convoluted images and colour channels. Wojek et al. combined Haar wavelets, Hog features and dense Implicit Shape Models [97, 59, 58] with different classifiers to test the performance multi-feature people detection. Their results indicated that using multiple features as opposed to single feature produced better results. Schwartz et al. [96] combined edge, texture and colour information using Hog, colour frequency histograms and co-occurrence matrices. The dimensions of the resulting features were then reduced using Partial Least Squares analysis. Wang et al. [115] combined Hog and LBP features for holistic human detection with parts-based detection being employed in regions where partial occlusions are detected. In
[13], the authors combined grayscale feature, colour feature, texture features (Gabor and Difference of Gaussian), edge features and gradient histogram features in a boosting framework for human detection. The authors proposed a variant of this detector which ran much faster in [11] with little loss in accuracy. Aharon et al. [3] presented a parts-based human detection method that made use of nine features with a feature selection algorithm. Walk et al. [113] combined Hog features, Lbp features and colour self-similarity features for human detection and tested the performance using SVM and MPLBoost classifiers. Although using multiple features has resulted in improved performance over single feature, the increase in computational complexity and memory requirements outweigh the potential benefits of the combination of these features.
Due to the different features available for human detection and the different evaluation frameworks for different benchmark data sets, there have been several survey papers published to unify the different features under a unified evaluation framework. Munder et al. [79] studied PCA coefficients, Haar Wavelet and local receptive field (LRF) features using Support Vector Machines (SVMs), feed-forward neural networks and \( k \)-nearest neighbour classifiers and a proposed evaluation framework for their data set. Their results showed that LRF features with SVM produced the best results. Paisitkriangkrai et al. [86] analyzed Hog, Covariance and LRF features on 3 different benchmark data sets using SVM classifiers with different kernels. Their results highlighted that Hog and Covariance features outperformed LRF feature and using non-linear SVM kernels produced the best results. Their work, however, did not propose a unifying evaluation framework for the data sets. The authors used the evaluation framework proposed by the authors of each data set to analyze.

Dollar et al. [14] proposed improved evaluation metrics for false positives per image evaluation and established a new benchmark data set. They also used their evaluation metric on other existing benchmark data sets to unify the performance of different detectors under a common evaluation metric. In [15], they further modified their evaluation methodology to obtain a more accurate and informative benchmark and tested more detectors under this modified evaluation framework (from 7 previously to 16). Hussein et al. [49] introduced another framework for comparing human detectors that consider the detector effect on a full image. They compared between testing on cropped windows and testing on whole images and between building a multisize image pyramid while keeping the scanning window size fixed and using a single image size and varying the the scanning window size. Their work however was tested on only one benchmark data set and hence, whether the framework is able to unify the other benchmark data sets is not known. Enzweiler et al. [19] tested 4 differ-
2.1. Feature Extraction

Different feature-classifier combination - wavelet-based AdaBoost cascade, Hog with linear SVM, LRF with Neural Network and combined shape-texture (Shape templates + LRF with Neural Network) and provided a new benchmark data set with its own evaluation framework. Gerónimo et al. [36] provided reviews, analyses and discussions on state-of-the-art sensors and benchmarking.

2.1.3 Other Feature Representations

Besides sparse and dense feature representations, other types of features have been proposed for human detection. One of them are shape features. For example, contour fragments of objects were used in [85] as features which were stored as codebook entries. Boosting was then performed to choose discriminative combinations of contour fragments to form a strong Boundary-Fragment-Model detector. This idea was further enhanced in [84] where the features were now learnt incrementally and shared across object categories. A similar idea to [85] was also presented in [29] by Ferrari et al. which used contour fragments for representation of objects. However, their method used a graph approach to connecting fragments. They further extended this in [28] where they proposed a family of local features based on the number of connected contour segments stored in codebooks. However, the problem with these approaches was that the contour fragments used were typically of a pixel width while practically, the object boundaries were of several pixels in thickness. This oversimplified the object appearance leading to suboptimal performance.

To describe humans with varying poses, binary contours in the form of templates have been proposed in many works. Broggi et al. [6] proposed using a symmetry-based segmentation to filter pedestrians from video frames. Using correlation, the upper body shape of pedestrians were then matched to an edge modulus image. Gavrila [31]
proposed using hierarchical holistic shape templates of humans for detecting pedestrians from a vehicle. This work was further improved in [32] where a probabilistic model was introduced for shape-matching (Fig. 2.7). Matching was performed using Chamfer distance. Nanda et al. [81] used probabilistic template matching over 3 scales and using 3 different templates to detect pedestrians in near infra-red video frames. In [70], the authors combined the Haar cascade classifier [111] with Chamfer Contour Matching which is a template correlation method based on the object shape. Zhao et al.
[139] proposed using shape templates for detection of people in videos with segmentation of foreground. However, in their work, the detection was mainly concentrated on upper bodies of humans. In [7], the authors used a Near Infra-Red camera to detect pedestrians. In their work, they used 2 different templates - one for to detect pedestrians holistically and the other to detect just legs. Lin et al. [64] proposed using hierarchical parts-based shape templates for human detection and segmentation in videos. In [18], pose-specific probabilistic shape [33] and texture models [76] were used to create a generative model. This model was then used to synthesize virtual samples of the target class to further populate the training set of a discriminative pattern classifier. However, these approaches were sensitive and performed poorly in cluttered images where humans were occluded.

2.2 Classification Framework

The classification framework also plays an important part in human detection. Most of the features described earlier use generic classifiers such as SVM [88, 77, 9, 63, 115, 95, 117, 23, 25] and and boosting-based classifiers like AdaBoost [12, 13, 11, 94, 111, 120], RealBoost [30] and LogitBoost [48, 68, 106, 109]. These features use the training procedures established by the authors of the benchmark data sets they are evaluated on. However, there are some methods that focus on developing better training methods/procedures. In [140], the authors proposed an AdaBoost-like cascade model with variable-size blocks for training and testing Hog. They were able to speed up detection without much loss in accuracy compared to the original Hog [9]. Hiromoto et al. [45] proposed speeding up Co-occurrence Histograms of Oriented Gradients classification by 2 ways. First, they divided the high dimensional feature vector into smaller feature vectors (broken down into block level feature vectors). A weak classifier was
Chapter 2. Literature Review

then assigned to each of the smaller feature vectors for evaluation. Then, the weak classifiers were combined in a cascade structure to speed up classification. Paisitkriangkrai et al. [87] proposed an AdaBoost classification model in Euclidean space with weighted Fisher linear discriminant analysis-based weak classifiers for Covariance features. However, boosting approaches suffer from long training times and the performance of the approaches is at best the same as the original non-boosting methods.

Zhang et al. [136] proposed a multi-resolution architecture for training a linear SVM classifier using Hog features. In their work, many different resolutions were computed using a coarse-to-fine feature hierarchy. During detection, the lower resolution features were used to reject majority of negative windows. This allowed a small number of windows to be processed in higher resolutions, which increased the detection speed. Park et al. [89] described a multi-resolution model that used deformable parts-based model when examining large resolution of humans and a holistic template when examining low-resolution humans. They also included contextual reasoning to further improve detections at small scales. Enzweiler et al. [17] proposed a mixture-of-level framework where a set of part-based expert classifiers was trained on intensity, depth and motion features with partial occlusion handling. In [20], this framework was further enhanced to include a view-related and sample-dependent combination of hybrid feature classifiers. Shen et al. [100] suggested an alternative feature selection using greedy sparse linear discriminant analysis instead of AdaBoost for training a detector. In their proposed algorithm, a new weak learner that maximized the inter-class separation was found. The coefficients of selected weak classifiers were updated repetitively during the learning process. In [124], a method to determine features to use for evaluating a region was proposed. Cascade structured detectors were learned by boosting local feature based weak classifiers. The weak classifier only moved to another feature when the predictions from the already examined features were not
sufficiently confident.

Several novel classifiers specifically for human detection have also been proposed. Following the work in [46], Wu et al. [123, 122, 125] proposed a novel Cluster Boosted Tree classifier for multi-view and multi-pose handling. They broke up the sample space by unsupervised clustering based on discriminative image features selected by boosting algorithm. At each boosting round, for each branch of the tree, one feature based weak classifier was selected to be attached to the branch. Then the discriminative power of the weak classifier was estimated. If it was too weak, the branch was split into two by unsupervised clustering, i.e. the training samples were divided into two subcategories. The splitting of the tree stopped once a target accuracy was reached. Maji et al. [72, 71] proposed additive kernels, such as Histogram Intersection and chi-squared kernels, for SVM classifiers. They devised a method whereby the non-linear additive kernels could run with runtime and memory complexity that is independent of the number of support vectors.

Ranxu et al. [130, 128] proposed L1-Norm minimization for linear and non-linear SVMs. The L1-norm minimization is a feature selection method. Their approach selected a subset of sparse features from a large dense feature set. They further extended this to a cascaded model in [129]. Dollar et al. [10] proposed a set classifier that is trained by a combination of boosting and weakly supervised learning through Multiple Instance Learning. Their classifier considers part-based representation of humans and learns individual component classifiers for each part before combining to form an overall classifier. Lin et al. [65], in contrast to the work by Dollar et al. which performed Multiple Instance Learning on a set of randomly selected parts, proposed using a “seed-and-grow” design to construct a multiple instance feature.
2.3 Discussion

In the early years, sparse feature representations were very popular as they offered compact feature representations and locations of interest points which produced high repeatability between different images containing the same objects. Furthermore, due to limitation of computational power and memory constraints, they were much preferred. However, with the improving hardware capabilities of existing computers and the availability of increasing and cheaper memory, dense feature representations are now popular as they describe objects richly compared to sparse feature representations [5, 56, 107]. However, it was difficult to compare between features as there was no established benchmark database and no de facto evaluation baseline.

This changed when Dalal et al. introduced Hog and the INRIA dataset [9]. In order to use their data set, the authors established an evaluation criteria that became the de facto criteria for many detectors. This methodology for evaluating detectors based on binary classifiers measures the per-window performance on cropped positive and negative image windows. This isolates the classifier performance from the overall detection system [14, 15]. This criteria was further improved by Dollar et al. in [14, 15] where they pointed out some misconceptions of the per-window methodology. Firstly, they found that better per-window performance does not necessarily translate to better detection performance. When implemented for a full image detection, there are many choices to be made in converting the binary classifier to a detector which includes values for spatial and scale stride and non-maximum suppression. Hence, they proposed an alternative evaluation framework, the per-image methodology, which was more practical to evaluate human detection systems and along with it, a more challenging data set (Caltech Pedestrian Detection Benchmark). Their evaluation framework is currently the most adopted one for comparisons between different human detectors.
Table 2.1: Comparison of Most Common Human Detectors

<table>
<thead>
<tr>
<th>Features</th>
<th>Learning</th>
<th>Detection Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>gradient hist.</td>
<td>classifier</td>
<td>INRIA</td>
</tr>
<tr>
<td>gradients</td>
<td>AdaBoost</td>
<td>log-average miss rate</td>
</tr>
<tr>
<td>grayscale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>color</td>
<td></td>
<td>72%</td>
</tr>
<tr>
<td>texture</td>
<td></td>
<td>95%</td>
</tr>
<tr>
<td>self-similarity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>motion</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>feature learn.</td>
<td>Caltech Pedestrian Data Set</td>
</tr>
<tr>
<td></td>
<td>part-based</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VJ</td>
<td>AdaBoost</td>
<td>log-average miss rate</td>
</tr>
<tr>
<td>SHAPELET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoseInv</td>
<td>AdaBoost</td>
<td>82%</td>
</tr>
<tr>
<td>LatSvm-V1</td>
<td>latent SVM</td>
<td>91%</td>
</tr>
<tr>
<td>FtrMine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HikSvm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hog</td>
<td>linear SVM</td>
<td>44%</td>
</tr>
<tr>
<td>MultiFtr</td>
<td>AdaBoost</td>
<td>80%</td>
</tr>
<tr>
<td>HogLbp</td>
<td></td>
<td>86%</td>
</tr>
<tr>
<td>LatSvm-V2</td>
<td>latent SVM</td>
<td>58%</td>
</tr>
<tr>
<td>MultiFtr+css</td>
<td></td>
<td>74%</td>
</tr>
<tr>
<td>FeatSynth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLS</td>
<td>PLS+QDA</td>
<td>43%</td>
</tr>
<tr>
<td>MultiFtr+motion</td>
<td></td>
<td>73%</td>
</tr>
<tr>
<td>ChinFtrs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiFtr+motion</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1 summarizes the differences and performances of the most common human detectors on INRIA [9] and Caltech Pedestrian Data Set [14, 15]. The first set of columns describes the features used by each detector. Almost all detectors make use of gradient histograms. Some detectors use a combination of features to produce much better results. The second set of columns gives details of the learning method used by the detector. SVMs and boosting classifiers are quite popular among the detectors listed. Some detectors learn a smaller set of features before classifier training (indicated by the tick in the “feature learning” column). Some detectors are part-based detectors such as LatSvm and FeatSynth. The last set of columns summarizes the detection performance on INRIA and Caltech Pedestrian Data Set. The detection performance is summarized using log-average miss rates (see Section 3.3.2 for details).
Many survey papers [14, 15, 79, 86, 49, 19] have concluded that the best performing feature for human detection is Hog by Dalal et al.. With combination of different features that use different contextual information, the performance of human detection can boosted greatly. However, this results in increased complexity and memory requirements. Therefore, it is necessary to find limitations of existing features and to improve on them so as to reduce the reliance on multiple features. In this thesis, we analyze the limitations of three such features - Hog, LBP and LTP - and provide new features that resolve their limitations. These will be discussed in greater detail in Chapters 3 and 4.

Human detection is such a challenging problem that conventional classifiers such as SVM and boosting classifiers offer only suboptimal performance in comparison to specialized classifiers such as additive kernels [72, 71], Multiple Instance Learning [10, 65] or Cluster Boosted Tree [123, 122, 125]. However, creating classifiers specialized for human detection using a bottoms-up approach is a tedious and time-consuming approach and due to the niche application, the wide use of the developed classifier is limited. In order to apply these classifiers to other forms of object detection, a lot of parameters need to be tweaked. Hence, other researchers have focused on improving classification frameworks for human detection using conventional classifiers. Some propose multi-resolutions architectures for training and testing to handle scale variations of humans [136, 89, 140] while others focus on combining subspace feature selection techniques with conventional classifiers [100, 87, 25]. These architectures have been shown to improve performance significantly. In this thesis, we propose an alternative classification framework which uses a subspace feature extraction technique, Asymmetric Principal Component Analysis, and a non-linear modified Minimum Mahalanobis Distance classifier. Compared to existing works where the imbalance in the number of samples available for the training data is not considered, we
acknowledge this imbalanced data and resolve this in our approach. Furthermore, we analyze the feature space and prove that a linear classifier is not always the best choice for a human detection problem. These will be discussed in greater detail in Chapter 5.
Chapter 3

Extended Histogram Of Gradients

This chapter describes the proposed *Extended Histogram of Gradients* (ExHoG) feature. The main conclusion is that ExHoG encoding provides better detection performance relative to other existing feature sets including Histogram of Gradients (Hg) [9] and Histogram of Oriented Gradients (Hog) [9].

We start by describing the Hg and Hog features and their limitations in Sect. 3.1. Section 3.2 provides details of the proposed ExHoG feature and its advantages over Hg and Hog. The overall performance results on benchmark data sets are quantified in Sect. 3.3. The chapter concludes with a discussion in Sect. 3.4.

3.1 Histogram of Gradients and Histogram of Oriented Gradients

3.1.1 Descriptor formation

Since the introduction of Scale Invariant Feature Transform (SIFT) by Lowe [69] and its phenomenal success in the field of object detection, researchers have been working on ways to specialize SIFT for specific object detections. Dalal et al. [9] was success-
ful in modifying SIFT to perform the task of human detection. There are 2 primary differences between the SIFT descriptor and the Hg descriptor [9]:

- SIFT descriptors describe information at scale-invariant key points which are rotated to align their dominant orientations. Furthermore, each descriptor is independent. However, Hg descriptors are computed in dense grids at a fixed scale without any orientation alignment.

- SIFT descriptors are used for sparse wide baseline matching while Hg descriptors are optimized for dense robust coding of spatial form.

The steps in computing the Hg descriptor for human detection as described by Dalal et al. are as follows (Fig. 3.1):

**Normalize input image**

The input image is first normalized by computing the square root of each colour channel. This reduces the effects of local shadowing and illumination variations.

**Compute first-order gradients**

The first order gradients in the $x$ and $y$ directions are then computed using $[1 \ 0 \ -1]^T$ masks for each colour channel. The colour channel gradient with the maximum response for each pixel is used. No smoothing is carried out prior to the gradient computation as it degrades performance.

**Divide the image into grids of cells**

An image window is scanned over the image with each window forming a Hg descriptor that describes the region. The window is broken up into grids of cells of $N \times N$ pixels.

**Group cells into overlapping blocks**

Within an image window, blocks of $M \times M$ cells are combined to form overlapping
blocks (usually, 50% overlap of blocks is used). A Gaussian spatial filter ($\sigma = 0.5 \times 2M$) is applied centered at the centre of the block to downweight the gradients of the pixels near the edges of the block. This minimizes the sudden changes in the block feature with small changes in the position of the detector window and to give less emphasis to gradients that are further away from the centre of the block.

**Compute histogram of edge directions for each block**

A histogram of $B$ bins of edge directions for each cell is computed. The votes into the bins are the gradient magnitudes of the pixels with the corresponding ori-
3.1. Histogram of Gradients and Histogram of Oriented Gradients

The votes are interpolated tri-linearly between neighbouring bin centres in both orientation and position to reduce aliasing. The $M \times M$ histograms in a block are normalized using clipped L2-norm. This minimizes the variation of gradient strengths due to local variations in illumination and foreground-background contrast. The histograms are then concatenated to form a $M \times M \times B$-D feature vector for the block.

**Concatenate histograms over overlapping blocks of cells**

The feature vectors from all the blocks are compiled to form a combined feature vector for the window.

Usually, $N = 8$ pixels, $M = 2$ cells and $B = 18$ bins, each representing an interval of $20^\circ$ in the range from $0^\circ$ to $360^\circ$, are values used in the computation of the Hg descriptor. In all, using these values, the Hg descriptor has 7560 dimensions. The stride of the sliding window is usually set to 8 pixels horizontally and vertically. Once the Hg descriptors for all the windows of the positive and negatives images have been processed, the descriptors a linear SVM classifier is trained using the descriptors. The negative images are then tested with the trained classifier to obtain the hard negative examples which are used for bootstrapping. The hard negative examples and the original training data are then used to train the linear SVM classifier. The trained classifier is used for testing. A flowchart of the steps for descriptor formation, training and testing is given in Fig. 3.2.

### 3.1.2 Limitations of Histogram of Gradients and Histogram of Oriented Gradients

Hg differentiates a bright object against a dark background and vice-versa as it considers gradient directions from $0^\circ$ to $360^\circ$. The intra-class variation of the humans
Figure 3.2: Flowchart of Hg computation procedure, training and testing. The SVM classifier training stage includes bootstrapping and retraining of the SVM classifier.

becomes larger with this differentiation. In Fig. 3.3, a dark object against a bright background and vice-versa in the two different cells are illustrated. As it can be observed, the Hg features for the 2 scenarios are different.

Dalal et al. [9] proposed solving the problem of Hg by using Histogram of Oriented Gradients (Hog) where the gradients of direction $\theta$ and $\theta + 180^\circ$, $0^\circ \leq \theta < 180^\circ$, are considered to be of the same orientation $\theta$ only. In order to solve the problem of Hg, Hog treats all gradients of opposite directions as gradients of a same orientation. This is illustrated in Fig. 3.3 where the Hog representations for both scenarios are the same. However, this causes Hog to be unable to discern some local structures that are different from each other. It is possible for 2 different structures to have the similar feature representation. This is illustrated in Fig. 3.4. The problem with Hog
3.1. Histogram of Gradients and Histogram of Oriented Gradients

Figure 3.3: Problem of Histogram of Gradients (Hg) and its solution by Histogram of Oriented Gradients (Hog). A dark object against a bright background and vice versa in a cell produces 2 different features which increases the intra-class variation of humans. Hog resolves this by considering gradients of direction $\theta$ and $\theta + 180^\circ$ to be of the same orientation $\theta$ only.

is that gradients of opposite directions in the same cell are mapped to the same bin in a histogram. In Fig. 3.4(a), the first pair of structures represent a slightly bent human
Figure 3.4: Problem of Histogram of Oriented Gradients (Hog). 6 local structures are shown. Hog makes the different structures in (a)(i) and (ii) similar as shown in (a)(iii) and different structures in (b)(i) and (ii) similar as shown in (b)(iii). A similar problem can be observed in (c).

torso against a background (edge) and human limbs against a background (ridge). Hog produces the same feature for these very different 2 structures. Similarly, in Fig. 3.4(b) and Fig. 3.4(c), it can be seen that, for each pair of structures, they are represented as the same by Hog.
3.2 Extended Histogram of Gradients

Consider an unnormalized \( H_g \) feature of a cell, \( b_k \) where \( k \) is the \( k^{th} \) cell in the block. Let \( i \) denote the bin of quantized gradient direction \( \theta \), \( h_{g_k}(i) \) the bin value of \( H_g \) and \( L \) the number of bins of \( H_g \) which is even. We find that \( H_{og} \) can be created simply from \( H_g \) as follows:

\[
h_{og_k}(i) = h_{g_k}(i) + h_{g_k}(i + \frac{L}{2}), \quad 1 \leq i \leq \frac{L}{2}
\]

(3.1)

where \( h_{og_k}(i) \) is the \( i^{th} \) bin value of \( H_{og} \). We see that \( H_{og} \), in fact, is the sum of two corresponding bins of \( H_g \).

Now, consider the absolute difference between \( h_{g_k}(i) \) and \( h_{g_k}(i + \frac{L}{2}) \) of \( H_g \) to form a Difference of HG (\( D_{hg} \)) as follows:

\[
h_{dg_k}(i) = |h_{g_k}(i) - h_{g_k}(i + \frac{L}{2})|, \quad 1 \leq i \leq \frac{L}{2}
\]

(3.2)

where \( h_{dg_k}(i) \) is the \( i^{th} \) bin value of \( D_{hg} \).

\( D_{hg} \) produces the same feature as \( H_{og} \) for patterns that contain no gradients of opposite directions. It differentiates these patterns from the ones that contain opposite gradients by assigning small or almost zero values to the bins that the gradients are being mapped to. The concatenation of these 2 histograms produces the Extended Histogram of Gradients (\( E\text{XHoG} \)).

In [9, 15, 19, 23, 66], \( H_{og} \) is clipped and renormalized after it has been created. This effectively reduces the illumination variations and noise. However, it presents a problem for some structures in different cells as their features may become similar. An example is illustrated in Fig. 3.5. The \( H_{og} \) features before clipping are different for the 2 different structures. However, after clipping, they become the same. This normalization procedure for \( H_{og} \) in [9, 15, 19, 23, 66] causes some local structures to
be similarly represented.

Consider the same normalization procedure in [9, 15, 19, 23, 66] for ExHoG. The magnitudes of the bins of HoG are much larger compared to DHoG since creation of HoG involves summation of two positive values while creation of DHoG involves an absolute difference of them. Hence, if there are noisy gradient pixels with large magnitudes or very abrupt intensity changes in the image, these large gradient magnitude peaks, which are captured in Hg (Fig. 3.6), are propagated into HoG and DHoG. These peaks are larger in HoG than in DHoG. If we perform normalization of the ExHoG feature after the concatenation of these 2 histograms as illustrated in Fig. 3.6 similar to [9], these large gradient magnitude peaks, typically, are only clipped in the HoG component of ExHoG and remain unclipped in the DHoG component of ExHoG. Hence, a different normalization scheme is proposed. In this scheme, the normalization is performed directly after the Hg block feature is created and before the summation and subtraction of the Hg. The normalization steps are described as follows:

\[
h_{g_{n}}(i) = \frac{h_{g_{k}}(i)}{\sqrt{\sum_{k=1}^{N} \sum_{i=1}^{L} (h_{g_{k}}(i))^2}}
\]  

(3.3)

\[
h_{g_{c_{k}}}(i) = \begin{cases} 
  h_{g_{n_{k}}}(i), & h_{g_{n_{k}}}(i) < T \\
  T, & h_{g_{n_{k}}}(i) \geq T,
\end{cases}
\]  

(3.4)

\[
h_{g_{c_{k_{n}}}}(i) = \frac{h_{g_{c_{k}}}(i)}{\sqrt{\sum_{k=1}^{N} \sum_{i=1}^{L} (h_{g_{c_{k}}}(i))^2}}
\]  

(3.5)

where \(N\) is the number of cells in the block and \(T\) is the clipping threshold. The HoG and DHoG features are then generated from this normalized Hg feature, \(h_{g_{c_{k_{n}}}}(i)\).

Fig. 3.7 shows the effect of this proposed normalization scheme on the 2 structures
3.2. Extended Histogram of Gradients

Figure 3.5: Example of 2 structures that have similar Histogram of Oriented Gradients (Hog) representations after clipping and normalization.
Figure 3.6: Problem of normalization directly on Extended Histogram of Gradients (ExHoG) [Best viewed in colour]. Histogram of Gradients (Hg) is created with a large peak due to the abrupt intensity change. One of the peaks remain unclipped in the formation of ExHoG.

shown in Fig. 3.5. It can be seen that the resulting Hog features for the 2 structures remain different. In [9], the bins of Hg are first merged to form Hog and then clipped. In the proposed normalization scheme, the bins of Hg are first clipped and then merged to form Hog. This allows the differentiation of some structures to remain after clipping and normalization. Furthermore, the proposed normalization scheme also clips the large gradient peaks before they can be propagated into the Hog and Dhg features. This allows the representation of ExHoG to be more robust to noise and abrupt intensity changes in the image (Fig. 3.8).

The proposed ExHoG of a cell is constructed from the clipped L2-norm normalized
3.2. Extended Histogram of Gradients

Figure 3.7: Effect of the proposed normalization scheme on the 2 structures in Fig. 3.5. The resulting Hog features for the 2 structures are different.
Figure 3.8: Proposed normalization scheme for ExHoG [Best viewed in colour]. The Hg feature is clipped as shown. It can be observed that the large peak has been clipped. The peak in the histogram of Difference of Hg has been suppressed.

\[
h_{egk}(i) = \begin{cases} 
  h_{gcn_k}(i) + h_{gcn_k}(i + \frac{L}{2}), & 1 \leq i \leq \frac{L}{2} \\
  |h_{gcn_k}(i) - h_{gcn_k}(i - \frac{L}{2})|, & \frac{L}{2} + 1 \leq i \leq L,
\end{cases}
\]  

(3.6)

where \(h_{egk}(i)\) is the \(i^{th}\) bin value of ExHoG. The cell features in a block are then concatenated to form the block feature. All the overlapping block features are then concatenated to form an overall feature descriptor for the window.

Unlike Hog, the proposed ExHoG differentiates gradients of opposite directions from those of same direction in the same cell. This is clearly illustrated in Fig. 3.9. Using the same local structures as discussed in Fig. 3.4, it is shown that the ExHoG representations for each of the local structure is unique. Hence, ExHoG represents
3.2. Extended Histogram of Gradients

Figure 3.9: ExHoG representations of local structures in Fig. 3.4. ExHoG differentiates the local structure pairs, 
\{\text{(a)(i),(b)(i)}\}, \{\text{(c)(i),(d)(i)}\} and \{\text{(e)(i),(f)(i)}\} misrepresented by Hog.
the human contour more discriminatively than Hog and has less intra-class variation than Hg. For instance, unlike Hog, ExHoG can differentiate between a human torso against a background and human limbs against a background. Furthermore, ExHoG also resolves the larger intra-class variation of objects caused by the intensity reversal of human and background as shown in Fig. 3.10.

3.3 Overall Results

We perform experiments on two data sets - INRIA [9] and Caltech Pedestrian Data Set [14, 15]. The performance of the proposed approach is compared against some state-of-the-art single type of holistic feature methods on these given data sets. There are 2 evaluation methodologies used in human detection depending on the types of data sets used. If a data set only contains cropped images for training and testing, the per-window performance is compared. If full images and human annotations for training and testing are available, then the per-image performance is compared as it is a better evaluation method [15].

Results are reported for the INRIA data set and Caltech data set using the per-image methodology. Linear SVM (using the modified version of SVMLight [54] provided by [9]) classifiers are used.
3.3. Overall Results

A cell size of $8 \times 8$ pixels is used with a block size of $2 \times 2$ cells. The number of bins for each cell for ExHoG is 18. A 50% overlap of blocks are used in the construction of the feature vectors. The Hg block feature is normalized using a clipping value of 0.08.

3.3.1 Training of classifiers

For INRIA and Caltech Data Sets, the training set of INRIA is used to train the classifiers. The training data set contains 2416 cropped positive training images and 1218 uncropped negative training images. The sliding window size is $128 \times 64$ pixels. We randomly take 10 samples from each negative image to obtain a total of 12180 negative samples for training the linear SVM classifier. Bootstrapping is then performed on the negative images across multiple scales to obtain 89400 hard negatives which are combined with the original training set to retrain the SVM classifier. During bootstrapping, a scale step of 1.05 is used. This training procedure is exactly the same as the ones described in [9] and [15].

3.3.2 Performance comparison of ExHoG against Hog and Hg

In order to demonstrate the effectiveness of the proposed normalization scheme as compared to the standard normalization scheme of Hog, we performed experiments on the INRIA data set for Hog and ExHoG using the different normalization schemes. The clipping value of the standard normalization scheme is set to 0.2 for both Hog and ExHoG. The INRIA testing set consist of 288 images. The images are scanned using the trained classifiers over multiple scales. The scale step used is 1.05. The window stride is 8 pixels horizontally and vertically. These parameters are the same as those used in [15] for test. The miss rate (MR) versus false positives per image (FPPI) is plotted to compare between different detectors. The log-average miss rate (LAMR) [15] is used
to encapsulate the detector performance which is found by averaging the miss rates at nine FPPI locations evenly spaced in the range $10^{-2}$ to $10^0$. If any of the curves end before reaching $10^0$, the minimum miss rate achieved is used [15].

Fig. 3.11 shows the results. It is seen that using the proposed normalization scheme improves the performance of both features. The LAMR of Hog decreases by 3% from 46% to 43% using the proposed normalization scheme. The LAMR of ExHoG is further reduced by 2% from 39% to 37% when the proposed normalization scheme is used. With the combined effect of the proposed normalization and the concatenation of Hog and Dhc histograms, ExHoG outperforms the standard Hog by 9% and the improved Hog by 6%.

We also perform experiments for Hg on the INRIA data set to compare with ExHoG. From Fig. 3.12, it can be seen that ExHoG outperform its predecessors. ExHoG performs the best at 37%.

### 3.3.3 Results on INRIA Data Set

The INRIA test set contains 288 images. We scan the images using the trained classifiers over multiple scales. The scale step used is 1.05. The window stride is 8 pixels horizontally and vertically. These parameters are the same as those used in [15] for test. To compare between different detectors, the miss rate (MR) versus false positives per image (FPPI) is plotted.

We compare the performance of ExHoG with VJ [111], SHAPELET [94], PoseInv [63] and HikSvm [72]. In order to keep the comparisons clearly within the domain of single type of holistic features, performance comparisons with methods that use hybrid features or parts-based modeling for human representation are omitted. These detectors are optimized, trained and tested by their respective authors. From Fig. 3.13, ExHoG
3.3. Overall Results

<table>
<thead>
<tr>
<th>Miss Rate</th>
<th>46% HOG Old Norm</th>
<th>43% HOG New Norm</th>
<th>39% ExHoG Old Norm</th>
<th>37% ExHoG New Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Positives Per Image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.11: Performance of ExHoG against Hog using different normalization schemes [Best viewed in colour]. ExHoG with the proposed normalization scheme outperforms all other methods.

achieve a LAMR of 37% which is significantly lower than all methods being compared with.

Fig. 3.15a show some ExHoG detections on INRIA.

3.3.4 Results on Caltech Pedestrian Detection Benchmark Data Set

The Caltech Pedestrian Detection Benchmark data set [14, 15] is the largest data set available for human detection. The data set has color video sequences and pedestrians in a variety of scales and more variable scenes than other human data sets like INRIA or Daimler. It is also the only data set which provides temporal correspondence between
bounding boxes and detailed occlusion labels. The data set has been created from a recorded video on a moving car through some densely populated human areas. As such, the data set contains artifacts of motion, blur and noise, and has various stages of occlusion (from almost complete to none). The data set is divided into 11 sessions. The first 6 sessions are designated as the training set while the remaining 5 are designated as the testing set. The testing set contains 155 000 annotated pedestrian samples in 65 000 images and 56 000 negative images.

In [15], the authors reported results whereby they used detectors trained on other data sets like INRIA for classification on their testing set. Here, we also present our results in a similar manner to [15] where our detectors are trained using the INRIA data
3.3. Overall Results

Figure 3.13: Performance of ExHoG against existing approaches on INRIA [Best viewed in colour]. ExHoG outperforms all other methods.

set (Section 3.3.1) and tested on the test sessions of the Caltech Pedestrian Detection Benchmark data set. The scale step used is the same as that used during test of the classifiers on the INRIA test set i.e. 1.05. The window stride is 8 pixels horizontally and vertically. The settings used here are similar to those used in [15]. Same as [15], in order to detect humans at smaller scales, the original images are upscaled. Only every 30th frame is evaluated so that our comparisons can be kept consistent with those in [15].

Detailed results are presented in Fig. 3.14. The detectors we compare with ExHoG are the same as those in Section 3.3.3 with the addition of Hog [9]. These detectors are optimized, trained and tested by their respective authors. The performance is analyzed
Figure 3.14: Performance under different conditions on test set of the Caltech Pedestrian Data Set [Best viewed in colour].
under six conditions as in [15]. Fig. 3.14 show the overall performance on the test set, on near and medium scales, under no and partial occlusions and on clearly visible pedestrians (reasonable). As in [15], the MR versus FPPI is plotted and LAMR is used for describing overall performance. The results are discussed under each condition in more details as follows.

**Overall:** Fig. 3.14(a) plots the performance on all 6 test sessions for every annotated pedestrian. ExHoG ranks first at 87%.

**Scale:** Fig. 3.14(b) plots the performance on unoccluded pedestrians corresponding to heights over 80 pixels. ExHoG has a LAMR of 40%.

Fig. 3.14(c) plots the performance on unoccluded pedestrians corresponding to heights between 30 - 80 pixels. ExHoG ranks first in performance at 81%. At this scale, ExHoG outperforms other approaches by a large margin. This highlights that at low- and medium-resolutions, our feature is more robust in detecting pedestrians than other approaches in comparison. This is an important aspect as in pedestrian detection problems, it is necessary to detect humans further away from the vehicle more accurately so that there is ample time to react to prevent an accident. 30-80 pixel height of pedestrians is the most appropriate image resolution for pedestrian detection in automotive applications [15].

**Occlusion:** Fig. 3.14(d) plots the performance on unoccluded pedestrians corresponding to heights over 50 pixels. Again, ExHoG ranks first at 59%. Fig. 3.14(e) plots the performance on partially occluded (1 - 35% occluded) pedestrians corresponding to heights over 50 pixels. ExHoG ranks first at 80%.

**Reasonable:** Fig. 3.14(f) plots the performance on reasonable condition that evaluates performance on pedestrians that are over 50 pixels tall under no or partial occlusion. ExHoG ranks first at 61%. ExHoG is able to handle low- and medium-resolutions
more robustly compared to the other methods and hence, this accounts for its strong performance under this condition.

Fig. 3.15b show some ExHoG detections on Caltech.

3.4 Discussion

ExHoG is derived by observing the inherent weaknesses of Histogram of Gradients (Hg) and Histogram of Oriented Gradients (Hog). Hg differentiates a bright human against a dark background and vice-versa. This increases the intra-class variation of humans. Hog maps gradients of opposite directions into the same bin of the histogram. As such, it is unable to differentiate some local structures and produces the same feature. ExHoG alleviates the weaknesses of the two features by considering both the sum and absolute difference of the Hg with the opposite gradients. ExHoG outperforms Hg and Hog.

In Chapters 5 and 6, further extensions of ExHoG will be shown. In Chapter 5, the usage of ExHoG for quadratic classification on subspace will be discussed which greatly improves performance. In Chapter 6, the application of ExHoG to other object detection problems will be shown and a parts-based modeling using the framework of [23] for ExHoG will also be discussed.
3.4. Discussion

(a) INRIA. Black bounding boxes indicate detection of pedestrians which are not considered for performance comparison.

(b) Caltech. Black bounding boxes indicate detection of pedestrians which are not considered for performance comparison.

Figure 3.15: Detection results of ExHoG on various data sets used in the experiments [Best viewed in colour]. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives. There are some missed detections (false negatives) in some of the images. These missed detections are due to small scales, partial occlusions or pose variations that were not seen in the training data.
Chapter 4

Discriminative Robust Local Binary and Ternary Patterns

This chapter describes a set of proposed novel edge-texture based features, Discriminative Robust Local Binary Pattern (DRLBP) and Discriminative Robust Local Ternary Pattern (DRLTP). The main conclusion is that DRLTP and DRLBP provides better detection performance compared to Local Binary Pattern (LBP) [83], Local Ternary Pattern (LTP) [101] and Robust Local Binary Pattern (RLBP) [82].

We start by describing the LBP, LTP and RLBP features and their limitations in Sect. 4.1. Section 4.2 provides details of the proposed DRLTP and DRLBP features and their advantages over LBP, LTP and RLBP. The overall performance results on benchmark data sets are quantified in Sect. 4.3. The chapter concludes with a discussion in Sect. 4.4.
4.1 Local Binary Pattern, Local Ternary Pattern and Robust Local Binary Pattern

4.1.1 Descriptor formation

Local Binary Pattern (LBP) is the most popular feature used for texture classification [83, 43, 61, 40, 91, 138]. It has also shown excellent performance for face detection problems [101, 137, 60, 41, 2]. The LBP [83] code at pixel location \((x,y)\) is computed as follows:

\[
LBP_{x,y} = \sum_{b=0}^{B-1} s(p_b - p_c)2^b, \tag{4.1}
\]

\[
s(z) = \begin{cases} 
1, & z \geq 0 \\
0, & z < 0 
\end{cases}
\]

where \(p_c\) is the pixel value at \((x,y)\), \(p_b\) is the pixel value in the \(b\)-th location on the circle of radius \(R\) around \(p_c\) and \(B\) is the total number of neighbouring pixels. If a position on the circle does not coincide with a pixel, the value at the position is estimated using bilinear interpolation from neighbouring pixels. For a block, a LBP histogram of \(2^B\) bins is computed for feature representation. In [83], it is shown that there are some patterns that occur more frequently than others and the number of state transitions between 0 and 1 for these patterns are at most two. Such patterns are termed as uniform patterns and the rest as non-uniform. For example, “1100 0001” and “0001 1000” are uniform codes while “1011 0100” is a non-uniform code as there are 6 state transitions between 0 and 1. By giving each unique uniform pattern a bin and collating all non-uniform patterns into a single bin, the number of bins is reduced accordingly. For \(B = 8\), the number of bins is reduced from 256 to 59. LBP is invariant to any monotonic changes to the image. Hence, it is robust to illumination and contrast variations. However, it is
sensitive to noise and small fluctuations of pixel values.

Therefore, Local Ternary Pattern (LTP) [101] has been proposed to handle this issue. In comparison to LBP, LTP has 2 thresholds which creates 3 different states for coding as compared to 2 in LBP. Within reasonable variations of pixel values, LTP is more resistant to noise and small variations compared to LBP. Like LBP, it has also been widely used for texture classification and face detection [27, 47, 62, 101, 134]. The LTP code at pixel location \((x, y)\) is computed as follows:

\[
LTP_{x,y} = \sum_{b=0}^{B-1} s'(p_b - p_c)3^b, \tag{4.2}
\]

\[
s'(z) = \begin{cases} 
1, & z \geq T \\
0, & -T < z < T \\
-1, & z \leq -T 
\end{cases}
\]

where \(T\) is a user-defined threshold. As defined by \(s'(z)\), LTP has 3 states while LBP has two. For a block, a LTP histogram of \(3^B\) bins is computed for feature representation. For \(B = 8\), the histogram has 6561 bins. This makes the feature dimensionality extremely high. Hence, in [101], the authors propose to split the LTP code into its “upper” and “lower” LBP codes. The “upper” code is computed as follows:

\[
ULBP = \sum_{b=0}^{B-1} f(p_b - p_c)2^b, \tag{4.3}
\]

\[
f(z) = \begin{cases} 
1, & z \geq T \\
0, & \text{otherwise} 
\end{cases}
\]
The “lower” code is computed as follows:

$$LLBP = \sum_{b=0}^{B-1} f'(p_b - p_c)2^b,$$

$$f'(z) = \begin{cases} 1, & z \leq -T \\ 0, & \text{otherwise} \end{cases}$$

By doing so, the dimensionality of the feature is reduced from 6561 bins to 512 bins. Using uniform LBP code representation, the number of bins is further reduced to 118 bins.

The steps in computing the LBP or LTP descriptors for human detection in this work are as follows:

**Gamma correction**

The input image is gamma corrected. This enhances the dynamic range in dark or shadowed regions while compressing the bright regions and highlights.

**Divide the image into grids of blocks**

An image window is scanned over the image with each window forming a LBP or LTP descriptor that describes the region. The window is split into grids of overlapping blocks of $N \times N$ pixels.

**Compute LBP or LTP histograms for each block**

A histogram of $2^B$ (LBP) or $3^B$ (LTP) bins for each block is computed. The block histogram is normalized using clipped square root of L1-Norm.

**Concatenate histograms over overlapping blocks**

The feature vectors from all the blocks are compiled to form a combined feature vector.
4.1.2 Limitations of LBP, LTP and RLBP

An issue with LBP and LTP is that they differentiate a bright human against a dark background and vice-versa as the codes for the 2 situations are different. This differentiation makes the intra-class variation of the humans larger. In Fig. 4.1(a) and 4.6(a), the 2 situations in a block are represented by LBP and LTP. As it can be seen, the LBP and LTP features for the 2 situations are completely different.

A solution for the above mentioned problem of LBP is proposed in [82] as Non-Redundant LBP. The authors propose mapping a LBP code and its complement to the minimum of the two. For instance, “1101 0101” and its complement, “0010 1010” will be treated as the same code “0010 1010” in the mapping. Hence, the code “1101 0101” becomes redundant as it is never used in the histogram. It can be seen that the states are changed from 0 to 1 or 1 to 0 during this mapping. By doing so, the feature is robust to the reversal in intensity between the background and the objects. Based on this robustness, in this thesis, we call this feature as Robust LBP (RLBP). RLBP can be computed as follows:

$$RLBP_{x,y} = \min\{LBP_{x,y}, 2^B - 1 - LBP_{x,y}\},$$  \hspace{1cm} (4.5)

Since the mapping reduces the number of codes by half, the number of bins for RLBP histogram is 128 for $B = 8$. Using uniform codes, the number of bins is reduced to 30. Fig. 4.1(b) illustrates how RLBP solves the issue of intensity reversal of object and background for LBP. It can be observed that for both situations, the RLBP feature is the same.

In order to solve the problem of brightness reversal of object and background of LBP, RLBP maps all LBP codes to the minimum of the code and its complement. However, this mapping function of RLBP makes it difficult for RLBP to differentiate some
Figure 4.1: Problem of Local Binary Pattern (LBP) and its solution by Robust LBP (RLBP). A dark object against a bright background and vice versa in a block produces 2 different features which increases the intra-class variation of humans. RLBP resolves this by mapping a LBP code and its complement to the minimum of the two.

local structures that are dissimilar. It is possible that 2 different structures may end up having a similar or even the same feature representation. This is illustrated in the second row of Fig. 4.2. This problem of RLBP is caused by merging the complement codes in the same block.
4.2 Discriminative Robust Local Binary and Ternary Patterns

4.2.1 The Proposed Discriminative Robust Local Binary Pattern

Besides texture, the human contour, which typically resides in high contrast regions between the human and the background in the absence of clutter and occlusions, also
contains discriminatory information. However, in order to be robust to illumination and contrast variations, Lbp does not differentiate between a weak contrast local region and a similar strong contrast one. It only captures the texture information of humans. The histogramming of Lbp codes only considers the frequencies of the codes i.e. the weight for each code in the block is the same. This form of histogram is unable to differentiate between similar regions of different contrast. Therefore, a weak contrast local region and a strong contrast one could have similar feature representations.

To mitigate this problem, a weighting scheme is proposed. Given an image window, following [9], the square root of the pixels is taken. Then, the first order gradients are computed. The gradient magnitude at each pixel is then computed and used to weigh its Lbp code. The stronger the contrast at the pixel, the larger the weight assigned to the Lbp code at that pixel. Consider a Lbp histogram for a $M \times N$ block. The value of the $i^{th}$ bin of the weighted Lbp histogram is as follows:

$$ h_{lbp}(i) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(LBP_{x,y}, i), $$

$$ \delta(m, n) = \begin{cases} 1, & m = n \\ 0, & \text{otherwise} \end{cases} $$

where $\omega_{x,y}$ is the gradient magnitude at location $(x,y)$.

We find that RLbp histogram can be simply created from (4.6) as follows:

$$ h_{rlbp}(i) = h_{lbp}(i) + h_{lbp}(2^B - 1 - i), \quad 0 \leq i < 2^B-1 $$

where $h_{rlbp}(i)$ is the $i^{th}$ bin value of RLbp.

In order to mitigate the issue of RLbp whereby, in the same block, all Lbp codes and their complements are mapped to the same bin, the following is proposed. Consider
the absolute difference between the bins of a LBP code and its complement to form absolute Difference of LBP (DLBP) histogram as follows:

\[ h_{dlbp}(i) = |h_{lbp}(i) - h_{lbp}(2^B - 1 - i)|, \quad 0 \leq i < 2^{B-1} \] (4.8)

where \( h_{dlbp}(i) \) is the \( i^{th} \) bin value of DLBP. For blocks that contain structures that have both LBP codes and their complements, DLBP assigns small values to the bins that the codes are mapped to. It differentiates these structures from those having no complement codes within the block.

The 2 histogram features, RLBP and DLBP, are concatenated to form Discriminative Robust LBP (DRLBP) as follows:

\[
h_{drlbp}(j) = \begin{cases} 
  h_{rlbp}(j), & 0 \leq j < 2^{B-1} \\
  h_{dlbp}(j - 2^{B-1}), & 2^{B-1} \leq j < 2^B 
\end{cases}
\]

\[
h_{drlbp}(j) = \begin{cases} 
  h_{lbp}(j) + h_{lbp}(2^B - 1 - j), & 0 \leq j < 2^{B-1} \\
  |h_{lbp}(j) - h_{lbp}(2^B - 1 - j)|, & 2^{B-1} \leq j < 2^B 
\end{cases}
\]

For \( B = 8 \), the number of bins for the block feature is 256. Using uniform pattern representation, the number of bins is reduced to 60.

The proposed DRLBP contains both edge and texture information in a single feature representation. Fig. 4.3 illustrates how DRLBP produces different features for the structures shown in Fig. 4.2. Hence, the proposed feature represent objects more discriminatively than RLBP. It also resolves the issue of intensity reversal of object and background as shown in Fig. 4.4.
4.2. Discriminative Robust Local Binary and Ternary Patterns

Figure 4.3: DRLbp representations of local structures in Fig. 4.2. DRLbp differentiates the local structure pairs, \{(a),(b)\}, \{(c),(d)\} and \{(e),(f)\} misrepresented by RLbp.
4.2.2 The Proposed Discriminative Robust Local Ternary Pattern

4.2.2.1 Robust Local Ternary Pattern

\( \text{LBP} \) is sensitive to noise and small fluctuations of pixel values [101]. \( \text{LTP} \) solves this problem of \( \text{LBP} \) using 2 thresholds to generate codes to represent the region. It is more resistant to small variations of pixel values and noise compared to \( \text{LBP} \). However, \( \text{LTP} \) also has the same problem as \( \text{LBP} \) whereby it differentiates a bright human against a dark background and vice-versa. \( \text{RLBP} \) [82] solves this problem for \( \text{LBP} \) by mapping an \( \text{LBP} \) code and its complement to the minimum of the two.
4.2. Discriminative Robust Local Binary and Ternary Patterns

Figure 4.5: Illustration of “upper” and “lower” LBP codes of LTP for 2 situations where the intensities are reversed. It can be seen that the “upper” and “lower” LBP codes are reversed for the 2 situations.

However, RLBp cannot be applied to the “upper” and “lower” LBP codes of LTP to solve the problem of LTP. Different from LBP, for a pair of human/background intensity inverted patterns, their “upper” LBP codes are not complements of each other. Similarly, their “lower” LBP codes are also not complements of each other. This is illustrated in Fig. 4.5 where 2 different cases of object/background inverted intensity pattern are shown. In Fig. 4.5(a)(i) and (a)(ii), a case illustrating a neighbourhood where all 3 states of LTP occur is shown. From the two LTP codes, it can be observed that the 2 patterns are simply intensity inverted. However, their corresponding “upper” LBP codes are not complements. Similarly, their corresponding “lower” LBP codes are also not complements. A similar observation is also seen in (b)(i) and (b)(ii) where only
2 states of \( \text{Ltp} \) are present. The “upper” and “lower” \( \text{Lbp} \) codes of the 2 \( \text{Ltp} \)s are not complements of each other. Hence, \( \text{RLbp} \) cannot be applied to the “upper” and “lower” \( \text{Lbp} \) codes of \( \text{Ltp} \) to obtain a feature that is invariant to the variation of a bright human against a dark background and vice-versa.

In order to alleviate this problem of \( \text{LTP} \), we need to analyze the 3-state definition of \( \text{Ltp} \) in (4.2). In \( \text{Ltp} \), there are 3 states: 1, 0 and -1. The state of 0 represent regions of small variations, noise and uniform regions. This state will not change when there is an inversion of brightness between the background and humans as the variations in the region representing this state remain the same. Therefore, for a pair of brightness inverted human/background patterns, only the state of -1 is inverted to 1 and vice-versa. Hence, for every \( \text{Ltp} \) code, we can find its corresponding inverted code. For instance, “-1-100 1100” has an inverted code “1100 -1-100”. If both codes are mapped to a same bin, a feature that is invariant to the reversal in intensity between the background and human can be obtained.

In this dissertation, the maximum of a \( \text{Ltp} \) code and its inverted representation is chosen as the feature. We name it as \textit{Robust \( \text{Ltp} \)} (\( \text{RLtp} \)). Mathematically, \( \text{RLtp} \) can be formulated as follows:

\[
\text{RLTP}_{x,y} = \max\{\text{LTP}_{x,y}, -\text{LTP}_{x,y}\}
\]  

(4.10)

The mapping defined for \( \text{RLtp} \) makes the feature \textit{robust} to the reversal in intensity between the background and humans. The \( \text{RLtp} \) code can then be split into “upper” and “lower” \( \text{Lbp} \) codes. The “upper” code can be expressed as follows:

\[
\text{URLBP} = \sum_{b=0}^{B-1} h(\text{RLTP}_{x,y,b}) 2^b,
\]  

(4.11)

\( h(z) = \begin{cases} 1, & z = 1 \\ 0, & \text{otherwise} \end{cases} \)
where \( RLTP_{x,y,b} \) represents the state value at the \( b \)-th location of the \( RLTP \) code. The “lower” code is computed as follows:

\[
LRLBP = \sum_{b=0}^{B-2} h'(RLTP_{x,y,b})2^b, \tag{4.12}
\]

\[
h'(z) = \begin{cases} 
1, & z = -1 \\
0, & \text{otherwise}
\end{cases}
\]

Here, \( LRLBP \) only has 7 bits as the state at \((B-1)\)-th location of \( RLTP \) is always 0 or 1.

Fig. 4.6(b) illustrates how \( RLTP \) alleviates the brightness reversal problem of human and background. It can be observed that for the two situations, the \( RLTP \) features are the same.

However, similar to \( RLBP \) in Section 4.1.2, \( RLTP \) also maps a \( LTP \) code and its inverted representation \textit{in the same block} to the same maximum value. This is illustrated in Fig. 4.7.

### 4.2.2.2 Discriminative Robust Local Ternary Patterns

\( LTP \) and \( RLTP \) are also robust to illumination and contrast variations and only capture texture information for human representation. Hence, the weighting scheme in Section 4.2.1 is also used. Consider a \( LTP \) histogram for a \( M \times N \) image block. The value of the \( k \)-th bin of the weighted \( LTP \) histogram is as follows:

\[
h_{LTP}(k) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(LTP_{x,y},k), \tag{4.13}
\]

It is not difficult to see that the \( RLTP \) histogram can be simply created from (4.13)
as follows:

$$h_{rltp}(k) = \begin{cases} h_{ltp}(0), & k = 0 \\ h_{ltp}(k) + h_{ltp}(-k), & 0 < k < \frac{2^B}{2} \\ \end{cases}$$

(4.14)

where $h_{rltp}(k)$ is the $k^{th}$ bin value of $RLTP$.

Similar to Section 4.2.1, we consider the absolute difference between the bins rep-
4.2. Discriminative Robust Local Binary and Ternary Patterns

Figure 4.7: Problem of Robust LTP (RLtp). 6 local structures are shown in the first row. RLtp make the different structures in (a)(i) and (ii) similar as shown in (a)(iii) and different structures in (b)(i) and (ii) similar as shown in (b)(iii). A similar problem can be observed in (c).

representing a LTP code and its inverted representation to form Difference of LTP (DLTP) histogram as follows:

\[ h_{dltp}(k) = |h_{ltp}(k) - h_{ltp}(-k)|, \quad 0 < k < \frac{3^B + 1}{2} \]  \hspace{1cm} (4.15)

where \( h_{dltp}(k) \) is the \( k^{th} \) bin value of DLTP.
Chapter 4. Discriminative Robust Local Binary and Ternary Patterns

RLtp and DLtp are concatenated to form Discriminative Robust Ltp (DRLtp) as follows:

\[
    h_{drltp}(l) = \begin{cases} 
    h_{rltp}(l), & 0 \leq l < \frac{3B+1}{2} \\
    h_{dltp}(l - \frac{3B-1}{2}), & \frac{3B+1}{2} \leq l < 3B \\
    h_{ltip}(l), & l = 0 \\
    h_{ltip}(l) + h_{ltip}(-l), & 0 < l < \frac{3B+1}{2} \\
    |h_{ltip}(l - \frac{3B-1}{2}) - h_{ltip}(\frac{3B-1}{2} - l)|, & \frac{3B+1}{2} \leq l < 3B
    \end{cases}
\]

Using (4.11) and (4.12), the DRLtp histogram is mapped to the split “upper” and “lower” Lbp histograms. Similar to DRLbp, DRLtp contains both edge and texture information in a single feature representation. Fig. 4.8 illustrates how DRLtp produces different features for the structures in Fig. 4.7. It also resolves the issue of brightness reversal of human and background as shown in Fig. 4.9.

4.2.3 Efficient computation of DRLtp using ULBP and LLBP

Using Ltp to find RLtp, DLtp and DRLtp is computationally intensive and requires a much larger storage requirement due to the large number of Ltp codes. For \( B = 8 \), the number of Ltp codes is 6561. In order to generate the RLtp and DLtp histograms from the Ltp histogram, there are 3280 addition and subtraction operations respectively. This is followed by 8 addition operations for each RLtp and DLtp code to find the “upper” Lbp code and 7 addition operations to find the “lower” Lbp code. If the “upper” and “lower” Lbp codes of RLtp and DLtp can be produced directly from the split Lbp codes of Ltp, the computational complexity and storage requirements will be greatly reduced.
Figure 4.8: DRL\textsubscript{TP} representations of local structures in Fig. 4.7. DRL\textsubscript{TP} differentiates the local structure pairs, \{(a), (b)\}, \{(c), (d)\} and \{(e), (f)\} misrepresented by RL\textsubscript{TP}. 

4.2. Discriminative Robust Local Binary and Ternary Patterns
The behaviour of \( ULBP \) from (4.3) and \( LLBP \) from (4.4) for human/background intensity inverted situations can be analyzed as follows. Suppose there is a bright human against a dark background. Consider a neighbourhood containing a human boundary. Assume that the centre pixel resides in the background. The differences between the human pixel values and the centre pixel value are larger than the threshold, \( T \). The differences between the background pixel values and the centre pixel value are in between \( T \) and \( -T \). The bits in \( ULBP \) corresponding to the human are 1 while the rest are 0. The bits in \( LLBP \) are all 0. If the brightness is now inverted, all bits in \( ULBP \) turn to 0 and the bits in \( LLBP \) corresponding to the human are 1 while the rest are 0.
We see that the brightness inversion turns the $LLBP$ code into $ULBP$ code and $ULBP$ code into $LLBP$ code.

Now, assume that the centre pixel does not belong to the background or human. Instead, it has a value between the bright human and dark background pixel values. The absolute differences of the human and the centre pixel and the background and the centre pixel are larger than $T$. The bits in $ULBP$ corresponding to the human are 1 while the rest are 0. The bits in $LLBP$ corresponding to the background are 1 while the rest are 0. If the intensity is now inverted for the situation, the bits in $ULBP$ corresponding to the background are all 1 while the rest are 0. Similarly, the bits in $LLBP$ corresponding to the human are 1 while the rest are 0. Again, we see that the intensity inversion turns the $LLBP$ code into $ULBP$ code and $ULBP$ code into $LLBP$ code.

From the above analysis, we find that the $ULBP$ and $LLBP$ codes for human/background intensity inverted situations are exchanged. If they are rearranged such that the “upper” and “lower” codes for both situations codes are the same, $RLtp$ is achieved. This can be done as follows. For any $Ltp$ code, the “upper” $Lbp$ code for $RLtp$ is defined as follows:

$$URLBP = \max\{ULBP, LLBP\},$$

The “lower” $Lbp$ code for $RLtp$ is defined as follows:

$$LRLBP = \min\{ULBP, LLBP\},$$

By producing these two rearranged $Lbp$ codes for any $Ltp$ code, $RLtp$ is obtained in the split $Lbp$ code representation. If $ULBP = 0$ and $LLBP = 0$, only 1 $Lbp$ result is considered and assigned to $LRLBP$. In Fig. 4.5(a) and (b), for each case, the $Lbp$ codes of the 2 intensity inverted $Ltp$ codes are reversed. For instance, in Fig. 4.5(a)(i), the
“upper” Lbp code is the “lower” Lbp code in (a)(ii). Similarly, the “lower” Lbp code is the “upper” Lbp code in (a)(ii). By following (4.17) and (4.18), we can obtain the URLBP and LRLBP easily from ULBP and LLBP for both cases.

Consider the ULBP and LLBP codes for an image block. The value of the $s^{th}$ bin, $0 < s < 2^B$, of URLBP can be generated from ULBP and LLBP codes as follows:

$$h_{urlbp}(s) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(\max(ULBP,LLBP),s),$$

(4.19)

The value of the $t^{th}$ bin, $0 \leq t < 2^{B-1}$, of LRLBP can be generated as follows:

$$h_{lrlbp}(t) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta(\min(ULBP,LLBP),t),$$

(4.20)

The split Lbp histograms, UDLBP and LDLBP, for DLtp can also be generated from the ULBP and LLBP codes. For every Ltp code whose ULBP and LLBP representations are swapped, the corresponding values of UDLBP and LDLBP bins are decremented by 1 accordingly. Otherwise, the bins are incremented by 1. The $s^{th}$ bin value of UDLBP is expressed as follows:

$$h_{udlbp}(s) = \left| \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta'(\lambda(ULBP,LLBP),s) \right|,$$

(4.21)

$$\lambda(p,q) = \begin{cases} p, & p > q \\ -q, & p < q \end{cases}$$

$$\delta'(m,n) = \begin{cases} 1, & m = n, m > 0 \\ -1, & |m| = n, m < 0 \\ 0, & \text{otherwise} \end{cases}$$

The function $\lambda$ determines whether the ULBP and LLBP codes are being swapped. If
a swap occurs, the negative maximum code is assigned to the result. The function $\delta'$ checks the value output from $\lambda$ with $s$. If the value is positive and matches $s$, the $s^{th}$ bin value is incremented. Otherwise, it is decremented. The $t^{th}$ bin value of LDLBP is determined as follows:

$$h_{ldlb}p(t) = \frac{1}{M-1} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \omega_{x,y} \delta''(\lambda'(ULBP, LLBP), t),$$

(4.22)

$$\lambda'(p, q) = \begin{cases} 
q, & p \geq q \\
-p, & p < q 
\end{cases}$$

$$\delta''(m, n) = \begin{cases} 
1, & m = n, m \geq 0 \\
-1, & |m| = n, m < 0 \\
0, & \text{otherwise} 
\end{cases}$$

The function $\lambda'$ determines whether the ULBP and LLBP codes are being swapped. If a swap occurs, the negative minimum code is assigned to the result. The function $\delta''$ checks the value output from $\lambda$ with $t$. If the value is zero or positive and matches $t$, the $t^{th}$ bin value is incremented. Otherwise, it is decremented. The URLBP, LRLBP, UDLBP and LDLBP histograms are then concatenated to form DRLtp.

A property of ULBP and LLBP is that any bit that is 1 in ULBP must be 0 in LLBP. Similarly, any bit that is 1 in LLBP must be 0 in ULBP. This property of ULBP and LLBP also applies to URLTP and LRLTP. Since RLtp is obtained from a max function, the most significant bit i.e. left-most bit of LRLBP must be 0. Thus, the number of possible codes of LRLBP is almost halved compared to URLBP. For $B = 8$, the URLBP have values in the range from 1 to 255. LRLBP only have values in the range from 0 to 127. Hence, the number of bins for uniform URLBP is 58 and that of uniform LRLBP is 30 (the bin representing 0 is in LRLBP). Hence, for a block, the total number of bins of RLtp using uniform “upper” and “lower” LBP is 88. Similarly, for DLtp using
Table 4.1: Number of hard negatives and feature dimensionality of each feature.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dimension size (Window)</th>
<th>Number of Hard Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP</td>
<td>6195</td>
<td>253 568</td>
</tr>
<tr>
<td>LTP</td>
<td>12390</td>
<td>129 276</td>
</tr>
<tr>
<td>RLBP</td>
<td>3150</td>
<td>302 753</td>
</tr>
<tr>
<td>DRLBP</td>
<td>6300</td>
<td>221 397</td>
</tr>
<tr>
<td>DRLTP</td>
<td>18480</td>
<td>80 780</td>
</tr>
</tbody>
</table>

uniform “upper” and “lower” LBP, the total number of bins is also 88. Overall, the number of bins of DRLTP is 176.

4.3 Overall Results

We perform experiments on two challenging data sets - INRIA [9] and Caltech Pedestrian Benchmark Data Set [15]. The performance of the proposed features is compared against some state-of-the-art single type of holistic feature methods on these given data sets. Results are reported for the INRIA and Caltech data sets using the per-image methodology suggested in [15] as the authors have shown it to be a better evaluation method. For both data sets, a block size of $16 \times 16$ pixels is used. A neighbourhood of 8 ($B$) pixels is considered using a circle of radius 1 ($R$). A 50% overlap of blocks are used in the construction of the features. Square root of L1-Norm normalization is used as preliminary experiments show that square root of L1-Norm gives the best results. The uniform pattern representation is used. The overlapping block features for the image window is concatenated to form the overall window feature for training the linear SVM classifier.

The training set of INRIA is used to train the classifiers for INRIA and Caltech data sets. The training data set contains 2416 cropped positive training images and 1218
4.3. Overall Results

uncropped negative training images. The sliding image window size is $128 \times 64$ pixels. We randomly take 10 samples from each negative image to obtain a total of 12,180 negative samples for training the linear SVM classifier. Bootstrapping is then performed on the negative images across multiple scales to obtain hard negatives which are combined with the original training set to retrain the SVM classifier. During bootstrapping, a scale step of 1.05 is used. This training procedure is exactly the same as the ones described in [9] and [15].

4.3.1 Performance comparison of DRLbp and DRLtp against Lbp, Ltp and RLbp

We perform experiments for Lbp, Ltp and RLbp on the INRIA data set to compare with DRLbp and DRLtp. In order to illustrate the benefit of the proposed weighting scheme, we also performed experiments on the INRIA data set for DRLbp and DRLtp without weighting. They are referred to as DRLbp W=1 and DRLtp W=1. For without weighting, we set $\omega_{x,y}=1$ for all pixels. For Ltp, DRLtp and DRLtp W=1 in our experiments, the threshold, $T$, is 3. Note that the RLbp feature representation in our work differs from the one in [82]. In our work, dense representation is adopted while sparse representation was used in [82] with probabilistic classification. Table 4.1 presents the number of hard negatives used for each feature for training the classifier.

The INRIA testing set consist of 288 images. The images are scanned using the trained classifiers over multiple scales. The scale step used is 1.05. The window stride is 8 pixels vertically and horizontally. These parameters are the same as those used in [15] for test. The miss rate (MR) against false positives per image (FPPI) is plotted for comparison between different detectors. The log-average miss rate (LAMR) [15] is used to encapsulate the detector performance which is found by averaging the miss rates at
nine FPPI locations evenly spaced in the range $10^{-2}$ to $10^0$. If any of the curves end before reaching $10^0$, the minimum miss rate achieved is used [15].

From Fig. 4.10, it is seen that our proposed features with weighting outperform its predecessors. $RLBP$ underperforms $LBP$ as there is a loss of information due to the mapping of $LBP$ codes and their complements to the same code. As a result, there are some structures misrepresented by $RLBP$ which results in a poorer performance. $DRLBP$ outperforms $RLBP$, $LBP$ and $DRLBP W=1$. $LTP$ outperforms $LBP$ thanks to its robustness to noise and small fluctuations of pixel values. Similarly, $DRLTP$ outperforms $LTP$ and $DRLTP W=1$. Overall, $DRLTP$ performs the best at 29%. In the subsequent sections, only $DRLBP$ and $DRLTP$ will be used for comparison against some other state-of-the-art detectors.

### 4.3.2 Results on INRIA Data Set

We compare the performance of $DRLBP$ and $DRLTP$ with VJ [111], SHAPELET [94], PoseInv [66], HikSvm [71] and Hog [9]. In order to keep the comparisons clearly within the domain of single type of holistic features, performance comparisons with methods that use hybrid/multiple features or parts-based model for human representation are omitted. The results of all compared detectors are given in [15]. These detectors are optimized, trained and tested by their respective authors. From Fig. 4.11, $DRLTP$ achieves a LAMR of 29% which is significantly lower than all the state-of-the-art methods being compared with. $DRLBP$ has a LAMR of 36%.

Fig. 4.13a and 4.14a show some $DRLBP$ and $DRLTP$ detections on INRIA.
4.3. Overall Results

![Performance graph](image)

Figure 4.10: Performance of DRLBP and DRLTP with/without weighting against LBP, LTP and RLBP [Best viewed in colour]. Without weighting (W=1), both DRLBP and DRLTP performs better than LBP, LTP and RLBP. With weighting, the performance improves further. DRLTP with weighting outperforms all other methods.

### 4.3.3 Results on Caltech Data Set

The Caltech data set [15] has color video sequences and pedestrians in a variety of scales and more scene variations. It has been created from a recorded video on a moving car through some densely populated human areas. As such, it contains artifacts of motion, blur and noise, and has various stages of occlusion (from almost complete to none). The data set is divided into 11 sessions. The first 6 sessions are designated as the training set while the remaining 5 are designated as the test set.

The authors in [15] reported results whereby they used detectors trained on other
Chapter 4. Discriminative Robust Local Binary and Ternary Patterns

Figure 4.11: Performance of DRLbp and DRLtp against existing existing approaches of dense representation on INRIA [Best viewed in colour]. DRLtp outperforms all other methods.

data sets like INRIA for classification on their test set. Here, the results are also presented in a similar manner to [15] where detectors are trained using the INRIA data set and tested on the test sessions. The scale step used is 1.05. The window stride is 8 pixels horizontally and vertically. The settings used here are similar to those used in [15]. Similar to [15], in order to detect humans at smaller scales, the original images are upscaled. Only every 30th frame is evaluated so that comparisons can be kept consistent with those in [15].

Detailed results are presented in Fig. 4.12. The detectors we compare with our implemented detectors are the same as those in Section 4.3.2. These detectors are optimized, trained and tested by their respective authors. The performance is analyzed
4.3. Overall Results

Figure 4.12: Performance under different conditions on test set of the Caltech Pedestrian Data Set [Best viewed in colour].

under six conditions as in [15]. Fig. 4.12 show the overall performance on the test set, on near and medium scales, under no and partial occlusions and on clearly visible pedestrians (reasonable). The MR versus FPPI [15] is plotted and LAMR is used for describing overall performance. The results are discussed under each condition in more details as follows.

**Overall:** Fig. 4.12(a) plots the performance on all 6 test sessions for every annotated pedestrian. DRLbp and DRLtp rank first at 86%. HOG is second at 90%. At lower FPPIs, DRLtp performs better than DRLbp. Thus, DRLtp is preferred as most detection systems require low false positive rates with lower miss rates.

**Scale:** Fig. 4.12(b) plots the performance on unoccluded pedestrians corresponding to heights over 80 pixels. Here, DRLtp performs the best at a LAMR of 30% with DRLbp the second best at 38%.

Fig. 4.12(c) plots the performance on unoccluded pedestrians corresponding to heights between 30 - 80 pixels. DRLbp ranks first in performance at 82% followed by DRLtp at 85%. At lower FPPIs, DRLtp performs better than DRLbp.

**Occlusion:** Fig. 4.12(d) plots the performance on unoccluded pedestrians corresponding to heights over 50 pixels. DRLtp ranks first at 56% and DRLbp ranks second at 60%. Fig. 4.12(e) plots the performance on partially occluded (1 - 35% occluded) pedestrians corresponding to heights over 50 pixels. DRLtp ranks first at 74% and DRLbp ranks second at 81%.

**Reasonable:** Fig. 4.12(f) plots the performance on pedestrians that are over 50 pixels tall under no or partial occlusion. DRLtp ranks first at 58% and DRLbp ranks second at 62%.

Fig. 4.13b and 4.14b show some DRLbp and DRLtp detections on Caltech.
4.3. Overall Results

(a) INRIA. Black bounding boxes indicate detection of pedestrians which are not considered for performance comparison.

(b) Caltech. Black bounding boxes indicate detection of pedestrians which are not considered for performance comparison.

Figure 4.13: Detection results of DRLbp on various data sets used in the experiments [Best viewed in colour]. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives. There are some missed detections (false negatives) in some of the images. These missed detections are due to small scales, partial occlusions or pose variations that were not seen in the training data.
Figure 4.14: Detection results of DRLTP on various data sets used in the experiments [Best viewed in colour]. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives. There are some missed detections (false negatives) in some of the images. These missed detections are either due to small scales, partial occlusions or pose variations that were not seen in the training data.
4.4 Discussion

This chapter has presented several key results. DRLbp outperforms Lbp and RLbp. DRLtp outperforms Ltp. This is because they resolve the issues that Lbp, RLbp and Ltp have with human detection. They do not differentiate a bright human against a dark background and vice-versa. Therefore, unlike Lbp and Ltp, they do not increase the intra-class variation of humans. Furthermore, by considering the weighted sum and absolute differences of the bins of the Lbp codes with their respective complement codes, DRLbp resolves the problem of RLbp whereby Lbp codes and their complements in the same block are mapped to the same value. Similarly, DRLtp resolves the problem of RLtp whereby Ltp codes and their inverted representations are mapped to the same value. In addition, the proposed features retain the contrast information of humans which contains discriminatory information.

In Chapter 6, the application of DRLbp and DRLtp to other object detection problems will be shown.
Chapter 5

Quadratic Classification Framework for Human Detection in Subspace

This chapter describes a nonlinear quadratic classification framework for human detection using the proposed features in Chapter 3. The distribution of the features in the feature space is analyzed and it is discovered that the boundary between human and non-human can be well approximated by a hyper-quadratic surface. The modified Minimum Mahalanobis Distance classifier (MMDC) [51] is proposed. However, for high-dimensional features, the estimated eigenvalues in some feature dimensions deviate greatly from that of the data population which results in overfitting of the quadratic classifier. Hence, there is a need to reduce the dimensions of the proposed features to minimize the overfitting problem. Furthermore, in human detection training sets, usually, there are much fewer positive samples than the negative ones which results in the negative covariance matrix being more reliable than the positive covariance matrix. Using Principal Component Analysis is inefficient as the unreliable dimensions from the less reliable covariance matrix are not effectively removed. In order to tackle the problems of dimensionality reduction and the asymmetry issue of human training sets, Asymmetric Principal Component Analysis (APCA) [51, 52] is proposed in this work for dimension reduction. As a result, the projected features allow for a
more robust classifier to be trained that less overfits the training data. In summary, the proposed classification framework enhances the detection performance on human detection for the proposed features and other features.

We start by describing the existing popular classifiers (Support Vector Machines and boosting-based) for human detection and their limitations in Sect. 5.1. Section 5.2 provides details of the proposed solution for the MMDC and the classification framework. Section 5.3 details the performance of the classification framework of ExHoG in this dissertation and compares the performance of the classification framework against other classification frameworks. The chapter concludes with a discussion in Sect. 5.4.

5.1 Popular Classifiers

5.1.1 Support Vector Machines

Support Vector Machines (SVM) [110] were developed primarily for the task of binary classification. Given 2 classes in the feature space, SVM classifiers find the optimal separating hyperplane that maximizes the margin between the data points of the classes at the boundary. The advantages of SVM classifiers are summarized as follows:

- SVM classifiers always obtain a global optimal solution using optimization theory. Unlike other classifiers which may have many local minima, the optimization problem of SVM is a quadratic problem and hence, a global solution is always obtained.

- To handle non-linearity of decision boundaries in the current feature space, using kernels, SVM classifiers can project the current data into higher dimensional feature space where the decision boundary could be linear and finds the hyperplane to separate the data. This does not explicitly increase the dimensionality of
the data. The kernel computations are still in the order of the current dimensions of the feature space.

- SVM classifiers tend to have better generalization capabilities with respect to unseen data with correct parameter settings.

- The statistical performance of SVM is independent of dimensionality of feature space. It only depends on the type of kernels and the number of support vectors.

SVM classifiers are most widely used for human detection [9, 15, 23, 25, 63, 66, 71, 72, 89, 90, 113, 115, 120, 136]. Non-linear kernels can be used with SVM classifiers for classification. However, the degree of non-linearity of the kernel SVM classifier is unknown and not easy to be controlled. Using an inappropriate degree of non-linearity of the kernel could lead to overfitting.

Furthermore, the computational complexity of non-linear SVM classifiers during classification heavily depends on the number of support vectors, the dimensionality of the features and the kernel that is used. Let $N_s$ the number of support vectors and $l$ the dimension of the samples. The number of operations is $O(MN_s)$ where $M$ is the number of operations required to evaluate the kernel. For polynomial and Gaussian kernels, $M$ is $O(l)$. For a human detection problem, $l$ and $N_s$ can be in values of several thousands. Hence, the complexity of non-linear SVM classifiers is extremely high.

In order to minimize the overfitting problem and for speed, linear SVM classifier is most widely used in human detection [9, 15, 23, 25, 63, 66, 88, 89, 90, 113, 115, 120, 136]. However, linear SVM classifiers only work well for distinct classification problems like cats versus dogs or cars versus motorcycles. For asymmetrical classification problems like human detection where it is one object versus all other objects, linear SVM may not perform well.
5.1.2 Boosting-based Classifiers

The idea behind boosting is to combine several weak classifiers in an ensemble such that their combined performance is improved. This combined ensemble classifier forms a \textit{strong} classifier from a series of combination of weak classifiers. Let \( h_1, h_2, \ldots, h_T \) be a set of weak classifiers and consider the ensemble decision as:

\[
f(x) = \sum_{t=1}^{T} \delta_t h_t(x)
\]

where \( \delta_t \) denotes the weight of the weak classifier and \( \sum_{t=1}^{T} \delta_t = 1 \). The weight, \( \delta_t \), and the weak classifier, \( h_t \), are learnt iteratively using a boosting procedure. Compared to SVM classifiers, cascade structured boosting-based classifiers enable very fast detection and the resultant strong classifiers are non-linear. There are many types of boosting-based classifiers used in human detection such as AdaBoost [14, 15, 10, 45, 75, 94, 111, 140], RealBoost [30, 37, 126], LogitBoost [39, 68, 109], MILBoost [12, 65, 112], Cluster Boosting Tree [125, 124], GentleBoost [67, 105, 135] and MPLBoost [114, 118].

However, boosting-based classifiers are based on a subset of features selected from a huge number of all possible features (usually numbering in hundreds of thousands or higher) by learning from the database. The feature subset selected by the classifiers may or may not yield a good classification result [57, 104], especially if the feature pool for selection is small. The work in this dissertation focuses on the further development of a single type of feature. It is not feasible to use a boosting-based classifier to select a feature subset from this limited number of specific features as the merit and the main strength of the boosting-based approach is to select a subset of features from a huge number of possible features.
5.2 Proposed Classification Framework

In human detection, the negative class comprises of all other classes that are not human. In the feature space, the positive class usually occupies a small space surrounded by the negative class. In order to illustrate this, the ExHoG features of the initial training set of 2416 positive samples and 12180 negative samples from INRIA [9] are projected to a lower-dimensional Asymmetric Principal Component Analysis (APCA) (Section 5.2.1) subspace. Fig. 5.1 shows the 2-dimensional scatter plots of the first 5 dimensions of the projected ExHoG features in the APCA subspace. Clearly, from the 4 scatter plots in Fig. 5.1, it is observed that a linear boundary is not optimal for separating the two classes. A hyper-quadratic boundary can be used to separate the two classes much better than a linear boundary. It is not difficult to understand the roughly quadratic surface boundary between the human class and the non-human class. Human class is one object while non-human class include all other objects. Therefore, the human samples are surrounded by the non-human samples in the feature space.

An example of a quadratic classifier is the Minimum Mahalanobis Distance classifier (MMDC) whose decision rule is given as follows:

\[
(X - \mu_n)^T \Sigma_n^{-1} (X - \mu_n) - (X - \mu_p)^T \Sigma_p^{-1} (X - \mu_p) > b, \tag{5.2}
\]

where \(X\) is the feature vector, \(\Sigma_n\) is the negative covariance matrix, \(\Sigma_p\) is the positive covariance matrix, \(\mu_n\) and \(\mu_p\) are the means of the negative and positive class respectively and \(b\) is a user-defined threshold for classification. The MMDC is the optimum or minimum error Bayes classifier for Gaussian distribution of the positive and negative data with arbitrary means and covariance matrices.

In general, the class means and covariance matrices of the human and non-human data population are unknown. It can only be estimated from the limited number of
5.2. Proposed Classification Framework

Figure 5.1: Distribution of ExHoG features in the APCA subspace [Best viewed in colour]. The first 5 dimensions (APCA 1, APCA 2, APCA 3, APCA 4 and APCA 5) after projection are chosen for the scatter plots. (a) shows the plot of APCA 1 vs APCA 2. (b) shows the plot of APCA 2 vs APCA 3. (c) shows the plot of APCA 3 vs APCA 4. (d) shows the plot of APCA 4 vs APCA 5. In (a) - (d), it can be clearly seen the positive samples occupy a small space while being surrounded by the negative samples.

...training samples. Hence, if the estimated variances deviate greatly from those of the data population, the MMDC will overfit the data as it uses the inverse of the covariance matrices. This will result in a poor generalization. This problem becomes very...
severe if the feature dimensionality is high [51, 52]. Therefore, in order to improve the classification performance, there is a need to reduce the feature dimensions so that the unreliable dimensions are removed before classification.

5.2.1 Dimensionality Reduction

Most dimensionality reduction approaches apply discriminant analysis (DA) [8, 44, 53, 73, 131, 133] to extract as few features as possible with minimum loss of discriminative information. They produce a portable feature vector for fast classification but are not directly targeted at solving generalization problem of the classifier in the high dimensional feature space. These approaches may improve the classification generalization if the highly nonlinear nearest neighbour (NN) classifier is applied. For human detection, there are only two classes and many training samples in each class. Obviously, the NN classifier is not feasible. As analyzed before, a quadratic classifier is devised in our system. The classification requires the inverse of the class-conditional covariance matrices. In the high dimensional space, the inverse of the class-conditional covariance matrices will cause a big classification problem as the unreliable small eigenvalues of the estimated class-conditional covariance matrices can largely deviate from the true values. Asymmetric Principal Component Analysis (APCA) [51, 52] is a dimensionality reduction technique that directly targets at solving such problem.

Suppose there are $q_p$ $l$-dimensional samples belonging to the positive class $\omega_p$ and $q_n$ samples belonging to the negative class $\omega_n$. It is studied in [51, 52] how Principal Component Analysis (PCA) can be used to enhance classification accuracy. The total scatter matrix, $\Sigma_t$, for the 2 classes is in fact a weighted linear combination of covariance matrices. If $\Sigma_t$ is decomposed such that $\Sigma_t = \Psi \Upsilon \Psi^T$ where $\Psi$ is the eigenvector matrix and $\Upsilon$ is the diagonal matrix containing the eigenvalues, the transformation...
5.2. Proposed Classification Framework

matrix, $\mathbf{\Psi}_m, \mathbf{\Psi}_m \in \mathbb{R}^{l \times m}, m < l$, of PCA keeps $m$ eigenvectors corresponding to the $m$ largest eigenvalues. Hence, PCA removes the unreliable dimensions of small eigenvalues [51, 52].

However, PCA does not remove the unreliable dimensions for classification. As $\Sigma_t$ is not constructed from the classifier point of view, PCA removes unreliable dimensions from the class more well-represented by the training samples. However, unreliable dimensions from the class less well-represented by the training samples should be removed. APCA solves this issue by weighting $\Sigma_p$ and $\Sigma_n$ differently. APCA proposes to define a covariance mixture to replace $\Sigma_t$ as follows:

$$
\Sigma_{t'} = \delta_p \Sigma_p + \delta_n \Sigma_n + \Sigma_m
$$

(5.3)

where $\delta_p + \delta_n = 1$, $\delta_p, \delta_n$ are the empirically estimated user-defined weights and $\Sigma_m$ is the covariance matrix of the class means. Typically, the less well-represented covariance matrix should have a larger weight so that unreliable dimensions from this class will be removed. This also addresses the asymmetry in the training data. In human detection, usually, the number of negative training samples far exceeds the number of positive training samples. Hence, a weight proportional to the number of negative training samples is assigned to the positive covariance matrix and vice-versa for the negative covariance matrix. The weights can then be fine-tuned using cross-validation. However, in the experiments of this paper, the weights are not fine-tuned so that a unified parameter setting is used for all data sets. $\delta_p$ is simply chosen to be proportional to the number of negative training samples and $\delta_n$ to be proportional to the number of positive training samples as follows:

$$
\Sigma_{t'} = \frac{1}{q_p + q_n} (q_n \Sigma_p + q_p \Sigma_n) + \Sigma_m
$$

(5.4)
Eigen-decomposition is performed on $\Sigma_{t'}$ as:

$$\Sigma_{t'} = \Phi \Lambda \Phi^T \quad (5.5)$$

and $m$ eigenvectors $\Phi$ are extracted from $\Phi$ corresponding to $m$ largest eigenvalues in $\Lambda$. The projected covariance matrices can then found as $\hat{\Sigma}_p = \Phi^T \Sigma_p \Phi$ and $\hat{\Sigma}_n = \Phi^T \Sigma_n \Phi$.

APCA removes the subspace spanned by the eigenvectors corresponding to the smallest eigenvalues of $\Sigma_{t'}$. By doing so, APCA removes the unreliable dimensions of both classes (more from the less reliable class) and keeps the large interclass distinction in the subspace spanned by the eigenvectors of the large eigenvalues of $\Sigma_{t'}$. As such, APCA alleviates the overfitting problem which lead to better generalization for the unknown query data [51, 52].

### 5.2.2 Quadratic classification in APCA subspace

After APCA, the eigenvalues in the APCA subspace are generally biased upwards. The bias is higher for the less well-represented class [51]. Hence, regularization of the covariance matrices are required for better classification. Classification is performed using a modified MMDC [51] which uses the regularized covariance matrices in the APCA space as follows:

$$(\hat{X} - \hat{\mu}_n)^T \hat{\Sigma}_{n'}^{-1} (\hat{X} - \hat{\mu}_n) - (\hat{X} - \hat{\mu}_p)^T \hat{\Sigma}_p^{-1} (\hat{X} - \hat{\mu}_p) > b, \quad (5.6)$$

where $\hat{X} = \Phi^T X$, $\hat{\mu}_n = \Phi^T \mu_n$, $\hat{\mu}_p = \Phi^T \mu_p$ and $\hat{\Sigma}_{n'} = \beta \hat{\Sigma}_n$. $\beta$, $0.5 \leq \beta \leq 2$, is a regularization parameter for the negative matrix. Compared to [51], the upper bound for $\beta$ is increased. For human detection training sets, the number of positive samples is
usually far smaller than the number of negatives samples. Therefore, the positive covariance matrix is less reliable than the negative covariance matrix. After APCA, the large eigenvalues of the positive covariance matrix are hence typically biased upwards more than the eigenvalues of the negative covariance matrix. As we need to suppress the positive covariance matrix, the weight of the negative covariance matrix, $\beta$, needs to be larger than 1.

5.3 Experimental Study

ExHoG is used to test the proposed classification framework. Experiments are carried out on two data sets - INRIA [9] and Caltech Pedestrian Data Set [15]. The performance of the proposed approach is compared against some of the state-of-the-art single type of holistic methods on these given data sets. Results are reported using the per-image methodology. The modified MMDC is used on the low-dimensional ExHoG. When presenting and discussing results, it will be referred to as ExHoG APCA.

A cell size of $8 \times 8$ pixels is used with a block size of $2 \times 2$ cells for ExHoG. The number of bins for each cell for ExHoG is 18. A 50% overlap of blocks are used in the construction of the feature vectors. The $Hg$ block feature is normalized using a clipping value of 0.08.

5.3.1 Training

For INRIA and Caltech data sets, the training set of INRIA is used to train the classifiers. The training data set contains 2416 cropped positive training images and 1218 uncropped negative training images. The sliding window size is $128 \times 64$ pixels.

In order to estimate the regularization parameter for the modified MMDC and to
determine the dimensions to reduce to, a 4-fold cross-validation is performed on the training set. The training set is created as follows. The negative images are scanned across different scales and at each scale, 7 samples are randomly selected from each negative image. A scale step of 1.2 is used during the multi-scale extraction. 47499 negative samples are obtained for training.

From the cross-validations experiments in Fig. 5.2, we find that $\beta = 1.4$ and a dimension of 200 give the best results for ExHoG. The selected $\beta$ is larger than 1 which indicates that eigenvalues of the positive covariance matrix are biased upwards more than the eigenvalues of the negative covariance matrix in the APCA subspace. This result verifies our analysis earlier whereby it was discussed that the positive covariance matrix will be less reliable due to the smaller number of training samples available. Hence, its large eigenvalues will be heavily biased upwards. In the INRIA training set, this is the case.

Using these found parameters, the modified MMDC is trained. Bootstrapping is performed on the negative images across multiple scales at a scale step of 1.05 to obtain 89400 hard negatives. They are then combined with the original training set to retrain the modified MMDC. The regularization parameter and the dimensions to reduce remain unchanged during the retraining of the classifier. This simplifies the training process.

### 5.3.2 Comparison of classifiers and features

The performance of ExHoG against Hog and/or Hg with linear SVM, HikSvm [71, 72] and APCA+MMDC is tested on INRIA. After cross-validation on INRIA, we find that $\beta = 1.4$ and a dimension of 200 give the best results for Hg. For Hog, $\beta = 1.25$ and a dimension of 200 give the best results.
5.3. Experimental Study

![Graph showing cross-validation results for different dimensions and parameters.]

Figure 5.2: Cross-validation results of different parameter for APCA [Best viewed in colour]. (a) Cross-validation results for different dimensions at $\beta = 1$. Dimension of 200 gives the best performance. (b) Cross-validation results for different $\beta$ at dimension of 200. $\beta = 1.4$ gives the best result.
The INRIA test set contains 288 images. We scan the images using the trained classifiers over multiple scales. The scale step used is 1.05. The window stride is 8 pixels horizontally and vertically. These parameters are the same as those used in [15] for test. To compare between different detectors, the miss rate (MR) versus false positives per image (FPPI) is plotted. In order to encapsulate the detector performance, the log-average miss rate [15] (LAMR) is used which is computed by averaging the miss rates at nine FPPI locations evenly spaced in the range \(10^{-2}\) to \(10^{0}\). If any of the curves end before reaching \(10^{0}\), the minimum miss rate achieved is used [15].

The results are shown in Table. 5.1. The table is divided into sections with each section corresponding to results of a particular classifier for both data sets. Rows 1 to 3 show the results of ExHoG against Hog and Hg using linear SVM classifiers. Rows 4 to 6 show those using our proposed classification framework. Rows 7 to 8 show those using HikSvm. In each section, it can be seen that ExHoG consistently outperforms Hog and/or Hg for a particular classifier for both data sets. This demonstrates that the proposed feature is better suited for human description compared to Hg and Hog.

Comparing between sections, the effectiveness of our proposed classification frame-
work can be compared against linear SVM and HikSvm. The proposed classification framework outperforms linear SVM for all 3 features. It also outperforms the non-linear HikSvm.

5.3.3 Results on INRIA Data Set

We compare the performance of ExHoG APCA with VJ [111], Shapelet [94], PoseInv [63], HikSvm and Hog [9]. In order to keep the comparisons clearly within the domain of single type of holistic features, performance comparisons with methods that use hybrid features or parts-based modeling for human representation are omitted. These detectors are optimized, trained and tested by their respective authors.

From Fig. 5.3, ExHoG APCA achieve a LAMR of 36% which is significantly lower than all the methods being compared with. Hog has a LAMR of 46% while HikSvm has a LAMR of 43%. Fig. 5.5 show some ExHoG APCA detections on INRIA in comparison to ExHoG.

5.3.4 Results on Caltech Data Set

The Caltech data set [15] has color video sequences and pedestrians in a variety of scales and more scene variations. It has been created from a recorded video on a moving car through densely populated human areas. As such, it contains artifacts of motion, blur and noise, and has various stages of occlusion (from almost complete to none). The data set is divided into 11 sessions. The first 6 sessions are designated as the training set while the remaining 5 are designated as the test set.

In [15], the authors reported results whereby they used detectors trained on other data sets like INRIA for classification on their test set. We also present our results in a similar manner to [15] where our detectors are trained using the INRIA data set.
Figure 5.3: Performance of ExHoG APCA against existing approaches on INRIA [Best viewed in colour]. ExHoG APCA outperforms almost all other methods.

(Section 5.3.1) and tested on the test sessions. The scale step used is the same as that used on the INRIA test set i.e. 1.05. The window stride is 8 pixels horizontally and vertically. The settings used here are similar to those used in [15]. Same as [15], in order to detect humans at smaller scales, the original images are upscaled. Only every $30^{th}$ frame is evaluated so that our comparisons is consistent with those in [15].

Detailed results are presented in Fig. 5.4. The detectors we compare with ExHoG APCA are the same as those in Section 5.3.3. These detectors are optimized, trained and tested by their respective authors. The performance is analyzed under six conditions as in [15]. Fig. 5.4 show the overall performance on the test set, on near and medium scales, under no and partial occlusions and on clearly visible pedestrians (rea-
5.3. Experimental Study

<table>
<thead>
<tr>
<th>False Positives Per Image</th>
<th>Miss Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^0</td>
<td>1.0</td>
</tr>
<tr>
<td>10^{-1}</td>
<td>0.9</td>
</tr>
<tr>
<td>10^{-2}</td>
<td>0.8</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>0.7</td>
</tr>
</tbody>
</table>

(a) Overall

(b) Near Scale

(c) Medium Scale

(d) No Occlusion

(e) Partial Occlusion

(f) Reasonable

Figure 5.4: Performance under different conditions on test set of the Caltech Pedestrian Data Set [Best viewed in colour].
sonable). As in [15], the MR versus FPPI is plotted and LAMR is used for summarizing performance. The results are discussed under each condition in more details as follows.

**Overall:** Fig. 5.4(a) plots the performance on all 6 test sessions for *every* annotated pedestrian. ExHoG APCA ranks first at 82%.

**Scale:** Fig. 5.4(b) plots the performance on unoccluded pedestrians corresponding to heights over 80 pixels. ExHoG APCA has a LAMR of 41%. Compared to Fig. 3.14, ExHoG APCA performs slightly worse compared to ExHoG. For both data sets (INRIA and Caltech), we use the same $b$ in Eq. (5.6) as we follow the criteria set by the authors of the Caltech data set [14, 15] where they simply use the original detectors trained on INRIA to test on their data set. However, ideally, $b$ is dependent on the data set and needs to be set accordingly. In our work, in order to ease the non-maximum suppression, we use $b$ to filter out or suppress negative windows before non-maximum suppression. Since we are using the same $b$ from INRIA and not optimized for Caltech, the results obtained after non-maximum suppression is suboptimal. $b$ is set too low such that there are too many negative windows. This is the reason the performance is poorer compared to ExHoG.

Fig. 5.4(c) plots the performance on unoccluded pedestrians corresponding to heights between 30 - 80 pixels. ExHoG APCA ranks first in performance at 75%. This highlights that at low- and medium-resolutions, our method using quadratic classifier in the APCA subspace is more robust in detecting pedestrians than other approaches in comparison. This is an important aspect as in pedestrian detection problems, it is necessary to detect humans further away from the vehicle more accurately so that there is ample time to react to prevent an accident. 30-80 pixel height of pedestrians is the most appropriate image resolution for pedestrian detection [15].
5.3. Experimental Study

Table 5.2: Summary of Runtime Performance of ExHoG (INRIA).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dimension</th>
<th>Extraction Speed (ms)</th>
<th>Classification Speed (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Linear SVM</td>
<td>HikSvm</td>
</tr>
<tr>
<td>ExHoG</td>
<td>7560</td>
<td>0.19</td>
<td>46.73</td>
</tr>
<tr>
<td>Hog</td>
<td>3780</td>
<td>0.13</td>
<td>23.37</td>
</tr>
<tr>
<td>Hg</td>
<td>7560</td>
<td>0.19</td>
<td>46.73</td>
</tr>
</tbody>
</table>

Occlusion: Fig. 5.4(d) plots the performance on unoccluded pedestrians corresponding to heights over 50 pixels. Again, ExHoG APCA ranks first at 55%. Fig. 5.4(e) plots the performance on partially occluded (1 - 35% occluded) pedestrians corresponding to heights over 50 pixels. ExHoG APCA ranks first at 81%. However, compared to Fig. 3.14, ExHoG APCA performs slightly worse compared to ExHoG. The training data consist purely of unoccluded humans from INRIA. During APCA, the reduced feature dimensions obtained are based on these unoccluded humans. However, during testing on Caltech, due to occlusions, there is significant information loss in most dimensions of the feature vector. After projection for classification, there is insufficient information in the resulting projected feature vector for detecting humans correctly. This is the reason the performance is poorer compared to ExHoG.

Reasonable: Fig. 5.4(f) plots the performance on pedestrians that are over 50 pixels tall under no or partial occlusion. ExHoG APCA ranks first at 58%. ExHoG APCA is able to handle low- and medium-resolutions more robustly compared to the other methods and hence, this accounts for its strong performance under this condition.

Fig. 5.6 show some ExHoG APCA detections on Caltech in comparison to ExHoG.
Figure 5.5: Detection results of ExHoG APCA against ExHoG on INRIA. Rows 1 and 3 show ExHoG detections. Rows 2 and 4 show ExHoG APCA detections on same images. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives.

5.3.5 Runtime Performance

Table 5.2 summarizes the extraction and classification time per window of ExHoG versus Hg and Hog for INRIA. The off-line training time of APCA+MMDC is approximately 12 hours which is slower than linear SVM but much faster than HIKSVM which takes around a day to train. Classification using MMDC is slower than linear SVM but is much faster than HIKSvm which is a fast non-linear kernel SVM classifier. For online
5.3. Experimental Study

Figure 5.6: Detection results of ExHoG APCA against ExHoG on Caltech. Black bounding boxes indicate detection of pedestrians which are not considered for performance comparison. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives. Rows 1 and 3 show ExHoG detections. Rows 2 and 4 show ExHoG APCA detections on same images.

human detection, both the computational and memory complexity is $O(l \cdot m)$ for the dimensionality reduction and $O(m^2)$ for MMDC. Overall, both the computational and memory complexity of APCA+MMDC is $O(l \cdot m + m^2)$. 
5.4 Discussion

This chapter has presented several key results. We propose a novel non-linear classification framework that is derived from analyzing the distribution of features in the feature space. It is found that the boundary between humans and non-humans can be described by a hyper-quadratic surface. Hence, modified Minimum Mahalanobis Distance classifier (MMDC) is proposed to be used which is a quadratic classifier. However, the class means and covariance matrices of the data population of humans and non-humans are unknown. It can only be estimated from the training samples which are limited in number. Hence, if the estimated variances (eigenvalues) deviate greatly from those of the data population, the MMDC will overfit the data as it uses the inverse of the covariance matrices in its classification. This will result in a poor generalization. This problem becomes very severe if the feature dimensionality is high. Therefore, in order to improve the classification performance, Asymmetric Principal Component Analysis (APCA) is used to reduce the dimensions of the features before classification. APCA is a dimensionality reduction technique that directly targets at solving such problem. Furthermore, APCA also considers the asymmetry present in the training sets of humans where there are only a few human samples and a huge number of negative samples by asymmetrically weighing the covariance matrices of each class. We demonstrate the effectiveness of our classification framework on several data sets using the proposed features in this thesis and against other classification frameworks.
Chapter 6

Extensions

This chapter describes extensions of the proposed features in Chapter 3 and 4 to general visual object recognition tasks. Here, the features are tested on some benchmark data sets of object detection and image classification to demonstrate that the feature are not just suited for human detection only. Furthermore, we also present a parts-based modeling approach using ExHoG for object detection using the framework of Felzenszwalb et. al. [25] and demonstrate superior results.

We start by describing the challenges associated with object detection and image classification tasks in Sect. 6.1. Section 6.2 provides results on some object detection and image classification data sets (University of Illinois at Urbana-Champaign (UIUC) Car, Caltech 101 and Caltech 256). Section 6.3 details the parts-modeling approach using ExHoG for object detection using the framework of Felzenszwalb et. al. [25] and results are shown for INRIA and PASCAL VOC 2007 data set to demonstrate its effectiveness. The chapter concludes with a discussion in Sect. 6.4.
6.1 Challenges with Object Recognition

Image classification and object detection are 2 essential parts of object recognition. In image classification, the objective is to classify a given image containing an object into one of several predefined categories. The goal of object detection is to distinguish objects of interest from the background in an image.

However, there are various challenges associated with object recognition. Typically, objects need to be detected in images containing cluttered, noisy backgrounds and other objects. Furthermore, the objects need to be detected indoors and outdoors in different illumination and contrast environments. The images could be under or over-exposed and specularities, shadows, noise, etc., are common. Objects may appear in images with different viewpoints and scale and the appearance of complex object classes can vary significantly with viewing angle. Moreover, within a given class, objects can exhibit widely varying appearances. For example, classes such as person, cat, dog, etc., have a wide range of articulated poses and non-rigid deformations.

Object detection remains somewhat harder compared to image classification because: (a) despite pose and appearance variations, occlusions, truncations, etc., the object locations and sizes need to be determined accurately; (b) it has to detect each object in the image regardless of the number; (c) the criteria for success can vary between applications for example, a small person in the background might be irrelevant for indoor surveillance purposes but highly relevant for a pedestrian detection application.

Proper feature representation is a very crucial step in any object recognition system. A highly descriptive feature can help to improve the performance of a computer vision system by discriminating the object from the background or other objects in different lightings and scenarios. In addition, a good feature also helps to ease the classification
framework by allowing the use of simpler and less computationally intensive classifiers.

In the subsequent section, we demonstrate the effectiveness of our proposed features (DRL_{tp}, DRL_{bp} and ExHoG) in this thesis for general visual object detection and image classification tasks.

### 6.2 Application and Results

In this section, we demonstrate the applications of DRL_{tp}, DRL_{bp} and ExHoG on some data sets for object detection and image classification - University of Illinois at Urbana-Champaign (UIUC) Car [1], Caltech 101 [26, 22] and Caltech 256 [38]. For all data sets for DRL_{tp} and DRL_{bp}, a neighbourhood of 8 (B) pixels is considered using a circle of radius 1 (R). A 50\% overlap of blocks are used in the construction of the features. Square root of L1-Norm normalization is used as preliminary experiments show that square root of L1-Norm gives the best results. The uniform pattern representation is used. For ExHoG, for all data sets, a cell size of 8 × 8 pixels is used with a block size of 2 × 2 cells. The number of bins for each cell for ExHoG is 18. A 50\% overlap of blocks are used in the construction of the feature vectors. The Hg block feature is normalized using a clipping value of 0.08.

All the image classification experiments for each data set are repeated 10 times with different randomly selected training and test images (15 and 30) and the average of per-class recognition rates is recorded for each run. The mean and standard deviation of the results from individual runs is reported as the final results. The framework of [56] is used in the image classification experiments where the dense SIFT features are replaced with the proposed features.
Table 6.1: Recognition Accuracy at Equal Error Rate on UIUC Cars data set. The results of HïkSvm for test set II is not provided in [71].

<table>
<thead>
<tr>
<th>Method</th>
<th>Performance(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test Set I</td>
</tr>
<tr>
<td>DRLtp</td>
<td>99.5</td>
</tr>
<tr>
<td>DRLbp</td>
<td>98.3</td>
</tr>
<tr>
<td>ExHoG</td>
<td>96.8</td>
</tr>
<tr>
<td>HïkSvm [71]</td>
<td>98.5</td>
</tr>
<tr>
<td>Agarwal et al. [1]</td>
<td>79.0</td>
</tr>
<tr>
<td>Mutch et al. [80]</td>
<td>99.9</td>
</tr>
<tr>
<td>Hae et al. [98]</td>
<td>88.1</td>
</tr>
</tbody>
</table>

Linear SVM classifier is used for all experiments. For the image classification experiments, the linear SVM classifiers are trained using a one-versus-all rule i.e. a classifier is trained to separate each class from the rest and a test image is assigned the label of the classifier with the highest response.

6.2.1 University of Illinois at Urbana-Champaign (UIUC) Car

This data set contains side views of cars partly taken with a camera and partly by extracting frames from video. The cars in this data set are of different resolutions and contain instances of partial occlusion, low contrast and highly noisy and textured background. The training set contains 550 cars and 500 non-car images of $40 \times 100$ pixels. A block size of $8 \times 8$ pixels is used for DRLtp and DRLbp.

There are 2 test sets. The first set consists of 170 images containing 200 cars which are of the same size as those in the training set i.e. single scale. The second set is made up of 108 images containing 139 cars which are of different sizes ranging from 0.8 to 2 times the size of the cars in the training set i.e. multi scale. During testing, the window stride is 5 pixels in the horizontal and 2 pixels in the vertical direction.
Table 6.2: Performance on Caltech 101 data set.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Performance (%)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(15 training)</td>
<td>(30 training)</td>
</tr>
<tr>
<td>DRLtp</td>
<td>72.59 ± 1.58</td>
<td>80.41 ± 0.98</td>
</tr>
<tr>
<td>DRLbp</td>
<td>57.64 ± 1.94</td>
<td>66.95 ± 0.62</td>
</tr>
<tr>
<td>ExHoG</td>
<td>61.26 ± 1.91</td>
<td>71.83 ± 1.30</td>
</tr>
<tr>
<td>Lazebnik et al. [56]</td>
<td>56.40</td>
<td>64.40 ± 0.80</td>
</tr>
<tr>
<td>HikSvm [71]</td>
<td>50.15 ± 0.61</td>
<td>56.49 ± 0.78</td>
</tr>
<tr>
<td>Feature Context [116]</td>
<td>69.63 ± 0.84</td>
<td>77.09 ± 0.74</td>
</tr>
<tr>
<td>Boiman et al. [5]</td>
<td>65.00 ± 1.14</td>
<td>70.04</td>
</tr>
<tr>
<td>Gehler et al. [35]</td>
<td>70.00</td>
<td>77.70 ± 0.30</td>
</tr>
<tr>
<td>Yang et al. [132]</td>
<td>67.00 ± 0.45</td>
<td>73.20 ± 0.54</td>
</tr>
</tbody>
</table>

Results are reported for UIUC Car data set using performance at Equal Error Rate (EER). Table 6.1 presents the performance at Equal Error Rate (EER) of DRLtp, DRLbp and ExHoG against some state-of-the-art methods for both test sets. In comparison with the other state-of-the-art methods, DRLtp achieves the best performance for multi scale test set and is the second only to [80] for the single scale test set. In [80], the features are created from multiple convolutions with 8 filters (4 Gabor and 4 3-D Max filters) over 10 scales. This is much more computationally intensive compared to our method.

Fig. 6.1 show some examples of DRLtp, DRLbp and ExHoG detections on both test sets at their respective EERs.

6.2.2 Caltech 101

Caltech 101 contains 101 different object classes like animals, vehicles, etc. with significant variance in shape and an additional “background” class. Examples of some object classes are shown in Fig. 6.2. The number of images per class vary from 31 to 800. The classifiers are tested on 15 and 30 random test samples per class. A block size
Figure 6.1: Detection results of DRL$\text{tp}$, DRL$\text{bp}$ and ExHoG on various data sets used in the experiments [Best viewed in colour]. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives.
6.2. Application and Results

Figure 6.2: Examples of some classes in Caltech 101 for category classification. It can be seen the classes are very different from each other and within the class, there are large intra-class variations.

of $16 \times 16$ pixels is used. Table 6.2 shows the performance of the proposed features against some of the state-of-the-art approaches. DRL$_{tp}$ outperforms all approaches for both 15 and 30 training samples. Furthermore, the proposed features outperform [56] which highlights that the proposed features are more robust compared to dense SIFT. It should be noted that, in [35], a combination of features is used with a boosting-based classifier. Even so, DRL$_{tp}$ with linear SVM outperforms it significantly.
Table 6.3: Performance on Caltech 256 data set.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Performance (%) (15 training)</th>
<th>Performance (%) (30 training)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL\text{tp}</td>
<td>$72.34 \pm 0.47$</td>
<td>$81.89 \pm 0.53$</td>
</tr>
<tr>
<td>DRL\text{bp}</td>
<td>$51.14 \pm 1.37$</td>
<td>$60.80 \pm 0.51$</td>
</tr>
<tr>
<td>ExHoG</td>
<td>$50.99 \pm 0.73$</td>
<td>$62.84 \pm 0.61$</td>
</tr>
<tr>
<td>Boiman \textit{et al.} [5]</td>
<td>35.00</td>
<td>42.00</td>
</tr>
<tr>
<td>Gehler \textit{et al.} [35]</td>
<td>34.00</td>
<td>45.80</td>
</tr>
<tr>
<td>Yang \textit{et al.} [132]</td>
<td>$27.73 \pm 0.51$</td>
<td>$34.02 \pm 0.35$</td>
</tr>
</tbody>
</table>

6.2.3 Caltech 256

Caltech 256 contains 256 different object classes with higher intra-class variability and object location variability than Caltech 101 and an additional “background” class. Examples of some object classes are shown in Fig. 6.3. The number of images per class vary from 80 to 827. The classifiers are tested on 15 and 30 random test samples per class. A block size of $16 \times 16$ pixels is used. Table 6.3 shows the performance of the proposed features against some of the state-of-the-art approaches. DRL\text{tp}, DRL\text{bp} and ExHoG outperform all approaches for both 15 and 30 training samples.

6.2.4 Discussion

Though ExHoG, DRL\text{bp} and DRL\text{tp} were developed for improving human detection, the performance of these features on UIUC Car, Caltech 101 and Caltech 256 have also shown that they are equally suitable for other object detection or classification tasks. This is not surprising as most object detection scenarios only require detection of the object in question regardless of the nature of the intensity of the background. For example, in Fig. 6.1, the cars can be bright or dark with respect to the background. Hence, a detector needs to be able to detect the cars regardless of the nature of the intensity of the background. Furthermore, since the proposed features are able to
distinguish patterns (Fig. 3.9, Fig. 4.4 and Fig. 4.8) that others cannot, we are able to get improved performance. It should be noted that the patterns discussed in Fig. 3.9, Fig. 4.4 and Fig. 4.8 are not unique to humans alone. They also exist in other objects.
6.3 Parts-based Representation using ExHoG

ExHoG, in a holistic feature representation, was proposed for human detection to alleviate the issues of Histogram of Gradients (Hg) and Histogram of Oriented Gradients. In comparison to other holistic feature representations, ExHoG consistently performed better. However, in the presence of occlusions and variability of human articulations, it performed sub-optimally in comparison to parts-based feature approaches [25, 3]. Here, we adopt the framework of [25] to perform a parts-based modeling using ExHoG for object detection. The Hog feature in [25] is replaced with ExHoG and a discriminative deformable part model is then trained using the latent SVM.

6.3.1 PCA and Analytic Dimensionality Reduction

In [25], Principal Component Analysis (PCA) [51, 52] was performed on the Hog cell features. It was discovered that the top few eigenvectors had some special properties that allowed for some dimension reduction without loss of information. Since Hog is being replaced with ExHoG in this work, PCA is also performed to analyze the top few eigenvectors. The eigenvectors are displayed in Fig. 6.4.

ExHoG is defined using four different normalizations of a 18-dimensional histogram over orientations for each cell. Therefore, it can be viewed as $4 \times 18$ matrix for a cell. The top 28 eigenvectors in Fig. 6.4 exhibit a special structure similar to the ones in [25]: They are each (approximately) constant along each row or column of their matrix representation. Hence, they lie (approximately) in a linear subspace defined by sparse vectors that have ones along a single row or column of their matrix representation. Therefore, the analytic dimensionality reduction as defined in [25] can be used to reduce the dimensions of ExHoG in a similar manner.
Figure 6.4: PCA of ExHoG features. Each eigenvector is displayed as a $4 \times 18$ matrix so that each row corresponds to one normalization factor and each column to one orientation bin. The eigenvalues are displayed on top of the eigenvectors.
In [25], a concatenation of Hg and Hog features was used. In our implementation, ExHoG is concatenated with Hg. Hence, for each cell, following the analytic dimensionality reduction in [25], the cell feature is recomputed by summing the 4 histograms of ExHoG and Hg and concatenating the four normalization factors (of Hg) to form a 40-dimensional vector. An ExHoG feature pyramid is then defined by computing ExHoG features of each level of a standard image pyramid. Features at the top of the pyramid capture coarse gradients histogrammed over large areas of the image while features at the bottom of the pyramid capture finer gradients histogrammed over small areas.

### 6.3.2 Parts-based representation

The parts-based representation adopted for ExHoG is described in brief as follows. For complete details on the representation, readers are referred to [25]. Every object is represented by a mixture of models, \( M = (M_1, \ldots, M_m) \) where \( M_c \) is the model for the \( c \)-th component and \( m \) is the number of components. A singular object model consists of a global “root” filter (which covers the entire object) and several higher resolution part models covering smaller parts of the object. Each part model specifies a spatial model and a part filter. The spatial model defines a set of allowed placements for a part relative to a detection window, and a deformation cost for each placement. All of the root and part models involve linear filters which are rectangular templates specifying weights for subwindows of an ExHoG pyramid. They are applied to a dense ExHoG array whose entries are feature vectors computed from a dense grid of locations in an image.

Let \( F \) be a \( w \times h \) filter. Let \( E \) be a ExHoG feature pyramid and \( p = (x, y, l) \) specify a position \((x, y)\) in the \( l\)-th level of the pyramid. Let \( \phi(E, p, w, h) \) denote the vector
obtained by concatenating the feature vectors in the \( w \times h \) subwindow of \( E \) at \( p \). The score of \( F \) at \( p \) is \( F' \cdot \phi(E, p, w, h) \), where \( F' \) is the vector obtained by concatenating the weight vectors in \( F \). Subsequently, \( w \) and \( h \) are dropped from the notations here on as these dimensions are already defined by the dimensions of \( F \).

An object model \( M_c \) with \( n_c \) parts is defined by a root filter, \( F_0 \), a set of part models \((P_1, ..., P_{n_c})\) where \( P_i = (F_i, v_i, d_i) \) and a bias term, \( b \). \( F_i \) is a filter for the \( i \)-th part, \( v_i \) is a two-dimensional vector specifying a position for part \( i \) relative to the root position, and \( d_i \) is a four-dimensional vector representing coefficients of a part deformation cost quadratic function.

The placement of a mixture model component \( M_c \) in an ExHoG pyramid is defined as \( z = (p_0, ..., p_{n_c}) \), where \( p_i = (x_i, y_i, l_i) \) specifies the level and position of the \( i \)-th filter. The score of a placement \( z \) can be expressed as a dot product, \( \beta_c \cdot \phi(E, z) \), of a vector of model parameters \( \beta_c \) and a vector \( \phi(E, z) \). \( \beta_c \) and a vector \( \phi(E, z) \) are defined as follows:

\[
\beta_c = (F'_0, ..., F'_{n_c}, d_1, ..., d_{n_c}, b),
\]

\[
\phi(E, z) = (\phi(E, p_0), ..., \phi(E, p_{n_c}), -\phi_d(dx_1, dy_1), ..., -\phi_d(dx_{n_c}, dy_{n_c}), 1),
\]

where \((dx_i, dy_i)\) is the displacement of the \( i \)-th part relative to \( v_i \) and \( \phi_d(dx_i, dy_i) \) are the deformation features. This relationship is used to learn the model parameters with the latent SVM framework. To detect objects in an image, the overall score is computed for each root location according to the best possible placement of the parts. High-scoring root locations define detections, while the locations of the parts that yield a high-scoring root location define a full object placement.
Figure 6.5: One-component model with 8 parts learned using INRIA. The model is defined by a coarse root filter (left most), several higher resolution part filters (middle) and a spatial model for the location of each part relative to the root (right most). The visualization of the spatial models reflects the cost of placing the center of a part at different locations relative to the root.

6.3.3 Results

Experiments are performed on two data sets - INRIA Human [9] and PASCAL VOC 2007 [21] - to demonstrate the effectiveness of parts-based modeling using ExHoG. Results are reported for INRIA using the per-image methodology suggested in [15]. For PASCAL VOC 2007, the average precision (AP) of a precision-recall curve for each object class is reported. For both data sets, a cell size of $8 \times 8$ pixels is used with a block size of $2 \times 2$ cells. The number of bins for each cell for ExHoG is 18. A 50% overlap
of blocks are used in the construction of the feature vectors. The Hc block feature is normalized using a clipping value of 0.2. The number of parts is set at 8. A latent SVM [25] classifier is used for both data sets. The training procedure in [25] is followed closely to train our root and part-detectors which includes mining for hard negatives to re-train the detectors. The overall detector is referred to as ExHoG LatSvm-V2.

6.3.3.1 INRIA Human

A one-component model for detecting humans is trained for INRIA using the INRIA training set which consists of 614 positive images and 1218 negative images. The slid-
ing window size is $128 \times 64$ pixels. Fig. 6.5 shows the model learned. The INRIA test set consist of 288 images. The images are scanned over multiple scales at a scale step of 1.07. The window stride is 8 pixels. The miss rate against false positives per image is plotted to compare between different detectors. The log-average miss rate (LAMR) [15] is used to summarize the detector performance.

The performance of ExHoG LatSvm-V2 is compared with LatSvm-V2 [25], CrossTalk [16] and VeryFast [4] in Fig. 6.6. ExHoG LatSvm-V2 achieves a LAMR of 13% which is significantly lower than all the methods being compared with. Fig. 6.7 show some ExHoG LatSvm-V2 detections on INRIA.

### 6.3.3.2 PASCAL VOC 2007

The PASCAL VOC 2007 data set contains 20 object classes and 9,963 images, containing 24,640 annotated objects. The data set is split into two parts: 5011 images for training and 4952 images for testing. A three component-model is trained for each class. Fig. 6.8 shows the model learned for 2 object classes.

During testing, the images are scanned over multiple scales at a scale step of 1.07. The window stride is 8 pixels in the $x$ and $y$ directions. The bounding boxes of all objects of a given class in an image are predicted. A bounding box is considered correct if it overlaps more than 50 percent with a ground-truth bounding box. Multiple detections are penalized. If several bounding boxes that overlap with a single ground-truth bounding box are predicted, only one prediction is considered correct. The others are considered false positives. We rescore detections using contextual information similar to [25]. This procedure has been noted to improve AP on several object classes.

Table 6.4 summarizes the results (AP score in %) of ExHoG LatSvm-V2 against
6.3. Parts-based Representation using ExHoG

Figure 6.7: Detection results of ExHoG LatSvm-V2 on INRIA. True detections are indicated in green while false positives are in red. The ground truth of each data set is used to determine the true detections and false positives.

LatSvm-V2\textsuperscript{1}. Both ExHoG LatSvm-V2 and LatSvm-V2 suffer a drop in performance in one object class each after rescoring using contextual information (sheep for ExHoG LatSvm-V2 and table for LatSvm-V2). The contextual information used to rescore annotated object detection is a 25-dimensional feature vector which consists of the normalized original score of the detection by the object detector, the top-left and bottom-right bounding box coordinates and the image context. The image context is a 20-

\textsuperscript{1}These results are obtained from the website of the authors of [25] at http://people.cs.uchicago.edu/ rgb/latent/. They are the most updated based on the latest version of their code.
dimensional vector which is made up of normalized scores of all 20 different object detectors of the annotated object. A quadratic SVM is used for the rescoring of detected objects. Based on this, it is not difficult to see that there are situations where the false positive scores (i.e. the scores of detectors that are not of the object class) are actually larger than the score of the object detector. When this happens, the correctly detected true positives becomes missed detections after contextual rescoring. Hence, some categories get worse with context. Out of the 20 object categories, ExHoG LatSvm-V2 performs better or same (only for boat category) than LatSvm-V2 in 16 object categories. This demonstrates that ExHoG can also be applied for detection of other objects besides humans and can achieve a better average detection performance.
6.4 Discussion

This chapter has presented several key results. We demonstrate the performance of the proposed features for general visual object recognition and show that the proposed features are not only suited for human detection purposes and be applied for other object detection and image classification tasks. In comparison to state-of-the-art approaches, the proposed features give an improved performance. Furthermore, we also present a parts-based modeling approach using ExHoG for object detection using the framework of Felzenszwalb et. al. [25] and demonstrate superior results on INRIA and PASCAL VOC 2007.
Figure 6.8: Three-component models with 8 parts learned for (a) car and (b) aeroplane classes.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we study the problem of human detection in still images. We investigate the limitations of some popular existing features and propose novel features that represent humans more accurately than existing features. We also analyze current classification frameworks and propose a new framework that is derived from analysis of the distribution of features in the feature subspace.

The first feature proposed is the Extended Histogram of Gradients (ExHoG). Histogram of Gradients (Hg) differentiates a bright human from a dark background and vice versa. Histogram of Oriented Gradients (Hog) mitigates this issue of Hg by considering unsigned gradients only. However, in the same cell/block, Hog maps all gradients of opposite directions and single directions to the same bin of the histogram. This causes some different structures to be similarly represented. ExHoG solves these issues of Hg and Hog by considering the sum and absolute differences of Hg. Compared to existing state-of-the art holistic approaches, the proposed feature provides a much improved detection performance on benchmark human detection data sets.

The next set of features we propose are the Discriminative Robust Local Binary Pat-
tern (DRLbp) and Discriminative Robust Local Ternary Pattern (DRLtP) which fuses edge and texture information into a single representation. We analyze the limitations of Local Binary Pattern (LBP), Local Ternary Pattern (LTP) and Robust Local Binary Pattern (RLBP) with respect to human detection. LBP and LTP differentiate a bright human from a dark background and vice versa. RLBP mitigates the issue of LBP. However, it maps LBP codes and its complement codes in the same block to the same bin in the histogram. This causes some different structures to be similarly represented. Furthermore, the contrast information is discarded in all 3 features. For human detection, the contour of the human, which typically resides in high contrast regions between the human and the background, contains discriminatory information. Discarding this information results in a less accurate detection performance. In order to alleviate the issues of LBP and RLBP, we consider the gradient weighted sum and absolute difference of a LBP and its complement. Furthermore, we analyze what should be the “inverted” representation for LTP and similarly, propose to solve the issue of LTP by considering the gradient weighted sum and absolute difference of a LTP code and its inverted representation. Compared with its predecessors and state-of-the-art holistic human detection approaches, the proposed features provide a much improved detection performance.

We propose a classification framework more suited for human detection by analyzing the distribution of features in the feature space. Based on our analysis, we find that the boundary between humans and non-humans can be approximately well described by a hyper-quadratic surface. We propose a modified Minimum Mahalanobis Distance classifier (MMDC) for classification of humans as it is a quadratic classifier. However, the class means and covariance matrices of the data population of humans and non-humans are unknown. It can only be estimated from the training samples which are limited in number. Hence, if the estimated variances (eigenvalues) deviate
greatly from those of the data population, the MMDC will overfit the data as it uses the inverse of the covariance matrices in its classification. This will result in a poor generalization. This problem becomes very severe if the feature dimensionality is high. Therefore, in order to improve the classification performance, Asymmetric Principal Component Analysis (APCA) is used to reduce the dimensions of the features before classification. APCA is a dimensionality reduction technique that directly targets at solving such problem. Furthermore, APCA also considers the asymmetry present in the training sets of humans where there are only a few human samples and a huge number of negative samples by asymmetrically weighing the covariance matrices of each class. The effectiveness of the proposed classification framework is demonstrated using the proposed features and against other classification frameworks.

We also show extensions to the applications of the proposed features to general visual object detection. We demonstrate superior results compared to the state-of-the-art methods. Furthermore, we also demonstrate a parts-based modeling approach using ExHoG for object detection. Superior results are obtained in comparison to existing state-of-the-art approaches.

7.2 Future Work

A summary of the possible directions for further study stemming from the present research are briefly summarized below.

1. The thesis illustrates the strengths of our single features as applied to human detection. Currently, most human detection approaches are now veering towards hybrid feature approaches where features representing different contextual information are combined into a single feature or are pooled by a classifier or classification algorithm to find the most appropriate feature for a region in the image. It would be interesting
to see the improved detection performance of the proposed features when combined with other powerful features.

2. Currently, the projection of the features to the lower-dimensional subspace is the time-consuming portion of our classification framework. In [25], the authors analyzed the features more thoroughly using Principal Component Analysis to find some relationship which allows for dimension reduction without projection. A similar approach could also be applied to our classification framework to remove the costly projection step and yet obtain a reduced feature which improves detection performance.

3. The application of features could be extended to video-based applications. Currently, all applications discussed in the thesis are image-based. Moving to video-based applications would present more challenges such as feature extraction in real-time and dynamic conditions. The application of the proposed features in such conditions would present further room for improvements. Furthermore, once the detection performance is properly tuned, tracking of people and objects in real-time can be worked on. Along with it, video analytics can also be carried out to provide a complete system architecture.
Author’s Publications

Journal Papers


Conference Papers


Bibliography


