PEDESTRIAN DETECTION BY USING WEIGHTED CHANNEL FEATURES WITH HIERARCHICAL REGION REDUCTION

Mr. Wittawin Susutti

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2014

การตรวจหาคนเดินเท้าโดยใช้ลักษณะเด่นของช่องสัญญาณแบบถ่วงน้ำหนักพร้อมกับการลดบริเวณแบบลำดับชั้น

นายวิธวินท์ สุสุทธิ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

Thesis Title	PEDESTRIAN DETECTION BY USING WEIGHTED CHANNEL FEATURES WITH HIERARCHICAL REGION REDUCTION
By	Mr. Wittawin Susutti
Field of Study	Computer Science
Thesis Advisor	Professor Chidchanok Lursinsap, Ph.D.
Thesis Co-advisor	Associate Professor Peraphon Sophatsathit, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

. . . . Dean of the Faculty of Science
(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

. . . . Chairman
(Assistant Professor Suphakant Phimoltares, Ph.D.)

. . . . Thesis Advisor
(Professor Chidchanok Lursinsap, Ph.D.)

. . . . Thesis Co-advisor
(Associate Professor Peraphon Sophatsathit, Ph.D.)

. . . . Examiner
(Assistant Professor Rajalida Lipikorn, Ph.D.)

. . . . Examiner
(Associate Professor Nagul Cooharojananone, Ph.D.)

. . . . External Examiner
(Chularat Tanprasert, Ph.D.)

วิธวินท์ สุสุทธิ: การตรวจหาคนเดินเท้าโดยใช้ลักษณะเด่นของช่องสัญญาณแบบถ่วงน้ำหนักพร้อมกับการลดบริเวณแบบลำดับชั้น. (PEDESTRIAN DETECTION BY USING WEIGHTED CHANNEL FEATURES WITH HIERARCHICAL REGION REDUCTION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ.ดร. ชิดชนก เหลือสินทรัพย์, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : รศ.ดร. พีระพนธ์ โสพัศสถิตย์ 73 หน้า.

วิทยานิพนธ์นี้ศึกษาปัญหาการตรวจจับคนเดินเท้าจากวีดิทัศน์งานประยุกต์ อาทิ การระบุคนเดินเท้าในเวลาจริงเพื่อหลีกเลี่ยงการชนสำหรับยานยนต์ไร้ผู้ขับเคลื่อนซึ่งเป็นสิ่งสำคัญ ปัญหานี้สามารถแปลงโดยหลักการทางคณิตศาสตร์ให้กลายเป็นปัญหาการจำแนกกลุ่มคนเดินเท้าและกลุ่มไม่ใช่คนเดินเท้าได้ กรอบงานที่เสนอเป็นเครื่องมือตรวจจับคนเดินเท้าบนพื้นฐานของสภาพปรากฏ และคุณลักษณะเด่นจากหลายช่องสัญญาณในวีดิทัศน์ของคนเดินเท้าด้วยการลดบริเวณแบบลำดับชั้น กรอบงานที่เสนอเตรียมการสำหรับสภาพแวดล้อมแบบกล้องตาเดียวเนื่องจากง่ายที่สุด สิ้นเปลืองน้อยและเป็นสภาพแวดล้อมที่เหมาะสมกับการประยุกต์ใช้งานจริง คนเดินเท้าถูกนำเสนอด้วยการรวมคุณลักษณะเด่นของช่องสัญญาณ การดำเนินงานแบ่งออกเป็นสองขั้นตอนคือขั้นตอนการเรียนรู้ระบบและขั้นตอนการทดสอบระบบ ในขั้นตอนการเรียนรู้ คุณลักษณะเด่นของช่องสัญญาณจะถูกถ่วงน้ำหนักโดยแม่แบบคนเดินเท้าเพื่อคัดเลือกคุณลักษณะเด่นที่สามารถใช้ในการวิเคราะห์ได้ การจัดการการโดนบังบางส่วนจะดำเนินการในขั้นตอนการทดสอบโดยการสร้างโครงสร้างการลดบริเวณแบบลำดับชั้น ดังนั้นคนเดินเท้าเต็มตัวจะถูกลดบริเวณเป็นบริเวณแบบแนวนอนและแนวตั้ง แต่ละบริเวณถูกออกแบบเฉพาะสำหรับระดับการโดนบัง และมีเครื่องมือตรวจจับเฉพาะของบริเวณนั้นๆ หลังจากนั้นเครื่องมือตรวจจับเฉพาะแต่ละบริเวณจะถูกรวมในรูปแบบลำดับชั้นเพื่อลดเวลาและขั้นตอนที่มากเกินไป ผลการทดลองให้ผลดีกับฐานข้อมูลมาตรฐาน การวัดประสิทธิภาพด้านอัตราความผิดพลาดความแม่นยำเฉลี่ยและ การแลกเปลี่ยนระหว่างความเร็วของการทำงานและประสิทธิภาพของเครื่องมือตรวจจับที่เสนอแสดงให้เห็นว่ากรอบงานเครื่องมือตรวจจับที่เสนอเป็นทางเลือกที่สมเหตุสมผลสำหรับการประยุกต์ใช้งานจริง

ภาควิชา ......คณิตศาสตร์และวิทยาการคอมพิวเตอร์...... ลายมือชื่อนิสิต ....

สาขาวิชา ....วิทยาการคอมพิวเตอร์............. ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

ปีการศึกษา ....2557.................... ลายมือชื่อ อ.ที่ปรึกษาร่วม ....

5073877523: MAJOR COMPUTER SCIENCE

KEYWORDS: OBJECT DETECTION / PEDESTRIAN DETECTION / DECISION TREE

WITTAWIN SUSUTTI : PEDESTRIAN DETECTION BY USING WEIGHTED CHANNEL FEATURES WITH HIERARCHICAL REGION REDUCTION. ADVISOR : PROF. CHIDCHANOK LURSINSAP, Ph.D., CO-ADVISOR : ASSOC. PROF. PERAPHON SOPHATSATHIT, Ph.D., 73 pp.

This dissertation studies the problem of pedestrian detection in video applications. Identifying a pedestrian in real time to avoid collision for unmanned vehicles is mandatory. This problem can be mathematically transformed to the problem of classifying objects as pedestrian and non-pedestrian classes. The proposed framework is an appearance-based multichannel detectors with hierarchical pedestrian region reduction. It is intended for monocular environment since it is the simplest, least expensive, and most practical environment to work in real applications. The pedestrian is represented by the combination of channel features. The operation is broken down into two steps, namely, training and testing. In training step, the channel features are weighted by pedestrian template for meaningful feature selection process. Handling partial occlusion is carried out in testing step by constructing a hierarchical region reduction structure. Then a full pedestrian image is reduced to the horizontal and vertical regions. Each region is designed for specific level of occlusion, having its own region detector. All region detectors are combined in hierarchical fashion to reduce time and redundant processes. The experiment yields good results using standard benchmark dataset. The performance evaluation on miss rate, average precision, and the trade-off between running time and performance of the proposed detector show that the proposed detection framework is a reasonable option for real world applications.

Department : ...... Mathematics and Computer Science ...... Student's Signature ....

Field of Study : .... Computer Science .... Advisor's Signature ....

Academic Year : .... 2014 .... Co-advisor's Signature ....

Acknowledgements

This dissertation is completed with the help of many people who provide valuable advices, knowledge, funding, and supports. First, I would like to thank my advisor, Professor Dr. Chidchanok Lursinsap and my co-advisor, Associate Professor Dr. Peraphon Sophatsathit, for their inspiration, knowledge, guidance for solving problems, and keeping me on time in every situation. Second, I would like to thank the committee such as Assistant Professor Dr. Suphakant Phimoltares, Assistant Professor Dr. Rajalida Lipikorn, Associate Professor Dr. Nagul Cooharojananone, and Dr. Chularat Tanprasert for their valuable advices and criticism that make this work better. Third, I would like to thank the program Strategic Scholarships for Frontier Research Network for the Ph.D. Program Thai Doctoral degree, Office of the Higher Education Commission, Ministry of Education, Thailand, as the funding source of this work. Fourth, the indispensable discussions from the members of AVIC research center, especially the member of image cluster. Finally, I would like to thank my parents for their support, trust, courage, and the opportunity in education, and my special friend for her will power and understanding.

Contents

List of Tables

List of Figures

Figure Page

CHAPTER I

INTRODUCTION

1.1 Problem and Motivation

Nowadays, pedestrian accident rates by vehicle are still high. Using technologies as tools to reduce accidents is a viable means to save life. These vehicles need to be equipped with state-of-the-art tools so as to prevent potential mishaps. Moreover, surveillance and action cameras are often installed on various type of vehicles such as cars, motorcycles, and bicycles to serve the purpose. Detecting pedestrian by the equipment becomes an on-going research. A number of techniques are applied to accommodate a wide range of real world applications such as finding human in an image, detecting human in surveillance videos using static cameras, surveillance system with moving cameras, robot vision, and automatic pedestrian detection system for vehicle design.

In this regard, pedestrian detection is one of the most active researches in computer vision. There are many published researches in this field that lead to a wide array of research diversifications. Among numerous researches, the areas of interest are grouped into several topics such as:

1) **Finding the features to represent pedestrian**

   Pedestrian class is one of the object that is non-rigid, different postures, variation of clothing, and always affected by illumination. These make pedestrian class complicated to represent.

2) **Computing time for real world applications**

   One of the major aim is pedestrian detection system to prevent accident since the accident can happen any time. The system should perform in real-time with acceptable performance.

3) **Partial occlusion problems**

   In real world situation, pedestrians can be occluded on the scene. The occlusion can be caused by many objects such as bushes, trees, cars, bags, and other pedestrians. So, the pedestrian detection system should be capable of handling partial occlusion problems.

From the topics mention above, pedestrian detection is an interesting and challenging problem. This dissertation proposes a pedestrian detection framework with partial occlusion handling. The proposed approach uses a weighted mean pedestrian silhouette for feature selection to determine the expected pedestrian location in a subwindow. Then, the appearance-based boosted decision trees are applied as classifiers with aggregated channel features (ACF). The partial occlusion problem is handled by a hierarchical pedestrian appearance scheme that integrated into boosted decision trees. From the experimental results, the proposed framework yields good results and handles partial occlusion problem well.

1.2 Objectives

The objectives of this dissertation are the following.

1) To detect pedestrian in moving monocular camera environment, focusing on the detection system which require as low resource as possible and can be applied in wide range of applications.
2) To detect partial occlusion pedestrian. From video observation in urban environment, partial occlusion problem is the major cause of more than half of pedestrians viewing in the scene blocked. So partial occlusion handling capability is required.

1.3 Scope and Limitations

This dissertation concerns two aspects. The first aspect is to determine a fully visible pedestrian. The second aspect is the pedestrian with partial occlusion. There are two issues worth considering prior to elaborating on the proposed method. First, many researches attempt to collect more types or channels of data by including more equipment to capture additional information such as stereo-based camera and infra-red sensor. Moreover, various types of image information such as depth, motion, and optical flow are taken into account in the classification process. Second, despite the use of this information which can improve the recognition accuracy, they are rather costly in terms of computations and practical use. To resolve the causes of these factors, the following basic constraints are imposed in the scope of work.

1) The proposed method operates in moving monocular camera environment. All images are taken from a monocular camera.

2) The camera is set perpendicular to the upright pedestrian's body. It pans on a hypothetical plane parallel to the ground. Therefore, the work will focus on applications in driving assistance. In this environment, the camera is attached to the car set parallel to the ground, and pans during the vehicle turns.

1.4 Summary of Contributions

The contributions of this dissertation are three folds. First, a semantic feature selection based on a mean smoothed pedestrian template is proposed. In training step of boosted decision trees, the feature points are selected using the mean smoothed pedestrian template where each pixel is weighted by the average pedestrian silhouette. Second, a combination of channel features for pedestrian representation is proposed by using two powerful image features as channel features. Third, a partial occlusion handing technique is established for use with multi-appearance based boosting decision trees. This approach combines several boosted decision trees in a hierarchical fashion. The classification result in each level activates the corresponding classifiers to handle the problem.

1.5 Research Procedures

1) Review and study fundamentals and related works of pedestrian detection.

2) Propose a partial occlusion handling method.

3) Experiment and compare results with other techniques.

4) Propose a pedestrian detection framework.

5) Experiment with standard datasets and compare the results with other techniques.

6) Analyse the experimental results, draw some concluding remarks and future works.

1.6 Dissertation Organization

The rest of the dissertation is organized as follows. The next chapter describes some backgrounds and related works. Chapter 3 establishes an architecture of pedestrian detection framework with partial occlusion handling. Chapter 4 describes the experimental setting and demonstrates the experimental results of the proposed framework. Finally, some discussions, conclusion, and future work of the dissertation are given in Chapter 5.

CHAPTER II

LITERATURE REVIEWS

In recent years, pedestrian classification is one of the most active research topics in computer vision. Some issues and surveys were discussed in [1–5]. Numerous efforts have been dedicated to various pedestrian recognition areas. Examples are gaining more information from the source by additional sensors and related instrument to a system, finding a feature that can capture pedestrian characteristics, integration with many classifiers, handling more complicated scenarios such as partial occlusion and night-time situation. This chapter reviews and discusses the related works including the recent works and algorithms in domain of pedestrian detection.

2.1 Camera and Additional Equipment

Most pedestrian detection researches utilize two types of camera, namely, monocular and stereo cameras. In a monocular system, the information directly obtained from a single camera is the sequence of images. Basic information that widely use is intensity or color image. Given image sequences or video data, a number of researches used the motion flow acquired from two consecutive frames computation as another source of information [6, 7]. Thus, the monocular system is inexpensive and easy to implement in real world settings [1, 8–10]. With the limited sources of the image information problem in monocular environment, a stereo-based system was proposed [11]. The most attractive benefit of stereo-based system is information from a depth channel. The depth channel can be applied to both classification and detection problems such as search space reduction and scene analysis [12, 13]. In [14, 15], three information channels were obtained, including intensity, flow, and depth, which yielded good classification results. More information can thus be added, depending on the additional equipment being incorporated into the system such as an infra-red camera [16, 17] and other sensors. However, the more equipment and information channels are added, the higher complexity of the system becomes. Consequently, the approaches including a stereo camera and additional equipment may be expensive and more complicated to implement in real world applications than the monocular system counterpart.

2.2 Features for Pedestrian Classification

Irrespective of the number of cameras used, the problem is how to represent the variation of pedestrian classes. Many features that capture the image information have been proposed such as histogram of oriented gradient (HOG) [18], local binary pattern [19], region covariance [20], edgelet [21], Haar-like rectangle [6, 22], and local receptive field [23, 24].

2.2.1 Histogram of Oriented Gradient

Shape of object is a crucial information widely used in image processing interpreted by the distribution of image gradients. This information can be derived from the gradients of image. Gradients of image are the values that describe a changing in color or intensity of image in directed manner. These values can be described in both how fast of the changing and the direction of the changing. The gradient of the image compose of two different information. First, the gradient magnitude that represent how fast the color or intensity of image is changing. More magnitude implied faster changing. Second, the gradient orientation represents the changing direction of color or intensity of image. Histogram of oriented gradient (HOG) is a very popular descriptor in domain of object detection, especially in pedestrian detection. HOG was first introduced by Dalal et al. [18] in 2005. HOG computes the appearance of gradient orientation in each small portions of an image. The input image is divided into small regions called cells. For each cell, compute the histogram of gradients according to the orientation and weighted by its magnitude. Neighbouring cells are formed to bigger region called block. The histograms in the block are concatenated and normalized. The combination of these histograms in all block then represents the HOG descriptor as shown in Figure 2.1.

2.2.2 Local Binary Pattern

The Local Binary Pattern (LBP) is a widely used image descriptor in pattern recognition. LBP is computed by comparing pixel's value of a pixel with its neighbours. The neighbour pixels having higher value are assigned to 1 and the lower value are assigned to 0. The binarized neighbour pixels form the binary number and the histogram bin associated with the decimal value of binary number is accounted. The final result is a histogram of binary pattern that represents the image. The process of LBP descriptor is illustrated in Figure 2.2. Therefore normal LBP feature need 256 histogram bins to represent image pattern. There is an extend version of LBP that reduce the size of histogram bin based on frequency occurrence binary patterns in

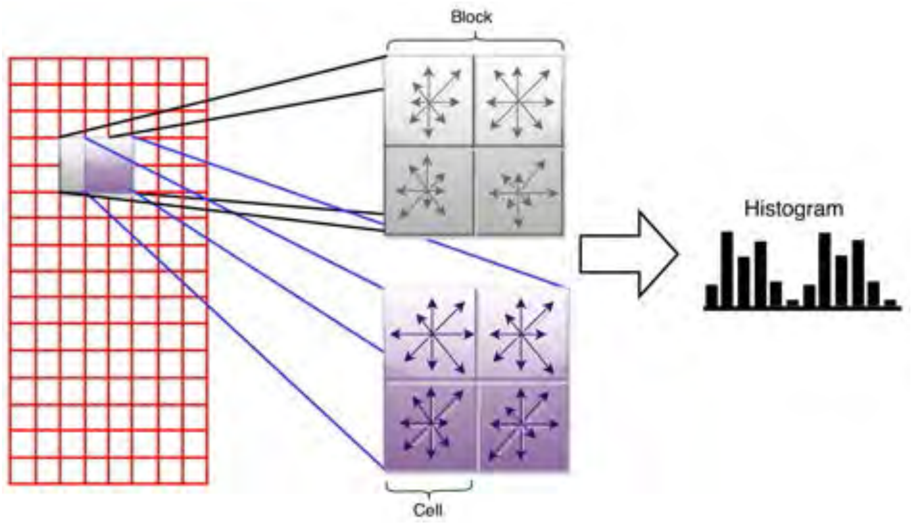


Figure 2.1: HOG calculation.

image called uniform pattern. The uniform patterns are the patterns which the number of bit transitions happen at most two times. Example of uLBP patterns are displayed in Table 2.1. From original 256 histogram bins, this uniform LBP (uLBP) compose of 59 histogram bins only.

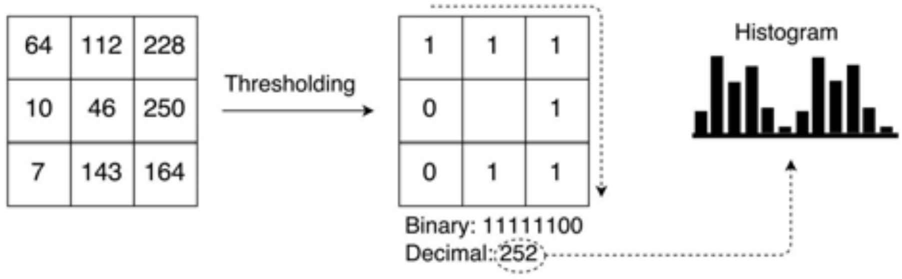


Figure 2.2: An LBP process.

2.2.3 Rectangle Filters

A simple but powerful rectangle filters was proposed by Viola et al.[22, 25]. The calculation is very simple by giving a set of rectangle as shown is Figure 2.3, the values denote the difference between summation of all pixel values in white area and summation of all pixel values in black area. This fast computation can be performed using integral images and learned using adaptive boosting.

Table 2.1: Example of uniform binary pattern.

Pattern	Number of transitions	Uniform
00000000	0	Yes
00111100	2	Yes
10001110	3	No
10101010	7	No
11100000	1	Yes

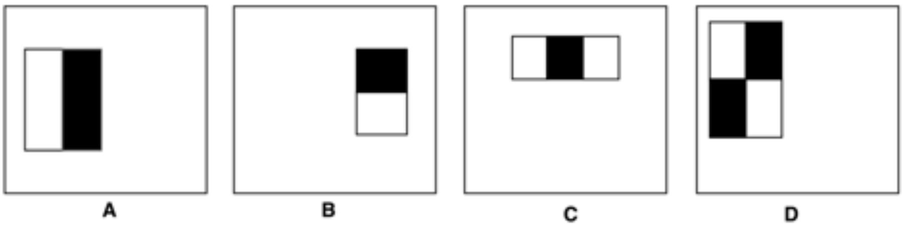


Figure 2.3: Examples of rectangle filters.

Since each individual image feature captures only one dominant information it represents, there is no feature that can encompass the pedestrian well in every situation. To overcome this problem, a multi-featured approach was employed. In [26], the different image features were concatenated to constitute one feature space. This was apparent in an example of mixture-of-experts [15] that made use of this multi-featured classification. The multi-featured with trainable classifiers was proposed in [27]. Wang et al. [28] combined two powerful image descriptors, HOG and LBP, for pedestrian detection with partial occlusion handling. The decision value of SVM is decomposed and distributed into small regions. These decomposed values are used to indicate the occlusion area of the detection window. The integral feature channels (ICF) that combined image information from many image channels and represented in one integral image was proposed in [29]. The ICF is the combination of the sum of many rectangle regions of the image in different channels and fast to compute. Benenson et al., [30] proposed roerei detector that applied ICF with all possible rectangle features and improve performance using global normalization. Informed Haar feature is introduced by Zhang et al.[63]. Informed Haar is a template pool formed by multiple size binary and ternary rectangle features. Each rectangle is created by sliding the rectangle over pedestrian shape and collect the pedestrian shape information as shown in Figure 2.4. Informed Haar composes of multiple features such as LUV color, gradient magnitude, and gradient histogram. This technique is applied with a boosted decision tree.

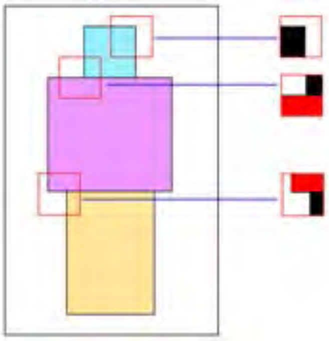


Figure 2.4: Examples of informed Haar rectangle feature.

2.2.4 Word Channel Features

Costea et al.[65] designed word channel feature for single classifier pedestrian detection. Word channel is high level visual word based on visual codebook with lower dimension of descriptors. Three types of feature are trained by the codebook to group visual words such as LUV, HOG, and LBP. The input image is extracted to these three types of feature. Each feature is matched with the corresponding codebook. The map will be decomposed to channel for each word of individual trained feature and perform sliding window classification with cascade of boosted decision tree. The steps for pedestrian detection using word channel are summarized in Figure 2.5.

2.2.5 Aggregate Channel Features

One of the recent image features called Aggregate Channel Features (ACF) which is a fast multi-scale image feature proposed by Dollar et al. [31]. The ACF is composed of several image features called channel which interpolatable with nearby scales such as color, gradient, and gradient histogram. Given an input image I, the channel features, $C_i = \varphi_i(I)$, is obtained by transforming I with image feature extraction φ_i. Each channel feature C_i is grouped into a block of pixels and summing and smoothed to lower resolution. Then, each channel is vectorized and concatenated to form a image feature. The boosted decision tree is applied with ACF to yield good performance for pedestrian detection. The procedure is summarized in Figure 2.6.

In multi-scale object detection, image features are normally computed in every scales as shown in Figure 2.7a. This is one of the bottlenecks in object detection system. The ACF

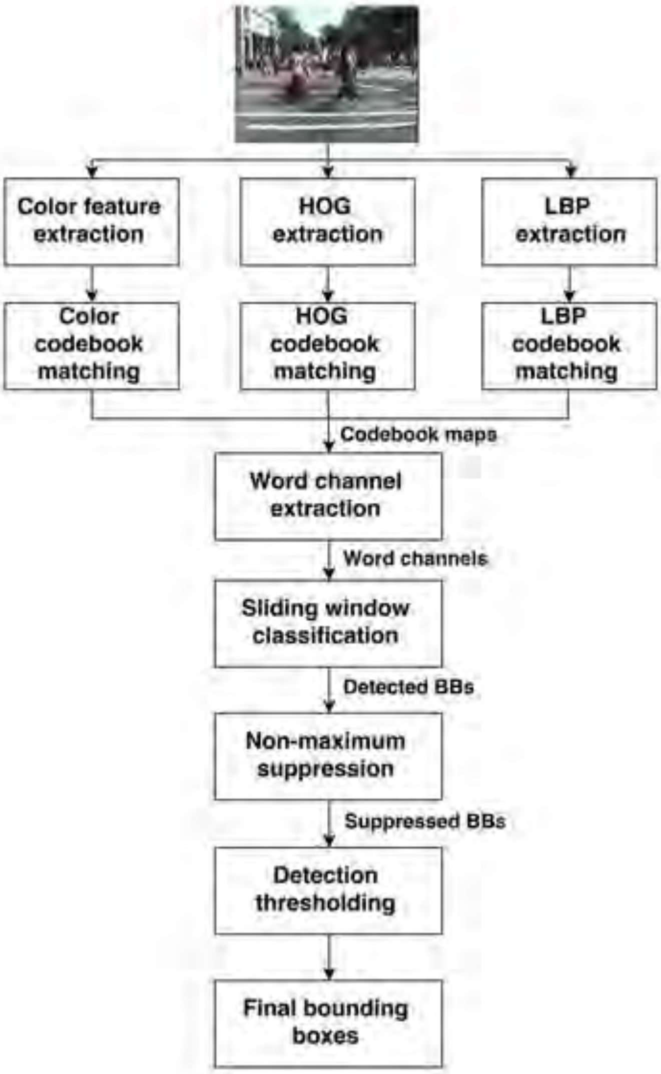


Figure 2.5: Steps of word channels pedestrian detection.

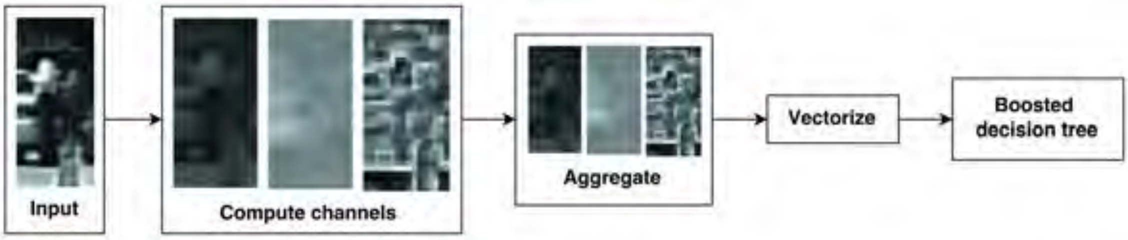
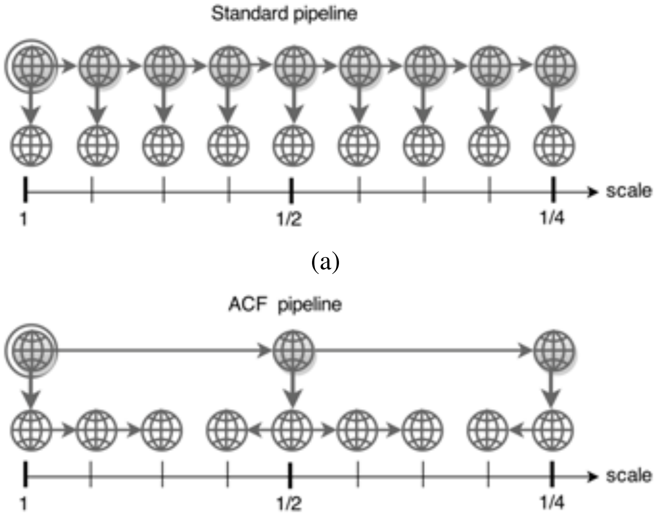


Figure 2.6: Steps of ACF detector.



(a)

(b)

Figure 2.7: Multi-scale feature approximation. (a) Standard approach. (b) ACF approach.

surpasses this issue by using fast feature pyramid technique which approximates the channel features in nearby scales during detection as shown in Figure 2.7b.

Recently, improvement on ACF has been attempted using locally decorrelated channel features (LDCF) [61]. This approach removes correlation of local image regions and estimates a local covariance matrix that shares information for all image regions.

2.2.6 Spatial Pooling Features

Spatial pooling feature (SP) was proposed by Paisitkriangkrai et al.[50, 64]. SP is composed of modified version of two features, namely, covariance descriptor and LBP. The input image is decomposed into several region of interests called pooling regions. SP covariance feature is extracted by computing covariance matrix of small patches in pooling region. The result of all patches represent covariance feature of pooling region. For SP LBP, LBP is extracted from each patch in pooling region and concatenated to represent the SP LBP of the pooling region. The SP feature concept is shown in Figure 2.8.

2.2.7 Motion Feature

Many pedestrian detectors use motion feature as an additional information [5, 6, 14]. Dalal et al. [7] adapted HOG computation from gradient based to optical flow based. These

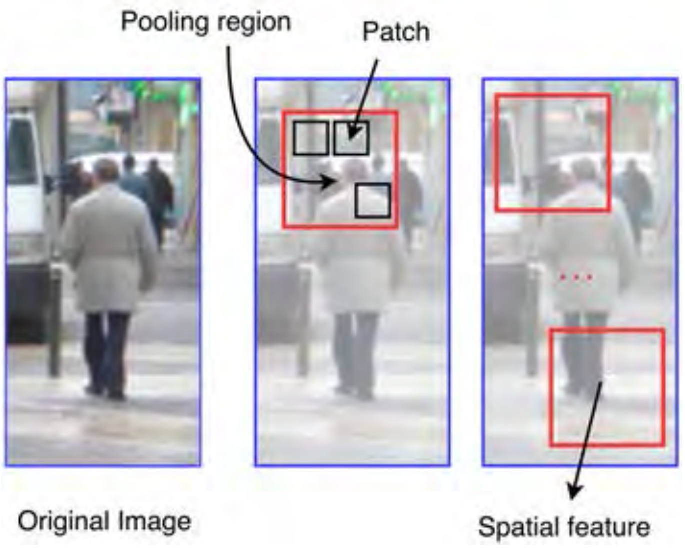


Figure 2.8: Concept of spatial pooling features

two types of HOG are combined and used for pedestrian detection. Enzweiler et al. [8] used parallax flow computation to generate regions of interest before performing pedestrian classification process. Park et al. [8] proposed a new motion feature extraction called Stabilized Difference temporal feature (SDt). The SDt is performed by weakly stabilized across multiple video frames. After stabilizing, the centric motion of both camera and objects are discarded and part centric motion is collected. The features are computed by normalizing the temporal difference of the stabilized image frames.

2.3 Type of Classifiers

A number of diverse classifiers can be applied on the pedestrian detection problem such as support vector machine (SVM) [7, 14, 15, 27, 32], neural networks [14, 15, 27], cascades of boosted classifiers [33, 34], decision tree and deep networks. An adaptive boosting [6, 35, 36] and bootstrapping technique are the popular approaches for enhancing these classifiers in training process. Details of each type of classifiers are described below:

2.3.1 Support Vector Machine

The support vector machine (SVM) is a supervised model for classification and regression problems. The optimal hyperplanes are constructed by the input data with associative labels. There are many SVM kernels that are widely used in pedestrian detection problem such

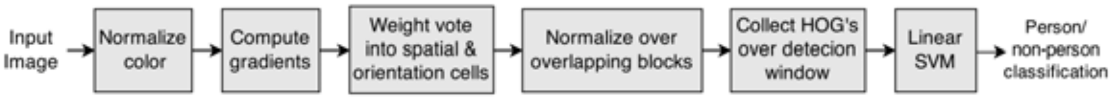

Input Image
Normalize color
Compute gradients
Weight vote into spatial & orientation cells
Normalize over overlapping blocks
Collect HOG's over detecion window
Linear SVM
Person/ non-person classification

Figure 2.9: Applying HOG with linear SVM.

as linear SVM, histogram intersection kernel, and latent SVM. Dalal et al.[18] proposed HOG feature and applied it with linear SVM as shown in Figure 2.9. Maji et al.[37] proposed the histogram intersection kernel with logarithmic runtime complexity. The latent SVM was proposed by Felzenszwalb et al.[38, 39] which was designed as a part-based model where each part was a latent variable.

2.3.2 Convolution Neuron Networks

A convolution neuron networks (CNN) is a type of neuron networks where each neuron responds with specific overlap regions. Individual neurons are grouped and formed as a layer called convolutional layer. The lowest layer learns the data from image features or pixel values and sent the outputs to the higher layer as inputs. In [40], the CNN are used for pedestrian detection as a high level feature extraction. The lowest layer learns from image pixel values. The unsupervised model in each layer is trained using the output of previous layer. The supervised model is trained with the extracted features from the CNN and used to classify the pedestrian. The structure of CNN learning proposed by [40] is illustrated in Figure 2.10. A CNN based networks called switchable deep network (SDN) is proposed by Luo et al.[41]. The SDN architecture composed of three type of layers, namely, a convolution layer, four switchable layers, and a logistic regression layer. The convolution layer extracts low and mid level features from input image while switchable layers extract high level representations based on pedestrian parts. The logistic regression layer collects the entire information from previous layers and predict the label of input image.

2.3.3 Cascade of Boosted Classifiers

Cascade of boosted classifiers, proposed by Viola et al.[22, 25], is a model that ensembles some boosted classifiers to a set of sequential classifiers. Each classifier is called a stage. In each stage, the input feature is classified and the positive result is sent to the next classifier, while the negative result is rejected from the cascade immediately. The process of cascade of

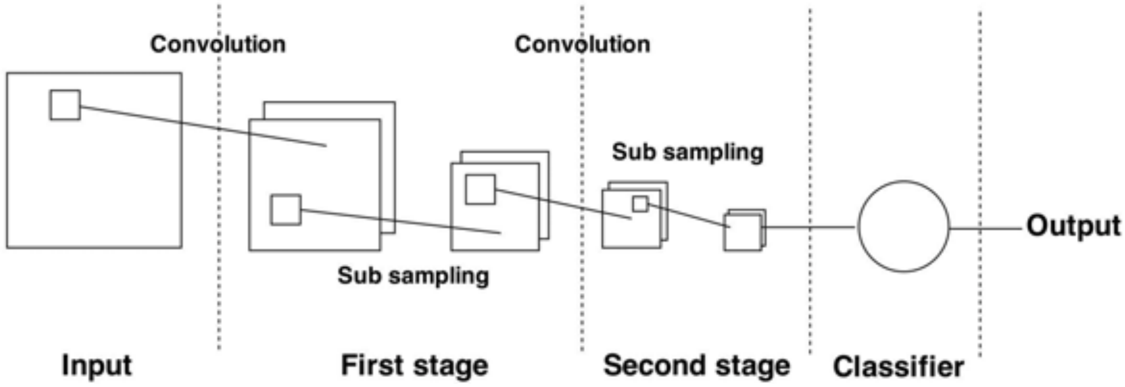


Figure 2.10: Structure of convolution neuron networks.

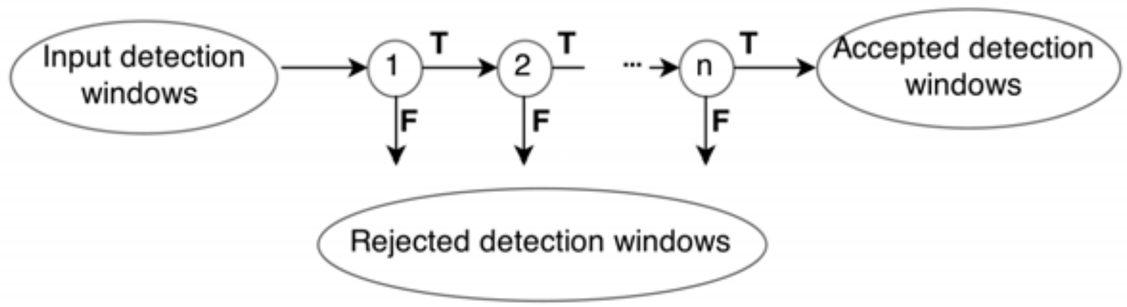


Figure 2.11: Overview of cascade classifiers.

boosting classifiers is shown in Figure 2.11.

2.3.4 Decision Tree

Decision tree is a well-known supervised classification learning technique. Each interior node of decision tree is called a decision node representing input variables or features. Edges or branches represent the alternative corresponding input according to the decision node. Each leaf or terminal node represents the label of classification problem or decision value for each class. There are many techniques called ensemble that combine decision trees together such as bagging [42], random forest [43], rotation forest [44], and boosted tree.

2.3.4.1 Adaptive Boosting

Boosting technique is an algorithm for creating strong classifier from combinations of weak classifiers. Weak classifier is a classifier that is slightly better performance than a random guess method. Basic procedure of boosting is iterating of learning weak classifiers. In each iteration, increase the weight of each misclassified instance and reduce the weight of the corrected one. So the weak classifiers will focus more on uncorrected training data from previous iteration. There are many boosting algorithms, for example, LPBoost [45], Gradient boosting [46], and the popular adaptive boosting (AdaBoost) that is adopted in this study. Adaboost was proposed by Y.Freund and R.Schapire in 1997 [47]. With Adaboost method, the weak classifiers are first trained by equal weight training data. The best classifier is selected and the training data are weighted according to the errors of the best classifier. Then, other weak classifiers are trained using weighted training data. These steps are repeated until the condition are met. The final strong classifier is formed by ensembling individual weak classifier from each iteration.

2.3.4.2 Boosted Decision Tree

A boosted decision tree is a boosted classifier where each weak classifier is a decision tree. Normally, training decision tree can gain upto 100% accurate on training data. In boosting approach, each weak classifier has to confine to performance slightly better than random guess. Thus, each decision tree is restricted to specific depth or height to reserve as weak classifier and a decision value from each decision tree is combined as the final result. The process of training boosted decision tree is summarized in Figure 2.12. Appel et al.[48] presented a fast training approach for boosted decision tree. This technique trains the features using multiple small subsets of training data. It prunes the features that guarantee to perform worse than other features called underachieving features. By pruning some features, the computation cost of training a boosted decision tree is reduced.

2.3.5 Partial Area Under the Receiver Operating Characteristic Curve Boosting

Partial area under the receiver operating characteristic (ROC) curve boosting (pAUCBoost) is an ensemble technique that focus on optimizing the areas under ROC curve in specific range proposed by Paisitkriangkrai et al.[49, 50]. pAUCBoost is mainly common with other boosting algorithms by combining several weak classifiers to formed the strong classifier. The major difference is pAUCBoost focused on the miss ordering of positive data and negative data in training set. The positive samples with lower rank than the negative samples will be assigned more weight in next iteration of boosting. The purpose is all positive samples should be ranked higher than the negative samples.

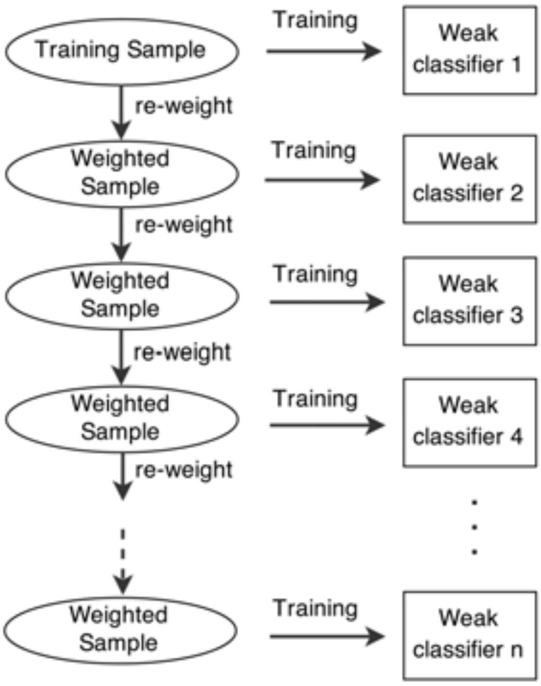


Figure 2.12: Boosted decision tree with Adaboost.

From a variety set of classifiers, the active choices that show the top results in major pedestrian dataset are the boosted decision tree (Adaboost), the deep networks, and pAUC-Boost.

2.4 Partial Occlusion Problem

There are two schemes of pedestrian detector. The first type is full-body classification that uses the information from full body appearance of pedestrian to classify a test image. The main advantage of this scheme is high accuracy rate because this technique collects all information from the entire body of the pedestrian image. But this technique is not efficient enough to handle real world situation. Thus, not only variations of pedestrian's postures and illumination are the problems, but also partial occlusion of pedestrian poses an additional challenge. The second scheme is the detector that can handle partially occluded pedestrian. The benefit of this scheme is suitable with real world problem. Due to the occlusion, the pedestrian appearance in the image changes and consequently effects performance of the detector. This section focus on how to handle with partial occlusion problem. In order to handle the partially occluded pedestrian, there are several perspective in this problem as follows:

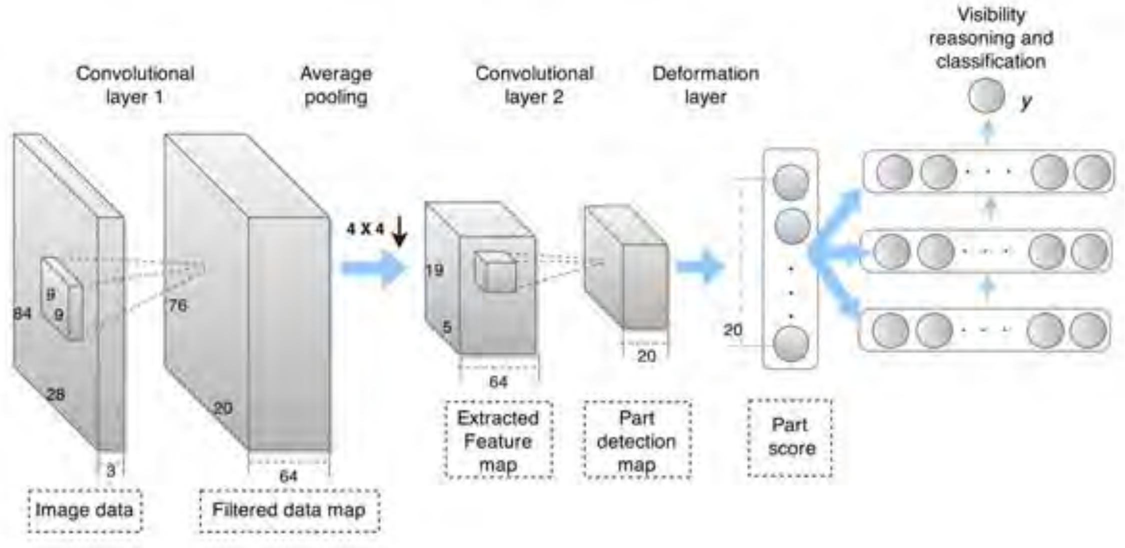
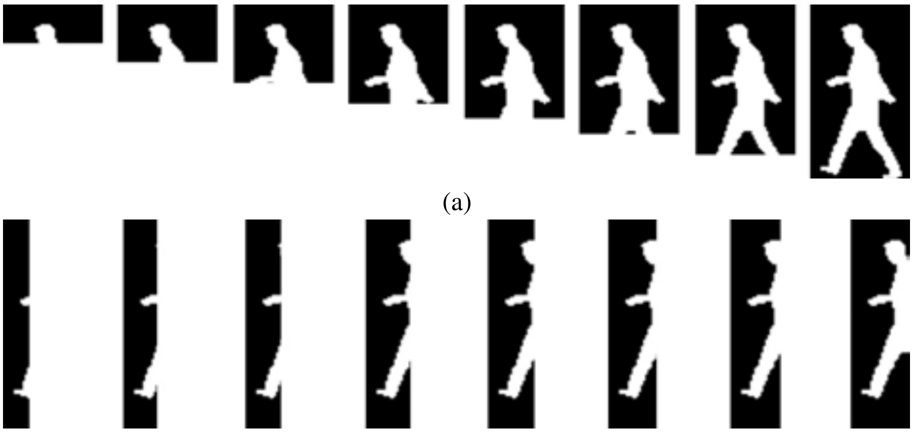


Figure 2.13: An overview of a joint deep network and DPM.

1) **Region-based approach:** A conventional region-based pedestrian classification and detection methodology were employed [38, 51–53], where a number of components or regions were extracted. Normally, a pedestrian composes of three regions, namely, head (including shoulder), torso, and legs. However, Rao et al.[36] and Xia et al.[33] divided a pedestrian into six regions, and Wojek et al.[54] used these six pedestrian regions with high resolution images to improve the performance. There are many techniques based on the region-based approach such as multiple component learning [35], probabilistic component assembly [55] using a Bayesian approach for grouping the components, and a region detector with edgelet feature [21]. The problem was handled by using several modalities such as intensity, motion, and depth, along with component-based classifiers [14] and a combination of features [28].

2) **Deformable part model:** A well-known deformable part model (DPM) was proposed by Felzenszwalb et al. [38, 39] that was applied in many researches such as multiresolution approach [56, 57], and multi-pedestrian as a cue to detect single one [58]. DPM is the SVM with latent structure. DPM defines the part position as a latent variable. Recently, Ouyang et al. [52, 59] applied this approach with deep networks. The image feature is extracted using convolution neuron network over HOG. The deformable part model is performed for each part and classification is done in final process. The process of joint deep model is shown in Figure 2.13.



(a)

(b)

Figure 2.14: Example of regions based on appearance approach. (a) Horizontal regions. (b) Vertical regions

3) **Appearance-based approach:** This method is similar to the region-based method but different in how to divide each region. This scheme splits the region in levels of appearance instead of region of pedestrian parts. Mathias et al.[60] used 33 regions with vary pedestrian appearance level including both horizontal and vertical region. Each region was classified by the corresponding detectors and combined the results. Figure 2.14 illustrates the example of appearance regions.

Major pedestrian detection algorithms are summarized in Table 2.2. In terms of feature, the channel features which represent image in LUV color space, gradient magnitude, and gradient orientation is the most popular choice. For a classifier, Adaboost technique is employed as the dominant classifier and INRIA person dataset are applied to the system as a training set. Most of top detectors such as ACF-Caltech+ [61], LDCF [61], Katamari [5], and SpatialPooling+ [50] use channel as the feature and Adaboost as the classifier while SpatialPooling+ uses multiple features and pAUCBoost as classifier. All top detector use Caltech dataset.

Based on these related works, an appearance-based multi-featured framework for pedestrian detection that uses the information from a monocular camera is proposed. The proposed framework uses ACF wih modified uLBP, boosted decision tree with Adaboost as the classifier on Caltech pedestrian dataset, and a combination of pedestrian appearance patterns to handle partial occlusion. The mean smoothed template is applied for efficient feature point selection. To support a wider range of real world applications, the framework is established based on the

Table 2.2: Summary of major pedestrian detection algorithms.

Algorithm	Feature	Classifier	Training dataset
ACF [31]	channels	AdaBoost	INRIA
ACF-Caltech+ [61]	channels	AdaBoost	Caltech
ACF+SDt [62]	channels	AdaBoost	Caltech
ChnFtrs [29]	channels	AdaBoost	INRIA
ConvNet [40]	pixels	CNN	INRIA
ChnFtrs [29]	channels	AdaBoost	INRIA
Franken [60]	channels	AdaBoost	INRIA
HikSvm [37]	HOG	HIK SVM	INRIA
HOG [18]	HOG	linear SVM	INRIA
HOG-LBP [28]	HOG+LBP	linear SVM	INRIA
InformedHaar [63]	channels	AdaBoost	INRIA/Caltech
JointDeep [59]	color+gradient	CNN	INRIA/Caltech
LatSvm-V1 [38]	HOG	latent SVM	PASCAL
LatSvm-V2 [39]	HOG	latent SVM	INRIA
LDCF [61]	channels	AdaBoost	Caltech
Katamari [5]	channels	AdaBoost	INRIA/Caltech
MT-DPM+Context [57]	HOG	latent SVM	Caltech+
pAUCBoost [49]	HOG+COV	pAUCBoost	INRIA
Roerei [30]	channels	AdaBoost	INRIA
SDN [41]	pixels	CNN	INRIA/Caltech
SpatialPooling [64]	multiple	pAUCBoost	INRIA/Caltech
SpatialPooling+ [50]	multiple	pAUCBoost	Caltech
VJ [25]	Haar	AdaBoost	INRIA
WordChannels [65]	WordChannels	AdaBoost	INRIA/Caltech

combination of various image features instead of relying on variety of information as in stereo environment or added equipment. Details on formulation and implementation of the proposed approach will be described in next chapter.

CHAPTER III

PROPOSED METHOD

The proposed pedestrian detection framework is composed of six major steps as follows:

1) A mean smoothed pedestrian template (MSPT) construction.
2) Pedestrian's region scheme based on appearance-based approach.
3) Training boosted decision trees with MSPT weighting using ACF and uLBP as features.
4) Node and path label assignment in decision tree and path label table construction.
5) A hierarchical region structure for pedestrian detection.
6) Combined detection score.

An overview of pedestrian detection framework is illustrated in Figure 3.1. Details are further described in the sections that follow.

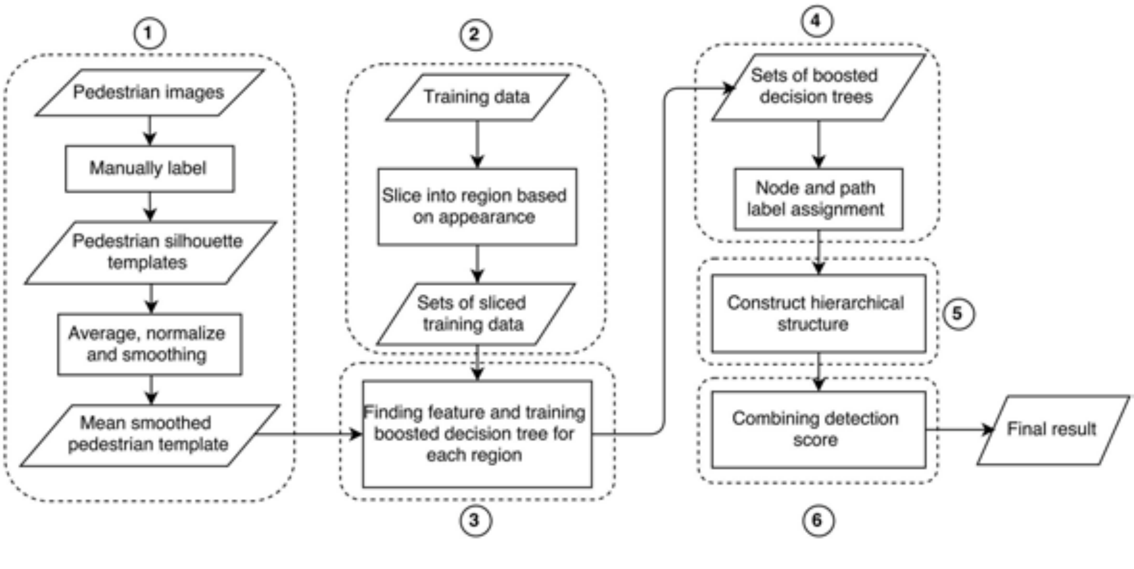

1
Pedestrian images
Manually label
Pedestrian silhouette templates
Average, normalize and smoothing
Mean smoothed pedestrian template
2
Training data
Slice into region based on appearance
Sets of sliced training data
3
Finding feature and training boosted decision tree for each region
4
Sets of boosted decision trees
Node and path label assignment
5
Construct hierarchical structure
6
Combining detection score
Final result

Figure 3.1: An overview of the proposed framework.

3.1 A Mean Smoothed Pedestrian Template (MSPT) Construction

A straightforward approach to determine whether there is a pedestrian in an image or not is by searching the formation of distributed pixels, where the shape is similar to pedestrian shape. The problem of this approach is the variation of pedestrian postures. This is handled by MSPT which is an estimated pedestrian shape template to represent pedestrian's posture. This template will be used in training step of boosted decision trees. The MSPT is constructed by averaging sum of a set of pedestrian silhouette images. The objective of constructing this template is to define a rough pedestrian shape and to find the average location of pedestrian that appear on the detection window. The silhouette images are generated by manually labelling the training data. The covered area is analyzed to determine possible existence of a pedestrian. An example of how to construct a template is shown in Figure 3.2. Let P be the set of N pedestrian images, S be the set of N pedestrian's silhouette images generating from P, and s_i be the binary pedestrian's silhouette image where $s_i \in S$. The procedure for generating MSPT is shown in Algorithm 1. The MSPT is done in data preparation step and applied in training step.

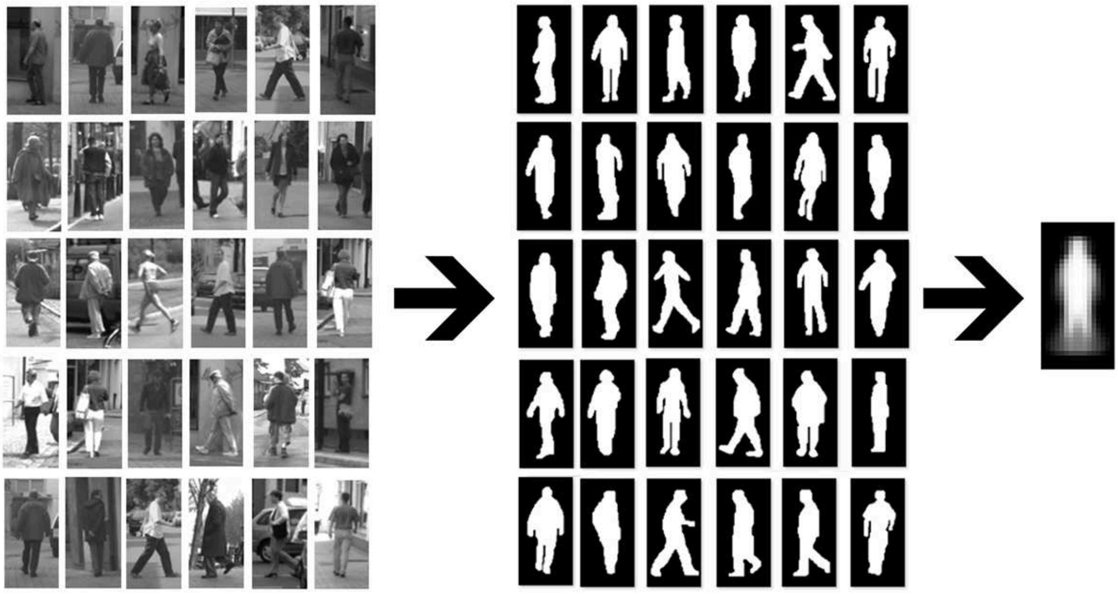


Figure 3.2: An example of MSPT construction.

3.2 Pedestrian Regions based on Appearance Approach

The problems concerning about the pedestrian regions are how many local regions, how large each region should be to permit extracting as much information as possible so as to achieve

Algorithm 1 Creating MSPT

```
Input: Set of pedestrian image $P$.
for $i$ = 1 to $N$ do
    Extracting silhouette image $s_i$ and manually label it.
end for
Compute a rough mean template by $M = \frac{\sum_{i=1}^{N} s_i}{N}$
Normalize and smooth $M$ using Gaussian smoothing filter.
Output: The mean smoothed pedestrian template $M$.
```

maximum recognition accuracy, and how to represent a pedestrian that is related to the actual pedestrian appeared in the video frames. To find the information corresponding with these problems, the occlusion statistics of pedestrian dataset in urban scene reported by Dollár et al. [66] are investigated. Some interesting points of this report are:

1) For all pedestrians appear in the scene, over 70% are occluded at least one frame. This means that the occlusion problem is very important and appears in the scene frequently. Pedestrians are most likely to be occluded.

2) Occlusion scenarios can be divided into four categories:

   - **Fully visible:** pedestrians have no occlusion.
   - **Partial occlusion:** pedestrians having 1-35% occluded area or 65-99% visible.
   - **Heavy occlusion:** pedestrians having 36-80% occluded area or 20-64% visible.
   - **Full occlusion:** pedestrians having over 80% area of occlusion.

3) The areas of occlusion most likely appear in lower part and left or right side of the pedestrian image. There are rare cases that the upper part of the pedestrian is occluded.

4) The pedestrian occlusion regions based on the observation along with the level of pedestrian visibility in detection window are visualized in Figure 3.3(a). The gray color regions represent pedestrian visible area. There are seven types of occlusion scenarios. The fully visible pedestrian type occurs approximately about 22% of all pedestrians in the scene. Four types of horizontal occlusion in bottom area of detection window occupy 69% of occlusion patterns. The two vertical left and right occlusion types of the detection window occur 2.6% and 2.7%, respectively.

From this observation, the occlusion regions can be grouped into three cases, namely, full visible region, horizontal regions, and vertical regions. The assumptions to be established on the proposed pedestrian regions are: 1) In the context of occlusion, the bottom region of a detection window that represents pedestrian's legs is not necessary in the detection process. From the occlusion statistics, over 69% of occlusion types are horizontal cases that occluded in the bottom area of the detection window. In addition for part-based approach, the leg part is insignificant because it will be covered in most horizontal cases. 2) The appearance-based approach with multiple levels of appearance from top to bottom is a reasonable choice and supported by the observation result of the horizontal occlusion that the lower area is occluded with multiple levels. 3) Dividing into too many regions makes both training and testing step time-consuming. In each region, the detector should handle only a few occlusion levels. So the occlusion types with nearby level of occlusion should be grouped. From these assumptions, the proposed framework sets up six pedestrian regions based on appearance approach. Each region represents pedestrian in a specific level of visibility in the detection window. The proposed region scheme composes of full, horizontal, and vertical regions. The details of each region are described below.

1) **Full appearance region** - This region supports the pedestrian with fully visible in the scene and cover 100% area of detection window.

2) **Horizontal level 1 region (H75)** - This region supports the pedestrian having partial occlusion in horizontal scheme, exposing about 75% from top to bottom of pedestrian detection window. This region is designed by grouping two nearby low levels of horizontal occlusion scenarios reported in occlusion statistics.

3) **Horizontal level 2 region (H50)** - This region supports 50% horizontal occlusion from the bottom and 50% visible region of pedestrian detection window.

4) **Horizontal level 3 region (H30)** - This is the smallest horizontal region to handle with heavy occlusion scenario, exposing approximately 30% visible region and covering approximately 70% occlusion from the bottom.

5) **Vertical left region (VL)** - This vertical region covers 62.5% from the right side of the pedestrian detection window, exposing the left side.

6) **Vertical right region (VR)** - This vertical region covers 62.5% from the left side of pedestrian detection window, exposing the right side.

The schematic visualization of the visibility region of pedestrian based on observation is shown in Figure3.3(b) and summarize in Table 3.1. The proposed region scheme is designed

Table 3.1: Summary of the proposed pedestrian region scheme.

Region	Occlude area (%)	Visible area (%)
Full visible	0	100
Horizontal level 1 region (H75)	25	75
Horizontal level 2 region (H50)	50	50
Horizontal level 3 region (H30)	68.75	31.25
Vertical left region (VL)	37.5	62.5
Vertical right region (VR)	37.5	62.5

to capture the information about occlusion scenarios from the occlusion statistics and to reduce the total number of regions by combining the nearby regions. In vertical scenarios, the statistics show a low chance approximately 5% of occurrence in the scene but there are special cases that cannot handle both full and all horizontal region schemes. So vertical regions are necessary supplement. The output of this region scheme applied to the pedestrian image is shown in Figure 3.3(c). The detection training proceeds as follows. Each image from the training dataset is divided into six regions. Then the data of each region is used in the training process according to the steps given in Figure 3.1.

3.3 Training Boosted Decision Tree with MSPT Weighting Using ACF and uLBP as Features

The features for training the detectors are the combination of ACF and spatial uLBP. The ACF is the same as the original proposed by Dollar et al. [31] with 10 channels including 3 channels of LUV color space, 1 channel of gradient magnitude, and 6 channels of gradient orientation. Therefore, ACF is a channel-based feature. The ACF represent features that combine multiple layers of image containing different information. Each channel has the same size as the original image. When ACF is applied with boosted decision tree, the classification process is performed in pixel-based comparison. To apply uLBP with ACF, the uLBP which is the vector of histogram, is adapted to pixel-based and represented in the form of channels being treated as ACF feature. A spatial uLBP is the extended version of the original uLBP that uLBP is computed in cell structure and represented in pixel-based channel feature like that of ACF. To compute the spatial uLBP, the input image is padded and divided into many small overlap cells with one pixel stride. So the number of cells is equal to the number of pixels in the image. In each cell, all pixels are transformed to uLBP pattern based on their neighbors. The uLBP histogram is generated in the cell, normalized, and represented as the pattern of the center pixel of its cell. So the spatial uLBP forms the 59 channels of features having channel size equals to the size of input image. By these steps, the uLBP histogram can be represented in pixel-based channels feature as ACF. Figure 3.4 illustrates an example of spatial uLBP process and the corresponding algorithm is shown below.

Algorithm 2 Computing spatial uLBP feature.

```
Input:
- Image $I$ with size $w \times h$.
- Cell size $c$.
Padding image with size $ceil(\frac{c}{2})$ in each side.
Dividing image into cells.
for each cell do
    for each pixel in the cell do
        Computing LBP.
        Transforming LBP to uLBP.
    end for
    Constructing histogram from uLBP in the cell.
    Storing histogram as the representation of center pixel of the cell.
end for
Output: Spatial uLBP feature with size $w \times h \times 59$.
```

In training process, a set of region specific pedestrian detectors are trained using a decision tree algorithm with Adaboost. In each iteration of boosted decision tree learning, the feature points that are used to train the weak classifiers are selected randomly and uniformly. So there is a chance that less informative features may be selected. To solved this problem, the proposed MSPT is applied in this step for bias weighting. Thus, the feature points are selected based on the value of the corresponding pixel in MSPT. The candidate features of Adaboost are the features which distribute on the pedestrian area. So this set of features should be more meaningful than the features from other areas.

3.4 Node and Path Label Assignment in Decision Tree and Path Label Table Construction

A boosted decision tree is a sequence of binary decision tree ensemble with the boosted algorithm as shown in Figure 3.5. Each node of the decision tree represents the location of a feature to be determined. During testing, the test image is extracted to determine the features and passed them to the boosted decision tree. Each binary decision tree checks the value of test sample and outputs the hypothesis value of that binary decision tree. This value is added to obtain the final decision result. Evaluation proceeds as follows. Let T_R be a boosted decision tree of region R consisting of t decision trees, u_i be an i^{th} individual decision tree in T_R. Suppose u_i is an n-depth decision tree and d_j is the j^{th} node of the decision tree. There will be 2^n leaf nodes that contain hypothesis values from each decision path. Thus, there are only $n-1$ levels of tree or 2^n-1 decision nodes that are used to determine the feature point for each decision tree. Focusing on the decision nodes, there are 2^{n-1} decision paths. In testing step, each decision tree has exactly one possible decision path to evaluate. Each decision path holds a set of decision features. When combine the boosted decision tree with region scheme, each path in the decision tree can be assigned to the corresponding region and the reported hypothesis can be represented as the hypothesis of the subregion. To make this idea possible, evaluation set up proceeds by locating the position of features in each node and each decision path. This path is then labelled associated with the region scheme. After all nodes in the path are labeled, they form the region-corresponding path in each decision tree. Finally, each decision tree will have associate region path label describing the group of feature locations along the decision path. An overview with example of how to assign node and path label is shown in Figure 3.6. Further classifications are given in the subsections that follow.

3.4.1 Assigning Horizontal Node Label

Horizontal node labelling starts from the decision nodes in the decision tree. Each of them will be assigned a region label according to predefined horizontal region in Section 3.2. There are three horizontal regions and one fully visible region that is counted both horizontal and vertical region. In each nearby region, the smaller horizontal region is a subregion of the bigger one. So the nearby horizontal regions can be grouped and the node label is assigned based on these groups. Let $H75$ represent horizontal level 1 region, $H50$ represent horizontal level 2 region, $H30$ represent horizontal level 3 region, and F represent fully visible region. There are three possible groups, namely, F with $H75$ as a fully visible region, $H75$ with $H50$ as a horizontal level 1 region, and $H50$ with $H30$ as a horizontal level 2 region. Let t be the number of decision trees in the boosted decision tree. There are 2^{n-1} decision paths from root node to interior nodes at level $n-1$. Therefore, the proposed framework deploys features as channels. The feature location in each decision node of the boosted decision tree can be any channel of features. The feature location of each node must be transformed to the same pixel location of image independent of the number of channels. Algorithms 3- 5 show horizontal node label assignment procedures. Note that the constant $\frac{3}{4}$ in Algorithms 3 is the level of visibility of $H75$ region. In the same manner, the constant $\frac{1}{2}$ in Algorithms 4 and $\frac{5}{16}$ in Algorithms 5 are the level of visibility of $H50$ and $H30$ region, respectively.

Algorithm 3 Assigning horizontal node label for a boosted decision tree of fully visible region.

```
Input:
- A boosted decision tree $T_F$.
- Height of detection window $H$.
for $i$ = 1 to $t$ do
    for $j$ = 1 to $2^n - 1$ do
        Transforming image feature at node $d_j$ to pixel location $(x_j, y_j)$.
        if $x_j \leq \frac{3}{4} * H$ then
            Assigning label for node $d_j$, $l_j$ = $H75$.
        else
            Assigning label for node $d_j$, $l_j$ = $F$.
        end if
    end for
end for
Output: Node label $l$ for each each node in a boosted decision tree of fully visible region.
```

Algorithm 4 Assigning node label for a boosted decision tree of horizontal level 1 region.

```
Input:
- A boosted decision tree $T_{H75}$.
- Height of detection window $H$.
for $i$ = 1 to $t$ do
    for $j$ = 1 to $2^n - 1$ do
        Transforming image feature at node $d_j$ to pixel location $(x_j, y_j)$.
        if $x_j \leq \frac{1}{2} * H$ then
            Assigning label for node $d_j$, $l_j$ = $H50$.
        else
            Assigning label for node $d_j$, $l_j$ = $H75$.
        end if
    end for
end for
Output: Node label $l$ for each each node in a boosted decision tree of horizontal level 1 region.
```

Algorithm 5 Assigning node label for a boosted decision tree of horizontal level 2 region.

```
Input:
- A boosted decision tree $T_{H50}$.
- Height of detection window $H$.
for $i$ = 1 to $t$ do
    for $j$ = 1 to $2^n - 1$ do
        Transforming image feature at node $d_j$ to pixel location $(x_j, y_j)$.
        if $x_j \leq \frac{5}{16} * H$ then
            Assigning label for node $d_j$, $l_j$ = $H30$.
        else
            Assigning label for node $d_j$, $l_j$ = $H50$.
        end if
    end for
end for
Output: Node label $l$ for each each node in a boosted decision tree of horizontal level 2 region.
```

3.4.2 Assigning Vertical Node Label

This process is similar to the previous one. In vertical region case, there are only three regions, namely, fully visible, vertical left, and vertical right regions. Unlike the horizontal region appearance, the vertical regions support different pedestrian appearances and are not subset of one another. However, some portions overlap. Hence, Algorithm 3 has to be augmented to handle this overlap region. Let VL represent vertical left region, VR represent vertical right region, and VLR represent overlap portions. The algorithm of vertical node label assignment is shown in Algorithm 6.

Algorithm 6 Assigning vertical node label for a boosted decision tree of fully visible region.

```
Input:
- A boosted decision tree $T_{VL}$ or $T_{VR}$.
- Height of detection window $H$.
for $i$ = 1 to $t$ do
    for $j$ = 1 to $2^n - 1$ do
        Transforming image feature at node $d_j$ to pixel location $(x_j, y_j)$.
        if $y_j < 6 * H + 1$ then
            Assigning label for node $d_j$, $l_j$ = $VL$.
        else
            if $y_j < 10 * H + 1$ then
                Assigning label for node $d_j$, $l_j$ = $VLR$.
            else
                Assigning label for node $d_j$, $l_j$ = $VR$.
            end if
        end if
    end for
end for
Output: Node label $l$ for each each node in a boosted decision tree of horizontal level 2 region.
```

3.4.3 Building Horizontal and Vertical Path Label Tables

After labelling all nodes of each decision tree in all boosted decision trees, the path label tables will be constructed. There are two type of path label tables, namely, horizontal and vertical path label tables. In this work, there are totally three horizontal and one vertical path label tables. For horizontal path label tables, each path of the decision tree will be assigned with the associative path label. The path label is defined by the biggest region of node label in the same path. All path labels in each tree are grouped and represented as path label tables. For vertical path label table, the path label is defined by using node label. There are 4 possible cases in vertical path label assignment. First, all node labels VL and VLR will have path label assigned as VL. Second, all node labels VR and VLR will have path label assigned as VR.

Third, all node labels VLR will have path label assigned as VLR. Fourth, both VL and VR nodes in the same path will have path label assigned as F. VLR is the case that can be counted as both VL and VR. However, F is the case that the path is neither VL nor VR. All path labels are collected and represented in vertical path label table. The algorithms for constructing horizontal and vertical path label tables are described in Algorithm 7 and 8.

Algorithm 7 Constructing of a horizontal path label table for a boosted decision tree.

```
Input: A boosted decision tree $T$ with corresponding node label $l$.
for $i$ = 1 to $t$ do
    for each path j in a decision tree do
        Path label $Ph_{i,j}$ = region label with maximum area.
    end for
end for
Output: A horizontal path label table $Ph$ for a boosted decision tree.
```

Algorithm 8 Constructing of a vertical path label table for a boosted decision tree.

```
Input: A boosted decision tree $T$ with corresponding node label $l$.
for $i$ = 1 to $t$ do
    for each path j in a decision tree do
        Let $V$ is a set of node label in path $j$.
        if $VL \in V$ and $VR \notin V$ then
            Path label $Pv_{i,j} = VL$.
        else
            if $VR \in V$ and $VL \notin V$ then
                Path label $Pv_{i,j} = VR$.
            else
                if $VL \in V$ and $VR \in V$ then
                    Path label $Pv_{i,j} = F$.
                else
                    Path label $Pv_{i,j} = VLR$.
                end if
            end if
        end if
    end for
end for
Output: A vertical path label table $Pv$ for a boosted decision tree.
```

3.5 Constructing the Hierarchical Region Structure

From the proposed appearance-based regions presented in Section 3.2 and path label presented in Section 3.4, provision for handling partial occlusion based on this scheme is set up below.

1) In horizontal region, each smaller region is a subset of the larger one. At any stage, if the testing detection window is recognized as a pedestrian by one of a detector, it is

unnecessary to activate other smaller detectors. On the other hand, if a larger region detector rejects the testing window, there are two possible scenarios that could happen. First, the testing window is negative as the result of the detector. In this case, the result is correct without misclassification. Second, there is occlusion in that detection window and the detector misclassifies it. This case is crucial and will lead to incorrect result. In this case, the detector from smaller region has to be activated on this window instead. The questions are how to know when the detector from the smaller region is activated, and when there is any cue that will hint the system to activate the smaller one rather than always perform all the remaining detectors as it is time consuming to test.

2) In the boosted decision tree model, each boosted decision tree consists of many decision trees. The features used to test each image will be determined by the decision tree sequentially. Each tree reports its hypothesis. All hypotheses will subsequently be combined and assessed the final result. Using path label in each decision tree as proposed in Section 3.4, each hypothesis precipitated from the corresponding decision tree will serve as a hint for the labelled region. Based on this hint, if the sum of hypothesis values in each region is more than the threshold value, the detector of that region will be activated.

From the above two provisions, assumptions on hierarchical region structure for pedestrian detection can be deduced as follows:

1) Pedestrian regions can be arranged in hierarchical fashion based on level of appearance in both horizontal and vertical schemes.

2) A boosted decision tree with path label technique can yield hidden information and give a subregion hypothesis about the smaller region in hierarchical model.

3) Due to weakness of each individual decision tree, only one succeeding step in the hierarchical model is decided.

Let A be the set of appearance regions, H_a be the total hypothesis of region a where $a \in A$. A partial hypothesis of subregion b of region c from d^{th} individual decision tree is represented by $h_{b,c,d}$. The total hypothesis of each region can be expressed using subregion hypothesis as follows:

A full appearance region in horizontal scheme composes of the sum of hypothesis of decision tree with path label F and the sum of hypothesis of decision tree with path label $H75$, i.e.,

$$H_F = \sum_i h_{(F_h,F,i)} + \sum_j h_{(H75,F,j)}$$

A full appearance region in vertical scheme composes of the sum of hypothesis of decision tree with path label F and the sum of hypothesis of decision tree with path label VL, VR, and VLR, i.e.,

$$H_F = \sum_i h_{(F_v,F,i)} + \sum_j h_{(VL,F,j)} + \sum_k h_{(VR,F,k)} + \sum_m h_{(VLR,F,m)}$$

A horizontal level 1 region composes of the sum of hypothesis of decision tree with path label $H75$ and the sum of hypothesis of decision tree with path label $H50$, i.e.,

$$H_{H75} = \sum_i h_{(H75,H75,i)} + \sum_i h_{(H50,H75,i)}$$

A horizontal level 2 region composes of the sum of hypothesis of decision tree with path label $H50$ and the sum of hypothesis of decision tree with path label $H30$, i.e.,

$$H_{H50} = \sum_i h_{(H50,H50,i)} + \sum_i h_{(H30,H50,i)}$$

For horizontal level 3 region, there is no subregion in this region. So the hypothesis is defined by the sum of all decision tree in the region.

$$H_{H30} = \sum_i h_{(H30,H30,i)}$$

For vertical left region, there is no subregion in this region. So the hypothesis is defined by the sum of all decision tree in the region.

$$H_{VL} = \sum_i h_{(VL,VL,i)}$$

For vertical right region, there is no subregion in the region. So the hypothesis is defined by the sum of all decision tree in the region.

$$H_{VR} = \sum_i h_{(VR,VR,i)}$$

Normally, when testing the boosted decision tree, there is a threshold for checking current cumulative hypothesis after individual decision tree is performed to increase the speed of the system. This threshold is used for fast negative sample rejection. Only positive and hard negative samples will be further determined in the boosted decision tree. The occluded pedestrian samples are most likely to be treated like negative samples by the full pedestrian detector. From this threshold technique, there is a chance that the occluded pedestrian will be early rejected from the boosted decision tree before collected some hidden or subregion information which is the process for handling partial occlusion problem. To solve this problem, each subregion is assigned a token that check the cumulative subregion hypothesis to ensure that the test data is performed by enough number of decision trees and collect enough information about subregions to decide partial occlusion situation. The token is a binary value, when positive indicates that the hypothesis of subregion is more than a specific predefined positive threshold or lower than a specific predefined negative threshold. This two predefined thresholds are set to guarantee that the subregion is determined already before detection window rejection. Algorithm 9 shows token computation and Algorithm 10 shows token checking. Both threshold values are set manually to minimize computation time and maximize the experimental results. Despite the boosted decision tree classification takes more time with high value of positive threshold and low value of negative threshold, it yields more the information for use in subregion hypothesis of occlusion handling.

Algorithm 9 A subregion token computation (*ComputeToken*).

```
Input:
- A previous cumulative subregion hypothesis $H$.
- A current subregion hypothesis $h$.
Compute cumulative subregion hypothesis $H = H + h$.
if $(H \geq Thr_p)$ or $(H \leq Thr_n)$ then
    Assign subregion token $Z = 1$.
else
    Assign subregion token $Z = 0$.
end if
Output:
- A current cumulative subregion hypothesis $H$.
- A token value $Z$.
```

The proposed hierarchical region structure for pedestrian detection is shown in Figure 3.7 and the algorithm is given in Algorithm 11. Note that the algorithm only shows the highest level of hierarchical region structure.

Algorithm 10 A region token checking (*CheckToken*).

```
Input:
- A horizontal token $Z_h$.
- A set of vertical token $Z_v$.
if ($Z_h = 1$) and ($\exists Z_{vi} \in Z_v | Z_{vi} = 1$) then
    Assign region token $Z_{tot} = 1$.
else
    Assign region token $Z_{tot} = 0$.
end if
Output: Region token $Z_{tot}$.
```

Algorithm 11 A boosted decision tree classification using hierarchical region structure

```
Input:
- Testing image feature $I$ of detection window.
- A set of boosted decision trees of all region.
- A threshold values for current region $Thr_y$.
- A set of threshold values for horizontal region $Thr_h$.
- A set of threshold values for vertical region $Thr_v$
Initialize: $H_y = 0$, $H_h = 0$, $H_v = 0$, $Z_h = 0$, $Z_v = 0$.
for $i$ = 1 to $t$ do
    Classified $I$ using $u_i$.
    Get a hypothesis $hy_i$, path label $Ph_i$ , and $Py_i$ according to the classification result.
    Compute a hypothesis of current region, $H_y = H_y + hy_i$.
    Compute a hypothesis based on horizontal path label and token:
    $(H_h(Ph_i), Z_{hi})$ =ComputeToken$(H_h(Ph_i), hy_i)$.
    Compute a hypothesis based on vertical path label and token:
    $(H_v(Pv_i), Z_{vi}) =$ ComputeToken$(H_v(Pv_i) + hy_i)$.
    if $H_y \leq Thr_y$ and CheckToken($Z_h, Z_v$) then
        if $\exists q | (H_h(q) > Thr_h(q))$ or $\exists r | (H_v(r) > Thr_v(r))$ then
            Activate a boosted decision tree corresponding to $q$ or $r$.
        else
            Reject this detection window.
        end if
    end if
end for
if $H_y > Thr_y$ then
    Accept this detection window as a pedestrian.
else
    Reject this detection window.
end if
Output: Pedestrian classification result.
```

Table 3.2: Details of weight for each region detector.

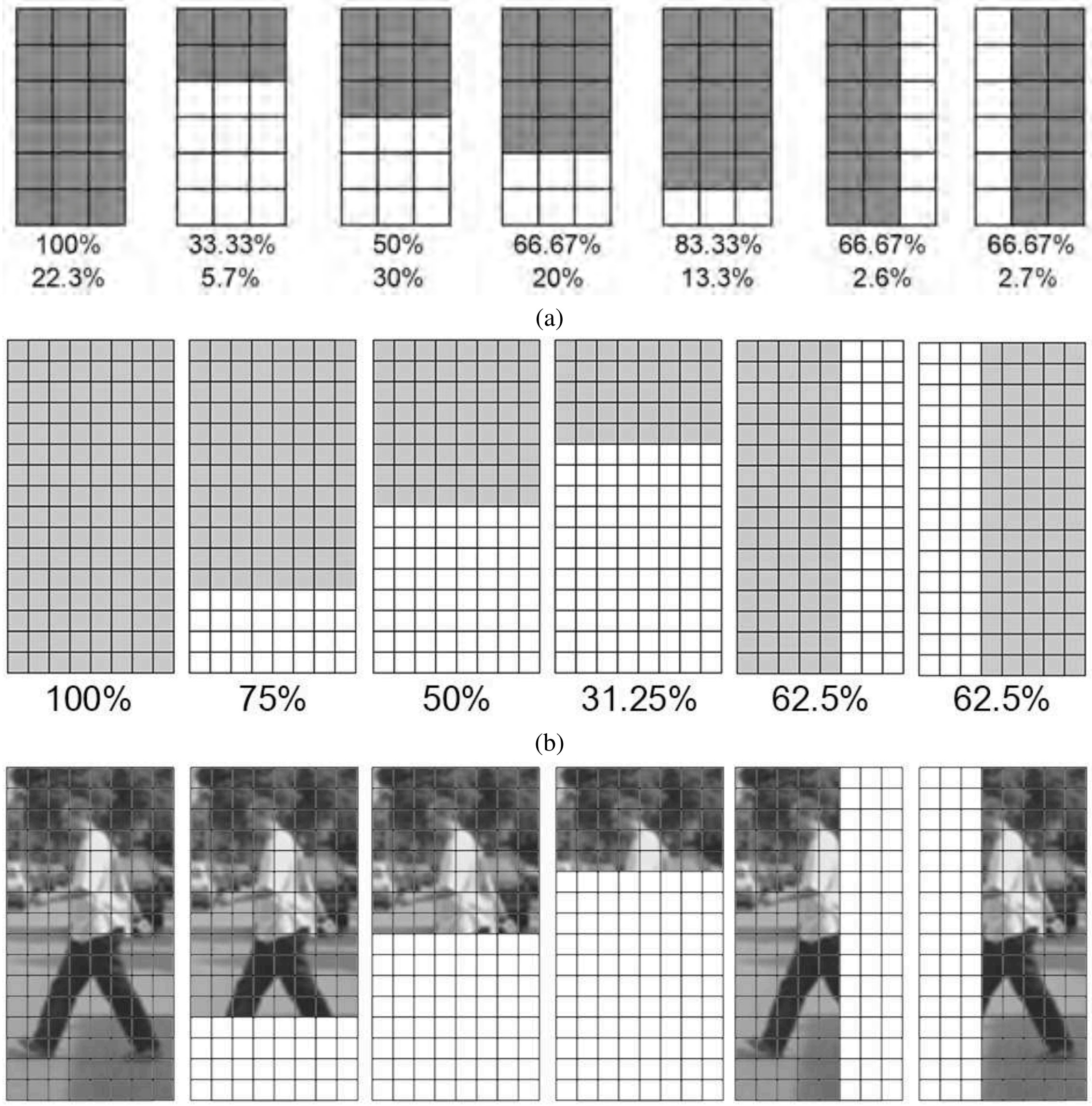
Region	Weight
Full	1
H75	0.75^2
H50	0.5^2
H30	0.3125^2
VL	0.625^2
VR	0.625^2

3.6 Combining Detection Score

A hypothesis or detection score represents the confidence of the detector about its result. In the proposed framework, there are many detectors performing detection tasks simultaneously. In many cases, the detection windows will be rejected by the first detector. With harder samples and positive samples, more detectors are required. Thus, a detection score is employed to accumulate the results obtained from all of these activated detectors. All subregion hypotheses along the hierarchical framework to the deepest activated detector are accumulated. Let B be the set of activated detectors, w be the weight of each detector defined by the square of level of visible region as shown in Table 3.2, and h be the accumulative hypothesis of activated subregion. The detection score D is computed by:

$$D = \sum_i w_i h_i \text{ where } i \in B$$

The objective of region weight is to set the priority for each hypothesis from each region. Based on the level of visibility of each region, detector of region with high level of visibility should have higher priority than the one with lower visibility level. So the proposed framework uses the level of visibility of each region as the base value. The accumulative hypothesis value should be moderately small because it serves as the suggested hypothesis value of the deepest activated detector. Squared of the level of visibility is used as weight for each subregion hypothesis. Computational results will be summarized in the experiments.



(a)

(b)

(c)

Figure 3.3: An example of pedestrian region schemes. (a) Occlusion scenarios as shown in occlusion statistics reported by Dollár et al. [66] including seven types of occlusion level. The upper numbers represent the level of visibility and the lower numbers show the proportion of occurrence in the video frames. (b) The proposed appearance-based pedestrian regions scheme along with level of visibility. (c) An example with actual regions when applied with the pedestrian image.

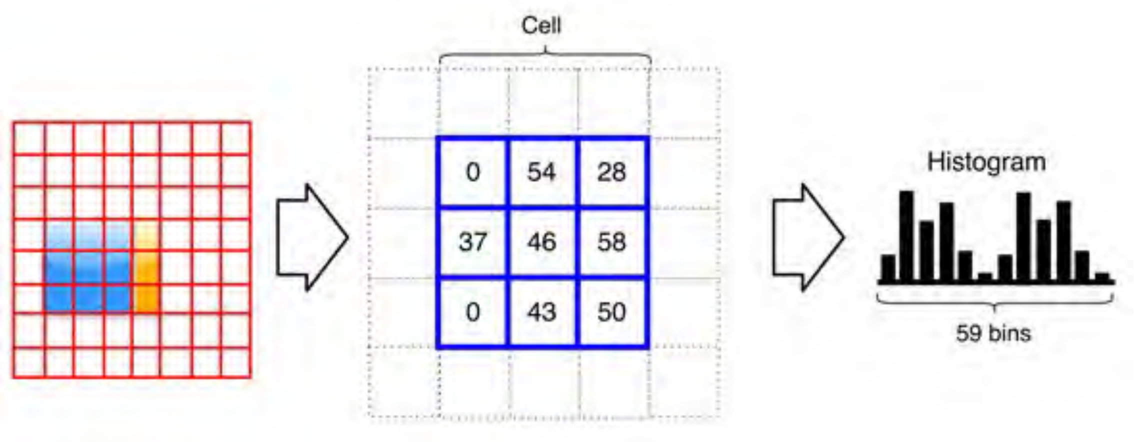


Figure 3.4: An example of uLBP process.

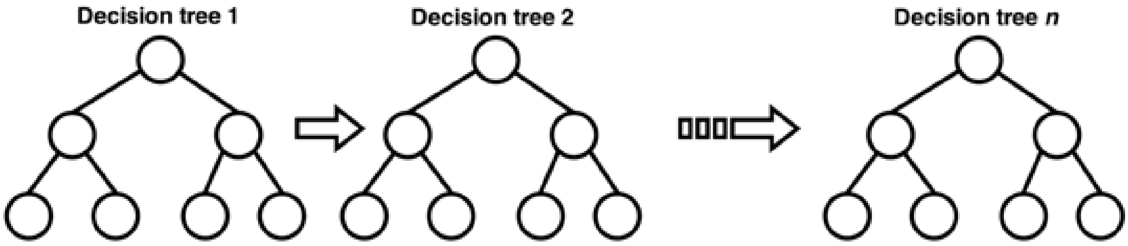
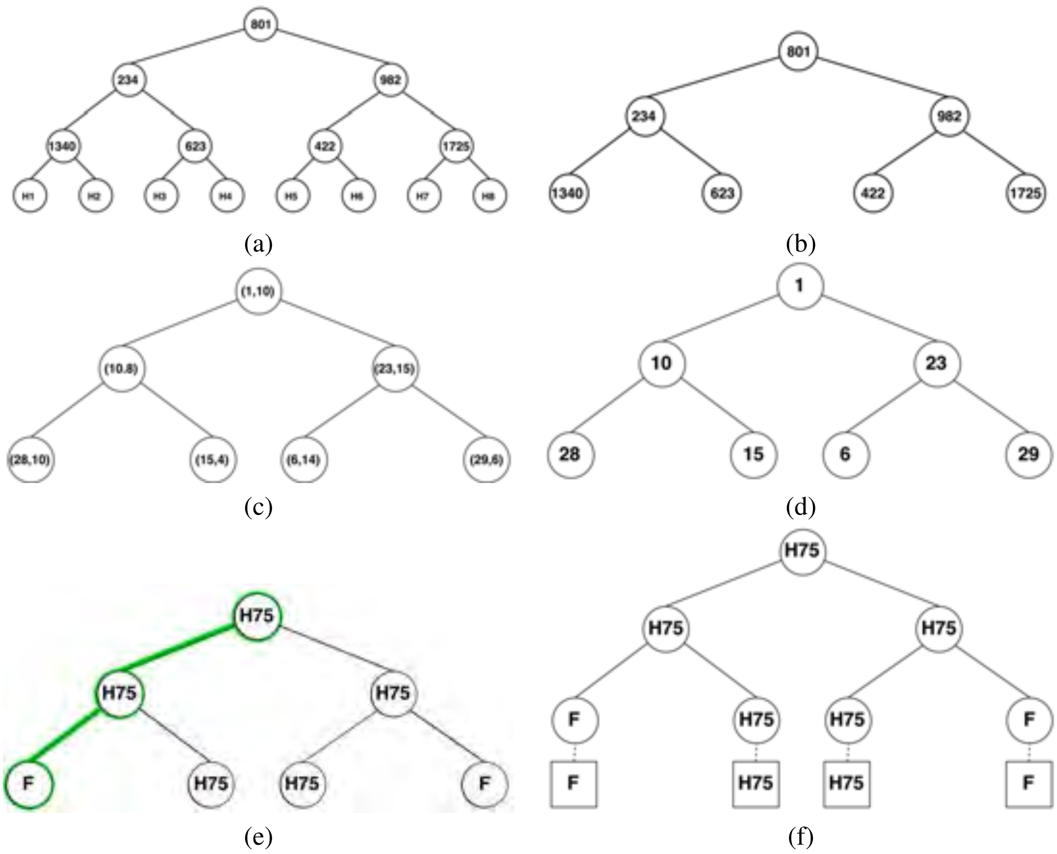


Figure 3.5: A structure of boosted decision tree with n decision tree.



(a) (b) (c) (d) (e) (f)

Figure 3.6: Examples of horizontal node label and path label process with three-depth decision tree. (a) A full decision tree. (b) Decision nodes. (c) Transforming feature to pixel location represented as (row, column). (d) Horizontal location of each node based on row value of pixel location. (e) Examples of decision path and assigning node label according to the related region. (f) Assigning path label for each path. The path label is the biggest region in the decision path.

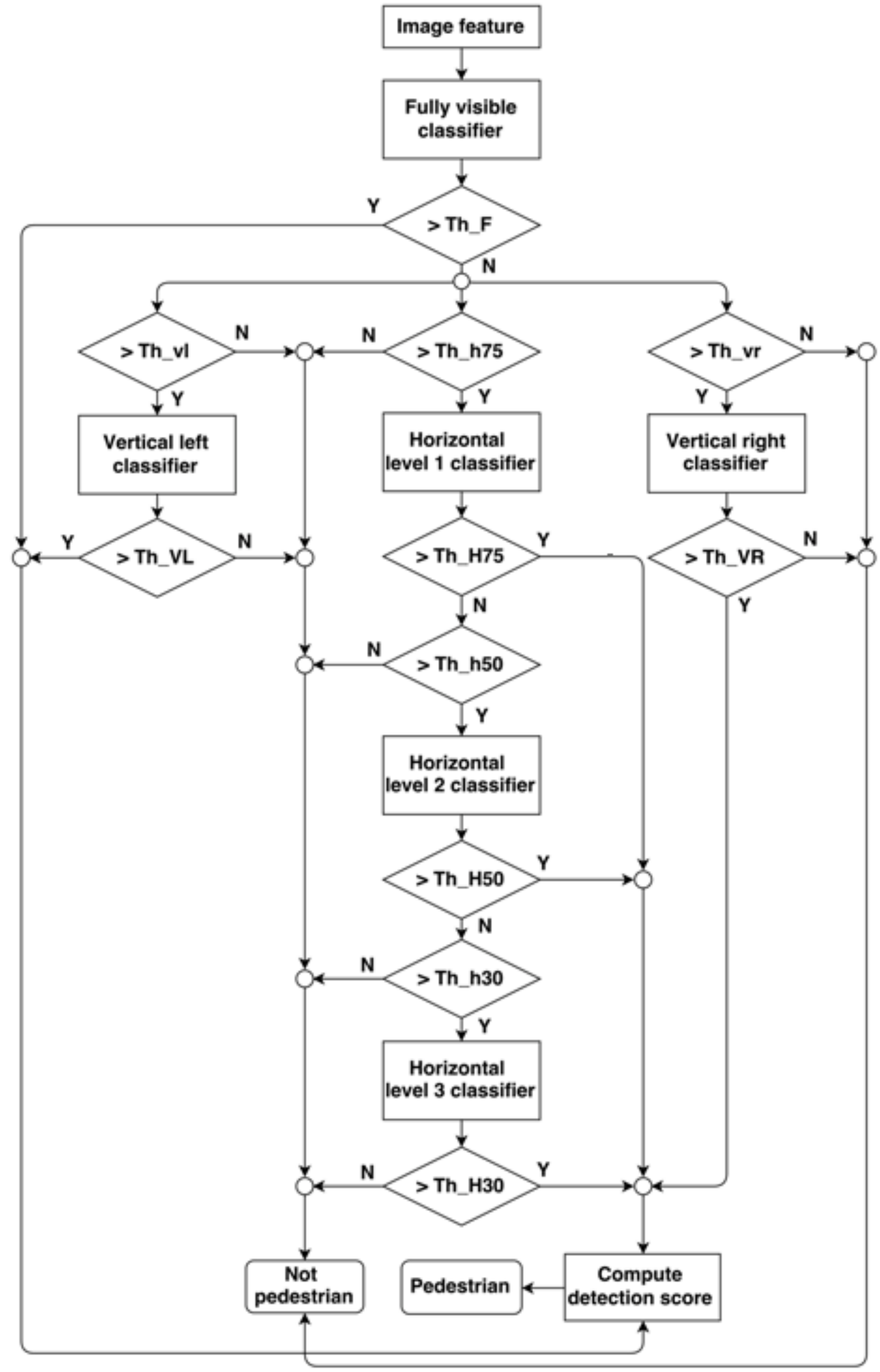


Figure 3.7: An overview of the hierarchical region structure.

CHAPTER IV

EXPERIMENTAL RESULTS

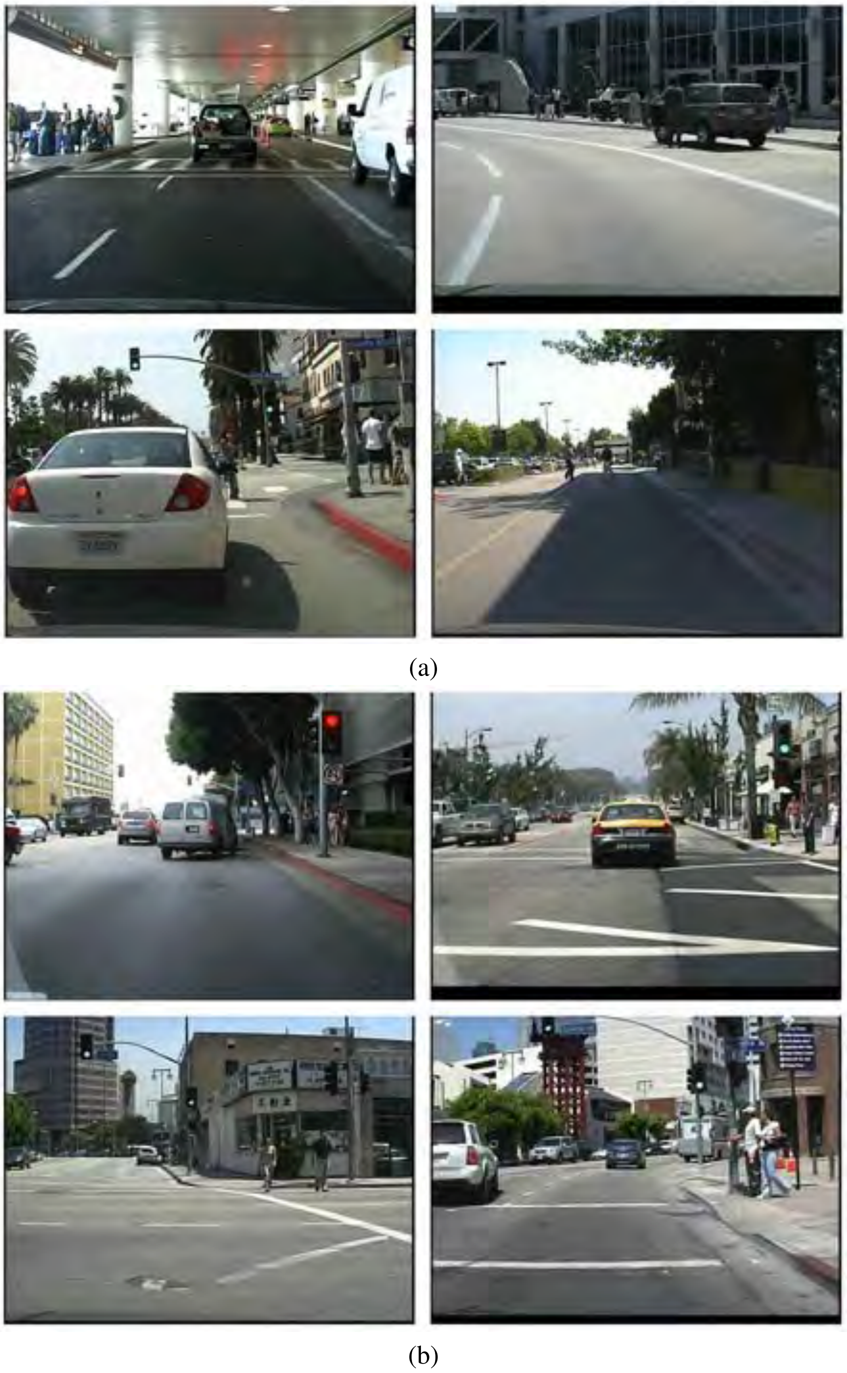
This section describes the details of datasets, experimental setting, and performance statistics. Some noteworthy issues are also discussed.

4.1 Dataset

There are many pedestrian datasets that provide standard input for this research such as INRIA person dataset [18], Caltech pedestrian detection benchmark [66], ETH Pedestrian Dataset [67], TUD-Brussels Pedestrian Dataset [26], and Daimler Mono Pedestrian Detection Benchmark Dataset (DPDB) [1]. Each dataset has a specific setting and environment. The most popular and challenging dataset is Caltech pedestrian detection benchmark (CPDB). The CPDB is a very large pedestrian dataset consisting of 10 hours of recording time. Each frame is captured at 30 Hz with dimension of 640x480 pixels. The total number of frames is over 240,000 frames from urban environments. There are approximately 2,300 unique pedestrians in the dataset with a total of 350,000 pedestrian bounding boxes (BB). The occluded pedestrians are bounded with occlusion label for testing reason. With this large data, the dataset is divided in to 11 sets. Each set is composed of image sequences running between 6-13 minutes. The first 6 sets are for training and last 5 sets are for testing purposes. Result summaries of the CPDB are shown in Table 4.1. The sample images from this dataset are displayed in Figure 4.1 and the positive samples are shown in Figure 4.2. The environment and setting of the CPDB dataset is the best matched with the objective and scope of this dissertation. Thus the proposed framework is trained and tested in the CPDB dataset.

Table 4.1: Characteristics of the Caltech pedestrian detection benchmark dataset.

	All images	Skip	Sampling image	Positive samples
Training set	128,382	4	32,077	24,498
Testing set	120,720	30	4,024	4,885



(a)

(b)

Figure 4.1: Example of images from the Caltech pedestrian detection benchmark dataset. (a) Images from training dataset. (b) Images from testing dataset.

Figure 4.2: Positive samples from the Caltech pedestrian detection benchmark dataset.

4.2 Performance Evaluation

Evaluation in this dissertation follows the CPDB dataset standard described in [66]. The detection system gets an image as an input, performs calculation in multiple-scale, and reports a set of answers including detection bounding boxes (BB_d) and corresponding detection confidence values. Some outputs may undergo the non-maximum suppression algorithm if necessary. A calculation of matching value (M_P) between BB_d with ground truth bounding box (BB_g) is based on PASCAL object detection measure defined as follows:

$$M_P = \frac{area(BB_d \cap BB_g)}{area(BB_d \cup BB_g)} > 0.5$$

The M_P values over 50% are counted as correct matched. Each mismatched BB_d is counted as false positive and unmatched BB_g is counted as false negative. Each BB_d and BB_g can be matched only once. If there are many BB_d candidates for each BB_g, the BB_d with the highest detection confident value will be matched first. The overall result is shown with the miss rate against false positive per image (FPPI) in log-log plot using varied threshold on detection confident values. The log average miss rate (LAMR) is an average miss rate of various FPPI rate in the range 10^{-2} to 10^{0}. Normally, the LAMR is the same as the performance at FPPI equals to 10^{-1}. The lower LAMR value is, the better performance evaluation becomes.

Due to the size of dataset, evaluation is performed on every 30^{th} frame in the test set to minimize high computations on all the detectors being compared. So there are 4,024 frames in total for testing. Most experiments utilize a reasonable test set that contains the ground truth of pedestrian over 50-pixel tall with unoccluded or partially occluded scenarios.

There are some noteworthy issues about how to compare the results with other detectors. First, to avoid any issues about other detectors that may be bias such as parameter tuning or incomplete implementation. There is no reimplementation of other detectors. Second, all detectors have to used the same evaluation code provided by the original dataset. Third, the

Table 4.2: Details of size and features for each region.

Region	Width	Height	Number of channels	Number of features
Full	16	32	10+59	35,328
H75	16	24	10+59	26,496
H50	16	16	10+59	17,664
H30	16	10	10+59	11,040
VL	10	32	10+59	22,080
VR	10	32	10+59	22,080

comparison has to use online detection results provided by the original authors of each detector.

4.3 Experimental Setting

Details of the experiment setting are described below.

Training data: The training data are dense sampling from the CPDB dataset. The training images are sampled from every fourth frames of the training set. A total of 32,077 image samples are collected. Positive examples are extracted from these data provided by the associated annotations from the dataset. The sampling rate of the training set affects the performance of the detector. With dense sampling with not more than every ten frames is preferred because there are more positive samples for training process. However, too low of dense sampling rate such as every one or two frames may result in overfitting detector. Negative samples are randomly generated from the training set with the size of 25,000 samples in the first round. In each round, bootstrapping process is performed to collect hard negative examples for the next round. There are 50,0000 accumulated negative samples after the second round of the training process.

Channel feature: The channel features of each regions are composed of 69 channels including 10 channels of ACF (3 channels from LUV color space, 1 channel of gradient magnitude, and 6 channels of six orientation bins of gradient histogram) and 59 channels of spatial uLBP. A total of 35,328 features are used per one fully detection window. The details about number of features for each region are shown in Table 4.2. The spatial uLBP calculation is performed on the luminance channel of LUV color space with the cell size of 3 × 3 pixels. Overlap stride between each consecutive cell is 1 pixel in both horizontal and vertical directions. For each pixel, the uLBP can be determined using its 8 neighbors.

Boosted decision tree: The boosted decision tree is trained with Adaboost in 4 rounds, each round is composed of different number of weak decision trees, such as 64, 256, 1,024, and 4,096, respectively. More number of training rounds will construct the detector with more fitting to the training set. More number of weak decision trees will have more accuracy but take longer time in both training and testing process. Each individual decision tree is a 4-depth tree. The depth of the tree is also one factor that affects the overall result. With more depth, each decision tree becomes more complicated and uses more combination of features to determine the hypothesis. However, for Adaboost approach each decision tree should be a weak classifier. So the decision tree with depth of 3 to 5 are the reasonable options.

Detector: The detector is designed with detection window of size 64 pixels in height and 32 pixels in width to support the pedestrian with 50-pixel tall. The detection window is shrunk with a scale factor of 2 to 32 pixels in height and 16 pixels in width to reduce computation load. The details for each region are shown in Table 4.2.

Detection strategy: The detectors are fixed in size but the input image is resampled in multiple scale for multiple scale detection purpose. The detection process is performed in 7 scales with additional 8 refinement scales per octave, for a total of 55 scales per image. A stride for each detection window is 4 pixels wide in both horizontal and vertical directions. While perform detection, the input image is resampled in 55 scales. The detectors perform in each resampled image for multi-scale detection that will support many size of pedestrian in the video frames, thereby smaller objects such as children can be handled accordingly.

MSPT construction details: The MSPT in this work is constructed using 1,280 pedestrian images. More number of pedestrian images are preferred to cover pedestrian variation. Each image is manually labeled to pedestrian shape silhouette image. Each silhouette image is modified to 10 minor variation silhouette images by using one pixel translation in four directions and flipped as shown in Figure 4.3a. The objective of this process is to increase the number of the silhouette images to handle more posture variations. Thus, there are 12,180 pedestrian silhouette images for MSPT construction. The MSPT is shown in Figure 4.3b.

Baseline detector: A baseline detector is the strong detector that is used to compare with the proposed detector. The baseline detector [61] is the modified version of ACF detector [31] trained with deeper decision tree (depth 5) with dense data sampling. This baseline detector called ACF-Caltech+ is ranked fifth in CPDB reasonable dataset.

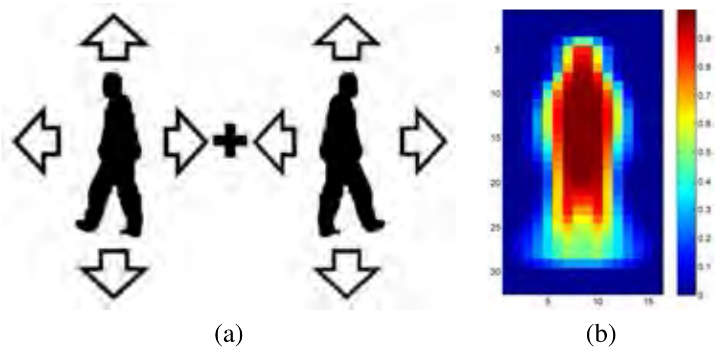


Figure 4.3: (a) Each pedestrian silhouette image is shifted and flipped. (b) Visualization of MSPT.

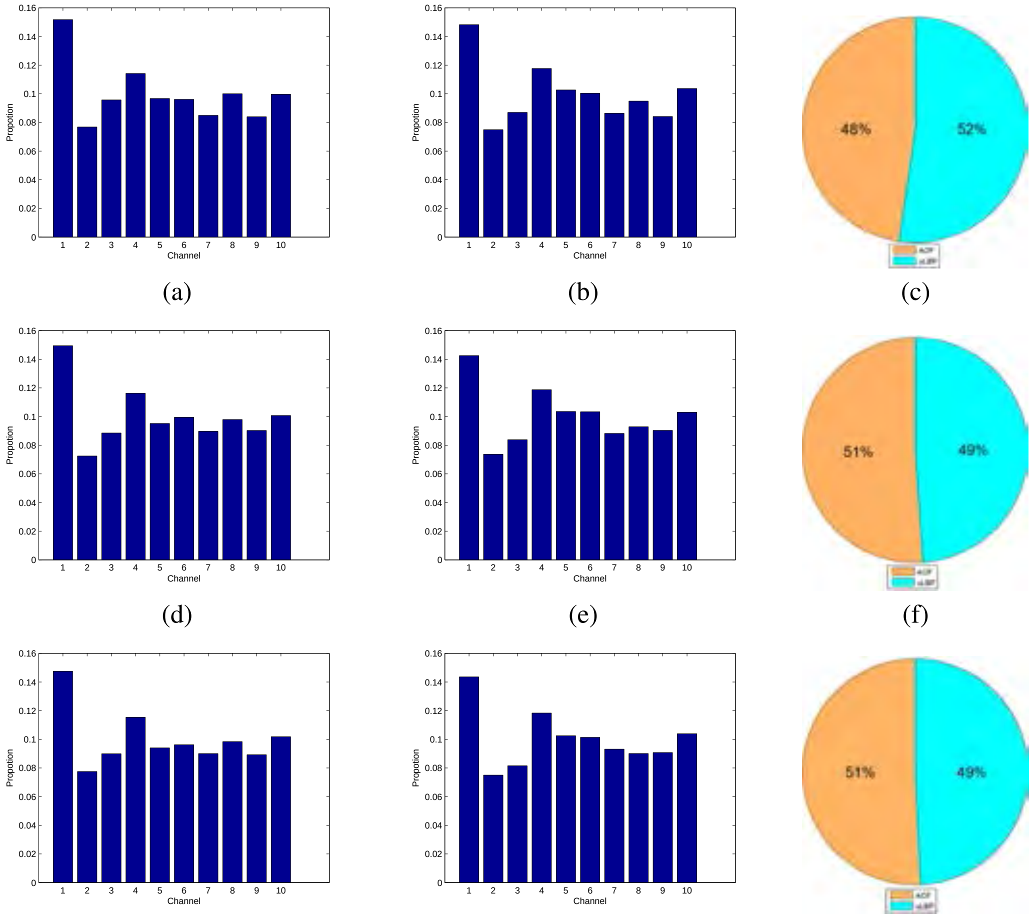
4.4 Experimental Results

The experimental results obtained from CPDB dataset are grouped into four categories to denote the representative effectiveness of the proposed framework in feature evaluation and effect of MSPT approach, performance of region detectors, comparative statistics results of proposed framework with other methods, and run-time performance.

4.4.1 Feature Evaluation and Effect of MSPT Approach

In this section, the channel features are analyzed in details with the effect of MSPT on the type of selected ACF features. As mentioned earlier, ACF is composed of 10 channel features that include 3 channels of LUV color, 1 channel of gradient magnitude, and 6 channels of gradient orientations. The power of each channel is explored based on the features being selected by Adaboost in training process. The overall results of each region are illustrated in Figure 4.4 and 4.5. In each figure, the left most figure represents features in baseline detectors without applying MSPT and the middle figure represents features in the proposed detector applying MSPT.

In case of individual feature comparison, the luminance channel of LUV color space is the top feature selected by Adaboost. The proportion of luminance channel being selected is approximately 15% of the total selected features, yielding the same results for every regions. This means that luminance channel is very informative and important feature for pedestrian classification. The runner-up feature is gradient magnitude feature. The U and V channels of LUV color are the least informative features. After applying MSPT, the number of selected features is different from the baseline detector but the projection is the same. The luminance



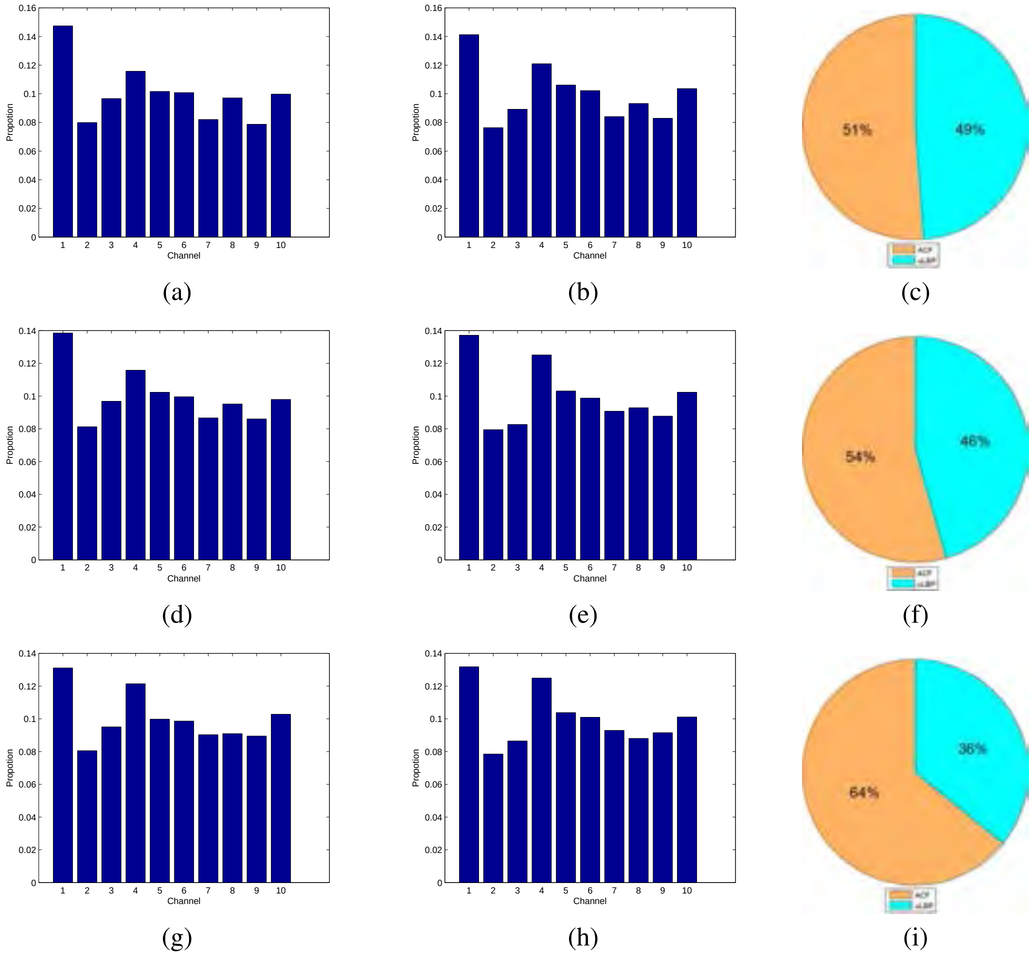
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.4: Proportion of feature channels in full and vertical regions. (a) ACF without MSPT in full region. (b) MSPT applied in full region. (c) Proportion between ACF and uLBP in full region.(d) ACF without MSPT in VL region. (e) MSPT applied in VL region. (f) Proportion between ACF and uLBP in VL region. (g) ACF without MSPT in VR region. (h) MSPT applied in VR region. (i) Proportion between ACF and uLBP in VR region.

channel is still the popular channel followed by gradient magnitude channel. The use of MSPT does not effect the type of selected features directly and the proportion of selected features change slightly. In contrast, MSPT approach affects the distribution of selected features in spatial domain. Visualization of selected feature distribution is displayed in Figure 4.6 and Figure 4.7. In each figure, the baseline detectors are shown in the left and the proposed detector with MSPT applied is shown in the right. From the experimental results, selected features using MSPT are more relevant to the pedestrian shape. As seen on the results of baseline detectors, many selected features are outside the pedestrian shape but some main features are in the same area as the proposed detectors. In the full region, the most attractive features are human face and shoulder. The results are the same in other regions. In addition, for small region such as



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.5: Proportion of feature channels in horizontal regions. (a) ACF without MSPT in H75 region. (b) MSPT applied in H75 region. (c) Proportion between ACF and uLBP in H75 region.(d) ACF without MSPT in H50 region. (e) MSPT applied in H50 region. (f) Proportion between ACF and uLBP in H50 region. (g) ACF without MSPT in H30 region. (h) MSPT applied in H30 region. (i) Proportion between ACF and uLBP in H30 region.

H50 and H30, the experimental results of detector without MSPT yield slightly better results than applying MSPT. From experimental observation, the feature selection process is forced to select small portion of candidate features based on MSPT which are very small due to the region size and feature availability. So the selected features are often chosen repeatedly with different combinations. This may limit the capability of the small region detector. Thus, MSPT is not applied with small regions. When combine ACF with uLBP, the proportion of 2 types of features selected by Adaboost are established. The last column of Figure 4.4 and Figure 4.5 show the experimental findings. Most regions report relatively the same proportion of ACF and uLBP being selected by Adaboost. In the small region detectors such as H30 and H50, ACF features are selected more than those of the uLBP. From the experiment, the MSPT approach organizes Adaboost to select the semantic features that corresponds to pedestrian shape without the proportion of selected features. The ACF and uLBP are selected at the same proportion by Adaboost means both ACF and uLBP are important features.

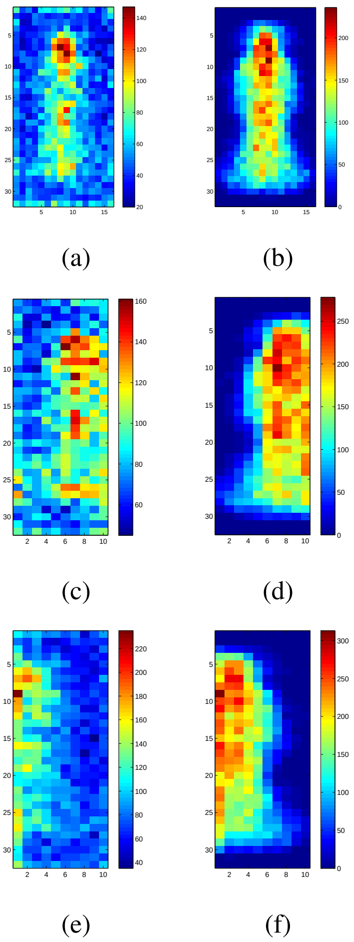
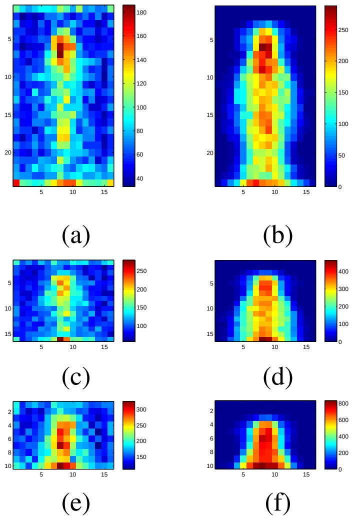


Figure 4.6: Visualization of feature distribution selected by Adaboost in full and vertical regions. (a) ACF without MSPT in full region. (b) MSPT applied in full region. (c) ACF without MSPT in VL region. (d) MSPT applied in VL region. (e) ACF without MSPT in VR region. (f) MSPT applied in VR region.

4.4.2 Performance of Region Detectors

In order to investigate the performance of each pedestrian appearance region detectors, all six region detectors are trained with training dataset and tested on CPDB reasonable dataset to compare the experimental results. Each region detector is train by using the same positive sample but difference in terms of size according to the region size. The ground truths of training and testing data are adjusted based on the size of the region detector for comparable purpose.

From the results in Figure 4.8, the region having the highest performance is the full



(a) (b)

(c) (d)

(e) (f)

Figure 4.7: Visualization of feature distribution selected by Adaboost in horizontal regions. (a) ACF without MSPT in H75 region. (b) MSPT applied in H75 region. (c) ACF without MSPT in H50 region. (d) MSPT applied in H50 region. (e) ACF without MSPT in H30 region. (f) MSPT applied in H30 region.

appearance region at 25% LAMR, followed by the VL at 41%, H75 at 41%, VR at 43%, H50 at 66%, and H30 at 79%, respectively. From experimental observations, the experimental results exhibit a direct variation with the area of pedestrian appearance regions in the image. In a larger region, the features can capture the essential information representing the pedestrian and the detector can learn from more areas of pedestrian. As expected, the smaller regions have the high LAMR and tends to generate more false positive results that worsen the performance of the system. This experiment supports the assumptions about part weights when the level of visibility decreases, the performance of individual detector also decreases. So reducing weight of small region detectors as proposed is acceptable and have to combine the hypothesis with other bigger region detectors to arrive at the final results.

4.4.3 Proposed Framework Results and Comparison with Other Methods

The proposed framework is composed of 3 enhancement approaches in both training time and testing time, namely, MSPT approach, the ACF-uLBP combination approach, and hierarchical approach. The experimental results of the enhancement are displayed in Figure 4.9. The custom ACF detector is the baseline result with 30% LAMR. The first MSPT approach enhance the result of baseline detector nearly 2% LAMR and does not affect in speed because it is applied in training process. The testing time enhancement using combination of ACF-uLBP is

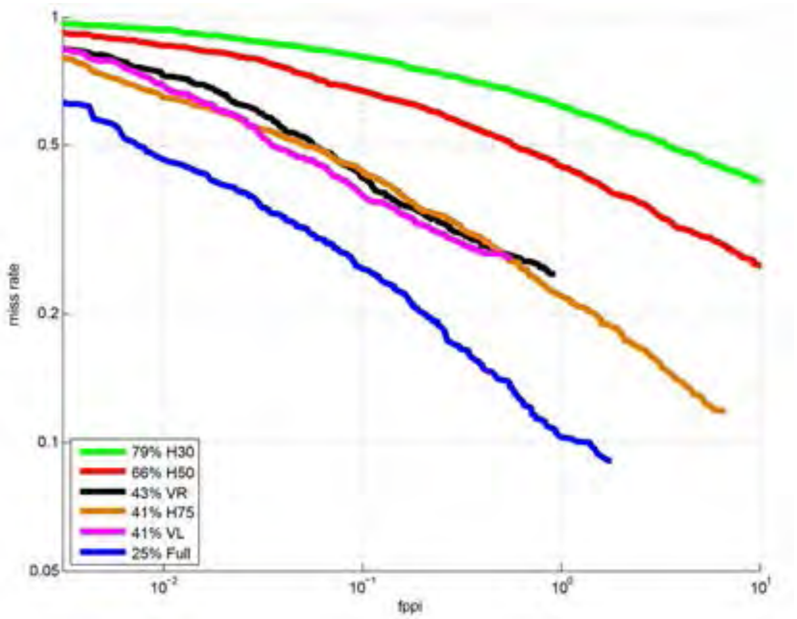


Figure 4.8: Experimental results with different pedestrian appearance regions.

applied which boosts the result by approximately 3% LAMR. After applying hierarchical approach to ensemble all region detectors, the result is slightly better by 0.15% but the framework becomes capable of handling partial occlusion problem. In this scenario, Mathias et al. [60] pointed out that the CPDB test set contained very low number of partial occlusion cases. So the effect of hierarchical framework is not boosted as expected. There is virtually no dataset that focuses directly on partial occlusion problem. This experimental results show that all 3 enhancements of the proposed method can speed up the overall performance of the baseline detector in nearly 5% LAMR from the baseline detector.

For comparative purpose, the proposed framework is compared with multiple methods. The results of 46 methods are reported on CPDB dataset and available online for comparison. In each experiment, only 8 results are displayed in the graph in terms of 6 best performance algorithms and standard HOG [18] and VJ [25] algorithms as two common baselines. There are 7 experiments tested on the CPDB dataset including reasonable, overall, near scale, medium scale, far scale, no occlusion, and partial occlusion experiments.

The reasonable test set is the most standard test set of CPDB dataset. All algorithms have to test and report the result of this dataset for standard comparison purpose. The reasonable dataset is tested on the pedestrian with size at least 50-pixel with non-occlusion and

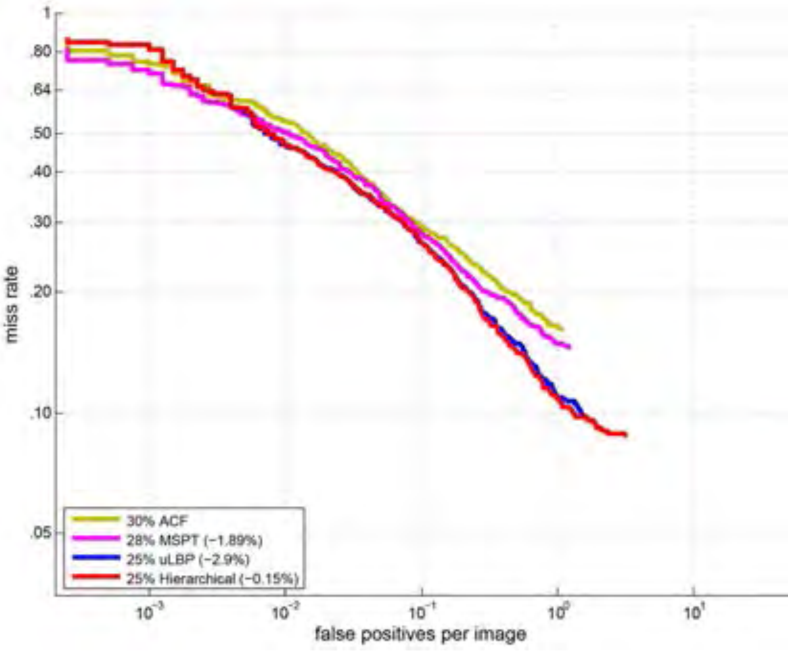


Figure 4.9: Experimental results show the enhancement of the proposed framework.

partial occlusion. The results of the proposed detector compare with other methods are displayed in Figure 4.10. The proposed method is ranked fourth in CPDB reasonable dataset after Spatialpoolng+ [50], Katamari [5], and LDCF [61], respectively. The proposed detector boosts the LAMR from the baseline detector shown as ACF-Caltech+ with an additional 5% LAMR. From the investigation, this boost up is resulted from MSPT and ACF-uLBP combination while the effect on hierarchical framework is small.

The example of detection results in reasonable dataset is shown in Figure 4.11. A green BB represents matched BB_d. A red BB represent mismatched BB_d or false positive. A yellow BB stands for unmatched BB_g or false negative. A cyan BB means missing ground truth, and a white BB is ground truth BB_g. Figure 4.12 shows more examples of the matched BB_d or true positive samples, while false positive samples are shown in Figure 4.13. There are some missing ground truth in the samples that affect in the overall results of the proposed results. False negative samples are displayed in Figure 4.14. In terms of missing ground truth, this is a crucial mistake of the dataset that may affect the overall results of the proposed detector because there is a pedestrian in the BB_d reported by the detector but there is no ground truth to support it. So the evaluation process will count this BB_d as false positive and degrades the performance of the proposed detector.

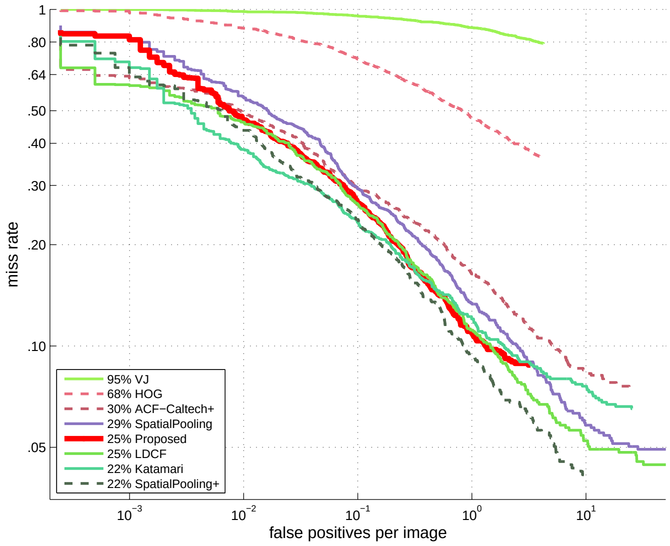


Figure 4.10: Experimental results on the CPDB reasonable dataset.

For detail evaluation on the CPDB dataset, the reasonable dataset is decomposed to 6 additional scenarios, namely, all dataset, large scale dataset, medium scale dataset, near dataset, non-occlusion dataset, and partial occlusion dataset. The proposed detector is tested on these additional test sets. The *all dataset* tests on all pedestrian annotated in the test dataset. The *large scale dataset* focuses on pedestrian of at least 100-pixel image. The *medium scale dataset* targets on pedestrian of 30 to 80-pixel image. The *near dataset* tests on at least 80-pixel pedestrian image. The *non-occlusion dataset* focuses on 50-pixel pedestrian image with no occlusion, and the *partial occlusion dataset* aims to the 50-pixel pedestrian image with partial occlusion. The experimental results are shown in Figure 4.15(a) - 4.15(f).

In all datasets, all detectors perform worse over 70% LAMR. The proposed framework is ranked fourth with 73% LAMR. In the large scale dataset, the result of the proposed framework is ranked first with only 5% LAMR. Some false negative samples of large scale dataset is shown in Figure 4.16. In addition, the large scale dataset corresponding to the size of pedestrian appears suddenly in the video frame and causes the accident. For medium scale dataset, all detectors perform at over 60% LAMR. The proposed detector is ranked fifth with 65% LAMR. In near dataset, the proposed framework is fourth ranked with 11% LAMR and ranked third with 21% LAMR on non-occlusion dataset.

For the partial occlusion dataset, the result of proposed method is ranked sixth. The detail

Table 4.3: Top pedestrian detectors with associate LAMR and reported BB_d.

Detector	Reasonable LAMR	Number of Reported BB_d
SDN [41]	37.8714	10,953
ACF-Caltech+ [61]	29.7591	106,855
Proposed	24.8877	14,890
LDCF [61]	24.7976	224,983
Katamari [5]	22.4898	104,732
SpatialPooling+ [50]	21.8875	44,548

of detection results on partial occlusion dataset is displayed in Figure 4.17-4.19. There are some missing ground truths that should be considered as partial occlusion case and detected by the proposed detector. These cases degrade the performance of the proposed detector as mentioned earlier. One interesting observation about this experiment is that the top detectors are not designed for partial occlusion problem but still yield good results in partial occlusion dataset. To investigate this issue, the results of other top detectors are evaluated with some examples shown in Figure 4.20. From the observation, those detectors report many false positive windows as shown in table 4.3. In real world applications such as driven assistance system, the number of false positives is very sensitive. An ideal system should report as low number of false positive as possible.

Other performance evaluation methods focusing on the number of false positives are also added. The average precision (AP) or average ratio of matched BB_d and reported BB_d describe the situation of the detector that reports too many false positive BB. The methods are defined below.

$$Precision = \frac{Number of matched BB_d + 1}{Number of reported BB_d + 1}$$

$$AP = \frac{\sum_{i=1}^{N} Precision_i}{N}$$

where N is the number of test images.

One of the best detectors is a comparative technique to use LAMR for theoretical analysis and AP for practical analysis. The results is shown in Table 4.4. By comparing these performance measures, the proposed framework reports low number of BB_d. The AP number is ranked second after SDN. Taking LAMR into consideration, the proposed method outperforms

Table 4.4: Top pedestrian detectors with associate LAMR and AP with standard deviation.

Detector	Reasonable LAMR	AP	Standard deviation
SDN [41]	37.8714	0.4521	0.2799
ACF-Caltech+ [61]	29.7591	0.0514	0.0312
Proposed	24.8877	0.4023	0.2823
LDCF [61]	24.7976	0.0239	0.0153
Katamari [5]	22.4898	0.0546	0.0433
SpatialPooling+ [50]	21.8875	0.1229	0.0826

Table 4.5: Top pedestrian detectors with associate LAMR, running time, and machine.

Detector	Reasonable LAMR	Running time	Machine
SDN [41]	37.8714	10 fps	NVIDIA GTX 760 GPU
ACF-Caltech+ [61]	29.7591	30 fps	Intel Core i7 CPU(3.4GHz)
Proposed	24.8877	4 fps	Intel Core i7 CPU(3.4GHz)
LDCF [61]	24.7976	4 fps	Intel Core i7 CPU(3.4GHz)
Informed Haar [5]	34.5980	0.625 fps	Intel Core i7 CPU(3.5GHz)
SpatialPooling+ [50]	21.8875	0.172 fps (only scanning time)	Parallelized quad core Intel Xeon processor

SDN by 13% LAMR. Thus, the proposed framework is a reasonable option for real world application.

4.4.4 Running Time Performance

Most of the top detectors is based on channel features. The detectors perform on a very large pool of features to enhance their performance. Strong powerful results come with more features and computation time that in effect become the bottle neck of implementation in real world applications. Balancing the trade-off between running time and performance has to be carefully considered. For the proposed method, the performance is enhanced in 3 steps. First, use MSPT in training step so that the running time during testing is not affected by this enhancement. Second, combine ACF and uLBP for feature representation. Third, apply hierarchical framework with more detectors to solving specific scenarios. In this case, performance

Table 4.6: Performance of variance enhancement of the proposed method.

Enhancement	Reasonable LAMR	Running time	Number of Reported BB_d	AP
MSPT	27.94	30 fps	6,578	0.5404
ACF-uLBP [61]	25.04	4.5 fps	8,929	0.6314
Hierarchical framework	24.89	4 fps	14,890	0.4023

of specific situations is enhanced but running time will be worsened due to more feature and detector computations. It is hard to compare running time of each detector because the original papers do not provide enough information of testing machine. The example of running time of various detectors are shown in Table 4.5 along with the details of testing machine and LAMR. From the Table 4.5, ACF-Caltech+ can run at 30 fps with an Intel Core i7 CPU (3.4GHz), while SpatialPooling+ is just very slow (the reporting time of SpatialPooling+ is only scanning time not including feature calculation and non-maximal suppression). This means that elegant results come from high computation time and very hard to implement in real applications. The proposed method is tested on an Intel Core i7 CPU (3.4GHz). Table 4.6 shows the trade-off between running time and performance of the proposed framework. With each enhancement, the LAMR is better, AP slightly increases but running time slows down considerably. Note that the running time can be improved by using another search space reduction and tracking algorithm. In so doing, the number of candidate windows will be reduced. With the various enhancements of the proposed method, adopting applications can be fined tune to boost their performance. Specially, the MSPT enhancement is best for speed; the ACF-uLBP is preferred for lower miss rate and running time; and the hierarchical framework is an attractive enhancement for handling specific scenarios like partial occlusion.

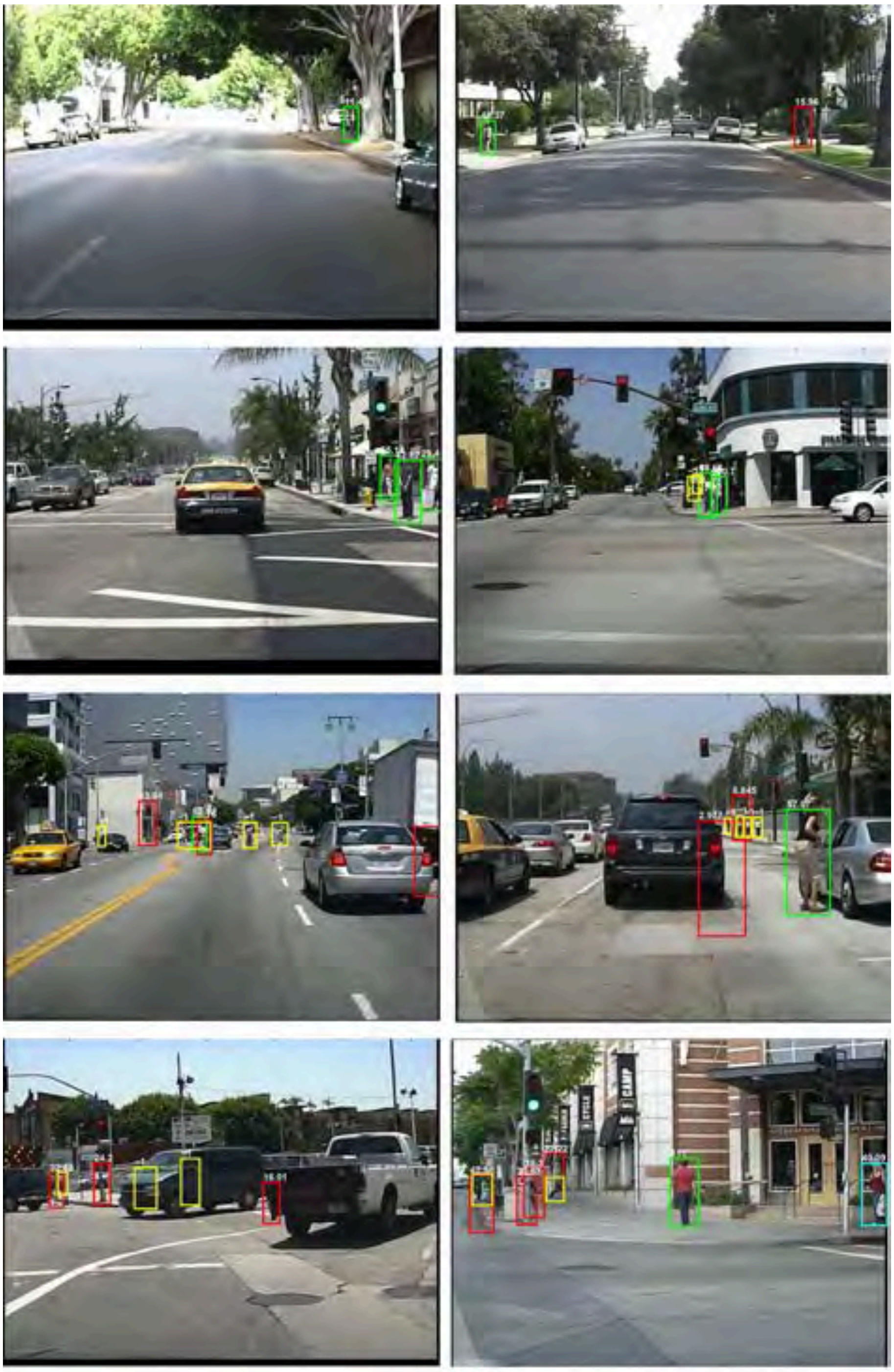


Figure 4.11: Example of detection results in CPDB dataset.

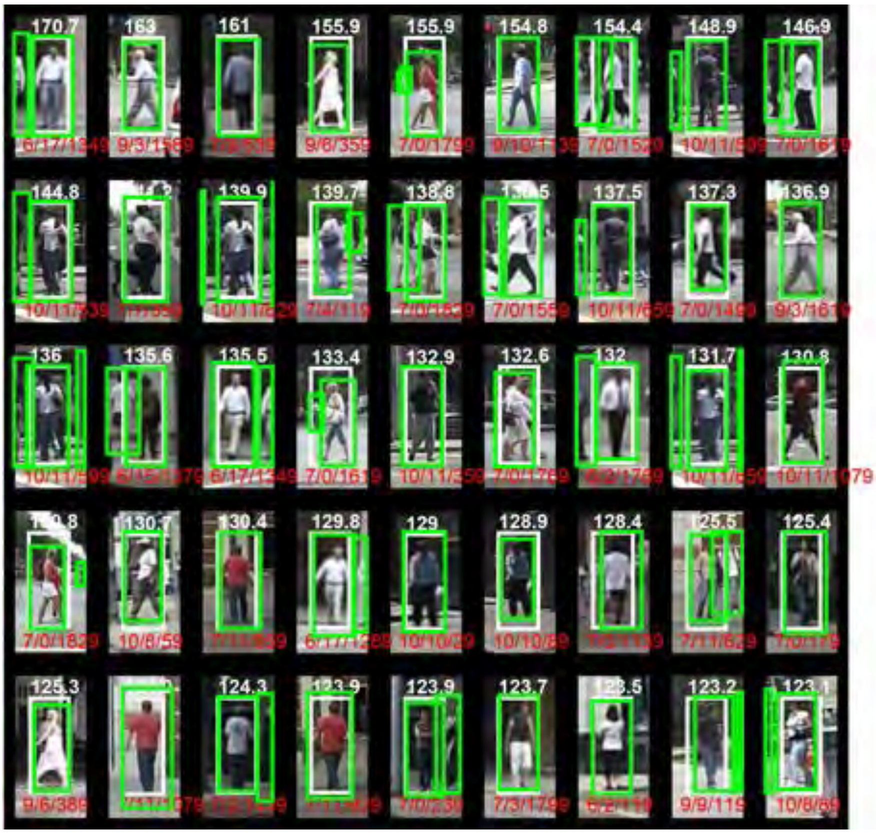


Figure 4.12: True positive detection results in reasonable CPDB dataset.

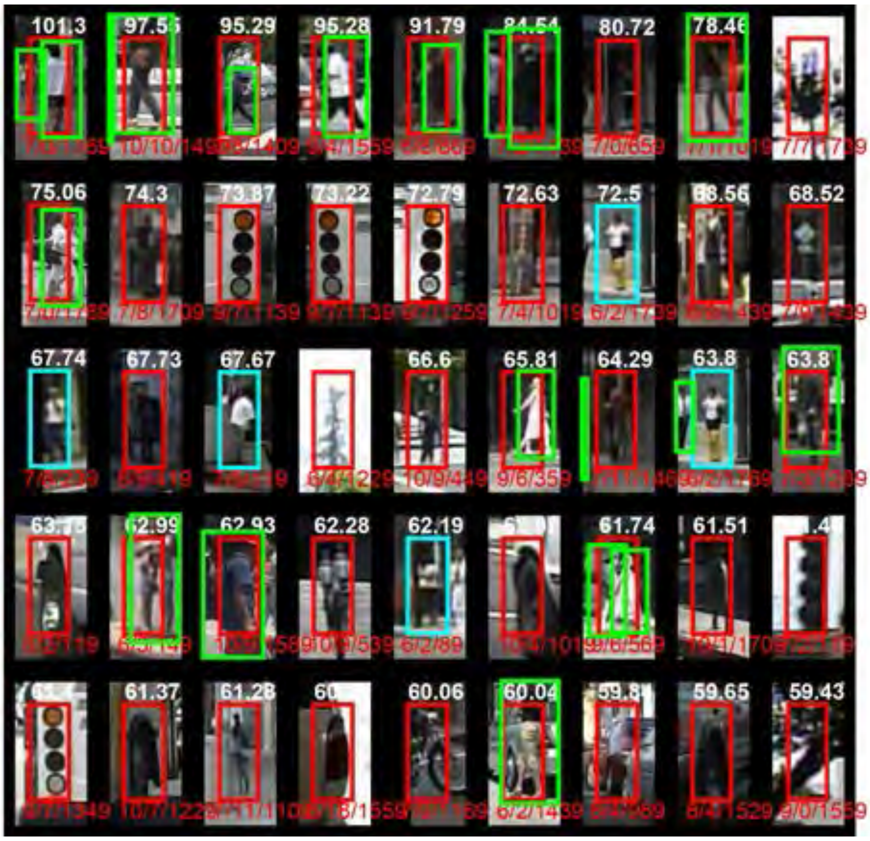


Figure 4.13: False positive detection results in reasonable CPDB dataset.

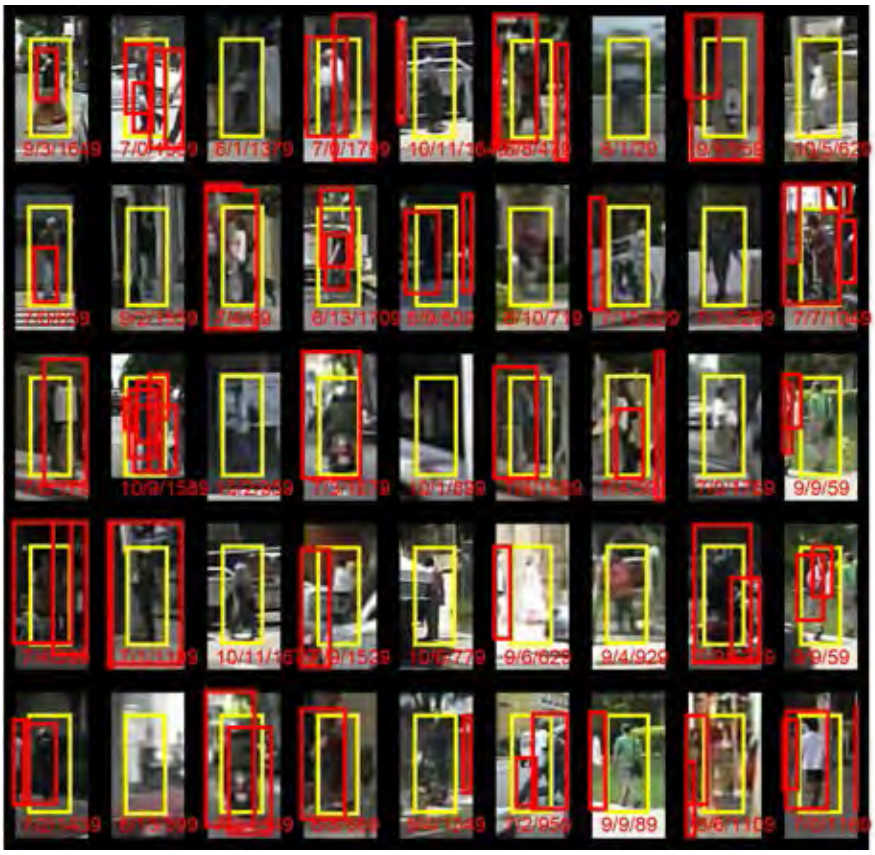


Figure 4.14: False negative detection results in reasonable CPDB dataset.

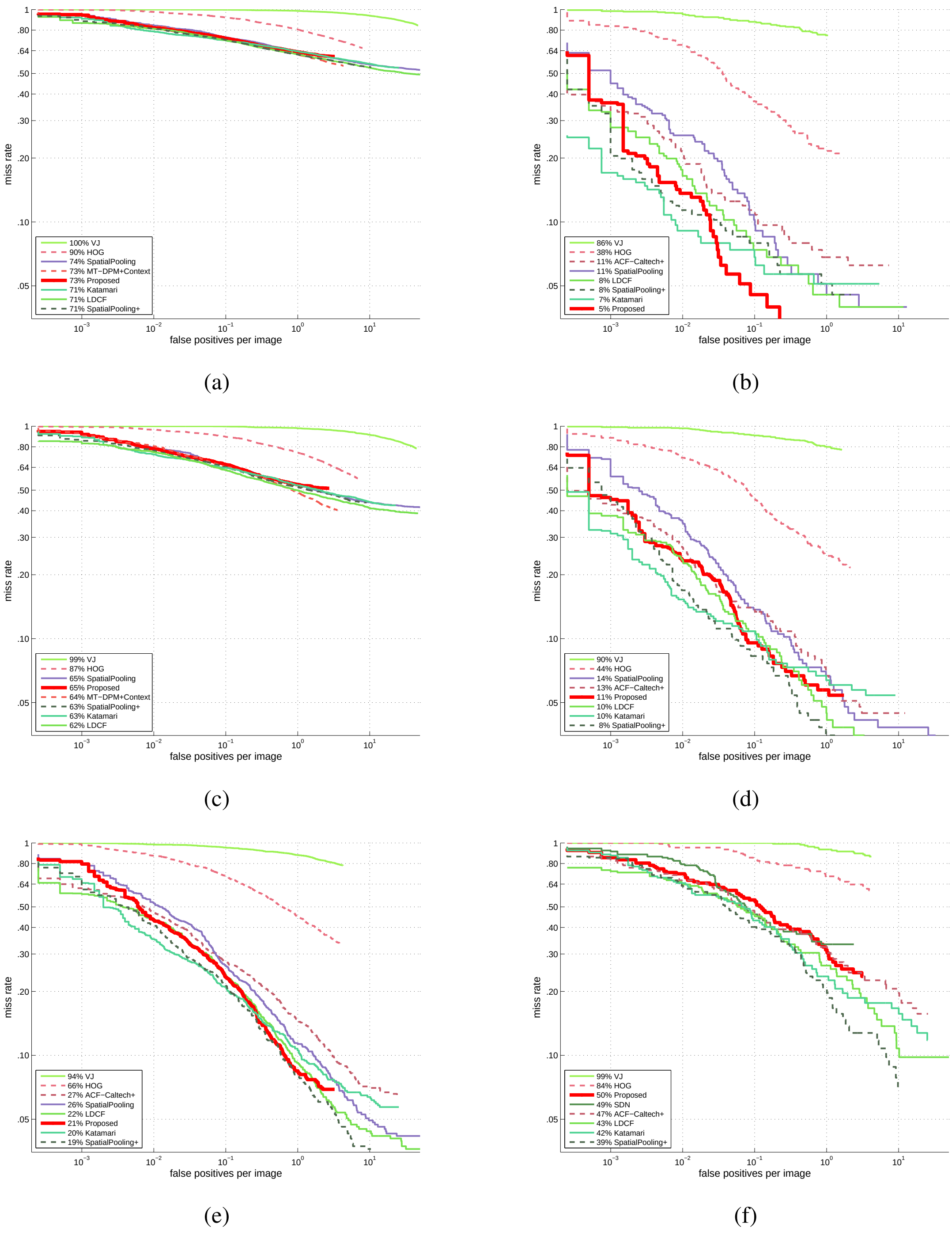


Figure 4.15: Results on CPDB dataset in details. (a) All dataset. (b) Large scale dataset. (c) Medium scale dataset. (d) Near dataset. (e) None occlusion dataset. (f) Partial occlusion dataset.

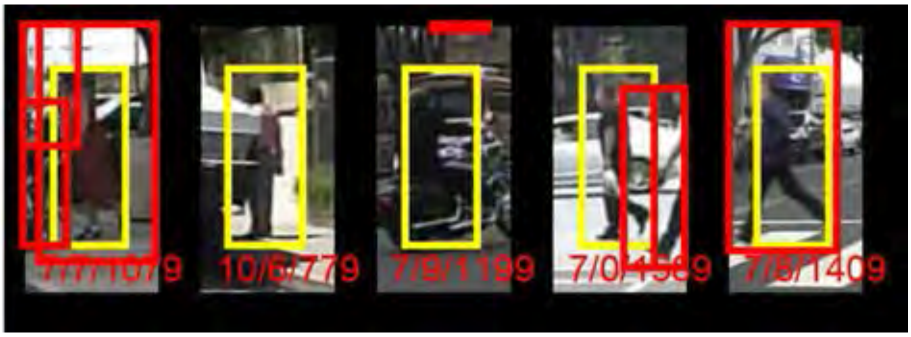


Figure 4.16: False negative detection results in large CPDB dataset.

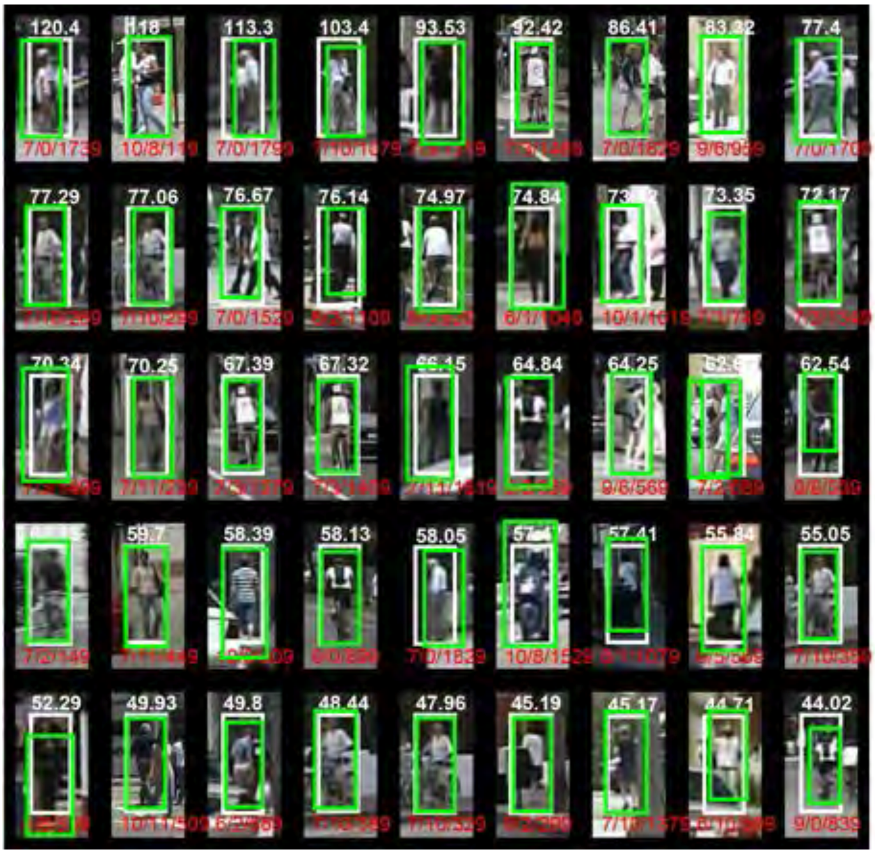


Figure 4.17: True positive detection results in partial occlusion CPDB dataset.

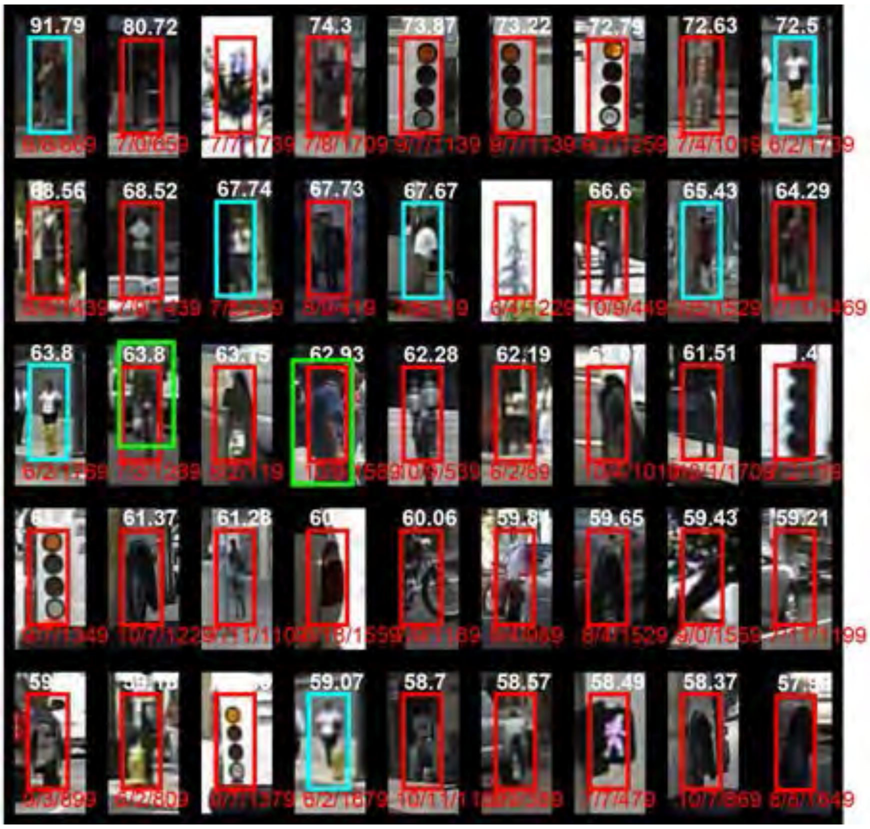


Figure 4.18: False positive detection results in partial occlusion CPDB dataset.

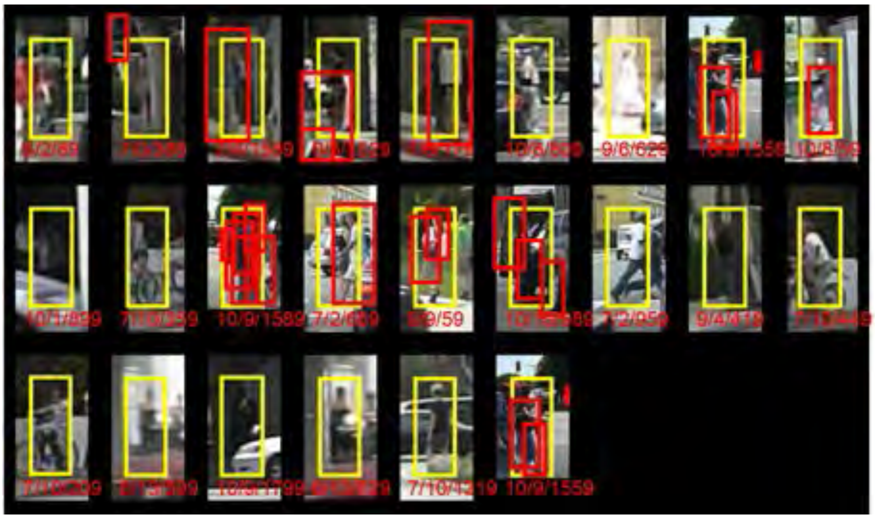
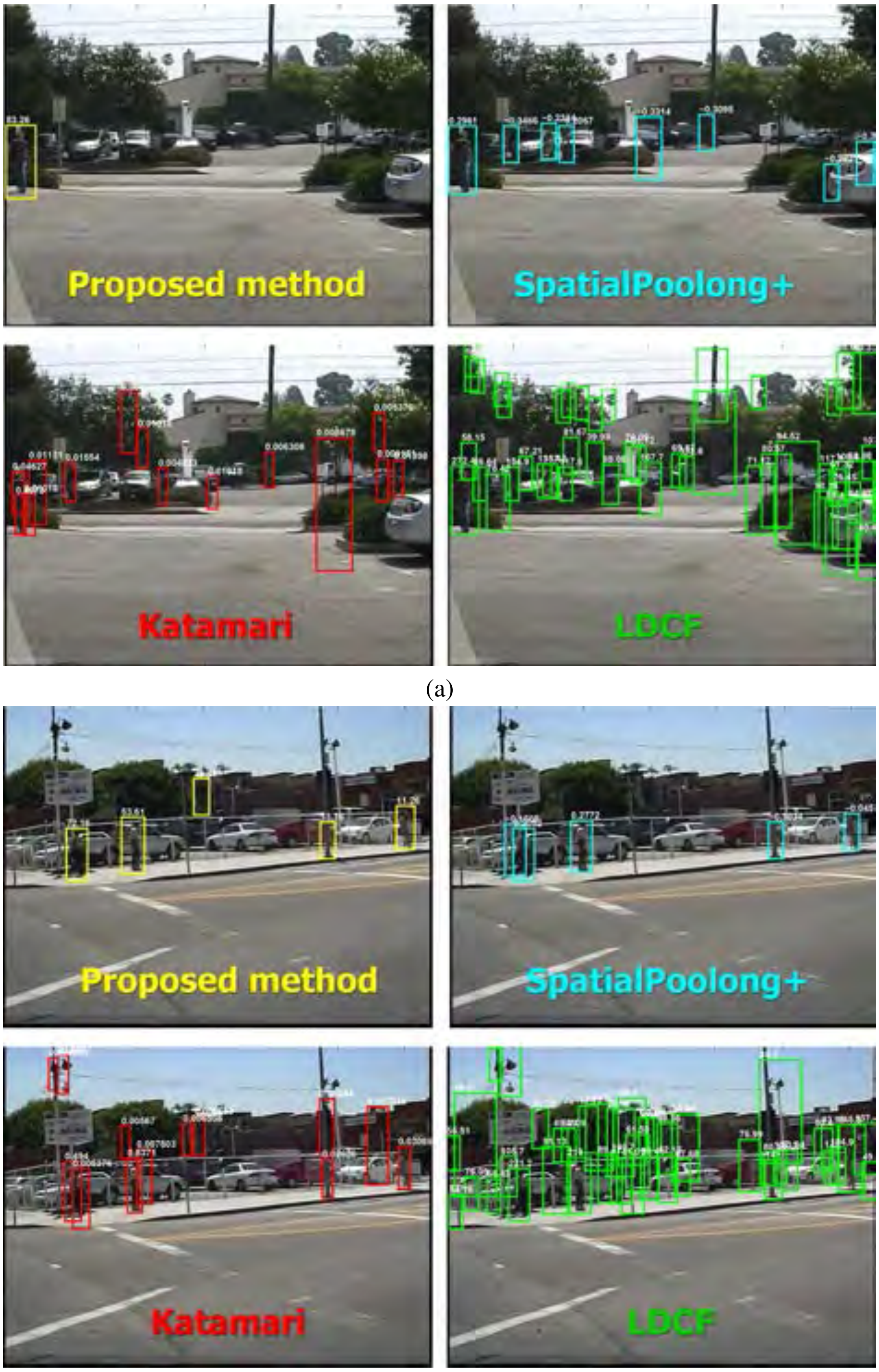


Figure 4.19: False negative detection results in partial occlusion CPDB dataset.



(a)

(b)

Figure 4.20: Comparison of detection results in CPDB dataset.

CHAPTER V

DISCUSSION, CONCLUSION, AND FUTURE WORK

5.1 Discussion

There are some human and pedestrian datasets provided for research purposed. Each dataset has specific setup of the camera and different in environment. Most of them do not contain enough partial occlusion examples needed for this work. Thus, the CPDB having some occulsion bounding boxes is the closest dataset to be adopted.

The proposed framework introduces 3 steps of enhancement for pedestrian detection. These enhancements including both training and testing enhancements. For the first enhancement which applied MSPT in feature selection process of Adaboost, this enhancement performs in training time that does not affect the speed of the system. Thus, MSPT yields good results in the experiment.

The second enhancement is the combination of ACF and spatial uLBP. From the experimental results, by adding more features to the system, the overall performance of the system is increased but the running time of the system drops considerably. This effect is found in other top detectors, wherein combinations of the large set of features are used. To obtain elegant results, they sacrifice the running time as a trade-off. The proposed method applies the combination of features as well but balances the trade-off to maintain the running time at 4 fps and low LAMR. Nevertheless, this still cannot be applied in real applications.

The third enhancement is hierarchical framework for partial occlusion handling. Experimental results show that this scheme speeds up the overall results slightly. There are some issues needed to be discussed. First, there are not enough partial occlusion samples in the CPDB dataset for testing and evaluation. Second, there are some missing ground truths that should be partial occlusion pedestrian. This issue degrades the performance of the proposed detector. In the proposed framework, each region detector is designed to use feature and classifier which require no calculation load in each region. All features are calculated once and used by every region detectors, hence faster speed. Other methods use different architecture for each region detectors which makes it very time-consuming. This problem is handled by hardware solution.

The number of false negative produced by the system is an another important issue for many applications. The ideal system should report false negative as low as possible but still yields an acceptable result. The proposed framework is designed based on such objective yet can be implemented in real applications. It maintains low LAMR, reports low number of false negatives, and acceptable running time.

5.2 Conclusion

This dissertation proposes a pedestrian detection framework with partial occlusion handling capability. The proposed framework is divided into three major enhancements, namely, enhancement in training stage using MSPT, enhancement in features using a combination of ACF and spatial uLBP as channel features, and handling partial occlusion problem with hierarchical structure of region detectors. The MSPT is created by using average pedestrian silhouette images. By using MSPT as weighting for feature selection process of training a boosted decision tree via Adaboost technique, the selected features are located in area of pedestrian shape and yield better classification results. The combination of ACF and spatial uLBP boosts the overall results of the proposed framework. To handle partial occlusion problem, the ensemble of specific region detectors in hierarchical fashion is proposed. Each region detector is constructed based on an appearance-based approach which represents the visible portion of pedestrian in detection window with different levels of visibility in both horizontal and vertical schemes. All region detectors are running together via hierarchical framework. Lower level region detectors of hierarchical framework are activated by the hidden region hypothesis value extracted from the upper level region detectors and the detection score is calculated from the hypothesis of all activated detectors. The results of the proposed framework performance on the real world CPDB dataset are ranked fourth at 25% LAMR and running at 4 frames per second with hierarchical framework and 30 frames per second with MSPT approach only. By comparing the proposed detector with other top performance detectors via LAMR and AP, the proposed detector is the reasonable option for real-time applications with low miss rate and less false negative produced.

5.3 Future Work

There are some issues and interesting problems to be discovered in the future work.

1. find the solutions to enhance the performance of region detectors.

2. increase the size of MSPT including the number of silhouette images and MSPTs.

3. find additional features to represent pedestrian with low computation time.

References

[1] Enzweiler, M. and Gavrila, D. Monocular Pedestrian Detection: Survey and Experiments. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 , 12 (2009): 2179-2195.

[2] Gandhi, T. and Trivedi, M. Pedestrian Protection Systems: Issues, Survey, and Challenges. Intelligent Transportation Systems, IEEE Transactions on 8 , 3 (2007): 413-430.

[3] Gavrila, D. The Visual Analysis of Human Movement: A Survey. Computer Vision and Image Understanding 73 (1999): 82 - 98.

[4] Geronimo, D., Lopez, A. M., Sappa, A. D., and Graf, T. Survey of pedestrian detection for advanced driver assistance systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 , 7 (2010): 1239-1258.

[5] Benenson, R., Omran, M., Hosang, J., and Schiele, B., Ten years of pedestrian detection, what have we learned?. Computer Vision for Road Scene Understanding and Autonomous Driving (CVRSUAD, ECCV workshop).

[6] Viola, P., Jones, M., and Snow, D., Detecting pedestrians using patterns of motion and appearance. Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, Oct 2003.

[7] Dalal, N., Triggs, B., and Schmid, C., Human Detection Using Oriented Histograms of Flow and Appearance. in Computer Vision ECCV 2006, vol. 3952 of Lecture Notes in Computer Science, pp. 428–441, 2006.

[8] Enzweiler, M., Kanter, P., and Gavrila, D., Monocular pedestrian recognition using motion parallax. Intelligent Vehicles Symposium, 2008 IEEE, June 2008.

[9] Cao, X.-B., Qiao, H., and Keane, J. A low-cost pedestrian-detection system with a single optical camera. Intelligent Transportation Systems, IEEE Transactions on 9 , 1 (2008): 58-67.

[10] Shashua, A., Gdalyahu, Y., and Hayun, G., Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. Intelligent Vehicles Symposium, 2004 IEEE, (2004): 1-6.

[11] Gavrila, D. M. and Munder, S. Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. Int. J. Comput. Vision 73 , 1 (2007): 41-59.

[12] Nedevschi, S., Bota, S., and Tomiuc, C. Stereo-Based Pedestrian Detection for Collision-Avoidance Applications. Intelligent Transportation Systems, IEEE Transactions on 10 , 3 (2009): 380-391.

[13] Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L., Dynamic 3d scene analysis from a moving vehicle. Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, (2007): 1-8.

[14] Enzweiler, M., Eigenstetter, A., Schiele, B., and Gavrila, D., Multi-cue pedestrian classification with partial occlusion handling. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, (2010): 990-997.

[15] Enzweiler, M. and Gavrila, D. A Multilevel Mixture-of-Experts Framework for Pedestrian Classification. Image Processing, IEEE Transactions on 20 , 10 (2011): 2967-2979.

[16] Bertozzi, M., Broggi, A., Fascioli, A., Graf, T., and Meinecke, M. Pedestrian detection for driver assistance using multiresolution infrared vision. Vehicular Technology, IEEE Transactions on 53 , 6 (2004): 1666-1678.

[17] Ge, J., Luo, Y., and Tei, G. Real-Time Pedestrian Detection and Tracking at Nighttime for Driver-Assistance Systems. Intelligent Transportation Systems, IEEE Transactions on 10 , 2 (2009): 283-298.

[18] Dalal, N. and Triggs, B., Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on 1 (June 2005): 886-893.

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

[20] Tuzel, O., Porikli, F., and Meer, P., Region Covariance: A Fast Descriptor for Detection and Classification. in Computer Vision ECCV 2006, vol. 3952 of Lecture Notes in Computer Science, pp. 589–600, 2006.

[21] Wu, B. and Nevatia, R., Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on 1 (2005): 90-97.

[22] Viola, P. and Jones, M., Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on 1 (2001): 511-518.

[23] Munder, S. and Gavrila, D. An Experimental Study on Pedestrian Classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 , 11 (2006): 1863-1868.

[24] Wohler, C. and Anlauf, J. An adaptable time-delay neural-network algorithm for image sequence analysis. Neural Networks, IEEE Transactions on 10 , 6 (1999): 1531-1536.

[25] Viola, P. and Jones, M. J. Robust real-time face detection. International journal of computer vision 57 , 2 (2004): 137–154.

[26] Wojek, C., Walk, S., and Schiele, B., Multi-cue onboard pedestrian detection. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, (2009): 794-801.

[27] Ludwig, O., Delgado, D., Goncalves, V., and Nunes, U., Trainable classifier-fusion schemes: An application to pedestrian detection. Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on, (2009): 1-6.

[28] Wang, X., Han, T., and Yan, S., An HOG-LBP human detector with partial occlusion handling. Computer Vision, 2009 IEEE 12th International Conference on, (2009): 32-39.

[29] Dollár, P., Tu, Z., Perona, P., and Belongie, S., Integral Channel Features. Proc. BMVC, (2009): 91.1-91.11.

[30] Benenson, R., Mathias, M., Tuytelaars, T., and Van Gool, L., Seeking the Strongest Rigid Detector. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, (2013): 3666-3673.

[31] Dollár, P., Appel, R., Belongie, S., and Perona, P. Fast feature pyramids for object detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 , 8 (2014): 1532-1545.

[32] Paisitkriangkrai, S., Shen, C., and Zhang, J., An experimental study on pedestrian classification using local features. Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, May 2008.

[33] Xia, X., Yang, W., Li, H., and Zhang, S., Part-based object detection using cascades of boosted classifiers. in Computer Vision–ACCV 2009, pp. 556–565, 2010.

[34] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S., Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on 2 (2006): 1491-1498.

[35] Dollár, P., Babenko, B., Belongie, S., Perona, P., and Tu, Z., Multiple Component Learning for Object Detection. in Computer Vision ECCV 2008, vol. 5303 of Lecture Notes in Computer Science, pp. 211–224, 2008.

[36] Rao, S., Pramod, N. C., and Paturu, C. K., People Detection in Image and Video Data. Proceedings of the 1st ACM Workshop on Vision Networks for Behavior Analysis, (2008): 85-92.

[37] Maji, S., Berg, A. C., and Malik, J., Classification using intersection kernel support vector machines is efficient. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, (2008): 1–8.

[38] Felzenszwalb, P., McAllester, D., and Ramanan, D., A discriminatively trained, multiscale, deformable part model. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, (2008): 1-8.

[39] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 , 9 (2010): 1627–1645.

[40] Sermanet, P., Kavukcuoglu, K., Chintala, S., and Lecun, Y., Pedestrian Detection with Unsupervised Multi-stage Feature Learning. Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, (2013): 3626–3633.

[41] Luo, P., Tian, Y., Wang, X., and Tang, X., Switchable Deep Network for Pedestrian Detection. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, (2014): 899-906.

[42] Breiman, L. Bagging predictors. Machine learning 24 , 2 (1996): 123–140.

[43] Breiman, L. Random Forests. Machine Learning 45 , 1 (2001): 5-32.

[44] Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. Rotation forest: A new classifier ensemble method. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 , 10 (2006): 1619–1630.

[45] Demiriz, A., Bennett, K. P., and Shawe-Taylor, J. Linear programming boosting via column generation. Machine Learning 46 , 1-3 (2002): 225–254.

[46] Friedman, J. H. Greedy function approximation: a gradient boosting machine. Annals of statistics (2001): 1189–1232.

[47] Freund, Y. and Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55 , 1 (1997): 119 - 139.

[48] Appel, R., Fuchs, T., Dollár, P., and Perona, P., Quickly boosting decision trees-pruning underachieving features early. JMLR Workshop and Conference Proceedings 28 (2013): 594-602.

[49] Paisitkriangkrai, S., Shen, C., and Van Den Hengel, A., Efficient pedestrian detection by directly optimizing the partial area under the roc curve. Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, (2013): 1057–1064.

[50] Paisitkriangkrai, S., Shen, C., and Hengel, A. v. d. Pedestrian detection with spatially pooled features and structured ensemble learning. arXiv preprint arXiv:1409.5209.

[51] Mohan, A., Papageorgiou, C., and Poggio, T. Example-Based Object Detection in Images by Components. IEEE Trans. Pattern Anal. Mach. Intell. 23 , 4 (2001): 349-361.

[52] Ouyang, W. and Wang, X., A discriminative deep model for pedestrian detection with occlusion handling. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, (2012): 3258-3265.

[53] Bar-Hillel, A., Levi, D., Krupka, E., and Goldberg, C., Part-Based Feature Synthesis for Human Detection. in Computer Vision ECCV 2010, vol. 6314 of Lecture Notes in Computer Science, pp. 127–142, 2010.

[54] Wojek, C., Walk, S., Roth, S., and Schiele, B., Monocular 3D scene understanding with explicit occlusion reasoning. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, (2011): 1993-2000.

[55] Rapus, M., Munder, S., Baratoff, G., and Denzler, J., Pedestrian Detection by Probabilistic Component Assembly. in Pattern Recognition, vol. 5748 of Lecture Notes in Computer Science, pp. 91–100, 2009.

[56] Park, D., Ramanan, D., and Fowlkes, C., Multiresolution Models for Object Detection. in Computer Vision ECCV 2010, Lecture Notes in Computer Science, pp. 241–254, 2010.

[57] Yan, J., Zhang, X., Lei, Z., Liao, S., and Li, S., Robust Multi-resolution Pedestrian Detection in Traffic Scenes. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, (2013): 3033-3040.

[58] Ouyang, W. and Wang, X., Single-Pedestrian Detection Aided by Multi-pedestrian Detection. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, (2013): 3198-3205.

[59] Ouyang, W. and Wang, X., Joint Deep Learning for Pedestrian Detection. Computer Vision (ICCV), 2013 IEEE International Conference on, (2013): 2056-2063.

[60] Mathias, M., Benenson, R., Timofte, R., and Gool, L. V., Handling occlusions with franken-classifiers. Computer Vision (ICCV), 2013 IEEE International Conference on, (2013): 1505-1512.

[61] Nam, W., Dollár, P., and Han, J. H., Local decorrelation for improved pedestrian detection. Advances in Neural Information Processing Systems, (2014): 424–432.

[62] Park, D., Zitnick, C. L., Ramanan, D., and Dollár, P., Exploring weak stabilization for motion feature extraction. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, (2013): 2882–2889.

[63] Zhang, S., Bauckhage, C., and Cremers, A., Informed haar-like features improve pedestrian detection. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, (2014): 947–954.

[64] Paisitkriangkrai, S., Shen, C., and Hengel, A., Strengthening the effectiveness of pedestrian detection with spatially pooled features. in Computer Vision–ECCV 2014, pp. 546–561, Springer, 2014.

[65] Costea, A. D. and Nedevschi, S., Word channel based multiscale pedestrian detection without image resizing and using only one classifier. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, (2014): 2393–2400.

[66] Dollár, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian Detection: An Evaluation of the State of the Art. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 , 4 (2012): 743-761.

[67] Ess, A., Leibe, B., Schindler, K., , and Gool, L., A Mobile Vision System for Robust Multi-Person Tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).

Biography

Name: Wittawin Susutti

Education:

- Ph.D. Program in Computer Science and Information technology, Chulalongkorn University, Thailand (June 2007 - July 2015).
- B.Sc. Program in Mathematics, King Mongkuts University of Technology Thonburi, Thailand (June 2000 - May 2004).

Publication:

- Susutti, W., Lursinsap, C., and Sophatsathit, P., ”Extracting salient visual attention regions by color contrast and wavelet transformation.” In: *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on.* IEEE, p. 1006-1011, 2009.

Scholarship:

- The program Strategic Scholarships for Frontier Research Network for the Ph.D. Program Thai Doctoral degree, Office of the Higher Education Commission, Ministry of Education, Thailand.