Enhanced Contour Description for People Detection in Images

by

Xiaoyun Du

Thesis submitted to the University of Ottawa
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Master of Applied Science degree in
Electrical and Computer Engineering

uOttawa

L'Université canadienne
Canada's university

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

People detection has been an attractive technology in computer vision. There are many useful applications in our daily life, for instance, intelligent surveillance and driver assistance system. People detection is a challenging matter as people adopt a wide range of poses, wear diverse clothes, and are visible in different kind of backgrounds with significant changes in illumination. In this thesis, some advanced techniques and powerful tools are presented in order to design a robust people detection system. First a baseline model is implemented by combining the Histogram of Oriented Gradients descriptor and linear Support Vector Machines. This baseline model obtains a good performance on the well-known INRIA dataset. Second an advanced model is proposed which has a two-layer cascade framework that achieves both accurate detection and lower computational complexity. For the first layer, the baseline model is used as a filter to generate several candidates. In this procedure, most positive samples survived and the majority of negative samples are rejected according to a preset threshold. The second layer uses a more discriminative model. We combine the Variational Local Binary Patterns descriptor, and the Histogram of Oriented Gradients descriptor as a new discriminative feature. Furthermore multi-scale feature descriptors are used to improve the discriminative power of the Variational Local Binary Patterns feature. Then we perform Feature Selection using the Feature Generating Machine in order to generate a concise descriptor based on this concatenated feature. Moreover Histogram Intersection Kernel Support Vector Machines is employed as an efficient tool of classification. The bootstrapping algorithm is used in the training procedure to exploit the information of the dataset. Finally our approach has a good performance on the INRIA dataset, with results superior to the baseline model.

Acknowledgements

I would like to take advantage this opportunity to express my great appreciation to Professor Robert Laganière, my supervisor, for giving me instructions on my thesis and projects. His patience and expertise have been very helpful during my graduate study in the University of Ottawa. I also extend my gratitude to Si Wu, the Post Doctoral fellow in the VIVA lab, for kindly sharing his experience in computer vision, and giving me suggestions and tips. I would like to thank Professor Shervin Shirmohammadi and Professor Ali Arya for being my thesis examiners. I would also like to give my thank to the chairman Professor Dimitrios Makrakisfor for taking the charge of my thesis defense. I would also like to thank the members of the VIVA Lab and the staff of the School of Electrical Engineering and Computer Science. They gave me helps during my studies and research in the University of Ottawa. Last, and most importantly, I appreciate my family for giving me encouragement and support for pursuing study in Canada.

Table of Contents

List of Figures

Chapter 1

Introduction

Computer vision now finds many applications in our daily life and it will even have a greater impact in many aspects of the society of the future. Typical tasks in computer vision are object detection, recognition, tracking, segmentation etc. In recent years, people detection has been a very active area of research in computer vision. This topic draws attention because it can be used in many applications, such as intelligent surveillance, driver assistance system, and events detection in video. Figure 1.1 shows several examples of applications of people detection systems: traffic surveillance at crossroads, vehicle assistance system, indoor security surveillance and outdoor crowd surveillance. People detection has been widely used in traffic intelligent surveillance in many countries, for example to identify dangerous intersections or people taking risky actions. In addition, people detection has been playing an important role in security surveillance, for instance, to detect people who appear in a restricted zone, like military sites. Furthermore, people detection is considered as a fundamental component to several high-level topics of computer vision, especially people tracking, people recognition, and people re-identification. People detection offers initial input to these applications and can help correct the procedures and make these tasks more effective.

(a)

(b)

(c)

(d)

Figure 1.1: Examples of applications of people detection systems, (a) traffic surveillance; (b) vehicle assistance system; (c) airport indoor security surveillance; (d) outdoor crowd surveillance.

1.1 Challenges in People Detection

Computers can process a large amounts of information. However our daily environment is still very difficult to understand for machines; obtaining high-level information from a scene turns out to be something very challenging for computers. People detection is one of these challenging tasks because of the wide variety of people clothing, sizes, colors and styles which result in very different appearances for people. Moreover human body is a non-rigid and complex object; that is observed in condition that complicated backgrounds formed by vehicles, trees, wire poles; in addition illumination conditions also vary

greatly. The objective of this thesis is to propose an approach for people detection that can accurately detect people in images using machine learning techniques. The solution we propose achieves performance results similar to more complex approaches but at a lower computational cost.

There are several approaches for people detection. These methods still need to be improved. One of my objective was to improve the performance while not increasing the complexity of the system when compared to the baseline linear SVM HOG model. Complexity is indeed important in people detection because these methods will be implemented in embedded architectures in the near feature.

1.2 General terms

In this section, we define some of the terms that will be used throughout this thesis.

In computer vision research, a *Feature* corresponds to a spatial image structure that is relevant for our computational task. *Features* usually are the specific structures in the images such as points, edges or objects. A *Descriptor* is a distinctive representation of the pattern of a feature. It is usually a vector of floating point or binary values.

In machine learning, a *Classifier* is a tool used to identify to the category to which belong an observation, on the basis of a training set containing data whose category membership is known. In the terminology of machine learning, supervised learning refers to classification where a training set fo correctly identified observations ia available. *SVM* refers to *Support Vector Machine*, which is a particular type of *Classifier*. An *SVM* model can divide the examples of the separate categories by a well defined boundary having a maximal margin. In addition to performing linear classification, *SVM* model can perform non-linear classification by employing a kernel function, which maps the input samples into a high-dimensional feature spaces. The principles of linear *SVM* will be explained in the Chapter 3 and the one of kernel *SVM* will be introduced in the Chapter 4.

During the training procedure, *Positive Samples* refer to annotated people image samples, which are obtained from the INRIA dataset[7] in our case. *Negative Samples* refer to those non-people images, which are also obtained from the INRIA dataset.

Hard Samples are defined as those examples (negatives or positives) which can not be classified correctly. Those hard examples are very useful to improve the performance using the bootstrapping training algorithm.

Bootstrapping is a training algorithm that re-trains the model by updating the dataset. It can improve the performance after several rounds of retraining. It will be explained in detail in Chapter 4.

Feature selection is a process that selects a subset of relevant features for model construction. This technique has several benefits: improving the prediction performance, reducing the measurement requirements, and reducing the training time.

FPPI refers to False Positive Per Image. *FPPW* refers to False Positive Per Window. The values of them are used in the evaluation of classification performances.

1.3 Proposed Work

In this thesis, a classic baseline model is first implemented based on the Histogram of Oriented Gradients (HOG) descriptor and the linear SVM algorithm. Figure 1.2 shows the procedure of this baseline model. We calculate the HOG descriptor by scanning the image with overlapping detection windows. Concatenated gradient orientation histograms are extracted from each window. A linear SVM executes an efficient classification to determine if a given window contains people or background. As our thesis focuses on images, we evaluate our model on the INRIA dataset, a well-know database of people images in various settings. This baseline model achieves a robust performance. However the result of the evaluation shows four major types of false detection as follows:

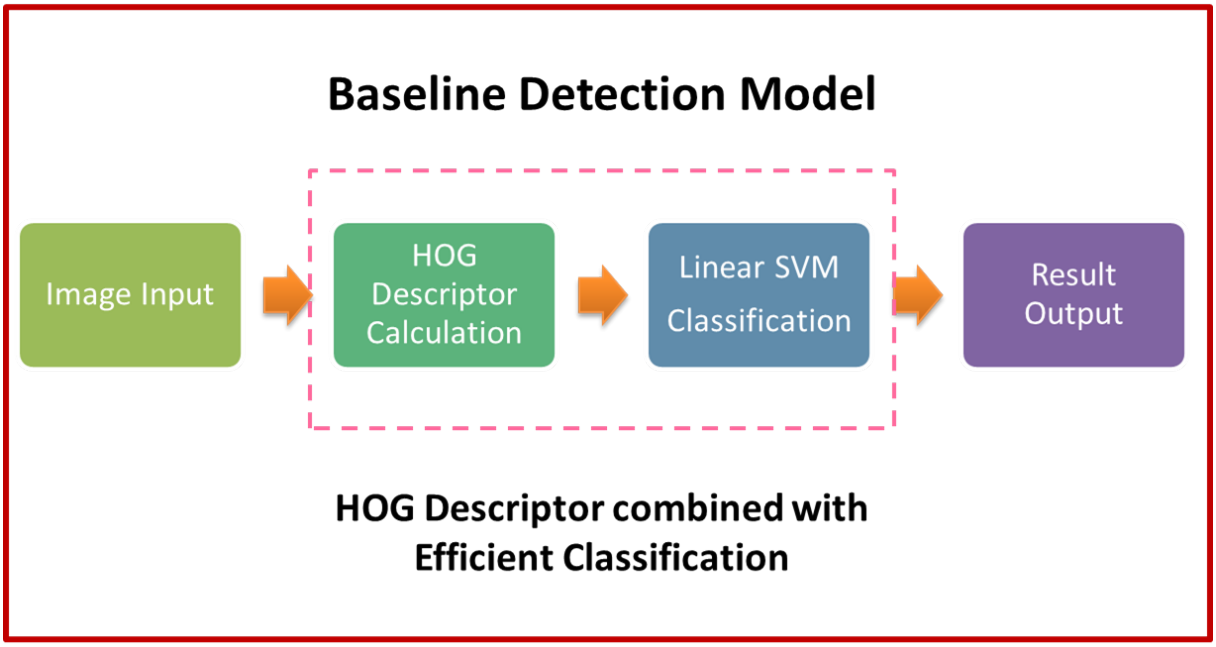


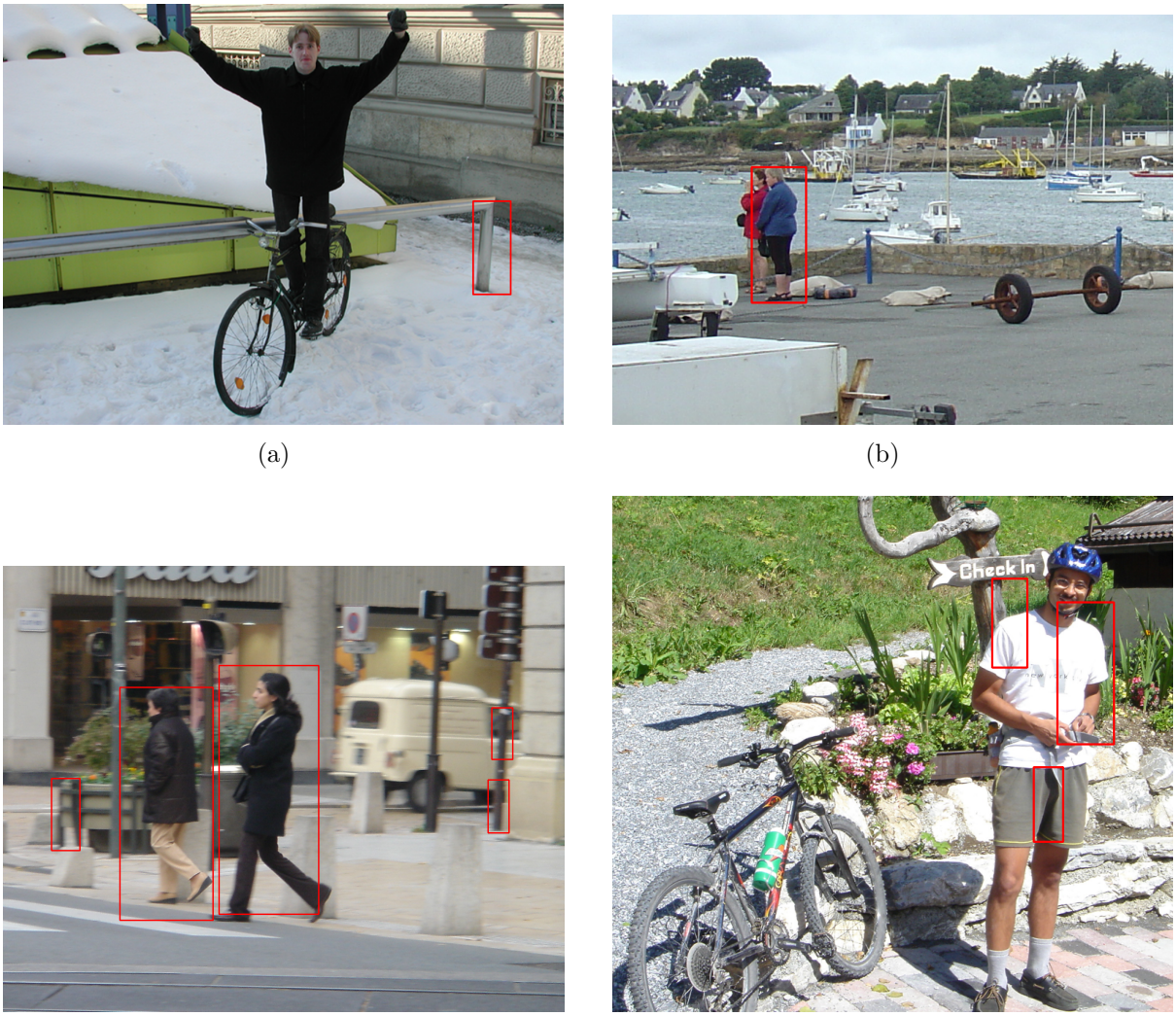
Figure 1.2: Illustration of the Baseline Model.

- Various poses of people: this type of false detection refers to those peoples whose pose is various and is different from that of in the training dataset. An example is shown in Figure 1.3 (a).

- Occlusion of people: this type of false detection refers to those peoples behind other objects. An example is shown in Figure 1.3 (b).

- Pilar-like object of background: this type of false detection refers to pillar-like objects which have proportions similar to the ones of peoples. An example is shown in Figure 1.3 (c).

- Part of body: this type of false detection refers to that part of the people's body are detected as a people such as legs, arms, and shoulder. An example is shown in Figure 1.3 (d).



(a) (b) (c) (d)

Figure 1.3: Examples of four major types of false detection, (a) various poses; (b) occlusion; (c) pilar-like object; (d) body parts.

In order to improve the detection results of the baseline model, our thesis proposes a two-layer cascade for people detection. The baseline model mentioned above is used as the first layer. It is defined as a fast filter, which can quickly identify candidate windows. A preset threshold is used on this filter such that most positive samples are accepted while rejecting the majority of negative samples. The second layer is designed as an enhanced model which re-scans these candidates generated by the baseline model and classifies them accurately. In order to better exploit contour information, a key feature in people detection, we combine the HOG and VLBP descriptor, and then propose a concise feature to improve

the detection performance. VLBP here is based on the gradient image, which focuses on contour information while discarding contextual details, conveyed by the original image. The enhanced descriptor better focuses on the more relevant contour information and filters out irrelevant background and foreground noise and textures. Figure 1.4 shows an overview of the computational process for the feature proposed in this thesis.

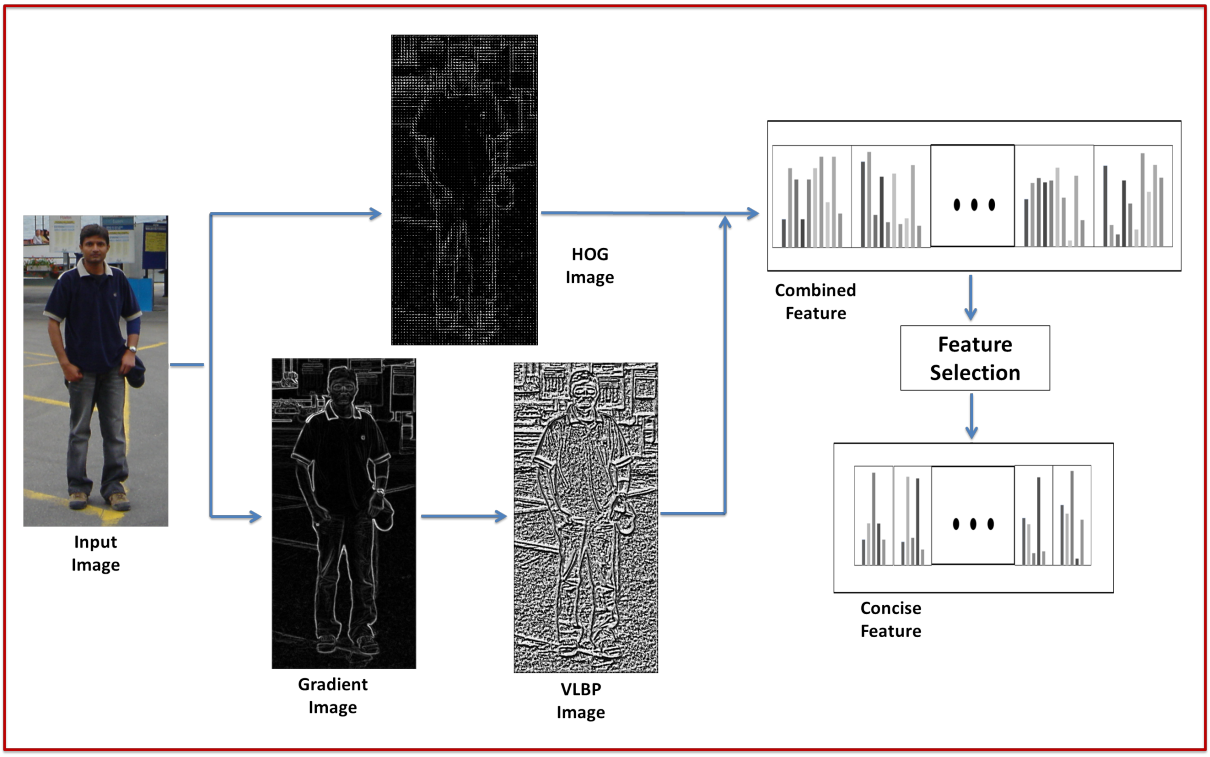


Figure 1.4: The illustration of our enhanced descriptor.

Feature selection techniques are generally employed to process high-dimension data. The Feature Generating Machine (FGM) [54] is used in this thesis as a dimension reduction algorithm to generate a concise descriptor which keeps the most important parts of the combined feature without impacting on the classification performance. The Histogram of Intersection Kernel (HIK) SVM [34] is used as an advanced and efficient algorithm for classification. Moreover, the training procedures employ a bootstrapping algorithm [16] [57] [7] [48] [52], which can improve the performance from a limited number of training samples.

We show in our experiments that our two-layer model outperforms the HOG baseline model on the INRIA dataset, and makes improvement for the four major type of false detection. Figure 1.5 illustrates the procedure of our advanced model.

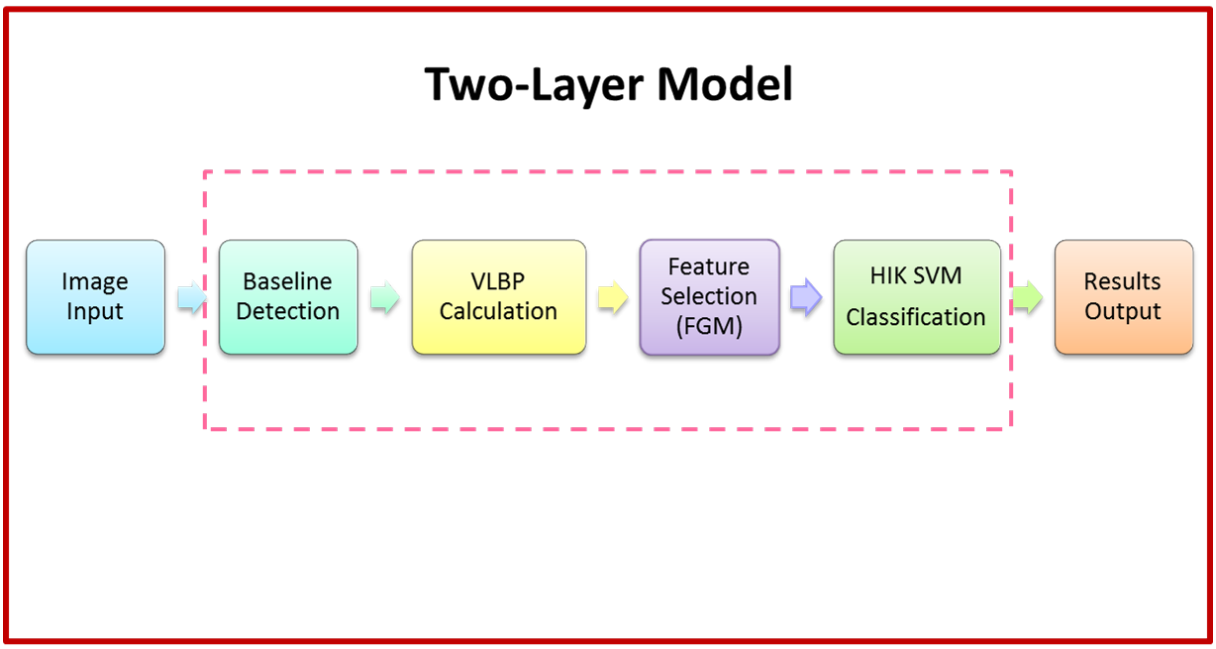


Figure 1.5: The illustration of Our two-layer Model.

1.4 Contribution

In this thesis, we combine HOG with VLBP in a concatenated. These features are integrated into a two-layer cascade of classifiers resulting in a classifier that produces classification results close to the state-of-the-art but at a lower computational complexity. Therefore our proposed approach is appropriate to be used in embedded system that are deployed, for example, in advanced driver assistance systems. The main contributions of our work are listed as follows.

- An enhanced descriptor is proposed which combines HOG and VLBP in a concatenated feature, in which a multi-scale approach is used. From the result of our evalu-

ation, these two descriptors are proved to be complementary to each other;

- A two-layer cascade mode for people detection is implemented. The first layer is a trained baseline model, which is used as a fast filter. The second layer is our proposed enhanced-description model which combine the HOG descriptor, the VLBP descriptor, HIK SVM, FGM feature selection and bootstrapping algorithm;

1.5 Evaluation Criteria

In this thesis, we evaluate our method when comparing to other methods by two criteria: the *Miss Rate* and the *False Positive Per Image (FPPI)*. Given a FPPI, a method with a lower Miss Rate achieves better performance.

1.6 Organization

The thesis is organized as follows. In Chapter 2, several related works are introduced. In Chapter 3, we describe the Baseline Detection Model. We evaluate this model on the INRIA dataset and analyze the performance. In Chapter 4, we propose an Advanced Detection Model, in which an enhanced contour descriptor is generated and optimized. Furthermore several powerful techniques are used for efficient processing, for example, FGM and HIK SVM. This well-designed model outperforms the baseline model according to the evaluation on the INRIA dataset, especially on the four major types of false detection. Experimental results are shown to illustrate the performance of this advanced detection model. Finally in Chapter 5, we make a conclusion for this thesis.

Chapter 2

Related Work

People detection has been the subject of several works in the recent literature. In an effort to improve the detection performance, these works have been focusing on two main aspects: feature description and classification algorithms. Improvement of either or both of them can result in enhanced detection rate.

2.1 Foundation Method

Most detectors use sliding window approaches to perform detection. The first sliding window detector was proposed by C. Papageorgiou and T. Poggio in [47]. In this work, multi-scale Haar wavelets are used in combination with SVM. Based on this theory, Viola and Jones proposed a method [56] based on fast feature computation using integral images and a cascade structure for efficient detection. This method is considered as a foundation for modern detectors.

2.2 Descriptors

A discriminative feature is a key component for a robust people detection system. By using a discriminative descriptor, the characteristic information of people will be obtained, and thus people can be distinguished from background. Some excellent descriptors have been presented in [11], [59] that meets this requirement.

Haar-like feature has been proposed by [55] and also used in [36] and [5] for people detection. A Haar-like descriptor considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image, thus we can detect peoples in an image. This descriptor can be fast implemented is considered as the first feature which can be calculated real-time.

Contour information of people contain the most discriminative information when we judge if an object is people. The Histograms of Oriented Gradient (HOG) descriptors is a milestone feature which extract the contour information of object. It has been first proposed by Dalal et al. in [7], and also has been mentioned in [51] and [28]. The fundamental aspect of this state-of-the-art descriptor is that a robust and invariant representation of a target can be obtained from the distribution of the intensity gradients or edge directions. Specifically, the detection window is divided into small connected regions ("cells"), and for each cell a histogram of oriented gradient or edge orientation of the pixels is calculated. It has been used in several existing approaches [23], [64], [4]. HOG shares some similarities with the Scale Invariant Feature Transformation (SIFT) [32]. HOG is calculated on a dense array of overlapped sample windows while SIFT uses sparsely distributed descriptors based on the interest points. SIFT usually identifies object categories using bag of words (BOW)[26]. Zhu et al. proposed fast calculation of HOG descriptor in [69] by using integral histograms.

Moreover a texture information can also be used to describe the characteristic of an

object. To describe textures, the LBP descriptor was first proposed in [41], and it has been successfully applied to people detection in [38], [67], [68]. It is nowadays a classic texture descriptors and has been used in several areas, for instance, face recognition [2], face detection [3], [27] , and object segmentation [24], [33]. This descriptor is invariant to change in illumination, which makes it to perform well and robustly in classification and segmentation tasks. Other types of LBP descriptors have been proposed to improve the performance and to meet specific requirements, including Elongated Local Binary Patterns (ELBP) [31], overlapped LBP (oLBP) [1], Rotation Invariant LBP [22]. Of particular interest is the variational LBP descriptor that have been proposed by Wu, et al. in [63]. This variational LBP (VLBP) descriptor, which is used to extract the texture feature on the gradient image, is particularly suited to describe the information of people's contour.

Furthermore shape features are also a cue for people detection. In [19] and [18], Gavrila et al. quickly match image edges to a set of shape templates by employing Hausdorff distance transform and a template hierarchy. Besides, a large pool of short line and curve segments proposed by Wu et al. [60], and named Edgelet features, is used to describe shape locally. There are other descriptors which also exploit shape information for people detection, for instance, Shapelet feature and Shape context. Shapelet feature has been proposed by Sabzmeydani et al. in [49]. This feature pays attention local regions of the image, and is built from low-level gradient information that distinguish people from non-people backgrounds. Shape context was fist proposed by Mori et al. in [37] and also has been mentioned in [30]. This method shows that the shape context, which makes use of local tangent information at point locations, contains more detailed information about the shape, and when the local tangent can be reliably estimated they outperform the original shape context.

Color information is also an useful cue for improving people detection accuracy. The Color Self Similarity (CSS) descriptor was introduced by Walk et al. in [57]. The author proposed a new feature descriptor that uses color information with the idea that color

patterns of people's clothes can potentially offer additional information that can be exploited by a classifier. This color self similarity feature is generally computed in HSV color space. The pairwise similarities of the features' histograms are computed and histogram intersection is used as the distance function. However by using co-occurrence histograms, a second order image statistics, the dimension of the feature descriptor becomes high. Note also that this descriptor is usually employed as a complement to other features such as the HOG. Color information and implicit segmentation are introduced in [44] by Ott et al. and show a improvement over HOG model.

Dollár et al. proposed a simple and uniform framework for integrating multiple feature types in [9] in which Haar-like feature are computed over multi-channels, including LUV, gray-scale, gradient magnitude and gradient magnitude quantized by orientation. Furthermore Dollár extended this framework to fast multi-scale detection in [8].

Multi-clue descriptors can represent more information. Although HOG descriptor is considered to be the best single feature, we can use one or more additional features to provide complementary information. Wang et al. [58] proposed a descriptor that combined LBP with HOG, and a modified Linear SVM algorithm for performing basic occlusion reasoning [65]. Wojek et al. in [59] showed that the performance of a combination of Haar-like features, shapelets, shape context and HOG features is better than that of any single of them.

2.3 Classification Algorithms

As an another important system component, an effective algorithm of classification is also essential to a powerful detection system. Support Vector machines (SVMs) is a leading technique used in vision tasks such as people detection [7], [39] , face detection [43], and car detection [47]. SVM is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, and is mostly used for classification and regression

analysis.

2.4 Techniques for People Detection

A well-designed model which describes the characteristic of people's shape is proposed in recent years. The deformable part models (DPM) [14] [16] [17] [20] [21] is probably the most successful approach for object detection based on HOG. It achieves state-of-the-art people detection [13] through the use of a latent SVM formulation. This DPM approach is trained using a discriminative procedure that only requires the bounding boxes of the objects, leading to an efficient and accurate detector. By using this elegant framework, DPM can capture variations in appearance. An important benefit of this approach resides in the fact that a part-based model is trained without an explicit labeling of the part locations. Part-based models are generally better at handling pose variations in the object model. However, they are computationally more complex even if efficient detection mechanism can be introduced such as the star-cascade presented in [15].

Another interesting SVM based detector is the one by [50]. In this paper, the authors focus on a Driving Assistance System application. They describe in detail an approach for single-frame classification that proceeds by breaking down the class variability through the repetitive training of a set of relatively simple classifiers associated with clusters of the training set. In addition, multi-frame decisions is also described. Together with a shift-invariant local description of image sub-regions and discriminant integration using Adaboost a powerful classifier is obtained. By pooling together many perceptual decisions the system can segment out pedestrians at a sufficiently reliable level. One key aspect of this approach lies in the integration of additional cues measured over time, situation specific features and additional object categories consisting of vehicles and stationary background structure. This system gives good results under daytime but can not achieve good performance on complex weather conditions.

An approach based on spatial pooling is proposed in [45]. In this method, spatial pooling algorithm is used to produce a new feature based on low-level descriptors. The incorporated spatial pooling is employed to improves the translational invariance. By using the ROC curve (pAUC) measure, the partial area is optimized. These techniques helps to improved the performance of pedestrian detection compared to other approaches.

A deep model automatically learn scene-specific features and visual patterns in static video surveillance [66] is proposed and can be executed without any manual annotations from the target scene. This method is proposed to solve the problem that the performance of a detector depends much on its training dataset and will decline when it is applied to a new scene. This algorithm learns based on a scene-specific classifier and the distribution of the target samples. Moreover a cluster layer is introduced to utilizes the scene-specific visual patterns. The characteristic of this approach is that by fitting the marginal distributions of target samples, it automatically weights the importance of the training samples. Finally this model shows significant important improvements on the standard datasets.

In this thesis, our objective was to design an enhanced people detector mainly by improving the contour description. We found that when HOG is combined with VLBP, the enhanced descriptor is better than either HOG, LBP or VLBP. Moreover when feature selection is employed, our enhanced descriptor is concise without impacting on the performance. Other techniques are used in our proposed model, such as bootstrapping, Kernel SVM, and our model achieves improved performance on the INRIA dataset superior to a single HOG detection system model.

Chapter 3

Baseline People Detection

People detection has been a popular topic of compute vision in the past few years. One of the challenges for this topic is that we need a robust model which can discriminate between peoples and non-peoples in real-world imagery even if the background is cluttered or the environment has difficult illuminations. A descriptor that can perform well on real world images and that has good view invariance properties is necessary. Furthermore, an efficient classifier that can accurately discriminate peoples from background is also a very important component of the detection system. This chapter focuses on implementing a robust detection model based on a discriminative feature that can distinguish people even in complex backgrounds with various illumination. Histogram of Oriented Gradients proposed by Dalal and Triggs in [7] is this kind of descriptors. We build in this chapter, a classic HOG detection model that combines the HOG descriptor with a linear SVM. This will be used as a baseline reference for the classification system we build in Chapter 4.

This model is illustrated in Figure 3.1. The objects in the left hand side are positive samples (people), and the objects in the right hand side are negative samples (non-peoples). The aim of this model is to optimally classify them by a hyperplane defined by a linear SVM. It should be noted that this HOG linear SVM model can achieve robust performance while keeping the computation manageable. As it will be reported later in this chapter,

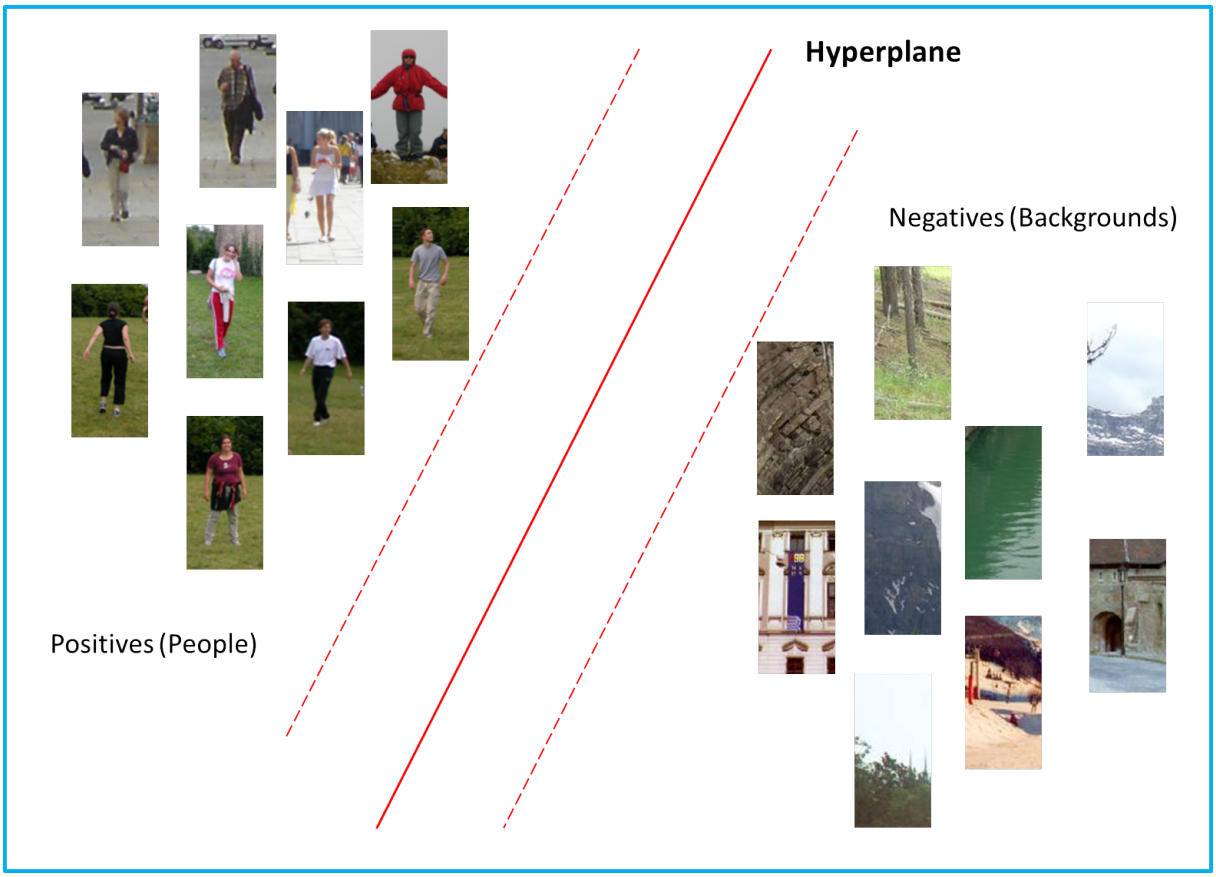


Figure 3.1: Linear separation between negative and positive samples

the results of the experiment on the INRIA dataset shows a 50% detection rate when the False Positive per Image (FPPI) is at 0.1.

This chapter is organized as follows. In Section 3.1 we introduces the HOG descriptor, its implementation and its characteristics. In Section 3.2, the linear SVM is introduced as a powerful tool of classification, which is an essential component of our HOG model. In Section 3.3, experiments on the INRIA dataset is performed, and the analysis of the results demonstrates that this HOG linear SVM model has a robust performance on the INRIA dataset. Finally, in Section 3.4, a summary of this chapter is drawn according to the work mentioned above and the result of the experiments.

3.1 HOG Descriptor

In this section, we introduce the HOG descriptor, which extract information on people's shape. The fundamental idea of this feature is that a robust and invariant representation of the object in an image can be obtained from the distribution of the intensity gradients orientation, even if we do not know the exact location of the object. Figure 3.2 shows a typical example of HOG representation of an image containing people. The HOG descriptor can also perform equally well in other shape-based classification although this chapter focuses on its characteristics and advantages for people detection.

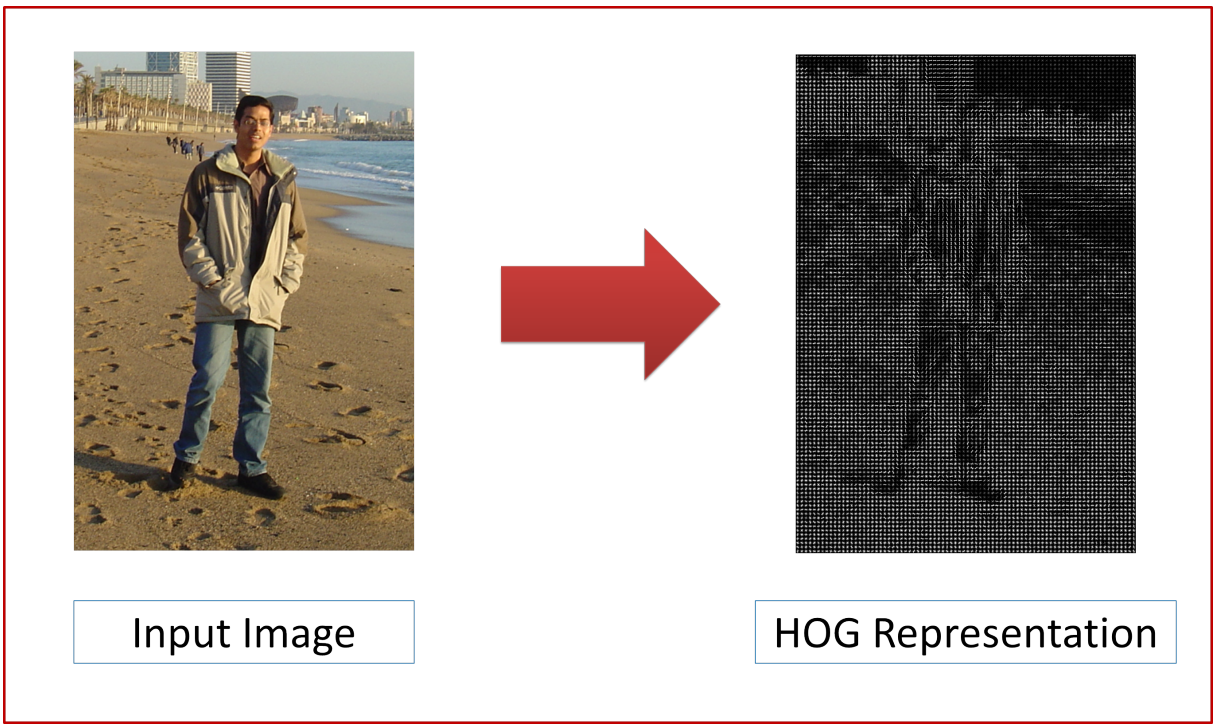


Figure 3.2: An example of the HOG representation of an image

3.1.1 Overview of HOG Computation

The HOG descriptor is computed on a dense and overlapping grid of equispaced cells. The HOG algorithm proposes that the appearance and the shape of local objects in an image can be characterized by the local intensity gradient distribution or edge direction.

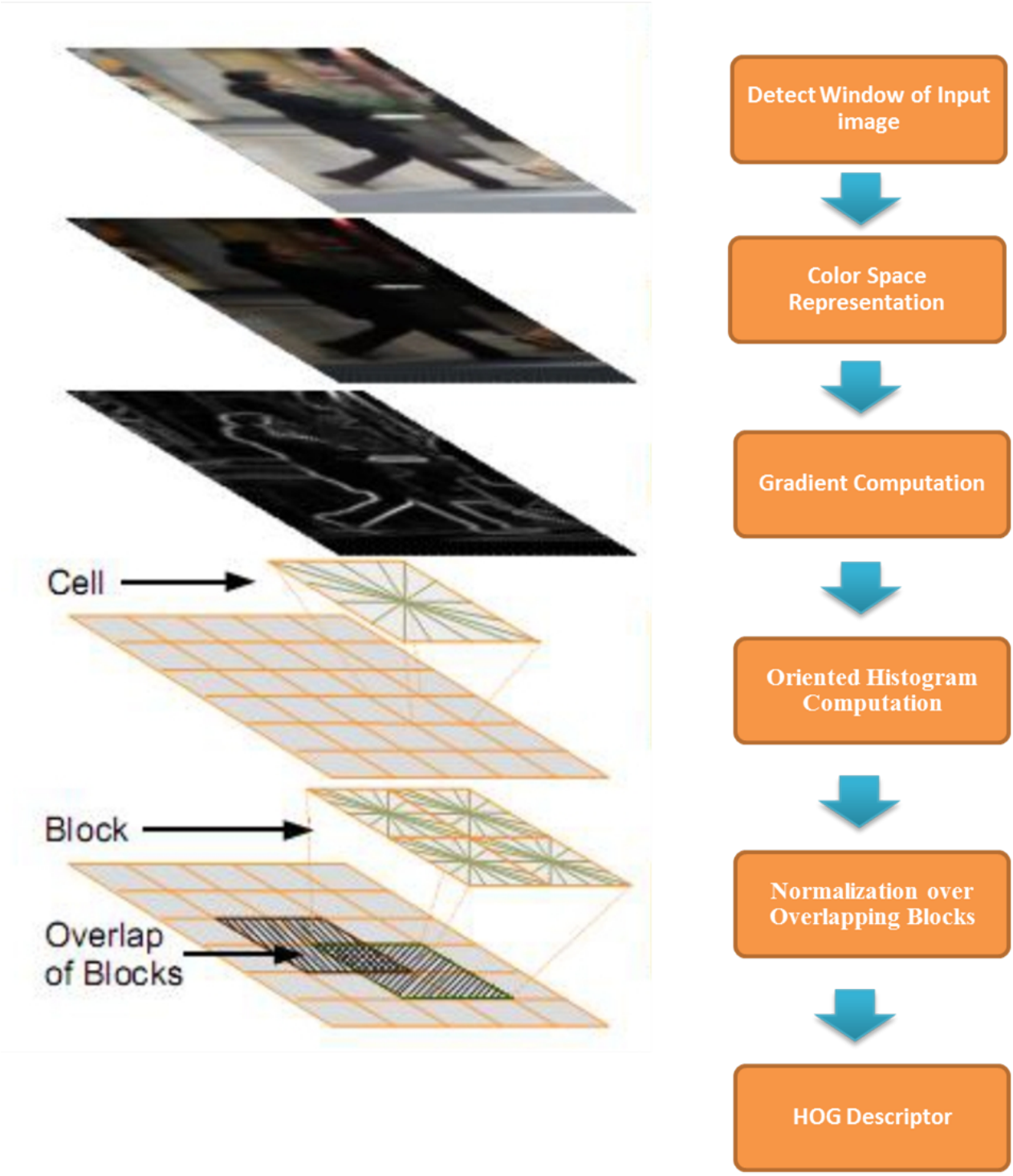


Figure 3.3: The main steps of HOG computation

The computation of this characterization is performed as follows: the image is segmented into small and connected regions (cells), the gradient is then computed inside these cells; then a larger area is defined as a block which groups cells. An histogram of the gradient orientation is computed for each of these block. A normalization is also applied to all the cells with the objective of making the HOG descriptor robust to illumination and shadow. The histogram of each block is finally concatenated together to form a vector, named the HOG descriptor. As shown in Figure 3.3, the above procedure can be summarized as the following four steps: gamma and color normalization, gradient computation, oriented histogram computation, and normalization over overlapping blocks. Each step is discussed in detail in the following subsections. The configuration of some parameters in HOG calculation is also discussed.

3.1.2 Color Space Representation

Color spaces describe the colors in an abstract mathematical model which is represented as tuples of numbers, usually three values or three-color components. In our case, the object images are encoded in the RGB model of color space. The RGB color space is defined from the three primary colors, which are Red, Green and Blue. This color model is specially designed for representing images in electronic systems. The most common representation of the RGB model uses 8 bits per channel, giving a total of 24 bits to encode a specific color, or equivalently $256^3 = 16{,}777{,}216$ colors.

In the gray-scale model, every pixel is encoded by using 8 bits while the pixels in the RGB model are encoded by 24 bits (8 bits for each of the three channels).The value of every pixel in gray-scale model ranges from 0 up to 255, giving 256 gray level.

From the wok of Dalal and Triggs in [7], some evaluations have been done on RGB, LAB and gray-scale image. The result shows that the performance of RGB space image and LAB space image are similar, and the detection performance of gray-scale image is

reduced by 1.5% at 10^{-4} False Positive per Window (FPPW), thus there is little difference between them. Thus we use RGB images in the work.

3.1.3 Gradient Computation

When calculating the gradients on a color image, each of the channels is calculated separately, and the one with the largest norm and its correspondent angle is used as the gradient vector of each pixel. Specifically, we calculate a value for the x-derivative, and another value for the y-derivative, for every pixel. These pairs are named as G_x and G_y, and they are defined by the following equations:

$$\begin{aligned} G_x(i,j) &= \frac{\partial I}{\partial x}(i,j) \\ G_y(i,j) &= \frac{\partial I}{\partial y}(i,j) \end{aligned} \tag{3.1}$$

In Equation 3.1, I is the input image, and (i,j) is the pixel coordinate.

The gradient magnitude $M(i,j)$ is computed as the square root of the quadratic sum of each gradient component; $G_x(i,j)$ and $G_y(i,j)$, as follows:

$$M(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)} \tag{3.2}$$

The gradient orientation $\Theta(i,j)$ can be obtained from the four-quadrant inverse tangent of $G_x(i,j)$ and $G_y(i,j)$ as follows:

$$\Theta(i,j) = \arctan \frac{G_y(i,j)}{G_x(i,j)} \tag{3.3}$$

To perform fast computation of gradient, there are several discrete derivative masks that can be used:

1-D Sobel masks:

$$\begin{aligned} Centered : M_c &= [-1, 0, 1] \\ Uncentered : M_{uc} &= [-1, 1] \\ Cubin - Corrected : M_{cc} &= [1, -8, 0, 8, -1] \end{aligned} \tag{3.4}$$

2×2 Sobel masks:

$$\begin{aligned} D_x &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \\ D_y &= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned} \tag{3.5}$$

3x3 Sobel masks:

$$\begin{aligned} S_x &= \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ S_y &= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \end{aligned} \tag{3.6}$$

According to the evaluation results of above 1D and 2D derivative masks reported in the work of Dala and Triggs in [7], the performances are sensitive to the way those gradients are computed. Before gradient computation, Gaussian smoothing is usually used. However, it has been experimentally demonstrated that no smoothing with the simplest 1-D derivative mask: centered [-1, 0, 1] in Equation 3.4 gives the best results; it achieves the lowest miss rate at 10^{-4} FPPW as shown in Table 3.1 from the work in Navneet Dalal's PhD thesis [6]. Thus in this work, we used: the gradient computed by [-1,0,-1] for the horizontal direction

and [-1,0,1]^T for the vertical direction.

Mask Type	1-D centred	1-D uncentred	1-D cubic-corrected	2×2 diagonal	3×3 Sobel
Operator	[-1,0,1]	[-1,1]	[1,-8,0,8,-1]	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Miss rate at 10^{-4} FPPW	11%	12.5%	10%	12.5%	14%

Table 3.1: Different gradient masks and their effects on detection performance. All results are without Gaussian smoothing ($\sigma = 0$). [6]

3.1.4 Oriented Histogram Computation

In order to compute an oriented gradient histogram, the input image is subdivided into several overlapping blocks, and those blocks are subdivided into smaller cells.

For every pixel in a cell, a weighted vote is issued for the bin corresponding to the angle of its gradient. These votes are accumulated to generate the final histogram of a cell. The orientation bins are evenly spaced over $0° - 180°$ ("unsigned" gradient) or $0° - 360°$ ("signed" gradient), depending whether the angles of histograms are signed or unsigned.

As Dala and Triggs have discussed in their work [7], the best results are obtained by using unsigned gradient and they set $20°$ between each bin resulting in a 9-bin histogram for each cell. The major reason why unsigned gradient is chosen is that people usually wear a wide range of clothing over a variety of background color, as a result, the signed gradient of the image appearance becomes then uninformative. It has been observed that the performance decreases when using signed gradients, even if the number of bins is doubled according to the evaluation performed in [7].

The vote weighting is done according to the gradient magnitude at the pixel. We can use the squared gradient magnitude, or the square root magnitude. In practice using the

magnitude itself is the best option. Equation 3.7, which computes the k th bin of the histogram, is as follows.

$$h_k = \sum_{i,j} M(i,j) 1[\phi(i,j) = k] \tag{3.7}$$

(Where 1 is the characteristic function that indicates if a particular orientation belongs to a given bin or not). In our case, k ranges from 1 to 9.

3.1.5 Normalization over Overlapping Blocks

As mentioned in the beginning of this chapter, people detection is a challenging task in computer vision because of the variety of background and illumination, and the occlusion of people. If we want to make our descriptor robust, illumination normalization is necessary.

Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local normalization turns out to be essential for good performance. A cell is composed of several pixels, and a block is composed from several cells. In a block, illumination is considered to be invariant. Several normalization schemes were discussed in [7]. If v is the vector of the histogram of a block, $\|v\|_k$ is the k-norm of the v for $k = 1, 2$, and let ϵ be a small constant, we have the following options:

L2-norm:

$$v \to \frac{v}{\sqrt{||v||_2^2 + \epsilon^2}} \tag{3.8}$$

L2-Hys: L2-norm first , then limiting the maximum values of v to 0.2 and re-normalization.

L1-sqrt:

$$v \to \sqrt{\frac{v}{||v||_1 + \epsilon}} \tag{3.9}$$

L1-norm:

$$v \to \frac{v}{||v||_1 + \epsilon} \tag{3.10}$$

The experiments in [7] shows that either L2-norm , L2-Hys, or L1-sqrt performs equally well. L1-norm decreases the performance by 5 %. Not normalizing the block vector reduces significantly the performance by around 27 % at 10^{-4} FPPW. The ϵ value in the above mentioned calculation may distort the discrimination effect of the descriptor but the results are unchanged over a wide range of ϵ values. As a result, we use the L2-norm in our experiments.

3.1.6 Configuration

From the work of Dalal and Triggs, it has been spotted out that shorter window size with less margin between the pedestrian and the image borders turned out in a loss of performance. Therefore every window present a margin of about 16 pixels from the pedestrian to all four sides. Reducing this margin to only 8 pixels (48×112 window) decreases performance by 6%. Increasing the pedestrian size within a 64×128 window also causes a similar loss of performance as the margin results decreased even though the person resolution is increased. The information in these margins provides useful context content that helps the detection.

The configurations and parameters of HOG used in our work are given as follows: To compute the HOG descriptor, an image is divided into non-overlapped cells of size of 8×8. For each cell, a histogram of gradient orientations with 9 orientation bins in the range 0 ° to 180 ° is computed and weighted by gradient magnitudes. Each block is composed of 2×2 neighboring cells, and the corresponding 4 histograms are concatenated into a 36-dimensional feature vector followed by a L2-normalization. For a 64×128 detection window, there are total 105 overlapping blocks when the block stride is set to 8×8 pixels, and the corresponding HOG descriptor is built by concatenating all the features of the 105 blocks. As a result, the HOG descriptor of this detection window has a dimensions of 3780. The complete descriptor size calculation equation is then as follows:

$$M = N_{bin} \times N_{cell} \times \{\frac{V_{winsize} - V_{blocksize}}{V_{blockstride}} + 1\} \times \{\frac{H_{winsize} - H_{blocksize}}{H_{blockstride}} + 1\} \quad (3.11)$$

(where N_{bin} is the number of bin in each cell, N_{cell} is the number of cell in each block, $V_{winsize}$ is the vertical of window size, $V_{blocksize}$ is the vertical of block size, $V_{blockstride}$ is the vertical of block stride), $H_{winsize}$ is the horizontal of the window size, $H_{blocksize}$ is the horizontal of the block size, $H_{blockstride}$ is the block stride.)

It should be noticed that the configuration of cell size and block size used in our work is different slightly from the best one identified in [7]. This best performing configuration consists of using 3×3 blocks of 6×6 pixels cells, the descriptor contains then a total of 8505 dimensions. This choice significantly increases the size of the vector of the descriptor (compared to that in our case, 3780 dimension). As we will show in chapter 4, we have been able to obtain better results with this simpler configuration.

3.2 Support Vector Machine

A discriminative classification should determine a decision boundary between pattern classes in a feature space. Support Vector machine (SVM) [39] is a kind of supervised learning model that is used to analyze data and recognize patterns, then a set of rules are built for classification in which similar dataset is assigned same label. A linear SVM forms a hyperplane which has the property that minimizing the empirical classification error while maximizing the geometric margin, in other word, this hyperplane has the largest distance to the nearest data points of each class.

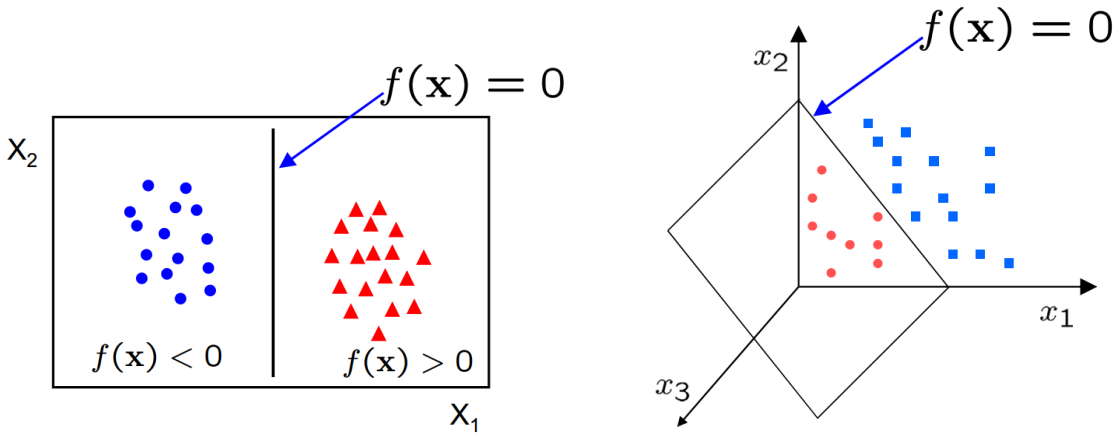
3.2.1 Linear Classification

Given a two-class dataset , a classifier is trained to predict the class labels of future data points. This is known as a "binary classification" problem that can be solved using machine learning algorithms. Support Vector Machine (SVM) is among the best performing classifiers, so we selected it as our classification algorithm.

Let $\{x_i,\ y_i\}$ for i=1,2, ... N be the training data, in which $x_i \subset R^d$ is a vector of descriptor of object "i" and $y \subset \{-1, 1\}$ be the class label.. Our aim is to find a rule that, to any given input x, will assign a class label y. Then we can learn a classifier g(x) such that

$$g(x_i) = \begin{cases} \geq 0, y_i = 1 \\ < 0, y_i = -1 \end{cases} \tag{3.12}$$

The linear classification is a subset of the binary classification when the data are linearly separable. Figure 3.4 is an example of linear classification:



(a) (b)

Figure 3.4: Examples of Linear Classification, (a) 2D Linear Classification, (b) 3D Linear Classification.

Here we assume that our data are linearly separable, which means that a hyperplane (or a line in 2D) can be drawn to split all the points (objects) which belongs to one class

from the other points which belongs to the other class. To get the optimal classification, the separating line or hyperplane should maximize the margins or distances to the closest points of each class. Equation 3.13 describes the hyperplane as follows:

$$f(x) = w^T x + b \tag{3.13}$$

In this form, w is the normal to the hyperplane and b is the bias. Usually w is known as the weight vector. Figure 3.4 (a) shows an example in 2D plane, in which the discriminant is a line. The line $f(x) = 0$ is the optimized classifier that maximizes the distances to the closest point of each class. In 3D the discriminant is a plane. Figure 3.4 (b) shows an example in 3D.

The above mentioned hyperplane can be defined as $w \bullet x + b = 0$, in which w is normal to the hyperplane and $\frac{b}{||w||}$ is the perpendicular distance from the origin to the hyperplane. There are two functions that describe this classification:

$$\begin{aligned} f(x_i) &= x_i \bullet w + b \\ f(x_i) &= \begin{cases} \geq +1, y_i = 0 \\ < 0, y_i = -1 \end{cases} \end{aligned} \tag{3.14}$$

3.2.2 Linear SVM

Here we want to find the best classification hyperplane, Figure 3.5 illustrates this problem. There are three hyperplanes representing three different classifications. h1 does not separate completely the two point sets. h2 separates the two point sets, but does not the meet the requirement of maximum margin. h3 separates the two point sets and maximizes the distance from the line to the support vectors, which corresponds to the optimal classification.

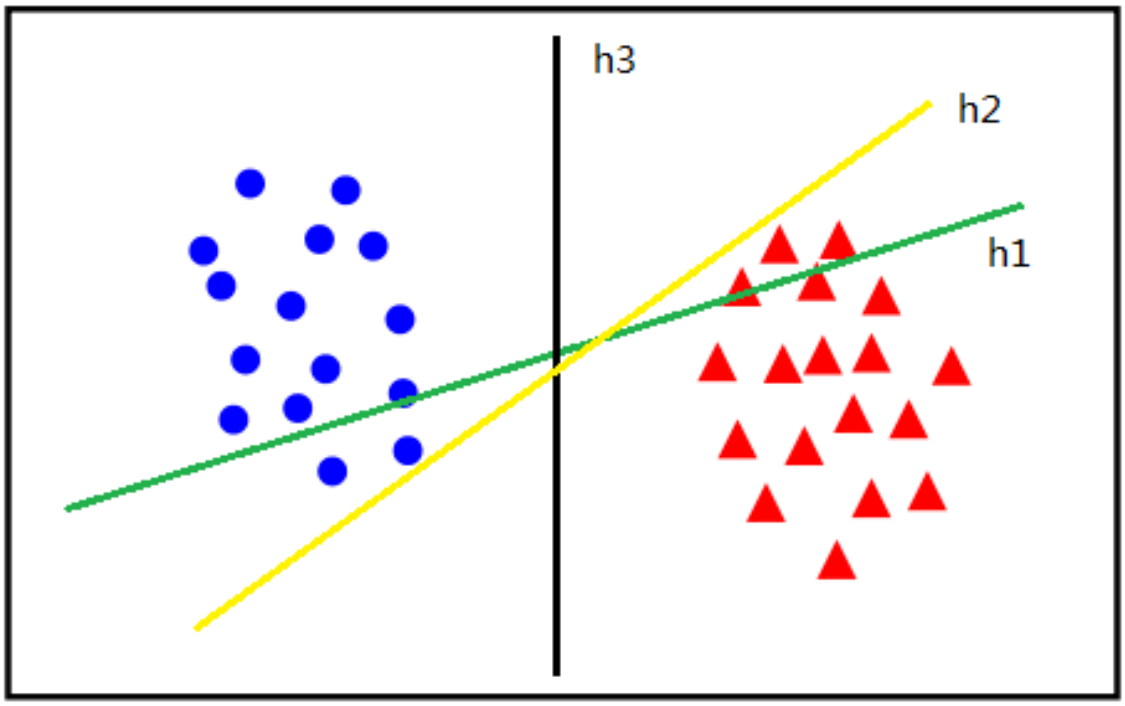


Figure 3.5: An example of different classifications in 2D

The aim of SVM is to maximize the distance (Margin) between the hyperplanes. Figure 3.6 is an illustration of an optimal classification in which a hyperplane separating the point sets can be drawn.

Those points on the blue and red dots line are the support vectors. We can see from Figure 3.8 that the distance from h1(the blue dot line) to the line is equal to $\frac{1}{\|w\|}$ and similarly for h2 (the red dot line). Thus the distance from h1 to h2 is equal to $\frac{2}{\|w\|}$. As we want to get the maximum of the distance, we should minimize the $\|w\|$, which is equivalent to minimizing $\frac{1}{2} \bullet \|w\|^2$. The transformed problem can be effectively solved as a Quadratic programming optimization (QP) problem as shown below:

$$\begin{aligned} &\min_{(w,b)} \frac{1}{2} \bullet \|w\| \\ &s.t.\ y_i(x_i \bullet w + b) - 1 \geq 0\ \forall i \end{aligned} \tag{3.15}$$

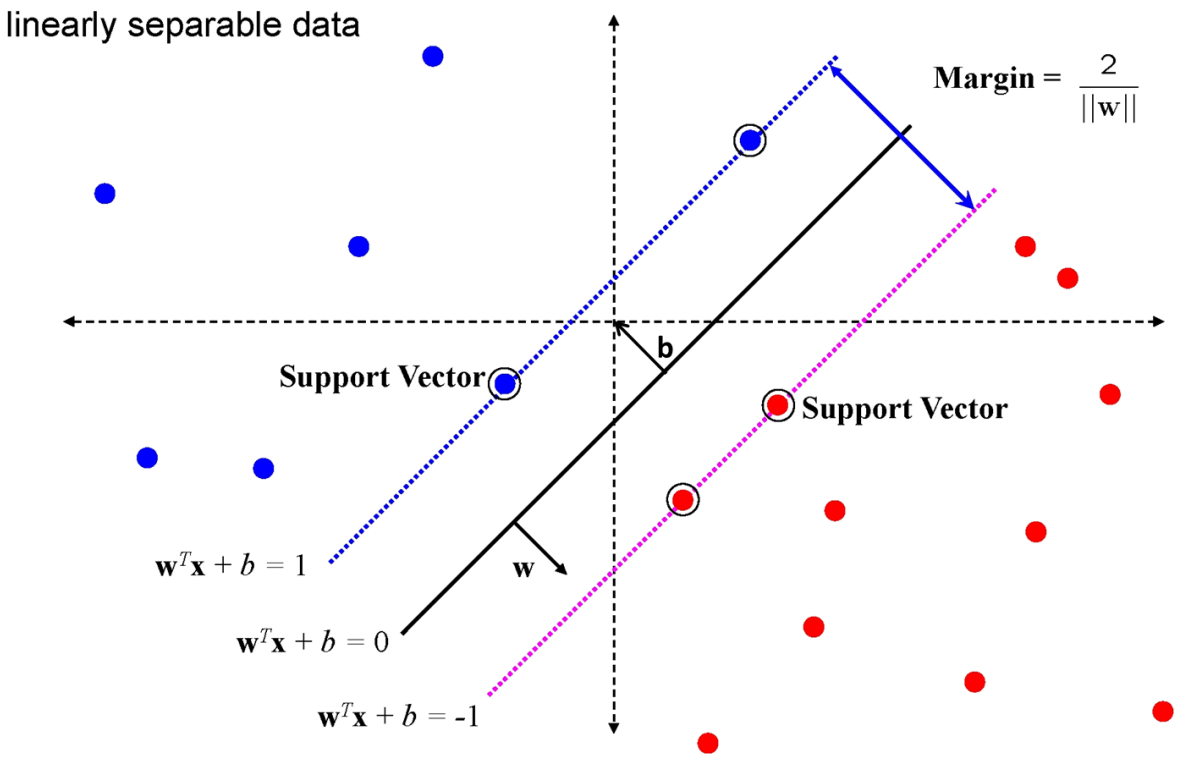


Figure 3.6: Illustration of Optimal Classification

3.3 Experiment

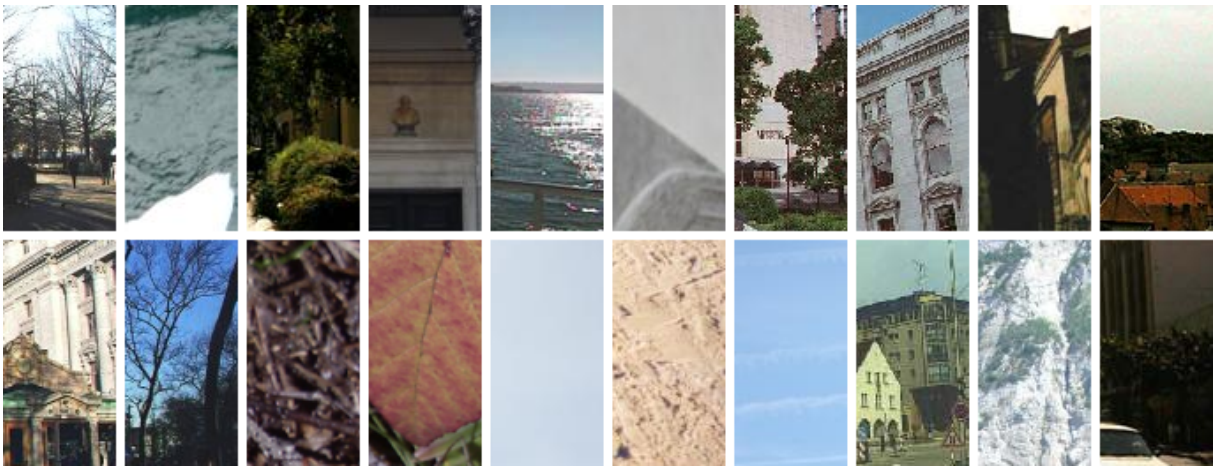
In this section, we evaluate the baseline (HOG+ Linear SVM) people detection model on the INRIA dataset. we will first introduce the dataset, and then present the obtained result. The result is presented in the form of a trade-off curve which contains the information about the miss rate and false positives per image.

3.3.1 Data Set

As people detection becomes a challenging task in computer vision, a highly realistic dataset plays an important role in the training procedure. Several image datasets can be obtained from computer vision institutes and research labs in the universities all around the world. There are several famous datasets, for example, the ETH dataset [12] recorded from cameras on a car; the Caltech dataset [11] which consists of approximately 10 hours video

taken from a vehicle driving through regular traffic in an urban environment; the MIT dataset [46], in which people images are obtained from color video sequences taken in different seasons with different video cameras; and the well-known INRIA dataset [7], which contains diversity of background with varied illumination. In this dataset, people in the images wear different kinds of clothes of different colors. Figure 3.7 shows some examples of INRIA dataset.

(a)



(b)

Figure 3.7: Some examples of INRIA dataset. (a) Positive samples, (b) Negative samples.

This dataset is split into two subsets: the training set and the test set. There are 1208 annotated peoples which are flipped horizontally to generate 2416 samples in 614 full size positive image set, and 1218 full size negative image samples in the training set. In

addition there are 1126 annotated peoples in 288 full-resolution positive images set and 453 full-resolution negative image samples in the test dataset. Here are some additional information about this dataset:

- 96 x 160 normalized and centered positive training images (left and right reflections)
- 1218 original negative training images
- 614 original positive training images
- 453 original negative test images
- 288 original positive test images
- Only upright persons with height large than 100 pixels are marked in each image
- All the images are obtained from different sources

In this thesis, we selected the INRIA dataset for the training and evaluation of our detection system because this dataset contains a large number of annotated upright people in still images taken from various viewpoints and with a variety of scenes containing several kinds of lighting condition. Detection result on the INRIA dataset is a benchmark in the majority of papers currently found in the literature. Our evaluation is only made on the 288 positive test images

3.3.2 Result

As there are a variety of poses, partial occlusions and different kind of illuminations in the INRIA dataset, reliable people detection is a challenge. However, the HOG linear SVM model works well on this dataset. The detection trade-off curve is shown in Figure 3.8, which is almost approximate to the original performance of the original work of Dala and Triggs.

From Figure 3.8 we can see that when we set a higher hit threshold, the False Positive per Image (FPPI) is at 0.1 and the HOG model achieves 50% detection rate. While if we

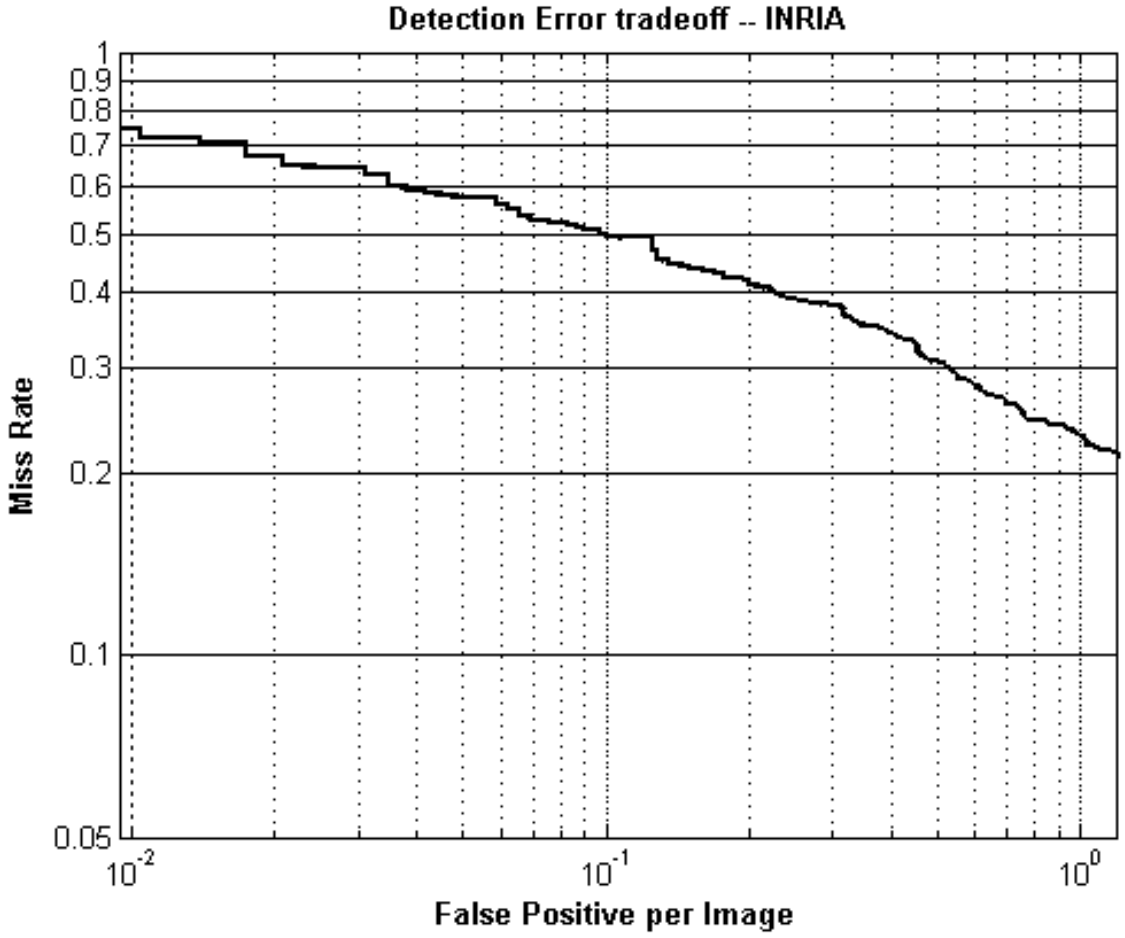


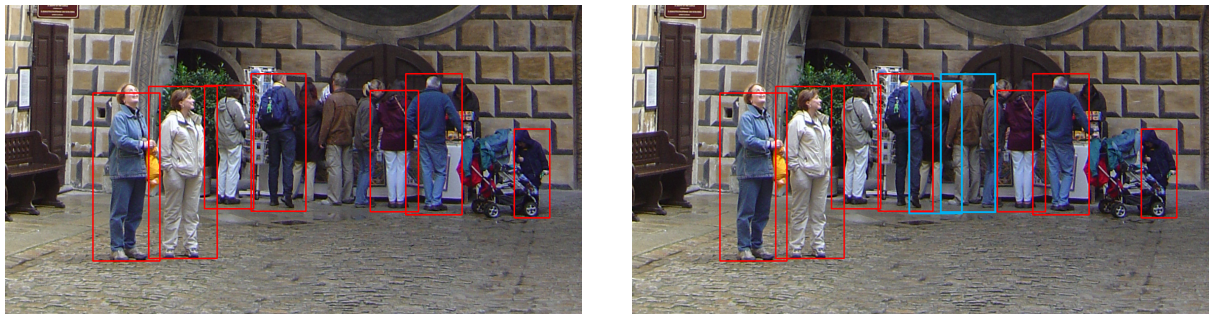
Figure 3.8: Performance of the HOG linear SVM Detector on INRIA dataset

set a lower hit threshold, the FPPI is 1 and the model achieves 78% detection rate. Figure 3.9 shows the detection results of the example images in these two cases.

Figure 3.9 (b) shows that two more persons are detected compared to the result in Figure 3.9 (a): the woman in black jacket and the man in brown jacket. These persons are in partial occlusion and they are detected only when a lower threshold is used. However when such low threshold is used, more false positives are produced. Given the detection trade-off curve, the threshold can be determined according to user's requirements.

Most people in the INRIA dataset can be correctly detected. Some representative results are shown in Figure 3.10.

We observed that the false detections can be grouped into four categories:



(a) (b)

Figure 3.9: Example of the detection for different threshold. (a) FIPPI = 0.1; (b) FPPI =1;

- *Type I*: Various poses of people.
- *Type II*: Occlusion of people.
- *Type III*: Pilar-Like Object of background.
- *Type IV*: Part of body.

Type I and *Type II* errors correspond to miss detection of people, and *Type III* and *Type IV* are false detection in the background scene.

Figure 3.11 shows two examples of *Type I* error. The peoples in these two images can be identified by our eyes easily but not by the HOG linear SVM model. The error is caused by the change in the shape of the people, different from what it has been observed in the training dataset.

Figure 3.12 shows two examples of *Type II* error. The persons occluded in these two images can be easily identified by our eyes because parts of their body are still visible while they can not be detected by the HOG linear SVM model. Although the occluded peoples stands upright, part of their body is covered by other people or objects which result in lost of information about peoples' whole body contours.

Figure 3.13 shows two examples of *Type III* error. These false detections are all pillar-like objects. The upright characteristics of these objects make their HOG descriptors

Figure 3.10: Some Results of the HOG linear SVM Detector on INRIA Dataset

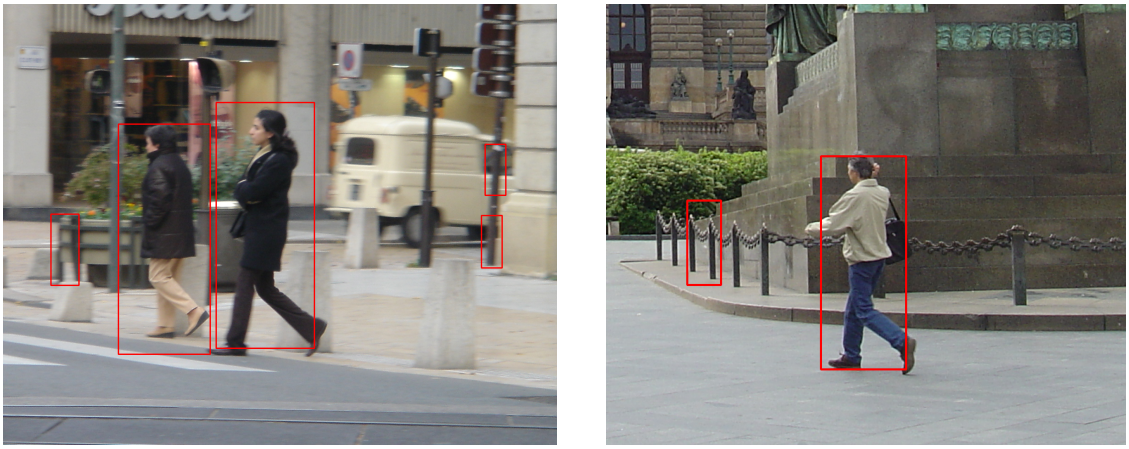
(a) (b)

Figure 3.11: Two examples of the results which is various poses of people (*Type I*)

(a) (b)

Figure 3.12: Two examples of people occlusion (*Type II*)

similar to people's descriptors.



(a) (b)

Figure 3.13: Two examples of the pillar-like objects (*Type III*)

(a) (b)

Figure 3.14: Two examples of part of people's body (*Type IV*)

Figure 3.14 gives two examples of *Type IV* error. These examples show that part of the people's body are detected as people. We can see that the false detections correspond to legs, arms, and shoulder. The shape of these parts are sometimes similar to people's silhouettes. Thus these parts are recognized as people according to their contour information.

3.4 Summary

In this chapter, we introduced the HOG descriptor which was proposed by Dalal and Triggs in [7]. This descriptor extracts contour information of people. Because the HOG descriptor is calculated on localized cells, the method is invariant geometrically. Moreover, by using contrast normalization, the HOG descriptor is invariant to illumination changes and robust to shadowing. Furthermore, a powerful classification tool, the linear SVM, has been employed in this chapter. This HOG linear SVM detection method has been evaluated on the INRIA dataset. While the observed detection performances are acceptable, this approach can be improved. The next chapter introduces an improved detection model that uses this HOG linear SVM classifier as a first layer and adds a second layer in order to make the detection results more accurate.

Chapter 4

Enhanced Contour Description for People Detection

In Chapter 3, we implement a HOG detection model which combined the HOG descriptor and a Linear SVM. Our work in this chapter aims at improving the performance of this classifier. To this end, we will both enhance the descriptors and the classifier.

In this chapter, we use Variational LBP (VLBP) as an auxiliary feature and combine it with HOG to generate a discriminative descriptor which is used in a two-layer model. VLBP is a kind of LBP-like feature. It describes the contour information of the objects in the image. Our experimental evaluation proves that it is complementary to the HOG descriptor. The computation procedure of VLBP is illustrated in Figure 4.1.

Furthermore multi-scale descriptors are used here to produce more discriminative VLBP feature so as to improve the performance of our model.

The Feature Generating Machine (FGM) is a well-know approach for feature-selection and is used in this chapter for dimensional reduction, keeping the most important part of a feature descriptor. By using FGM, our combined discriminating feature, of high dimension, can become more concise.

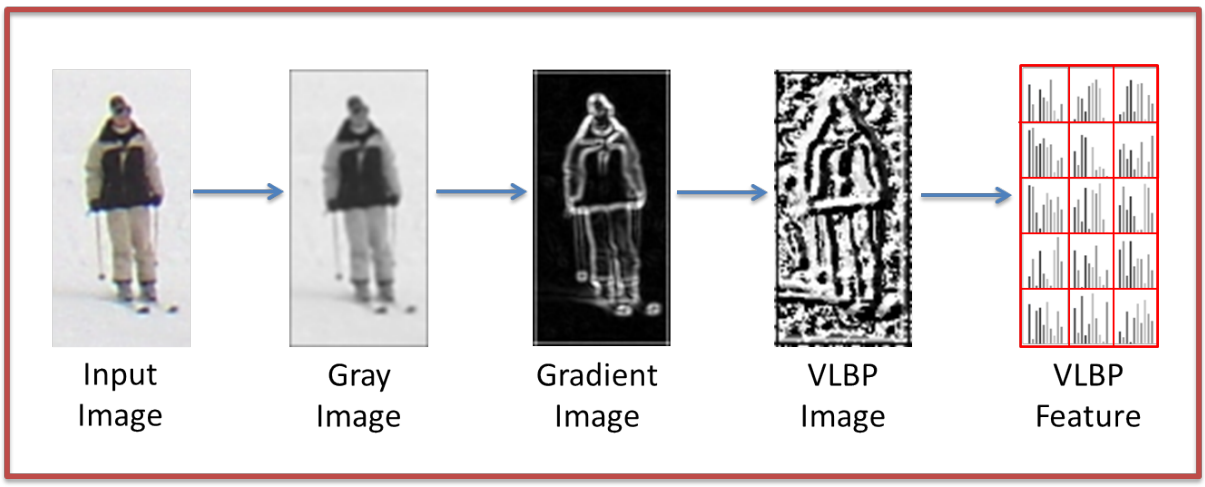


Figure 4.1: An example of VLBP computation.

To improve the classification, we introduce Histogram Intersection Kernel SVM(HIK SVM), which is much more effective than the Linear SVM used in Chapter 3. As HOG and VLBP are both histogram features, they can work well with the HIK SVM. In addition we implement fast HIK to compensate for its computational complexity.

An well-designed framework is very important in a detection system to ensure that all the implemented components work effectively. In this chapter, a two-layer cascade model is proposed to get a precise detection result while limiting the complexity of the procedure. Figure 4.2 shows the framework of this cascade model. The first layer is a fast filter based on the baseline HOG linear SVM. It is used to generate several candidates for the second layer. So we set a lower threshold than usual to make this layer to pass almost all positive samples while rejecting most negative samples. This way, only a limited amount of candidates need to be classified by the next layer. In the second layer, the HOG and the Variational LBP features are used to generate a more discriminative descriptor. Then FGM is used for feature selection. Moreover, HIK SVM is employed as a powerful tool for classification. The experimental evaluation presented in section 4.7 shows that our proposed model achieves a significant improvement.

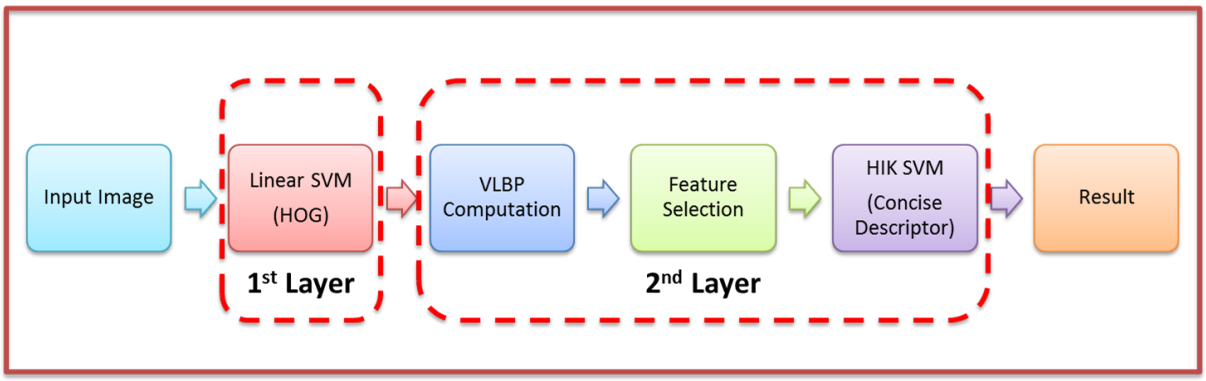


Figure 4.2: The framework of our advanced detection system.

This chapter is organized as follows. We introduce the local binary pattern in Section 4.1. In Section 4.2, we explain the variational LBP. In section 4.3, configuration of the descriptors is proposed. In Section 4.4, feature selection is presented as a useful tool for optimizing our descriptor data. In Section 4.5, Histogram Intersection Kernel is employed for powerful classification. In Section 4.6, we introduce two techniques in our procedure, which is fast implementation of HIK SVM and the bootstrapping method in the training procedure. In Section 4.7, we evaluate our proposed model on the INRIA dataset and analyze the results. Finally in Section 4.8, we make a summary for this chapter.

4.1 Local Binary Pattern

Local Binary Pattern (LBP) was first proposed in the works [41] and [42] and is well known as a texture descriptor with efficient computation. It has been used in various application, such as detections and recognitions. Figure 4.3 shows an example of LBP image. By thresholding the neighborhood of each pixel in an image, the pixels are transformed into a binary number. The most important characteristic of the LBP descriptor is that it is robust to monotonic gray-scale changes caused by the variation of illumination.

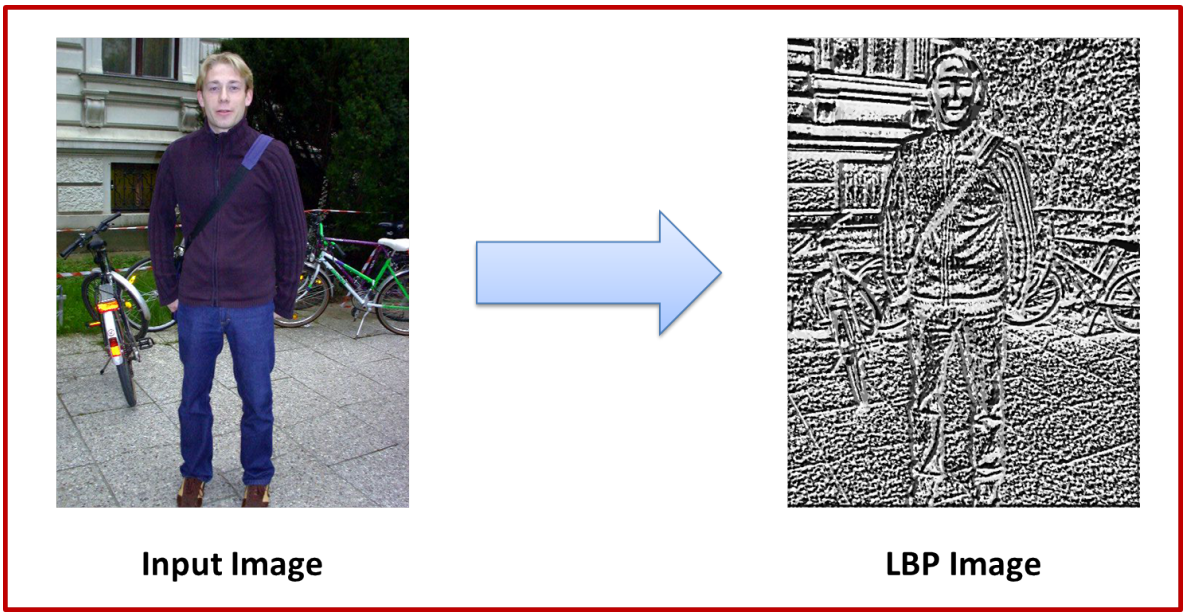


Figure 4.3: An example of LBP Image.

The original LBP encodes the local structure around each pixel. it proceeds by thresholding the 3×3 neighborhoods of each pixel around the center value by subtracting the center pixel value. The resulting negative values are encoded with 0, and the others with 1. For each given pixel, a binary number is obtained by concatenating all these binary values in a clockwise direction, which starts from the one of its top-left neighbor. The corresponding decimal value of the generated binary number is then used for labeling this given pixel. The binary numbers are considered as the LBP codes. Figure 4.4 shows the computation of basic model of LBP. The histogram of these 256 (2^8) different labels can be used as a texture descriptor.

To deal with different scales, the LBP descriptor can be extended to use neighborhoods of different size [3]. Using circular neighborhood and bilinearly interpolating values at non-integer pixel coordinates allow using any radius and number of pixels in the neighborhood. This kind of LBP is defined as the extended LBP (ELBP). Figure 4.5 shows some examples of ELBP.

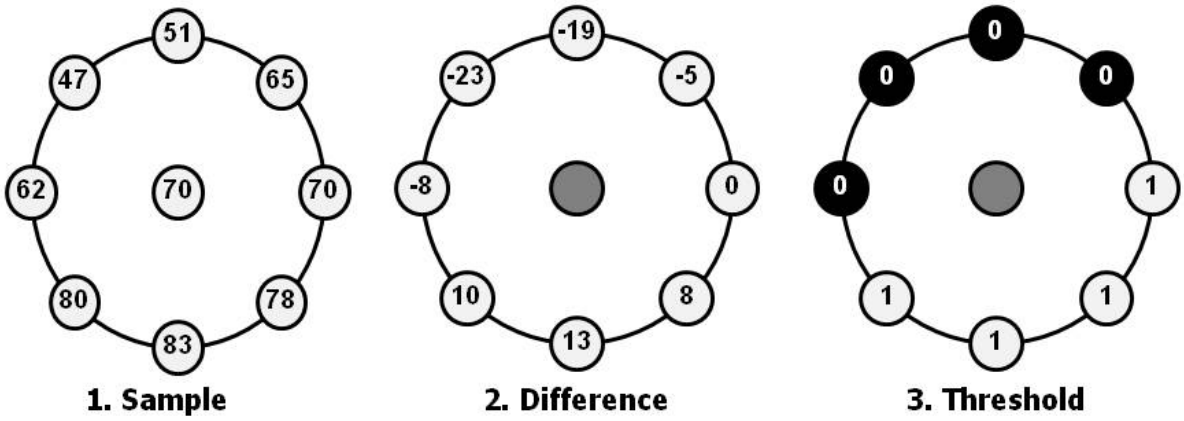
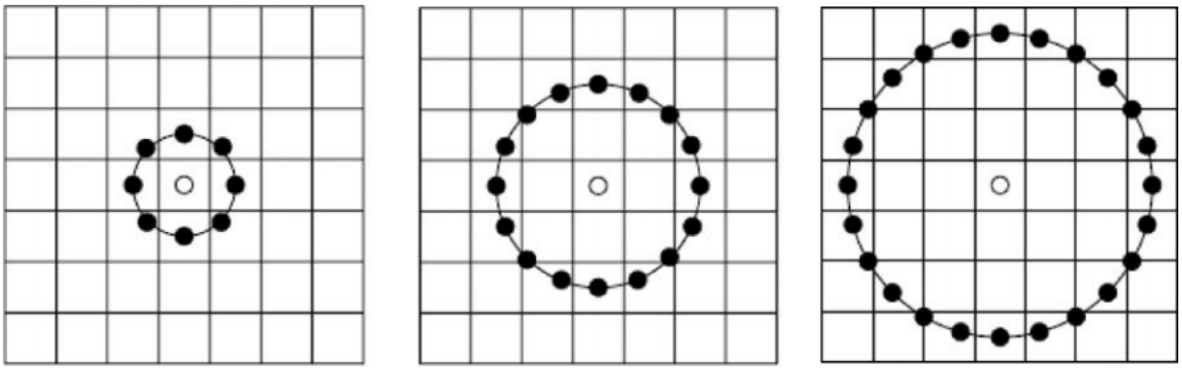


Figure 4.4: An example of basic LBP descriptor for the illustrated LBP descriptor, its value is $(00011110)_{(Binary)} = 30_{(Decimal)}$.



(a) (b) (c)

Figure 4.5: Examples of ELBP, (a) the circular (8,1), (b) the circular (16,2), (c) the circular (24,3).

4.2 Variational LBP

The Variational LBP (VLBP) descriptor was proposed by Wu et al. in [62]. It emphasizes the edge information of peoples' contour in a detection window. Instead of using the original gray image, the computed gradient magnitude image is used to produce the corresponding LBP image. The VLBP descriptor is then composed of the histogram of the blocks in the LBP image.

Specifically, we first compute the gradient G from a pair of Sobel filters (x, y) given an

input image I. Let $M = |G|$ denotes the magnitude of G. We compute the LBP image L on M, where the LBP value of a given pixel at (x_c, y_c) can be obtained as follows:

$$L_{P,R}(x_c, y_c) = \sum_{P=0}^{P-1} s(i_p - i_c)$$
$$s(i_p, i_c) = \begin{cases} 1, if\ i_p \geq i_c; \\ 0, otherwise; \end{cases} \tag{4.1}$$

Here P and R denote a neighborhood of P sampling points on a circle of radius of R. Moreover, i_c and i_p are gray-level values of the central pixel and p, one of the surrounding pixels in the circle neighborhood respectively. In this chapter, we use 3×3 cells.

After the LBP labeled image L has been obtained, we compute the VLBP descriptor, denoted by $V = (H_1, H_2, \ldots, H_n)$, where H_i, $i = 1, 2, \ldots, n$, are computed as follows:

$$H_i = \sum_{x,y} I\{L(x, y) = i\}, i = 1, \ldots, n\,. \tag{4.2}$$

Here n is the number of different labels produced by the LBP operator $I\{A\}$ is 1 if A is true and 0 if A is false. The VLBP descriptor is finally normalized by L2-norm $V = \frac{V}{\|V\|}$.

4.3 Configuration

For a 64×128 detection window, we crop the center 48×96 region for VLBP feature extraction because we experimentally found that the border region fails to provide discriminative information. Similar to the HOG descriptor computation, a histogram of LBP values with 64 bins is computed in each 16×16 block. We use a block stride of 8×8 to obtain a total of 55 blocks, and concatenate the corresponding histograms to build our 3520-dimensional

VLBP descriptor. To capture more discriminative information, multi-scale representation is usually adopted.

Considering that the 48×96 (pixels) window only encode limited information, some useful patterns tend to spread over different scales. In this work, we use three window sizes: 12×24, 24×48, 48×96. In the first case, the block sizes and strides are changed to 12×12 and 6×6 respectively. As a result, the obtained descriptor has 4352 dimensions. An example of computation of the multi-scale VLBP descriptor is illustrated in Figure 4.6.

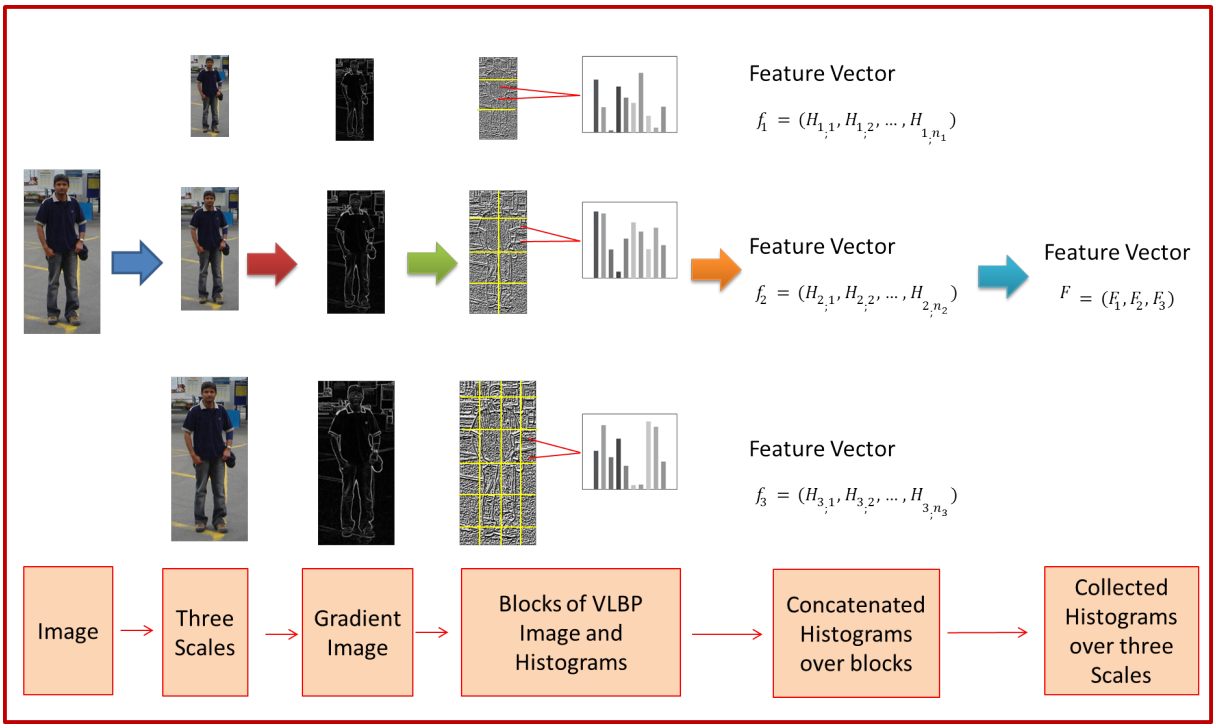


Figure 4.6: An example of multi-scale VLBP computation

4.4 Feature selection

The detection result can be improved through the use of two descriptors as we mentioned above, but the dimension of our combined descriptor vector is very high; it has a total dimension of 8132(HOG dimension 3780 + three-scale VLBP dimension 4352). This imposes a significant computational load on the second-layer classifier. Furthermore, most

part of the high dimensional features produces noisy information and thus have a negative effect on generalization performance. We therefor introduce feature selection to simplify our combined descriptor.

In order to identify the most discriminative features without impacting on the classification performance, the Feature Generating Machine (FGM) is applied. This is a very useful tool to perform feature selection on a large-scale and high dimensional data. FGM based feature selection is performed by repeatedly generating a pool of informative feature subsets. It works by solving the problems on a small number of multiple kernels. Specifically, the problem of feature selection is formulated as a SVM problem, in which the features are selected or reject according to a binary vector. A mathematical expression is defined here to gain a simplified sparse representation:

$$\begin{aligned} \min_{d\in D} \ \min_{\omega,\epsilon,\rho} \ & \frac{1}{2}\|\tilde{\omega}\|_2^2 + \frac{C}{2}\sum_{i=1}^{n}\epsilon_i^2 - \rho. \\ s.t. \ & y_i\tilde{\omega}'(x_i \odot d) \geq \rho - \epsilon_i, i = 1, ..., n. \end{aligned} \tag{4.3}$$

Here, d is defined as a vector for feature selection and is $\in$ D which is defined as: $D = \{d | \sum_{a=1}^{b} d_n \leq B, d_n \in 0, 1, n = 1, ..., m\}$; $\omega = [\omega_1, \ldots, \omega_m]' \in R^m$; $C > 0$ is a regularization parameter; $\epsilon_i > 0$ is a positive constraint.

In our problem, L0-norm Sparse SVM is converted to a mixed integer programming. To solve the problem of convex relaxation, a Cutting Plane Algorithm is introduced and combined with multiple kernel learning. After convex relaxation, the problem can be transformed into a convex multiple kernel learning problem and solved by an efficient and scalable cutting plane algorithm.

4.5 Histogram Intersection Kernel based SVM

As we have mentioned previously, SVM is a powerful tool for the classification of datasets. In Chapter 3, we discussed a common case: linear classification. In this case, a hyperplane is used as a separation boundary. But it should be pointed that, a more complex separation is required for the classification of non-linearly separable dataset. Figure 4.7 shows an examples of non-linearly separable dataset in 2D.

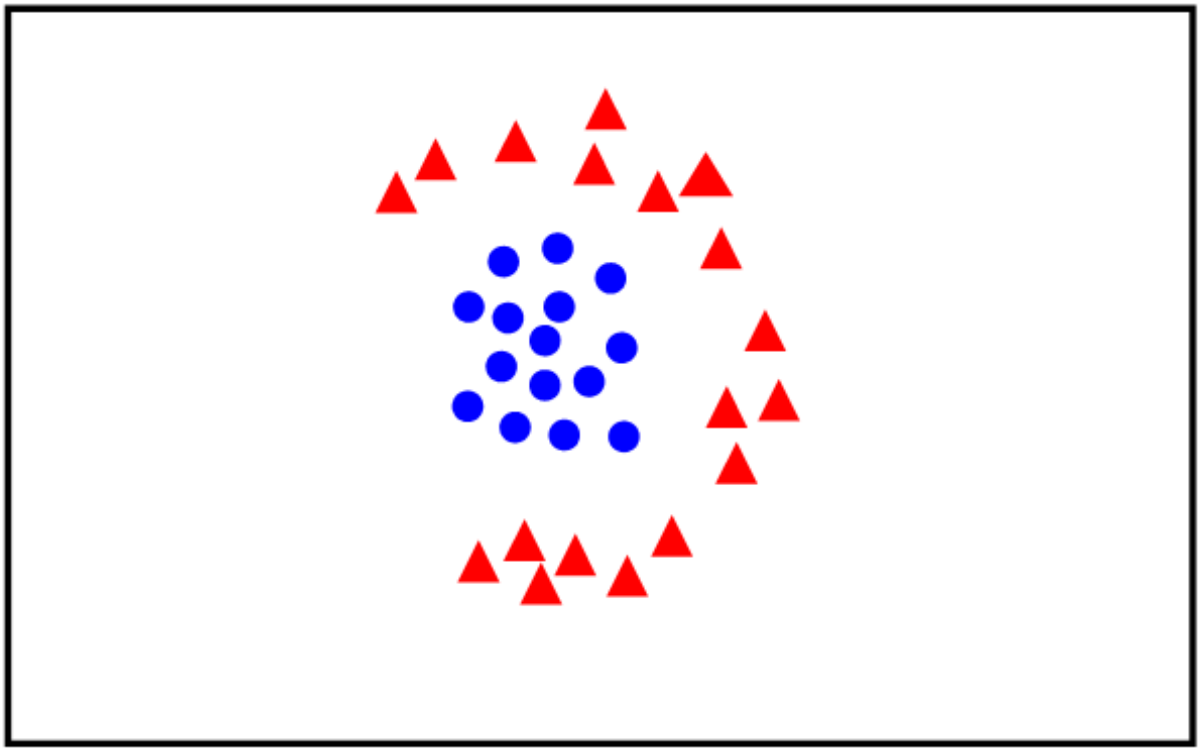


Figure 4.7: An example of the non-linearly separable datasets in 2D

Thus a more complex separation boundary is needed when we deal with such kind of dataset. Usually when we mapped the feature to a higher dimensional space, non-linearly separable feature can be transfer to the linearly separable. In the SVM algorithm, this high-dimensional mapping is possible through the introduction of the concept of kernel.

One of the useful kernel is the Histogram Intersection Kernel SVM (HIK SVM) mentioned in [40], which is well known as an effective similarity measure for objection classi-

fication and recognition in computer vision. HIK SVM was first proposed by Michael J. Swain and Dana H. Ballard in [53], and also mentioned by S Maji, AC Berg, and J Malik in [35].

Suppose that we have a binary classification problem $y_i \in \{-1, 1\}$ for $i = 1, 2, ...l$, the dual optimization problem is described in Equation 4.4:

$$\begin{aligned} &\max_{a} \sum_{i=1}^{l} |a_i| - \sum_{i,j=1}^{l} a_i a_j K(x_i, x_j) \\ &s.t. \sum_{i=1}^{t} a_i = 0,\ 0 \leq (y_i a_i) \leq C \end{aligned} \tag{4.4}$$

here a_i are the coefficient to be learned. A new sample can be classified according to the function of $\sum_{i=1}^{l} a_i K(X, X_i) + b$, here K is the *kernel* function .

Thus the histogram intersection can be defined as follows. We introduce two examples X and Z, and the histograms of them X_{in} and Z_{in}. Here these two histograms consists of n bins, and x_i and z_i $(i = i, 2, ...n)$ represent the $i - th$ bin for X_{in} and Z_{in} respectively. We assumes that X_{in} and Z_{in} have the same size, M, which means that $\sum_{i=1}^{n} x_i = M$ and $\sum_{i=1}^{n} z_i = M$, then we can obtain the histogram intersection K_{HI} by using the following Equation :

$$K_{HI}(X, Z) = \sum_{i=1}^{n} min(x_i, z_i) \tag{4.5}$$

Since the two features (HOG and VLBP) we used in this chapter are both features, for which we calculate histograms, the Histogram Intersection Kernel should work well with these two features. So we employ Histogram intersection Kernel (HIK) SVM in our classification model.

4.6 Implementation

In this section, we detail the implementation of our classification model. We use Fast HIK SVM in order to speed up the classification. We also use the bootstrapping algorithm, such to better exploit the information of the dataset thoroughly and then improve the performance of the classification.

4.6.1 Fast HIK SVM

Histogram intersection kernel combined with SVM is employed in this thesis for fast and effective classification of our feature descriptors. Many works have demonstrated that HIK SVM has an excellent performance [35], [34], [61]. In this paper, because HOG and our variational LBP are both histogram features, they are very suitable for being combined with HIK SVM.

However, the HIK SVM is computationally more expensive that the linear SVM. Its complexity is proportional to the number of support vector. To solve this problem, we employ an algorithm to speed up the classification. Works by M.Herbster [25] and Maji, et al. [34] proposed an approach for fast implementation of HIK SVM, these approaches are summarized here.

First, as we have mentioned above, the histogram intersection kernel $K_{HI}(X, Z)$ can be defined as follows:

$$K_{HI}(X, Z) = \sum_{i=1}^{n} min(x_i, z_i) \tag{4.6}$$

The criterion expression that classifies the new sample is defined as follows:

$$h(Z) = \sum_{j=1}^{m} a_j y_j K_{HI}(Z, X_j) + c \tag{4.7}$$

Thus we can get that

$$h(Z) = \sum_{j=1}^{m} a_j y_j \left(\sum_{i=1}^{n} min(z_i, x_{i,j}) \right) + c \tag{4.8}$$

in Equation 4.7, the non-linearity of "$\min(z_i, x_{i,j})$" prevent "collapsing" weight vector. Moreover, the complexity of evaluating $h(Z)$ is $O(mn)$. To achieve fast implementation, we can change the Equation 4.8 as follows:

$$\begin{aligned} h(Z) &= \sum_{j=1}^{m} a_j y_j \left(\sum_{i=1}^{n} min(z_i, x_{i,j}) \right) + c \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{m} a_j y_j \ min(z_i, x_{i,j}) \right) + c \\ &= \sum_{i=1}^{n} h_i(z_i) + c \end{aligned} \tag{4.9}$$

we define : $h_i(t)$ as:

$$h_i(t) = \sum_{j=1}^{m} a_j y_j \ \min(t, x_{i,j}) \tag{4.10}$$

The complexity of computing $h_i(t)$ is $O(m)$ while the total complexity of evaluating $h(Z)$ is $O(mn)$. However the computation of $h_i(t)$ can be reduced to $O(\log m)$ by proceeding as follows.

Let's now consider the i of $h_i(t)$ as a fixed value, then let $\bar{x}_{i,j}$ represents the sorted values of $x_{i,}$, in ascending order with the corresponding $\bar{a}_j$ and $\bar{y}_j$ for a and y in ascending order also. Let r is the largest integer for which $\bar{x}_{r,i} < t$, then we can transform the Equation 4.10 above as follows:

$$
\begin{aligned}
h_i(t) &= \sum_{j=1}^{m} \bar{a}_(j)\bar{y}_j \ \min(t, \bar{x}_{i,j}) \\
&= \sum_{1\le j\le r} \bar{a}_j\bar{y}_j\bar{x}_{i,j} + t \sum_{r<j\le m} \bar{a}_j\bar{y}_j \\
&= A_i(r) + tB_i(r)
\end{aligned}
\tag{4.11}
$$

here we define

$$
A_i(r) = \sum_{1\le j\le r} \bar{a}_j\bar{y}_j\bar{x}_{i,j} \tag{4.12}
$$

and

$$
B_i(r) = \sum_{1<j\le m} \bar{a}_j\bar{y}_j \tag{4.13}
$$

Equation 4.12 and Equation 4.13 show that both parts of $h_i(t)$ is linear. Moreover, $h_i(t)$ is continuous according to the equation as follows.

$$
\begin{aligned}
h_i(\bar{x}_{r+1}) &= A_i(r) + \bar{x}_{r+1}B_i(r) \\
&= A_i(r+1) + \bar{x}_{r+1}B_i(r+1)
\end{aligned}
\tag{4.14}
$$

In these functions above, A_i and B_i only depend on the support vectors and a, which mean that they are independent of the input and can be precomputed. Then we can compute $h_i(t)$ by using linear interpolation between $h_i(\bar{x}_r)$ and $h_i(\bar{x}_{r+1})$ to find r , which corresponds to the position of t in the list $\bar{x}_{j,i}$ which are sorted in ascending order. The values of $\bar{x}_j$ and the $h_i(\bar{x}_j)$ are stored in a lookup table. The runtime complexity of computing $h(X)$ is therefore $O(n \log m)$. If the number of support vectors is large, the

improvement of speed is remarkable. Thus we employ this effective and efficient HIK SVM implementation inside the second layer of our framework.

4.6.2 Training Procedure

In this chapter, we use bootstrapping, which was proposed in statistics first and consists of oversampling the sample data and applying some results back to the training data. In our case we use an initially trained classifier to rescan the testing samples and randomly select the falsely detected results to update the training dataset. We then iteratively update the training dataset, and make it more relevant for training.

Given the pre-trained detector of HOG+linear SVM as the first layer, the initial training data for the second layer is generated from the first layer which includes all positive samples and some randomly selected negative samples. We employ FGM as the feature selection tool to remove redundant components of our concatenated descriptor composed of HOG and VLBP. The HIK SVM is trained by using the concise feature on this initial training set. Next, a test is done on non-people images to extract hard negative samples and used them to update our negative training data. After that, we start over the above training process based on the updated training dataset. There are two criteria for stopping this process. The first one is that the false positive rate on a relatively small test set should be closed to 0 (in our cases, it should be less than 0.0001). The second one is that the number of hard negative samples generated from the test should be smaller than a preset threshold (in our case, it should be less than 30). Bootstrapping, which exploits the hard examples in the training procedure, is not always considered in classifier training; however, it does contribute much to the training procedure. The bootstrapping procedure is illustrated in Figure 4.8.

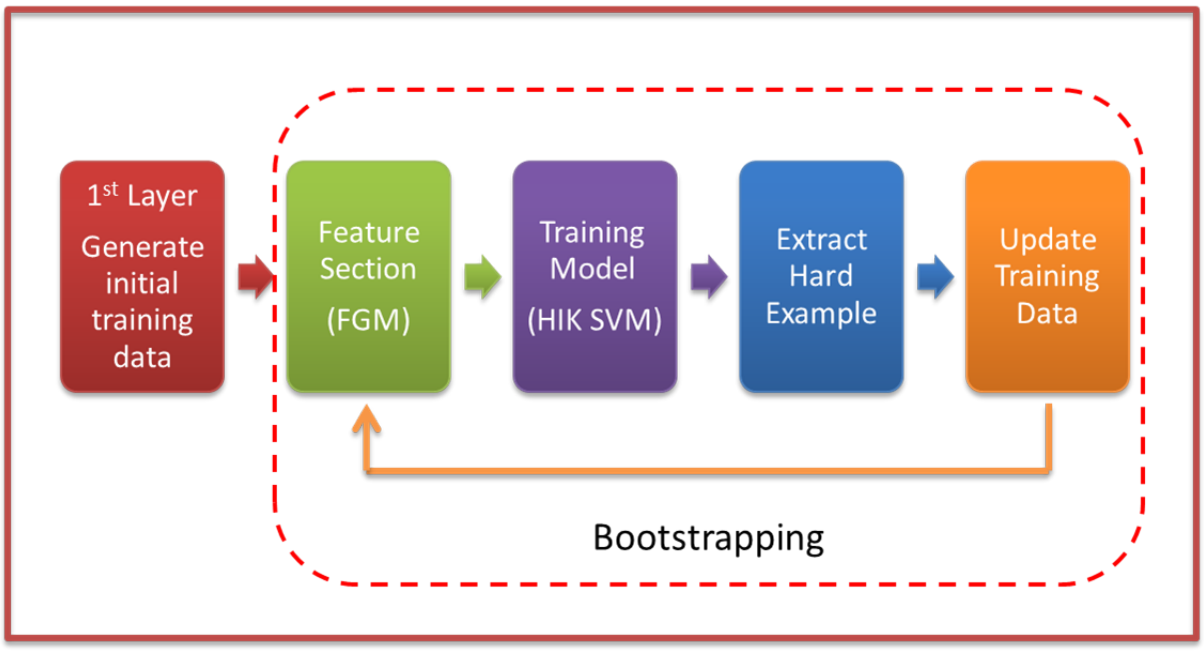


Figure 4.8: The procedure of bootstrapping.

4.7 Experiment

In this section, we evaluate our enhanced contour description for people detection on the INRIA dataset. The performance of our model is presented in the form of a trade-off curve which presents the miss rate (Y axis, for example 0.7 means miss rate is 70%) and the number of false positive per image (FPPI) (X axis, for example, 10^{-1} means 0.1 false positive per image).

4.7.1 Bootstrapping

The results of the experiment on bootstrapping is as follows: from Figure 4.9, the round 0 curve is the result of the model trained by 1126 initial random selected negative samples and all 2416 positive samples. The other curves are the results of the subsequent rounds retrained by the former negative samples combined with additional hard negative examples and the same positive samples. 2416 positive samples and 1126 negative samples are used

in the initial round. There are 2416 positive samples and 2372 negative samples used in the first round of bootstrapping, which includes the original 1126 negative samples and the hard negative examples extracted by the trained model. As it can be seen in Figure 4.9, the curve of the first round of bootstrapping is lower than the initial one, which means that the performance of the first bootstrapping round is much better than the one of the initial round.

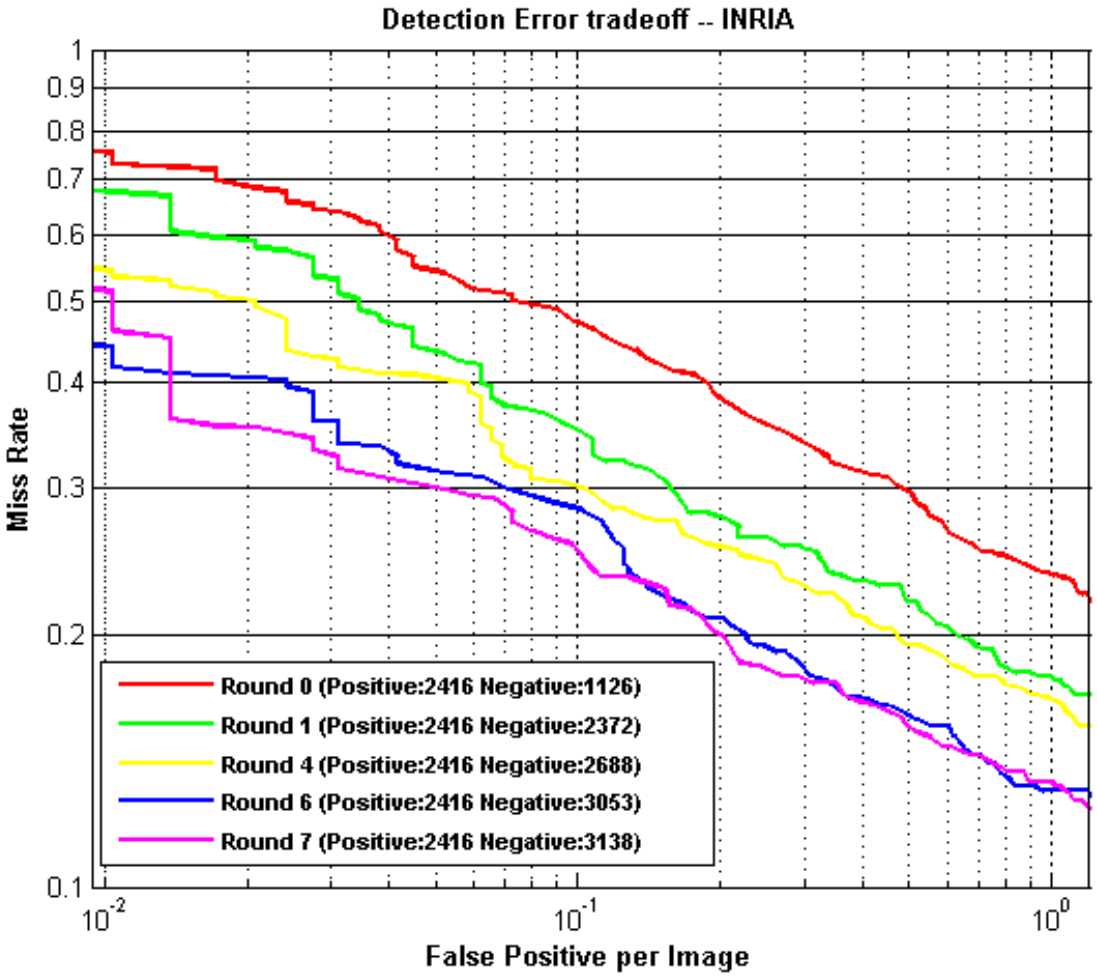


Figure 4.9: Comparison of bootstrapping rounds.

We performed seven rounds of bootstrapping. The number of negative samples used in the 7th round is almost the same as that of the 6th round, which means that only a very small number of hard negative samples are extracted and the 7th round improve only

marginally. The model has then become stable. Figure 4.9 indeed shows that the curve of the 7th round is very closed to that of the 6th round.

The 7th retraining round curve is our final result which shows much better performance than the initial curve. From Figure 4.9, we can see that there are 3183 negative samples and 2416 positive samples used in the 7th round, which means that the additional 2012 hard negative samples (compared to the initial 1126 negative samples) contribute significantly to the retrained model.

4.7.2 Comparison with baseline methods

In order to validate our approach, we compare our approach with the following detectors: HOG detector, VLBP detector, HOG_LBP detector, HOG_VLBP detector, HOG_VLBP (Multi-Scale) detector (our proposed model). All these detectors are tested on the INRIA dataset. From Figure 4.10, we can see that the performance of VLBP is not much better than HOG. When we combine HOG and VLBP descriptors, the result is improved. This demonstrates that HOG and VLBP features are complementary to each other. The result of the HOG feature combined with the VLBP feature is also better than the HOG feature combined with the LBP feature, which shows that our variational LBP is better than original LBP. We used the multi-scale VLBP features in the second layer of our model. From Figure 4.10, we can see that the performance of the curve of the multi-scale VLBP feature is better than that of only a single scale VLBP feature. Also by using the FGM algorithm, we reduce the dimension of our enhanced descriptor by almost 50% (from 8132 to 4000 dimensions) without significantly impacting the performance. The speed of our proposed cascade model achieve average 0.5 frame per second on the INRIA dataset.

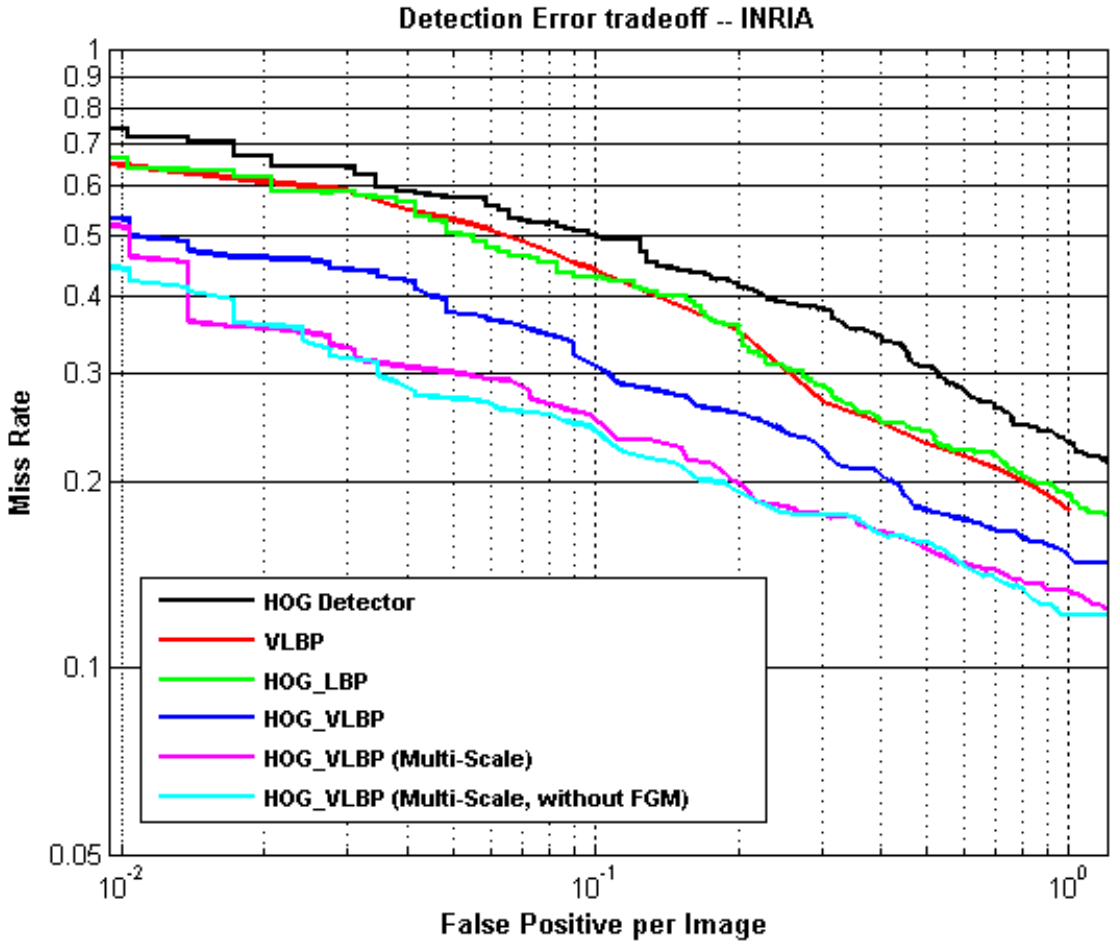


Figure 4.10: Comparison of our proposed method with baseline methods.

4.7.3 Comparison with other approaches

We also compare our proposed method with other classic detector: The Viola-Jones detector [56] , DPM detector [14], and ConvNet detector [29]. From the Figure 4.11, we can find that our model is much better than the Viola-Jones detector, which is a widely used detection system. Besides our approach is better than ConvNet in some cases (when FPPI <0.1). Although our model can not performance as well as the DPM algorithm, it is not far from it.

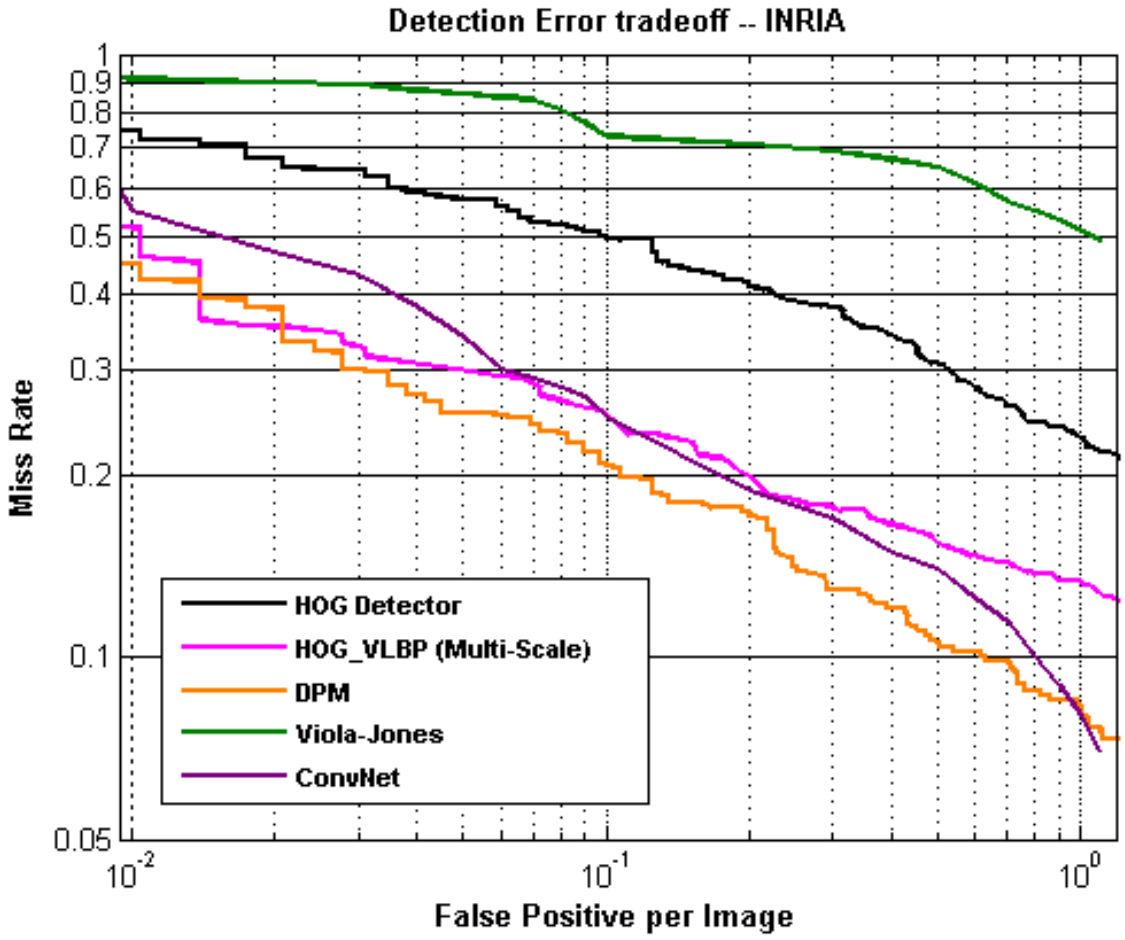


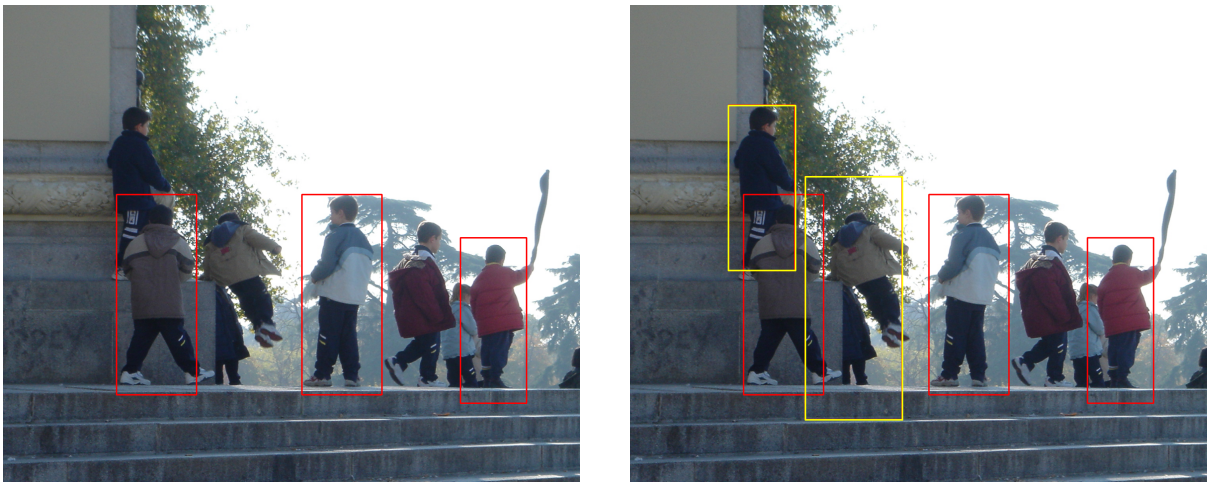
Figure 4.11: Comparison of our proposed method and other classic detectors.

Figure 4.12 shows three examples of detections using a HOG linear SVM and our model when $FPPI = 0.1$. As it can be seen our advanced model can detect more people than the baseline HOG linear SVM model.

From Figure 4.10 we find that when we set a higher threshold, the False Positive per Image (FPPI) is 0.1 and our model achieves around 75% detection rate. While if we set a lower threshold, the FPPI is 1 and our proposed model can achieves 87% detection rate. Figure 4.13 shows the detection results of the example images in these two cases.

(a) (b)

Figure 4.12: Comparison examples of the results of HOG model and our enhanced model when FPPI=0.1. (a) HOG linear SVM detection model; (b) Our enhanced model;



(a) (b)

Figure 4.13: Examples of the results for different threshold. (a) FPPI = 0.1; (b) FPPI =1;

Two more peoples are detected at FPPI = 1: the boy who stand beside the column, and the boy who is jumping. In general, allowing a higher FPPI results in better detection results (lower miss rate) at the price of increasing the false detection. Given the detection trade-off curve, the threshold can be determined following the user's specific requirements.

Some representative results of various environment, illuminations and postures of people are also shown in Figure 4.14.

Figure 4.14: Some representative Results of our proposed model on INRIA Dataset

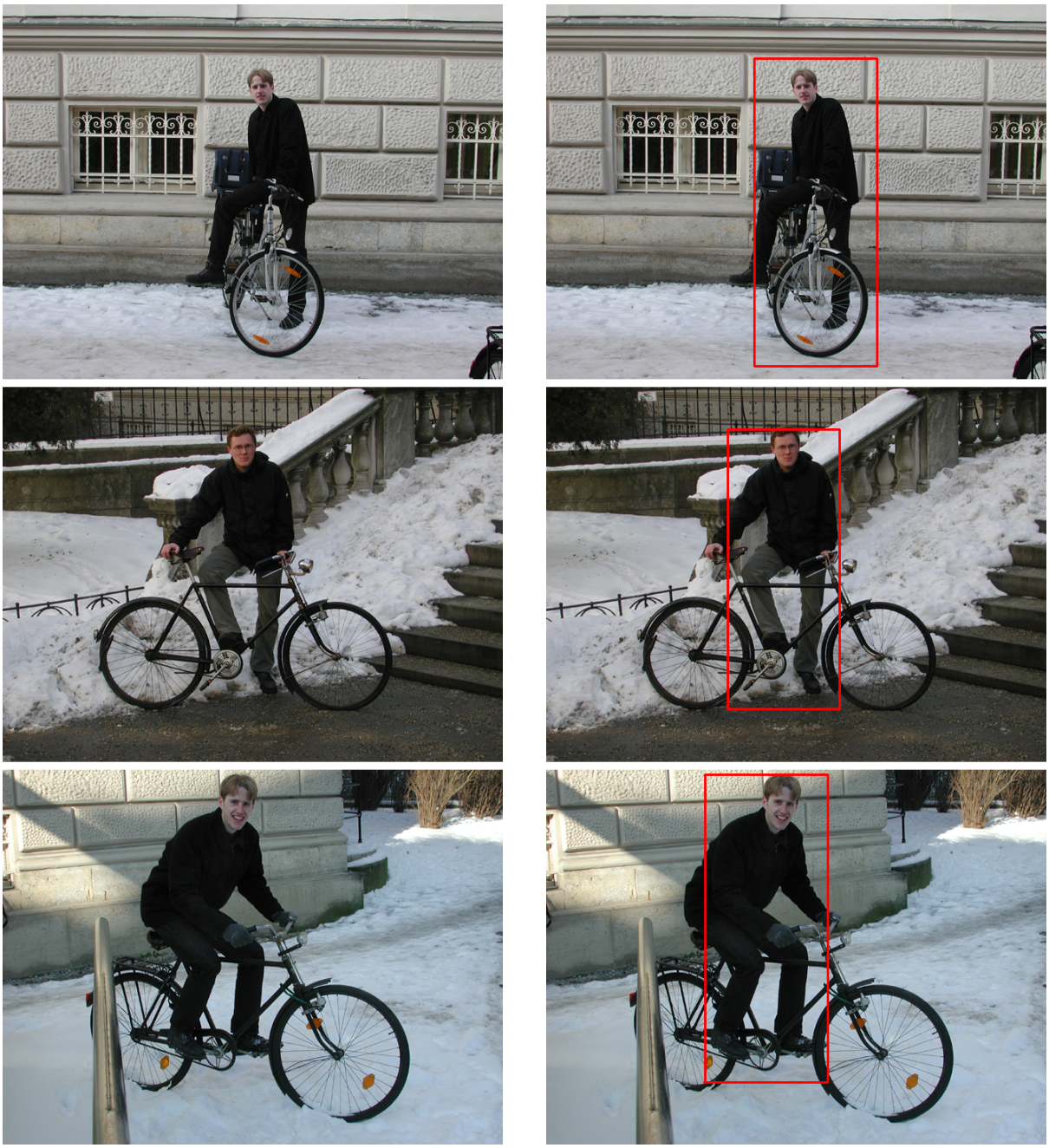
Most false positives can be grouped into 4 major types of errors:

- *Type I*: Various poses of people.
- *Type II*: Occlusion of people.
- *Type III*: Pilar-Like Object of background.
- *Type IV*: Part of body.

Type I and *Type II* errors correspond to miss detection of people, while *Type III* and *Type IV* are false detection of background. Our advanced model shows certain improvements on all four detection types.

Figure 4.15 (a) shows three examples of *Type I* that are undetected by the linear SVM HOG model. From Figure 4.15 (b) , we can see that by using our model, those hard examples can be detected.

Figure 4.16 (a) shows three examples of *Type II* errors that can not be detected by the HOG linear SVM. Figure 4.16 (b) shows the improvements brought up by our model.



(a) (b)

Figure 4.15: Comparison examples of the results of various poses of people (*Type I*). (a) the HOG linear SVM model; (b) our proposed model.

(a) (b)

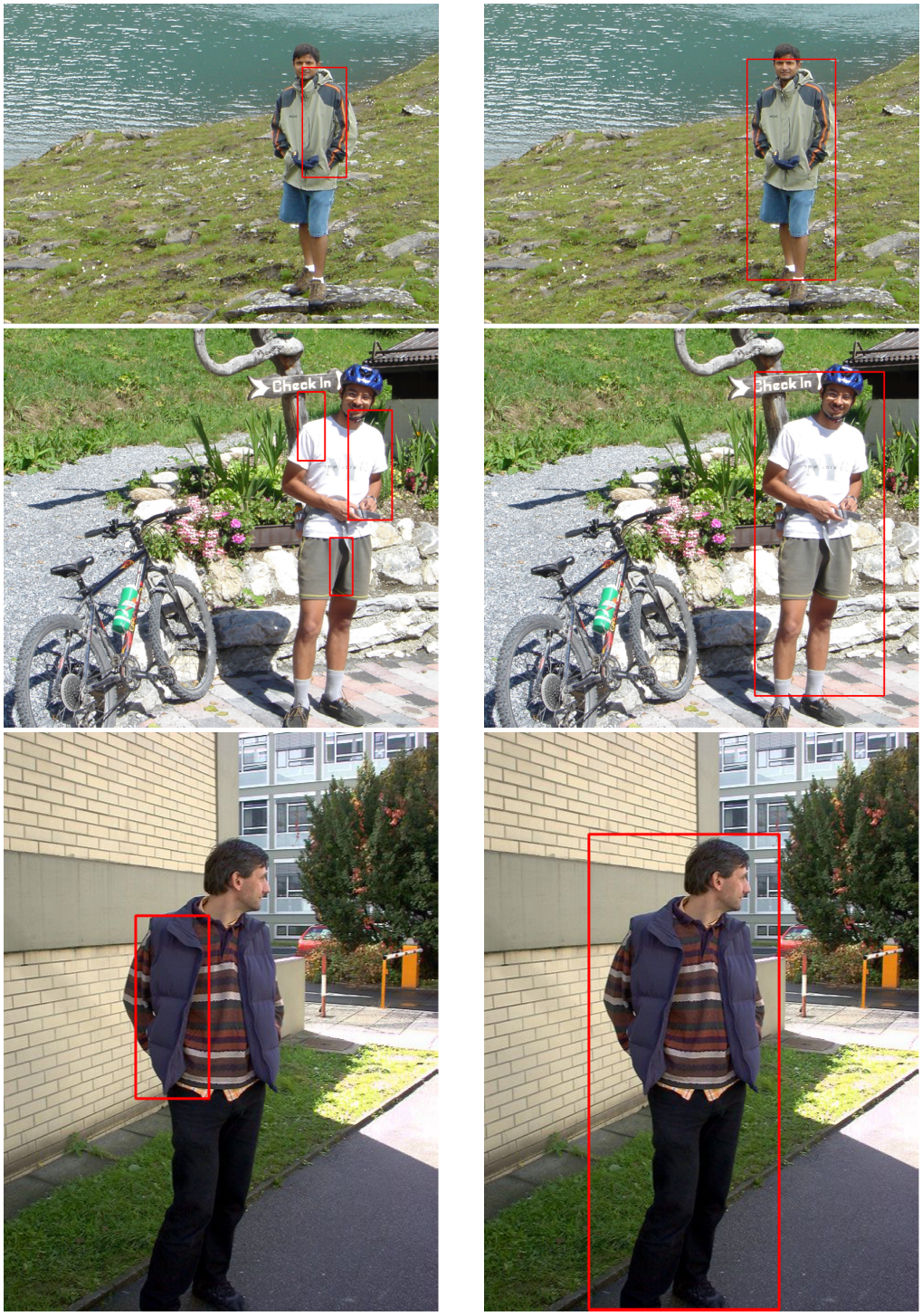
Figure 4.16: Comparison examples of the results which is occlusion (*Type II*). (a) the HOG linear SVM model; (b) our proposed model.

Figure 4.17 (a) shows three examples of *Type III* errors that produce false positives in the HOG linear SVM case. In Figure 4.17 (b), the lamp-standards and the column blocks are not detected. Thus our improved model can overcome these flaws and achieve a better performance.

Figure 4.18 (a) gives three examples of *Type IV* that are falsely detected by the HOG linear SVM. These examples show parts of the people's body detected as a peoples. Figure 4.18 (b) illustrates three improved results provided by our proposed model.

(a) (b)

Figure 4.17: Comparison examples of the results which is pillar-like objects (*Type III*). (a) the HOG linear SVM model; (b) our proposed model.



(a) (b)

Figure 4.18: Comparison examples of the part of people's body (*Type IV*). (a) the HOG linear SVM model; (b) our proposed model.

4.8 Summary

In this chapter, we improved our contour description and proposed a two-layer framework based on our discriminative descriptor for people detection. We combined the HOG descriptor and the VLBP descriptor as an enhanced feature, and employed FGM to generate a concise descriptor. Given a previously trained HOG+SVM detector on the first layer to quickly generate candidates, we then trained an auxiliary detector to make a final decision. It is noted that the second layer is efficient because a HIK SVM is applied and can be quickly implemented. The well-designed training procedure guarantees best performance. As a result, our model significantly outperforms the linear SVM HOG model, which is the baseline detection introduced in the previous chapter.

Chapter 5

Conclusion

In this thesis, we implement a baseline detection model for people detection. This detection system is robust and has an acceptable accuracy. An advanced detection model is then proposed. This model has a two-layer cascade framework, which employs the baseline model as the first layer and a finely discriminative model as the second layer. A discriminative feature is obtained, which combines the HOG descriptor and the Variational LBP descriptor, to better describe the contour information of peoples in images. Feature Generating Machine was used for dimension reduction without impacting on the classification performance. HIK SVM was introduced as a powerful tool for classification, and can be used in an efficient way through a fast implementation. Bootstrapping was used to produce an optimally trained classifier. Finally our experiments showed that advanced people detection system outperforms the baseline model on the INRIA dataset. Moreover, our proposed method can also be used to executed on the video dataset, such as Caltech Pedestrian Dataset[10].

In our future work, we can try to introduce more discriminative descriptors, for instance, combining more features and exploring color information. Moreover other algorithms for classification such as AdaBoost, Random Decision Forest and Deep Learning can also be explored for improved classification.

References

[1] Tamirat Abegaz, Gerry V Dozier, Kelvin S Bryant, Joshua Adams, Brandon Baker, Joseph Shelton, Karl Ricanek, and Damon L Woodard. Genetic-based selection and weighting for lbp, olbp, and eigenface feature extraction. In *MAICS*, pages 221–224. Citeseer, 2011.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer vision-eccv 2004*, pages 469–481. Springer, 2004.

[3] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[4] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[5] Xinyi Cui, Yazhou Liu, Shiguang Shan, Xilin Chen, and Wen Gao. 3d haar-like features for pedestrian detection. In *ICME*, pages 1263–1266, 2007.

[6] Navneet Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[8] Piotr Dollár, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *BMVC*, volume 2, page 7. Citeseer, 2010.

[9] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *BMVC*, volume 2, page 5, 2009.

[10] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[11] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[12] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. Moving obstacle detection in highly dynamic scenes. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 56–63. IEEE, 2009.

[13] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Visual object detection with deformable part models. *Communications of the ACM*, 56(9):97–105, 2013.

[14] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[15] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010.

[16] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[17] Pedro F Felzenszwalb and David McAllester. Object detection grammars. In *ICCV Workshops*, page 691, 2011.

[18] Dariu M Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1408–1421, 2007.

[19] Dariu M Gavrila and Vasanth Philomin. Real-time object detection for smart vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87–93. IEEE, 1999.

[20] Ross B Girshick, Pedro F Felzenszwalb, and David A Mcallester. Object detection with grammar models. In *Advances in Neural Information Processing Systems*, pages 442–450, 2011.

[21] Ross Brook Girshick and Pedro F Adviser-Felzenszwalb. *From rigid templates to grammars: Object detection with structured models*. University of Chicago, 2012.

[22] Zhenhua Guo, Lei Zhang, and David Zhang. Rotation invariant texture classification using lbp variance (lbpv) with global matching. *Pattern recognition*, 43(3):706–719, 2010.

[23] Feng Han, Ying Shan, Ryan Cekander, Harpreet S Sawhney, and Rakesh Kumar. A two-stage approach to people and vehicle detection with hog-based svm. In *Performance Metrics for Intelligent Systems 2006 Workshop*, pages 133–140, 2006.

[24] Marko Heikkilä, Matti Pietikäinen, and Janne Heikkilä. A texture-based method for detecting moving objects. In *BMVC*, pages 1–10, 2004.

[25] Mark Herbster. Learning additive models online with fast evaluating kernels. In *Computational learning theory*, pages 444–460. Springer, 2001.

[26] Nakamasa Inoue, Tatsuhiko Saito, Koichi Shinoda, and Sadaoki Furui. High-level feature extraction using sift gmms and audio models. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3220–3223. IEEE, 2010.

[27] Hongliang Jin, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. Face detection using improved lbp under bayesian framework. In *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 306–309. IEEE, 2004.

[28] Takuya Kobayashi, Akinori Hidaka, and Takio Kurita. Selection of histograms of oriented gradients features for pedestrian detection. In *Neural Information Processing*, pages 598–607. Springer, 2008.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[30] Kang-Dae Lee, Mi Young Nam, Kyung-Yong Chung, Young-Ho Lee, and Un-Gu Kang. Context and profile based cascade classifier for efficient people detection and safety care system. *Multimedia Tools and Applications*, 63(1):27–44, 2013.

[31] Shu Liao and Albert CS Chung. Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *Computer Vision–ACCV 2007*, pages 672–679. Springer, 2007.

[32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[33] Arko Lucieer, Alfred Stein, and Peter Fisher. Texture-based segmentation of high-resolution remotely sensed imagery for identification of fuzzy objects. In *Proceedings of GeoComputation*, 2003.

[34] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[35] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Efficient classification for additive kernel svms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):66–77, 2013.

[36] Gonçalo Monteiro, Paulo Peixoto, and Urbano Nunes. Vision-based pedestrian detection using haar-like features. *Robotica*, 24:46–50, 2006.

[37] Greg Mori, Serge Belongie, and Jitendra Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[38] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[39] Stefan Munder and Dariu M Gavrila. An experimental study on pedestrian classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1863–1868, 2006.

[40] Francesca Odone, Annalisa Barla, and Alessandro Verri. Building kernels from binary strings for image matching. *Image Processing, IEEE Transactions on*, 14(2):169–180, 2005.

[41] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE, 1994.

[42] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[43] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 130–136. IEEE, 1997.

[44] Patrick Ott and Mark Everingham. Implicit color segmentation features for pedestrian and object detection. In *Computer vision, 2009 IEEE 12th international conference on*, pages 723–730. IEEE, 2009.

[45] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *Computer Vision–ECCV 2014*, pages 546–561. Springer, 2014.

[46] Constantine Papageorgiou and Tomaso Poggio. Trainable pedestrian detection. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 4, pages 35–39. IEEE, 1999.

[47] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[48] Henry A Rowley, Shumeet Baluja, Takeo Kanade, et al. *Human face detection in visual scenes*. School of Computer Science, Carnegie Mellon University Pittsburgh, PA, 1995.

[49] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[50] Amnon Shashua, Yoram Gdalyahu, and Gaby Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 1–6. IEEE, 2004.

[51] Frédéric Suard, Alain Rakotomamonjy, Abdelaziz Bensrhair, and Alberto Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 206–212. IEEE, 2006.

[52] Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):39–51, 1998.

[53] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[54] Mingkui Tan, Li Wang, and Ivor W Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1047–1054, 2010.

[55] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

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

[57] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1030–1037. IEEE, 2010.

[58] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[59] Christian Wojek and Bernt Schiele. A performance evaluation of single and multi-feature people detection. In *Pattern Recognition*, pages 82–91. Springer, 2008.

[60] Bo Wu and Ramakant Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 90–97. IEEE, 2005.

[61] Jianxin Wu. Efficient hik svm learning for image classification. *Image Processing, IEEE Transactions on*, 21(10):4442–4453, 2012.

[62] Jianxin Wu, Christopher Geyer, and James M Rehg. Real-time human detection using contour cues. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 860–867. IEEE, 2011.

[63] Jianxin Wu and Jim M Rehg. Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2011.

[64] Yuji Yamauchi, Hironobu Fujiyoshi, B-W Hwang, and Takeo Kanade. People detection based on co-occurrence of appearance and spatiotemporal features. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[65] Chengbin Zeng and Huadong Ma. Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2069–2072. IEEE, 2010.

[66] Xingyu Zeng, Wanli Ouyang, Meng Wang, and Xiaogang Wang. Deep learning of scene-specific classifier for pedestrian detection. In *Computer Vision–ECCV 2014*, pages 472–487. Springer, 2014.

[67] Yongbin Zheng, Chunhua Shen, Richard Hartley, and Xinsheng Huang. Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection. In *Computer Vision–ACCV 2010*, pages 281–292. Springer, 2011.

[68] Yongbin Zheng, Chunhua Shen, and Xinsheng Huang. Pedestrian detection using center-symmetric local binary patterns. In *ICIP*, pages 3497–3500, 2010.

[69] Qiang Zhu, M-C Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.