UNIVERSITY OF SCIENCE

ADVANCED PROGRAM IN COMPUTER SCIENCE

Đặng Trương Khánh Linh - Bùi Huỳnh Lam Bửu

PEDESTRIAN DETECTION AND LOCALIZATION

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

HO CHI MINH CITY, 2011

UNIVERSITY OF SCIENCE
ADVANCED PROGRAM IN COMPUTER SCIENCE

Đặng Trương Khánh Linh	0612743
Bùi Huỳnh Lam Bửu	0612733

PEDESTRIAN DETECTION AND LOCALIZATION

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THESIS ADVISOR
Lê Hoài Bắc

HO CHI MINH CITY, 2011

ACKNOWLEDGEMENTS

We greatly want to express my gratitude to my thesis advisor, Professor Le Hoai Bac, who have helped and guided us a lot when we start to do this thesis. Professor Bac aroused our interest in soft computing, especially in computer vision. Moreover, he helped us get the background and have the overall view of computer vision. They also give us some very incredible ideas and advices, as well as encouragement when we got stuck.

Moreover, we would like to give our thanks to Vo Dinh Phong and Tran Ngoc Trung for their objective argument and assistances to our thesis. We also received very incredible ideas from them in workshop at every weekend.

Finally, our classmates and friends from computer vision group of Mister Vo Dinh Phong also are very important to us because their ideas about our thesis are very meaningful. In addition, we would like to deeply express our thankfulness to all people who read and revise our thesis draft.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This thesis' goal is to build up the automatic system which enables to detect object in static image, particularly pedestrian. Our main concern is on exploring robust feature extraction algorithms which encode image regions to high dimensional feature vectors that have outstanding performance. Our extracted feature set robustness is tested by using linear Support Vector Machines which classify each image region as an object or as a non-object. Our approach is purely data-driven based on bottom-up perspective using low level appearance to detect objects. In evaluating process, we choose the most challenged case which is pedestrian to evaluate our approach's performance. Our thesis is based on well-known Dalal work of Normalized Histogram of Oriented Gradients (HOGs).

This thesis makes some following contributions. Firstly, we re-implement and evaluate HOG method and give some insights about its strengths and weaknesses. Secondly, from these insights, we modify some aspects of HOG descriptor to enhance its strength as well as eliminating its weakness. In the first modified HOG descriptor, we disregard non-informative region which is the center of image window. In the second one, instead of using uniform grid of point like in original work of Dalal, we use non-uniform grid to enhance its strength and reduce its weakness. The final one is called multi-level in which we utilize multi-level concept when divide image window as grid of points. From experiment, the first two contributions have the performance as good as the original one, while reduce the length of feature vector. In the third contribution, we enhance the performance by 3% compared to Dalal's works on HOG descriptor.

Chapter 1. Introduction

1.1. Thesis introduction

Computers have been used intensively in our daily lives. In the past, people used them to speed up complex calculation. Moreover, computers nowadays are not only the big calculators, but it also can simulate human perspectives. From the virtual Turning machine, many researchers believe that capacity of human and computer are comparable if human's thought and decision based on step by step process [3]. Computer scientists have tried to extend computer's capability in order to allow smart machine to substitute human in some certain jobs such as dangerous or poisonous ones. They have added artificial intelligent to computer to make it seems to have perspectives as human. Among enormous applications of computer, computer vision is the subject which draws most attention of computer scientists.

Our daily lives is filled of millions of objects ranging from big ones such as human, car, bicycle,… to tiny ones like cells. And the task of recognition and classification each object to its catalogue is the fundamental task for any intelligent based system [4]. The difficulty is that a given class has a huge intra class variation. For example, human is usually thought as an object consists of up-right shape, two legs, two hands, and an omega shaped head. However, in reality, human probably appear in diverse shapes. For example, people who sit down, stand up, lie down, or play sport have totally different shape. In addition, illumination, points of view are also the significant factors affect to recognition and detection process.

Thus, recently, the goal of researchers working in computer vision and intelligent based machine is to invent algorithms or facilities in order to allow computer has the ability to see and analysis a given images or videos. And one of the primary tasks is the detection and catalogue objects in images. Such ability allows us to have numerous applications such as human computer interaction, robotics, smart autonomous vehicle as well as image retrieval.

In this chapter, we begin section 1.1 with the brief discussion about our goal and applications. Section 1.2 is going to mention the challenges of object detection task. Section 1.3 presents a brief background as well as the general perspectives on object detection. Section 1.4 provides the overall framework of our approach. Section 1.5 discusses some our key contributions as well as our observations when doing this research. Section 1.6 gives the outline of the structure of this thesis.

1.2. Goal and Applications

1.2.1. The goal

The main target of this thesis is to build up an automatic system which is possible to detect and localize pedestrians in static image. For more specific, it is the issue of creating object detection from the view of point of computer, in which detector scan all the given images and bound the box around object if it appears in image. We use the approach which utilize robust extraction algorithm to extract a region of an image, and then use a classifier to decide whether this region contains pedestrian or not. In this cope of thesis, we just concentrate on how to encoding image regions into feature vectors which is robust on illumination, slight change and osculation.

Unlike matching one word with another word in which we can easily see they are identical or different, but matching object with object (for example, human) is the totally different matter. Natural object such as human, cat, dog and man-made object such as car, bicycle have diverse of shapes, so it is difficult for computer to distinguish two catalogues. In this thesis, we use an approach which does not make strong assumption on context. For example, the context of car in cartoon is wider perspective than the car in show room. So, if we heavily depend on context of the car, we will miss the car in other view of point, such as car in painting or cartoon. Overall, the goal is to build a detector which can detect general object in wide perspective.

- Input: arbitrary image.
- Output: boundary box which contains pedestrian if image has that one.

1.2.2. Applications

Robust extraction algorithm is not only useful in finding pedestrians in images, it also can be used to extract characteristic of any object. So, we can use this descriptor as a core in system of analyzing and cataloguing images in album. We obviously see that the advent of digital camera has allowed people to take photograph more easily. In 2-3 years, one personal digital camera can take as many as 10,000 photos, and which is impossible for human to manually search and locate these photos in short time. Consequently, Intelligent Management Software which can automatically add tags to these images to facilitate search is dispensable.

Moreover, person detectors are also being employed for detect pedestrians in smart cars. For instance, a warning message will appear in windshield to arouse drivers whenever the car tends to hit pedestrians or obstacle. Another application in smart car system is that cameras can detect the behaviors and consciousness of drivers in order to execute some proper assistance.

Information detected in multi-cameras will be fused together; and with the training knowledge in system, detectors will make reasoning decision to whether take a certain action or not. However, there is almost no detector which is good performance can execute in real time. For example, with the limited capacity of processing unit of portable devices, it is really hard for them to use good performance detectors in real time. Fortunately, in recent years, by the breakthrough in chip processing, and associating with some good detectors, building the software for smart cars is the subject that draws a lot of attention of researchers.

In biology field, computer vision, or object detection in particular has been applied a lot. Nowadays, object detection enables biologists effortlessly classify different types of cell. Before the advent of computer vision, it took a lot of effort and time of scientists to count and classifier cells. It is very difficult because the number of cell is enormous and more than 200 types of cell exist in reality. By the mean of object detection which has good performance would probably help a lot for biologists to speed up their experiment and analysis.

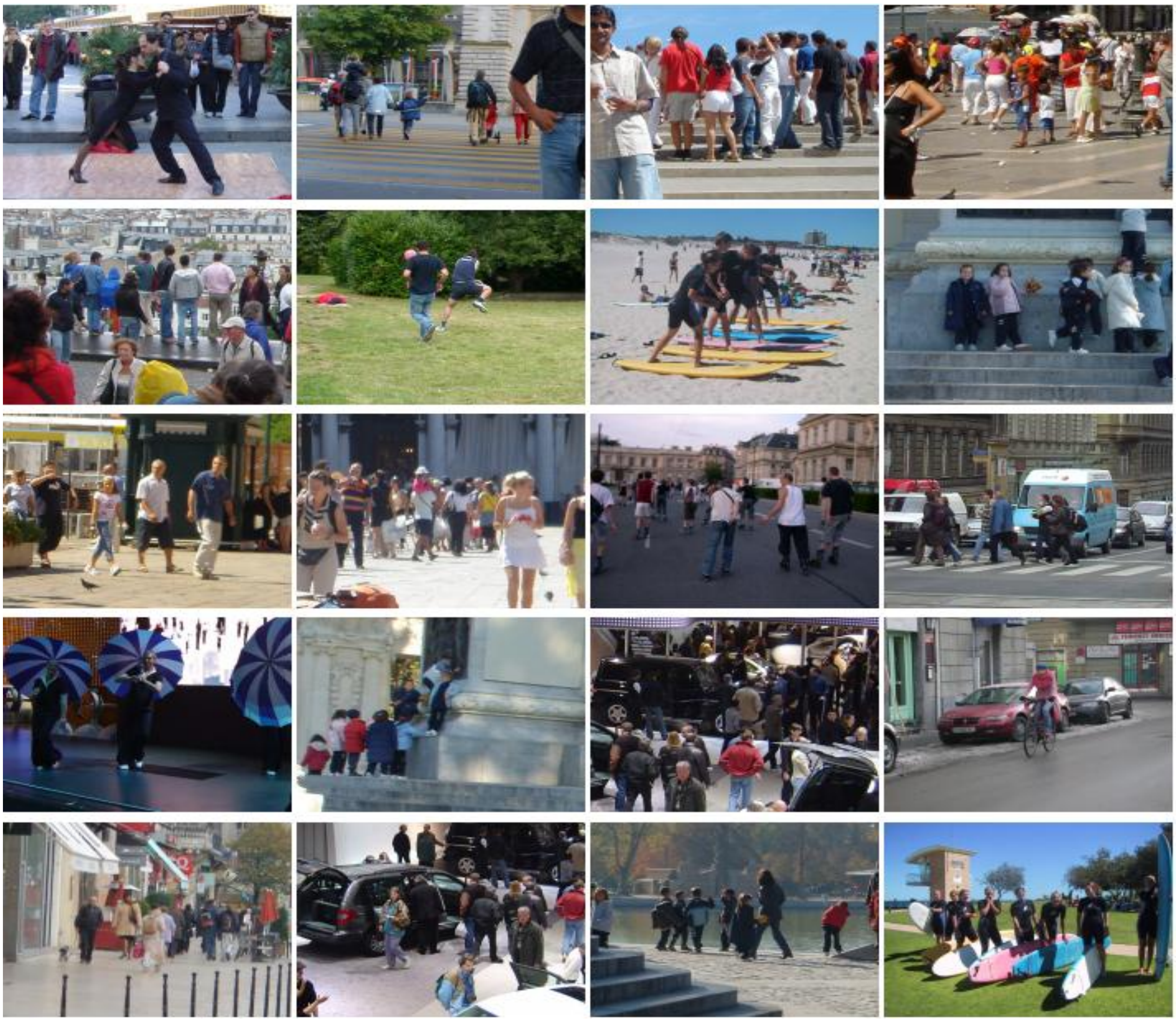


Figure 1:some images from collection of personal digital camera. This is INRIA pedestrian dataset, is the benchmark data for every analysis in this thesis. The collection consists of people from wide range of variation on pose, appearance, clothing, illumination

1.3. Challenge

The most difficulty of building an object detector is the diverse of variation in images. These following factors effect on object detector are described below.

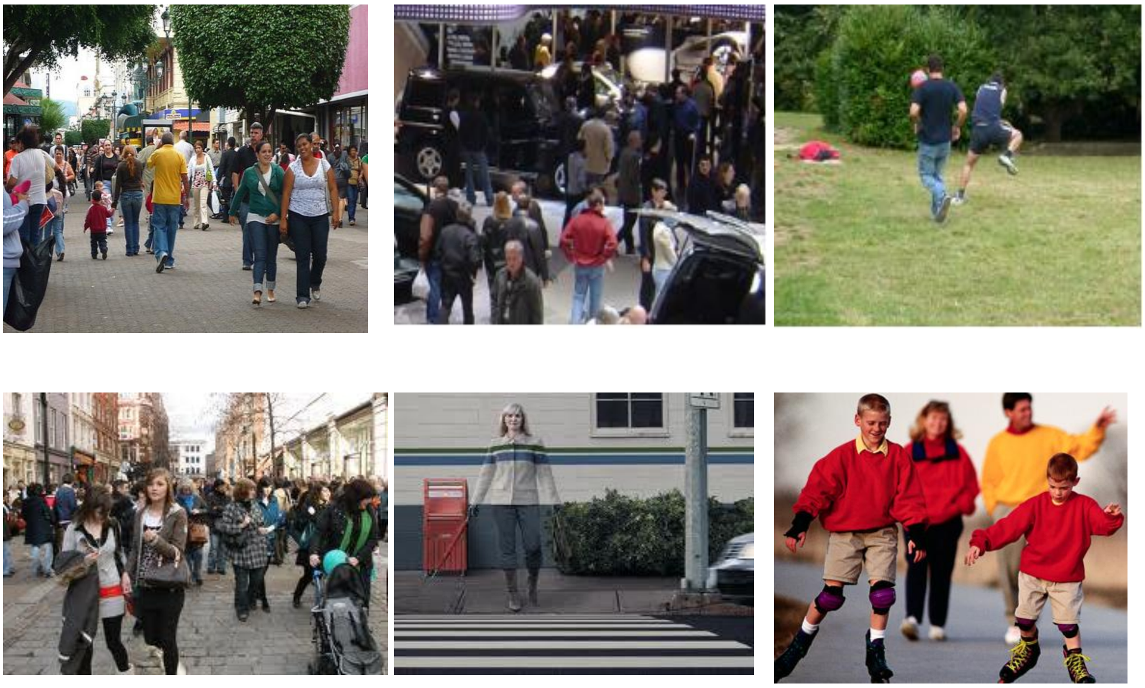


Figure 2:Some challenge images that make pedestrian detection is a difficult task

Firstly, image is just a matter of pixel, and it lacks of motion knowledge like in video. Object in image suppress 3-D information and depend on viewpoint of camera as well as the scale. As mentioned above, most natural object classes have huge variation in intra-class. Although two instances belong to one object class, they probably appear different on account for illumination, viewpoint, and shape distorting.

Background information is also the vital key to prevent us from building robust detectors. Background clutter varies from image to image. For example, images can be taken from indoor, outdoor, and under diverse natural factors such as illumination, viewpoint. So, the desirable detectors have to have the ability of distinguishing object in complex background.

Moreover, in image, color and illumination of objects in one class probably varies considerable. Let’s think of a photograph taken in day with direct sunlight and shadows versus one taken in night with dim light, you easily see how the big gap they have. So, the robust detector must have capacity of resisting of changing color and illumination in object.

Finally, partial occlusion is an inevitable in real images. In this situation, just only a part of object can be visible. That is the reason why creating a good performance detector is very difficult.

1.4. Some background of object detection

Object detector is indeed the combination of an image feature set and a detection algorithm. Feature extraction can be spare or dense representation of image region as feature vector. Typically, feature extraction is the way of capturing intensity patterns, texture details, shape, and contour information. Nowadays, feature vectors are under two perspectives which are described below.

Spare feature extraction based approach is the first one. This is an approach taking advantages of a set of salient image regions. It is based on the assumption that not all image regions contain useful information, which is uniform, textureless, or too cluttered to use. The motivation of this idea comes from the studies of physiology which observe the human eye-tracking. Form these studies, scientists find out that the gaze preferably fixates on image regions with corners and multiple superimposed orientations [5,6,7], and that local spatial contrast is significantly higher at these points than at random locations, while image uniformity and pixel correlations has less effect on human’s gaze.

On the other perspective, the alternative approach is to densely compute feature vectors on image regions. Unlikely the first approach, the second one assumes that, in the early stage of visual scene analysis, all image regions might be of equal importance and small details should not be eliminated because it can adversely affect the performance. Instead, the second stage will decide which regions are the most relevant. This idea derives from the studies of mammalian visual system which say the first level of visual coding in mammals involves the computation of dense and overlapping center-surround receptive-fields of different scales [8,9,10].

Both spare and dense approaches require to fully scanning whole image. The indeed difference between them is that the final encoding feature vector. Salient points based approach such as SIFT [21] or *shape context* [20] scan through the image to find the blobs, and the final encoding feature vectors are calculated on gradient and contour information. On the other hand, spare approach has no awaveness of how to choose in the first phase. Instead, spare ones will keep the relevant information in the second phase.

We now move to the framework of detector; several detector models have been employed nowadays. Although there are some differences among them, all detector models are seemly divided into two categories.

Discriminative model:

This approach is to learn to recognize classes of image regions which commonly occur in the given object class. It can be viewed as "parts based approach" [11,12] which takes advantages of structure of object. For example, when I see two wheels, one handle bar, some rods connecting two wheels, a seat and no motor, I can know that it is bicycle. The notion what defines a part is also not clear. Some approaches try to detect physical parts (e.g: human body consist of head, legs, arms, and torso), while others attempt to define small region or use the salient image regions as representation of parts.

Generative model

The simpler approach is to implicitly encode spatial information in the form of rigid templates of feature vectors. This one is usually based on densely computed image representations. After that, state of the art machine learning methods such as SVMs are employed to create the complete detector.

1.5. Overview of our approach

In give image, we use sliding window to densely scan at all position at different scales. At each position, we get its score, and we decide this window contains object or non-object via classifier. This method is purely based on statistic approach which disregards the fore-given context of any object class. When extracting region containing object, we assume that there are some invariants which are not change dramatically within one type of object. These invariants become the main characteristics for classifier to distinguish this object class with other object class. So, by extracting invariants of object or non-object, we can represent them in high dimensional vector. And we assume that it is possible to build up a hyper-plane which separates object, non-object points as far as possible.

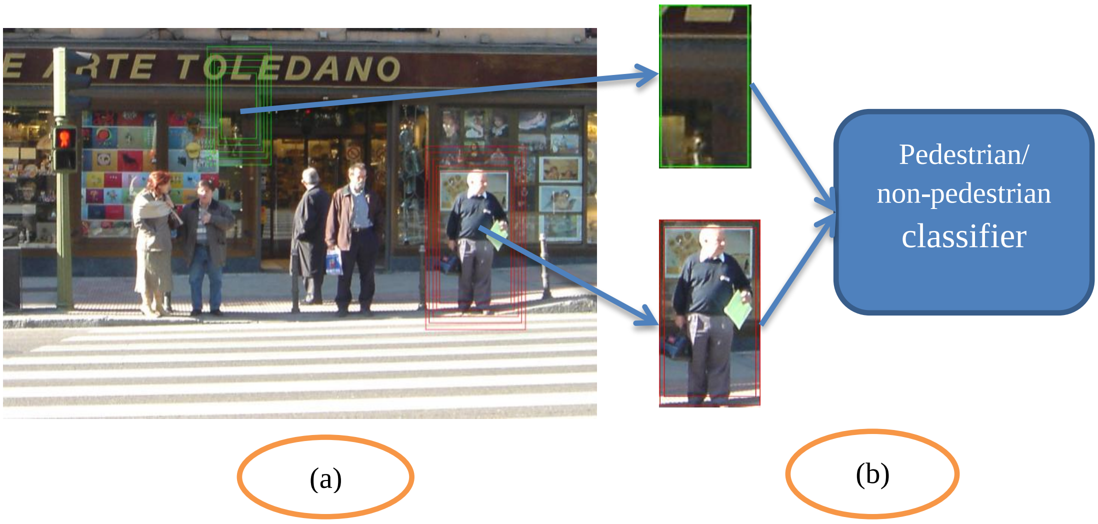


Figure 3: Overview of detection method

(a) At each point in image, we densely scan with multi-scale. (b) Sliding window is detected and extracted to feature vector which is input to pedestrian/non-pedestrian classifier.

We just focus on method of represent robust features in order to robust from slightly changes in shape, illumination and scale. The classifier used in this thesis is Linear Support Vector Machine (stand for SVM) [13,14,15]. Recently, SVM have widely used in machine learning. And in computer vision, it is intensively used in learning process. We use SVM because it is simple, runs fast, and has good performance.

For more specific, in extracting feature process, we use locally normalized Histogram of Oriented Gradients (HOG) as a descriptor [16]. HOG is computed from gradients of image and has the characteristic that robust to (1) small changes in image contour locations and directions, (2) significant change in image illumination and color, (3) remaining as discriminative and reparable as possible. We use weighted histograms gradient orientations over spatial neighborhood [16,17,18,19,20] to calculate HOG features. Before calculate histogram of gradients, we do some pre-process to eliminate the effects of illumination and color changes. So, the histogram of oriented gradients has information of the contour of the object.

Once we densely scan image, we will get a bulk of windows at level classifier which means that each window is now represented as high dimensional feature vector. Note that we scan all position at multiple scales, so there are probably some windows overlap each other. After that, we suppress all window whose score below the threshold, and keep and positive windows (exceed the threshold). Because HOG is robust to slight changes in shape and contour, it is possible to have many positive windows contain same object. To resolve this problem, we fuse all positive ones and use non-maxima suppression [23,24] to find only one window most likely contains object of a class. In this thesis, Mean Shift [76,77] is used as a suppression algorithm in this process.

.

1.6. Summary of Contributions

This thesis investigates in field of finding the robust and fast descriptor based on the Histogram of Oriented Gradients method. These following contributions are:

Firstly, we re-implemented the work of HOG and got similar performance. The second contribution is that we propose several methods of reducing the length of feature vector, while maintain the good performance. The third one is that we get better performance than HOG original method by adding more layers. In our experiments, we find out that the performance is enhanced by 3% if we add two more layers in the encoding feature vector. The details of this method will be described in section 5.3.

1.7. Outline

This chapter gives the overall object detection problem, and describes the applications as well as the goal of this thesis. In addition, in this chapter we also give a brief introduction about computer vision, object detection background, outline approach to the problem, and some contributions. The remaining chapters are organized as follows:

- Chapter 2: we will review the state of the art in object detection field, focusing particularly on pedestrian detection
- Chapter 3: we present the overview of HOG approach to object detection. We have not given the details implementation yet, but we describe the overall detection framework and give an overview of key experimental results.
- Chapter 4: describe in details the computation of HOG feature vectors. Different types of HOG are proposed and discussed in this section. And we take "pedestrian" object as a test case. Moreover, we define two more models which are the improvement of HOG original descriptor. This section studies the weakness of the original HOG in order to find the way to enhance its performance. So, we propose three approaches. In the first one, we observe that the center of image is less informative than the others, so we ignore it. The result is that computing time is significantly reduced, while the performance is remained. The final method is the one that employs multiple

levels which have more information about the shape and structure of object. The experiment shows that the performance is enhanced approximately by 3%. However, the cost of this method is the increment in feature vector length.

- Chapter 5: fusion overlapping detections algorithm is described in details. This chapter will explain how we choose Mean Shift algorithm, parameters and its effect. In addition, we also explain why Mean Shift is suitable for densely representative encoding descriptor. To deal with partly invisible object or crowded objects, we utilize some tricks to deceive them.
- Chapter 6: conclusions will be presented in this chapter. We review our main contributions as well as show the limitations. We also propose some promised future works in this chapter.

Chapter 2. Related work

As mentioned in chapter 1, object detection which can be used for many potential applications has drawn a lot of attention of computer researchers. In this, we will review some well-known automatic object detection and localization, with particular attention on human detection.

According to most well-known works on object detection, it can be viewed as the combination of two stages: the image descriptor or feature vectors that they use and the detection framework which is built over these descriptors.

The following sections will present some related works which work on these categories. Section 2.1 provides some different image feature sets that have been used in human detection. Section 2.2 gives some useful binary classification methods used extensively in object/non-object decision. Section 2.3 mentions a little bit about fusion methods, and section 2.4 is about our motivation when choose image feature sets in this thesis.

2.1. Image features

As human’s perspective, a certain kind of object is represented as shape and texture; and the process of distinguishing one type of object with the others is not the trivial process. A child, for example, at the beginning of life, has been taught several years to distinguish between duck and chicken. Similarly, the learning process of computer shares some identical aspects. Representation of image region in order to guide computer distinguish object/non-object region is not the trivial one. Due to the importance of image representation or sometimes called image features, the research in this field is very active.

In image feature sets, most relevant features for object detection or classification which provide invariance to illumination changes, different of viewpoint and shifts in object contours are extracted. To achieve this, instead of directly using raw image intensities or gradients, more advanced methods usually used which called local image descriptors. Such feature sets can be based on points [25,26], blobs [27,28], intensities [29,30,31], gradients [32,33], or combinations of several or all of these [34]

Despite of the diversity of image feature representation, people usually divide them into two broad categories: sparse representations base on points, blobs, image fragments or part detectors; and dense representations using image raw intensities, gradients, or order of pixels.

2.1.1. Spare Local Representations

Spare local representations are based on the local descriptors of possibly most relevant local image regions. These regions can be salient point such as blobs or parts detectors.

Key Point Detectors

Key point detectors have a long history [35→45]. This approach is simulated by the observation process of human beings [5,6,7]. So, key point detectors' hypothesis is that the selected points are stable and reliable image regions, which are especially informative about local image content. Representations using key point extract local image feature at a spare set of salient and most valuable image points, which are sometimes called interest points. Then, the final detectors are based on feature vectors computed from these interest points. Overall, the performance of final detector is strongly depended on reliability, accuracy, and repeatability with which these interest points can be found in the given object class and the informativeness of these interest points. Many type of key point detector has been being invented such as Forstner-Harris [46,47,25], Laplacian of Gaussian [27], Difference of Gaussian [42], and scale-invariant Harris-Laplace [50]

Some advanced key point detectors such as DoG or Harris-Laplace add some additional aspects such as local scale and dominant orientation information. In the past, the approaches which utilize salient and informative image regions as key points have attracted many attention because they has some advantages which are discussed below.

Obviously, compactness of representation is the first benefit. It is for sure that there are many fewer key point descriptors than image pixels, so it enables us to speed up the latter stage of classification process. Secondly, it's robust to scale and rotation of objects. Thirdly, point detectors are based on a very strong hypothesis which is salient image regions are most valuable and repeatable one [51].

However, there are some inherent drawbacks. Note that most key point detectors are designed to be repeatedly on particular objects and may have limitations when generalizing to object classes or categories. This means that they may not be repeatable for general object classes. In addition, one key point detector is designed to suitable to particular object class, but not all. This means that a key point detector may work very well for one certain object, but probably has a very poor result for the other object classes.

After detecting key points, the second stage is the computation of feature vectors over local image surroundings of key points. Among enormous approaches trying to solve this issue, there are two popular ones which called *Scale Invariant Feature Transformation* (SIFT) [19,21,28,38,42] and *shape context* [20]. Both are image gradient based descriptors which compute histograms of image gradients or edges. SIFT use the local scale and dominant orientation from selected key point detector to vote into orientation histograms with weighting based on gradient magnitudes. SIFT descriptor computes histograms over rectangular grids, while shape contexts utilizes log-polar grids.

2.1.2. Dense Representation of Image Regions

Instead of using spare representations, the other approach called dense representation can be alternative. It extracts image features densely over an entire image or detection window and to collect them into high dimensional descriptor vector which can be used for image classification or object/non-object window decision. Typically, the dense representation is based on image intensities, gradients, or higher order differential operators.

Edge and Gradient Based Detectors

Another method use gradient and edge to get object information. A well-known approach is the pedestrian detection system invented by Gavrila and Philomin [22] which extract edge images and match them to a set of learned exemplars using chamfer distance. Besides, gradient image descriptors have been used by Ronfard et al [52] and Mikolajczyk et al [33] to build an automatic body detector. Similarly, Felzenszwalb and Huttenlocher [54], Ioffe and Forsyth [55], and Mikolajczyk et al [33] design human detection system which propose 7 parts detectors: head, upper body and legs.

Order of Pixel Detector

Order of pixel is an alternative approach of extracting local image regions to feature vectors. And Local Binary Pattern (LBP) [56 → 63] is the most famous method in this field. Local Binary Patterns is a type of feature used for classification in computer vision. LBP was first described in 1994 by T. Ojala, M. Pietikäinen, and D. Harwood. It has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) classifier, it yields the best classifier of humans and many other object classes [Xiaoyu Wang, Tony X. Han, Shuicheng Yan, ICCV 2009] [64,65].

The concept of Local Binary Pattern is really simple. It is based on the order of pixel’s surroundings. Eight neighborhood pixels’ intensities are compared with center pixel. Where the center pixel's intensity is greater than the neighbor, write "1", otherwise, write "0". This gives an 8-digit binary number which is usually converted to decimal for convenience. After that, the histogram is computed by the frequency of each number occurring. Recently, the augmentation of LBP which is Local Ternary Pattern (LTP) was developed and it shows a better performance.

Hybrid detector

As mentioned a little bit above, the Hybrid model detector consisting of LBP and HOG give the state of the art in object detection and classification [64,65]. HOG and LBP descriptor are independently extracted, and then they are combined together. But before the combination, the LBP feature vector need to be carefully pruned to eliminate useless dimensions as well as reduce the time consuming in second phase of hybrid detector.

2.2. Classification Methods

In reality, there are enormous machine learning methods (or sometimes substituted by classification methods). The first recognized one is developed by Arthur Samuel [66]. After that, this challenge field has attracted many researches and diversity of machine learning methods have been designed which include statistical learning, decision tree learning, PAC learning and so on.

Classification methods of local part descriptors can be divided into two categories which are discriminative approaches such as Support Vector Machines (SVMs) and generative approaches like graphical models. In this section, we just focus on the discriminative one.

2.2.1. Discriminative Approaches

In computer vision field, SVM and AdaBoost are two most popular classifiers among enormous ones due to their abilities to automatically select relevant features from large feature sets. In this section, we will examine these classifiers.

2.2.1.1. Support Vector Machine Classifier (SVM classifier)

In last decade, SVMs classifier has been used widely for object recognition and classification. The essence of SVM is to find the hyper-plane that separates and maximizes the margin between the object and non-object class. The simplest form of SVM is linear SVM which is very simple and efficient in term of computing cost, while its performance is a bit low because feature image sets are rarely linearly separated. In the past few years ago, non-linear or kernel SVMs has drawn a lot of researchers' interest [53,68,69]. Moreover, Hichem Sahbi [67] has designed a context dependent kernel which is based on characters of particular object to enhance the classification process.

2.2.1.2. Cascaded AdaBoost

The core idea of AdaBoost (adaptive boosting) is that many week classifiers are combined to form a strong one. The term "cascade" reveals the one thing that at each level the most relevant features will be maintained, while the irrelevant ones will be eliminated. Despite of time consuming in training process of Cascaded AdaBoost, it provide significant improvement compared to SVMs in term of final classifier.

As mentioned, Viola and Jones [70], Viola et al [71] use AdaBoost to train cascades of weak classifiers for face and pedestrian detection. Opelt et al [43] use AdaBoost framework for interest point based weak classifiers. Schneiderman and Kanade [72] designed a more elaborate model in which they define parts as functions of specific groups of wavelet coefficients, represented with respect to a common coordinate framework. In this one, geometric relationships between parts are implicitly captured.

Recently, Zhu et al [73] used cascade of rejecters based approach to speed up detector using HOG descriptor. They use an integral array representation [Viola and Jones 2001] and AdaBoost to achieve significant improvement in running time, while maintain the same performance as original HOG work.

2.3. Motivations of Employment HOG Methods

Feature extraction has two popular approaches based on the inside characteristics of object and statistic. In the first catalogue, many well-known feature sets such as Harris [25] and LoG [27] are unable to cope with the sheer variation in human clothing, appearance, and articulation [48]. The approaches to resolve person detection via edge detection also have attracted many researchers.

But Leibe [49] states that these detectors do not perform well for natural images. On the other hand, the second catalogue which is statistical approach has recently used very much. The statistical approaches such as HOG has outstanding performance to the first catalogue one such as SIFT, Harris [16].

Non-man-made objects have diverse shapes in reality, but they are usually rigid in some finite shapes. For example, the most common shape of people is up-right in outdoor, lie down in bed room, and sit down in living room. Hence, by taking these rigid shapes of object, we can use the statistical model to detect them. Instead of exhausting effort to find salient regions and extract most relevant feature for classifier like SIFT [19,21] or LoG [27], we can get object's shape and distinct information through statistic. Recently, some statistical approach has got the brilliant result [16, 65] , this is the one inspires us to explore in this field for object detection.

Our purpose is to take advantage of rigid shape of object in order to build a robust detection algorithm. Hence, we explore and study deeply inside about HOG [16] to get its insight. Not stop at this stage, we propose some improvements to reinforce the strength of HOG, as well as eliminate its fragile. Our contributions are on two folds. Firstly, we ignore the less informative region in image window to shrink feature vector's length, while remain same performance. Secondly, we define new levels of image window, getting more information about shape and contour to enhance the performance.

Chapter 3. Overview of Detection Methodology and results

In this thesis, we use a method to map local image regions to high dimensional feature spaces. To encode the static image, we use HOG approach which heavily base on image gradients. The following sections will describe in more details of detector framework of HOG.

3.1. Overall framework

Our object detector consists of two main phases called training phases and detection phase. In training phase, we use training dataset to create binary classifier which provides object/non-object decision for fixed size image's region (usually call window). And in detection phase, we densely scan whole image at multiple scale and use the classifier derived from training phase to explore positive region in test image (positive region is the one that likely contains objects). After receiving all positive windows, they are fused together to have final detections by non-maxima suppression algorithm [76]. The performance of final step is mostly depended on the reliability and robustness of classifier.

3.1.1. Training phase

The training set consists of positive and negative windows which have the fixed and same sizes. Positive training windows contain object at the center, while negative training windows are the arbitrary sub-sample that does not contain any instance of object [16]. So the number of positive training sample is very limited. On the other side, the number of negative training sample can be very huge because for example, one natural scene image can generate approximately 10,000 negative windows.

As the framework in figure 4, positive and negative training windows are collected to prepare for the first training phase. These windows are then extracted using Histogram of Oriented Gradients [16](see details in Chapter 4) to map each training window to dimensional feature vector. After that, some Machine Learning techniques [79] are used to create a binary classifier from these vectors (in the figure 4, the flow work is (1) + (2) → (3) → (4) → (5) → (6)). However, this classifier cannot be applied right now due to its sensitivity to false positive rate. The reason is that the negative windows set used in first learning phase is very small sub-set of negative windows in all negative images in dataset.

Hard examples

The full name of hard examples is hard negative examples which are the false positive windows in negative images. We densely scan through all negative images in dataset and use above binary classifier to find hard negative examples. And then, these hard ones will be push to the initial training negative windows for the second learning phase (in the figure 4, the flow work is (11) → (6) → (8) →(1) →(3) → (4) → (5) → (7)). The purposes of generating hard examples are in two folds. Firstly, it greatly reduces the false positive rate, which enhances 7% at 10^{-4} FPPW (Fig. 5). Secondly, it is impossible to load all high dimensional vectors of positive and negative examples (usually millions of windows) to RAM, so we have to get hard examples for second learning.

However, there are two main drawbacks of this process. Firstly, it takes considerable time for finding out hard negative samples. According to our experiment, the more hard samples we collect, the better performance is. However, due to the enormous number of windows in image, the total number of windows which we have to scan through is very huge. From the pedestrian INRIA dataset, we have to examine approximately 2,000,000 windows to draw a few thousands of hard samples, and it takes us nearly 3 days. Secondly, while we gather hard negative samples and put them into negative training set, we also increase the miss rate of detector.

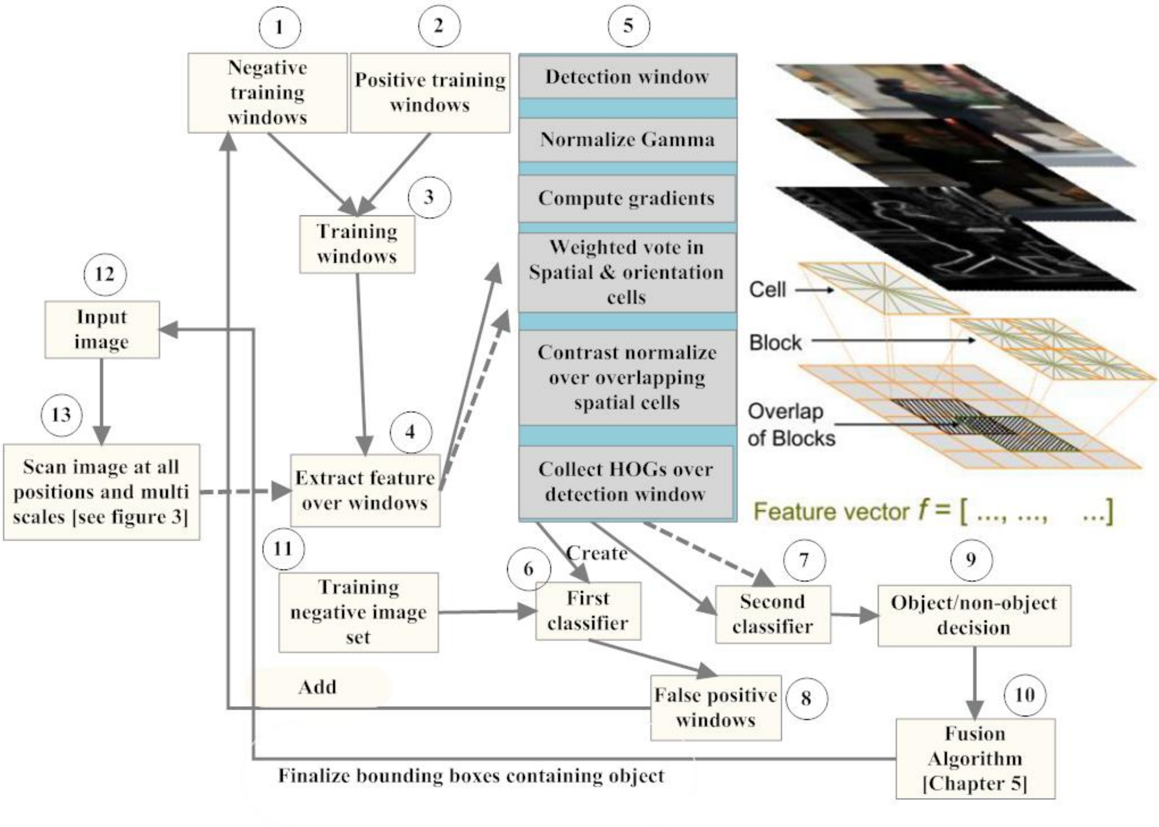

1
Negative training windows
2
Positive training windows
3
Training windows
5
Detection window
Normalize Gamma
Compute gradients
Weighted vote in Spatial & orientation cells
Contrast normalize over overlapping spatial cells
Collect HOGs over detection window
Cell
Block
Overlap of Blocks
Feature vector f = [..., ..., ...]
12
Input image
13
Scan image at all positions and multi scales [see figure 3]
4
Extract feature over windows
11
Training negative image set
Create
6
First classifier
7
Second classifier
9
Object/non-object decision
Add
False positive windows
8
10
Fusion Algorithm [Chapter 5]
Finalize bounding boxes containing object

Figure 4: Overview of framework

Input: normalized and fixed resolution positive windows, and negative training images Output: the trained binary classifier for object/non-object on resolution image windows
First phase learning: • Create initial negative examples once by randomly selecting windows locations on each negative image at different scale. • Calculate descriptor vector of all positive and negative training windows • Learn a linear SVM classifier on these supplied descriptor vector
Generate hard negative examples:

• In the negative image set, randomly select many windows of multi-scale at random locations. • For each scale (window at certain scale and location), we do • Rescale the input window as resolution positive windows size. • Apply encoding algorithm and use the classifier in first phase learning for object/non-object decision. • Push all the detections with score larger than 0 (i.e hard examples) to a list
Second phase learning • Push all hard examples to the initial negative set • Learn a linear SVM on the new dataset.

Table 1: Algorithm of generating hard examples

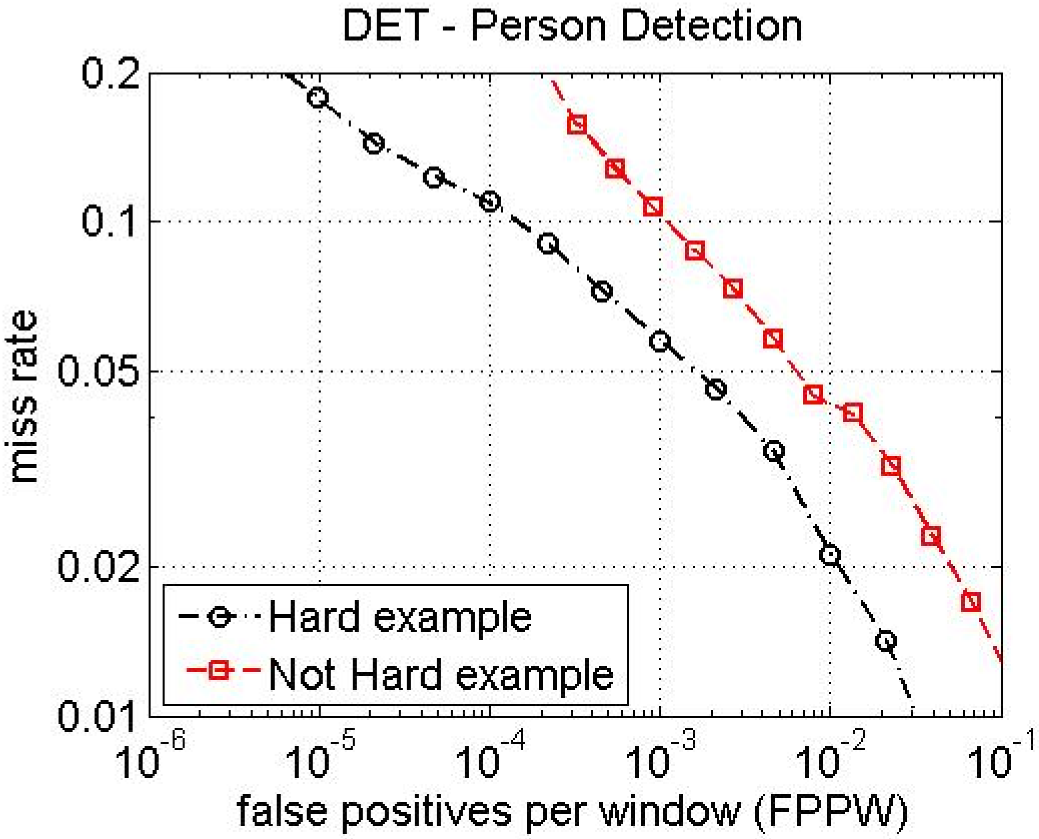


Figure 5: The performance of detector after using Hard examples procedure is better than detector does not use hard example procedure

3.1.2. Detection phase

The goal of this phase to find out all positive windows of given test image. During detection procedure, the given test image is densely scanned at all scales and locations. ((in the figure 4, the flow work is (12) → (13) → (4) → (5) → (7) → (9) → (10) →(12)). For each scale and location (called window), we compute the feature vector of detection window. And we use the classifier derived from training phase to make the decision of whether this window is positive or negative (contain object or non-object). Because it is possible that many overlapping positive windows contain same object, it is necessary to fuse all detection windows to find the final ones.

3.2. Overview of feature sets

The feature sets used in this thesis based on dense and overlapping encoding of image regions using Histogram of Oriented Gradients descriptor. This descriptor is a statistical approach which regards the orientation of gradients in image. Dalal has proposed two types of HOG called static and motion HOG [16]. Static HOG descriptor is used to extract image region feature, while motion HOG one is used in video. And because our target is to detect object in static image, we will use static HOG descriptor to extract image regions characteristics. In this thesis, we use term HOG instead of static HOG to indicate static HOG descriptor.

HOG which is based on the characteristics of well-normalized local histogram of orientation of gradients will be described by following steps (block 5 in figure 4). Apply normalization to the image to reduce the influence of illumination effects. In this project, we use square root method to each color channel. By observation, normalize image by square root has increased the performance a lot by prevent the effects of shadow and illumination. In the second step, we compute the first order image gradients. These gradients contain information of contour and some texture of object. In addition, gradient is resistant on illumination and color variation. Once completely compute first order image gradients of each channel, we choose the dominant color channel. After that, like SIFT descriptor; local image region is encoded into high dimensional vector by concatenate many local spatial histograms of gradients. Image window is divided into small non-overlapping regions called “cell”. For each cell, we compute the histogram of gradients over all pixels in the cell by accumulating the magnitude of each pixel gradient into bins which are the range of orientation of gradients. The detail will be described in Chapter 4.

After receiving histograms of each cell, we take a local group of cells and normalize them. This normalization step will help the feature vector resists to variation of illumination, shadowing, and edge contrast. The group of local cells is call “block”. In this stage, many blocks can be overlapping each other, so they share some same cells. This seems redundant, but in practice this can enhance the performance of descriptor because this gives us more information about image region.

Finally, collect all HOG descriptors from all dense overlapping blocks of detection window into big feature vector for use in the window classifier.

The HOG descriptor has several advantages such as the followings. First of all, it captures the contour information. For example, in HOG descriptor, information of edge or shape of object is stored in histogram of cells and blocks. So HOG contains the characteristic of local shape. In addition, when put together all overlapping blocks, we probably get relevant information while still maintain invariant. Next, when object translates or changes a little bit, it make little different in histogram if these changes are smaller than the local spatial or orientation bin size. The third one is that illumination invariant is assured by gamma normalization and contrast normalization. Finally, the overlapping blocks has a benefit that it allows little information can be missed during the encoding process.

3.3. Fusion of multiple detections

In the detection phase, image is densely scanned at all locations and scales. This probably creates a lot of overlapping detections for one instance of object. The reason is that a detection window probably gets positive score although it is slight off object center. So, the detection windows need to be fused together. Intuitively, during the detection phase, we observe that although the number of detection windows is much larger than the number of instance object, these windows are most likely concentrate around objects. Hence, from this observation, we can employ clustering algorithm to find the right position of instance object. There are two well-know and traditional cluster algorithm which are K-mean [80] and mean shift. And then, we decide to use Mean Shift because the number of object in image is unknown.

3.4. Classification Methods

There are three method are put in our consideration. Initially, we thought that we can employ maximum likelihood paradigm to create a linear separator between positive and negative samples, but we find out that it is unrealistic to do that because the derived vector is huge, over than 2,000 dimensions. In maximum likelihood method, we have to calculate inverse matrix, so it is impossible for moderate computer to find inverse matrix of huge one in short period of time.

The other two learning methods are Ada-boost and linear SVM which are all common. When doing this thesis, our main target is to investigate and find out the solid descriptor that can transform image region to vector. So, we choose linear SVM because it is simple and reliable classifier. There are three properties of linear SVM which make it valuable are: (1) it converges reliably and repeatedly during training process, (2) it handles large dataset gracefully, (3) and it has good robustness towards different choices of feature sets and parameters.

Support vector machines are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier. Since an SVM is a classifier, then given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

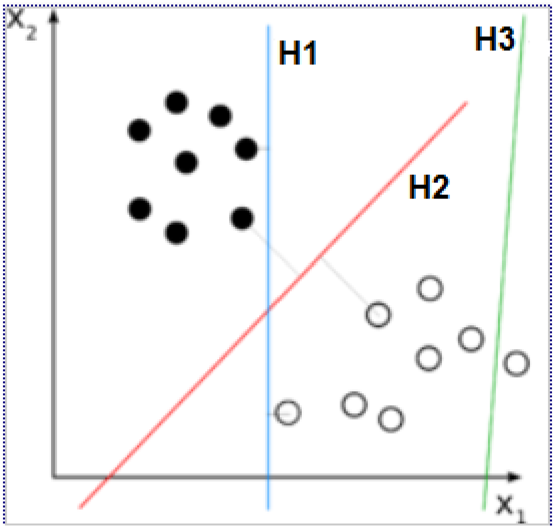


Figure 7: H3 (green) does not separate two classes. H1 (blue) does with small margin, H2 (red) with maximum margin.

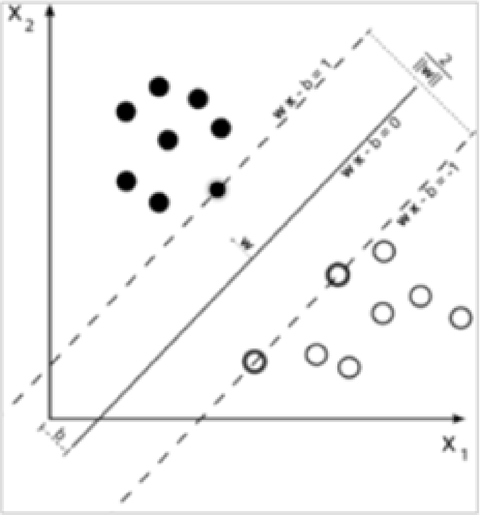


Figure 6: maximum margin hyperplane and margins for an SVM trained with samples from two classes

We use soft linear SVM with default c = 0.01. Non-linear SVM enhances the performance by 3% at 10^{-4} FPPW, but the tradeoff is that it takes much more time in computation [81].

3.5. Overview of results

The main purpose of this thesis is to rebuild the HOG descriptor and improve some aspects in HOG, so we will compare our results with Dalal's work.

At first, we re-implement HOG detector of Dalal, and its performance is comparable with Dalal's one. After that, we propose some slight contributions which are reduction of dimension of feature vector and increasing performance by adding multi-level.

The first contribution is that we try to reduce dimension of feature vector, and we get the performance still as good as Dalal's one. In the second contribution, we enhance HOG detector performance by adding some information; and its performance are higher than original one approximately 2%. These contributions will be discussed in detail later in section 5.

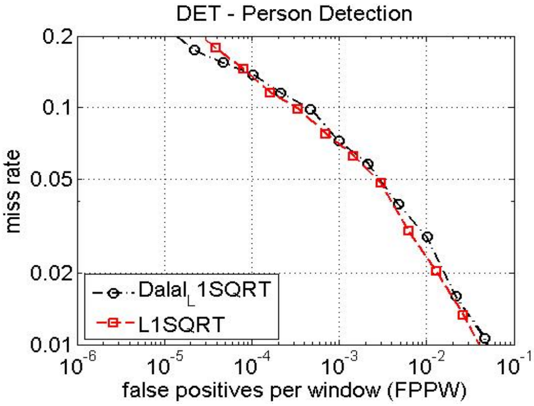


Figure 9: Our performance (red curve) is approximately as good as Dalal's.

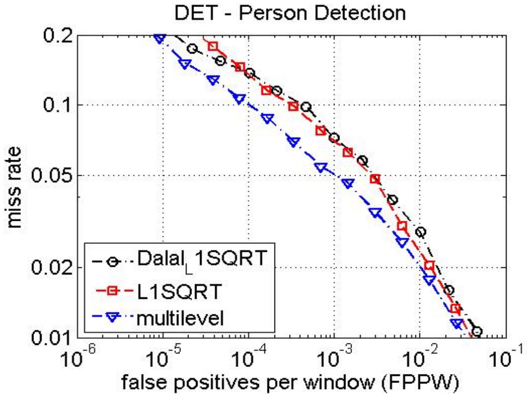


Figure 8: The performance of multi-level is higher by 2-3% than the original at 10-4 FPPW.

Chapter 4. Histogram of Oriented Gradients

In this section, we are going to review HOG feature sets. There are a lot of parameters in HOG feature, and their effects are very different. For example, some parameters slightly effect on HOG performance, while other some ones considerably influence on the efficiency of HOG. Moreover, we define two new approaches based on HOG, which are the improvements of original HOG. Overall, one conclusion we can draw out is that HOG encoding feature sets gives outstanding performance compared with other existing method such as SIFT, Haar wavelets.

4.1. Static HOG Descriptor

Histogram of Oriented Gradients is indeed the dense and overlapping description of image region. There are four HOG variants which are followings.

4.1.1. Rectangle HOG(R-HOG)

R-HOG looks like SIFT descriptor, blocks use overlapping square/rectangle grids of cells. The descriptor blocks are computed over the dense uniform grids. And each block is normalized independently. The parameters of R-HOG descriptor are ς x ς, η x η, β which are number of cells in one block, number of pixels in one cell, and number of bins respectively.

4.1.2. Circular HOG(C-HOG)

C-HOG seems to be similar to Shape-Context. In C-HOG, cells are defined into grids of log-polar shape instead of square or rectangle. At each center of grid point, we divide local image patch into a number of angular and radial bins. The angular bins are uniformly distributed over the circle, and bins will be increased as big as they are far from the center.

4.1.3. Bar HOG

Bar HOG is similar to HOG, but it also uses second order derivative instead of first derivative. After that, we collect histograms of both first and second order derivative. The advantage of this approach is that Bar HOG has additional information about bar and blob.

4.1.4. Center-Surround HOG

In R-HOG and C-HOG, each block is normalized independently, so one cell can be normalized redundantly. It seems that optimal computation cost will not reach. In order to overcome this issue, in Center-Surround HOG, every cell is normalized just only one time. So that it speeds up computation.

4.2. Implement and Performance Study

In this thesis, we choose R-HOG as our default descriptor because of shortage of time and its excellent performance. We now describe details about how to implement R-HOG as well as give out the effects of parameters. For all experiments, we use Detection Error Tradeoff Curve to show the performance.

As mentioned, there are several variants influence on HOG description performance. In this section, we are going to describe effects of main factors.

4.2.1. Color channel

This section will give the evaluation of pixel representations including gray-scale, RGB, LAB, and HSV. According to our experiments, the performances of RGB and LAB are similar and they are outstanding the rest. While the pure gray-scale and HSV reduces the performance 2% at 10^{-4} FPPW (false positive per window).

4.2.2. Color/Gamma Normalization

As mentioned above, images of dataset have to be pre-implemented before encoding HOG feature vector. A raw test image can be beautiful with human’s viewpoint, but it can be very difficult and vague for computer’s perspective due to effects of illumination and shadowing. Hence, normalizing gamma and contrast of image is necessary. There are two popular normalization methods which are “square root” and “log scale”. Their results are similar. In this thesis, we “square root” method because it is faster than the other. By experiment, performance will be boosted by 7% when using “square root” normalization method.

4.2.3. Gradient Computation

Gradient is the term that indicates the change of pixel in image. Hence, by employing gradient information, it allows us to get and encode the shape and contour of object in image.

We compute gradient by calculating first order derivative of pixels in image. In computer field, there are several ways to estimate the changes of pixels in image. There are plenty of masks used in convolution of image. However, the simple mask [-1 0 1] give the best outcome according to experiment.

Compute first order derivative of each pixel on Ox, Oy coordinates:

One sided: $S_x = f'(x) = lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$

$$S_y = f'(y) = lim_{h\to 0}\frac{f(y+h)-f(y)}{h}$$

Corresponding mask:

-1	1

Two sided: $S_x = f'(x) = lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h}$

$$S_y = f'(y) = lim_{h\to 0}\frac{f(y+h)-f(y-h)}{2h}$$

-1	0	1

Corresponding mask:

Note: ‘h’ is usually taken as 1.

Calculate Gradient: after getting S_x , S_y which are two first order derivatives on Ox, Oy coordinates respectively, we can use them to calculate the magnitude and orientation of pixel.

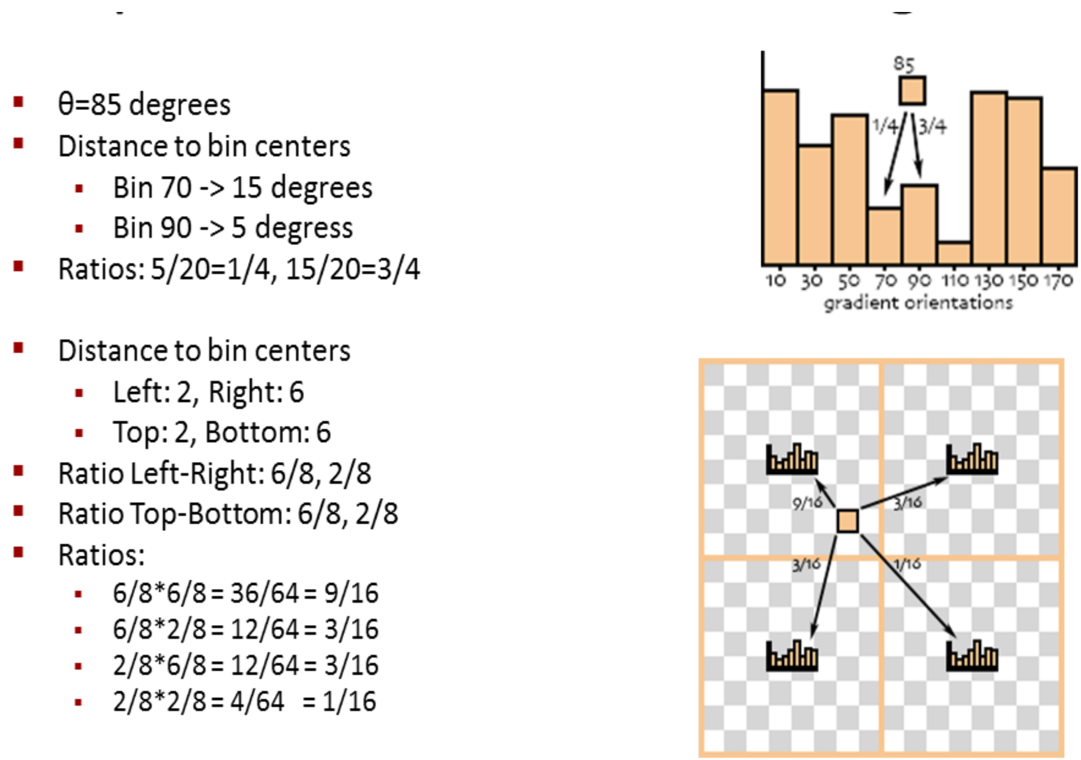
- Magnitude: $S = \sqrt{{S_x}^2 + {S_y}^2}$
- Orientation: $\theta = \arctan(\frac{S_y}{S_x})$

And one importance notice is that we definitely should not smooth or blur image before computing gradients. The most likely reason is that edge informative is essential to descriptor, and if we blur image, we lose a lot of edge information.

4.2.4. Spatial/Orientation Binning

After calculating gradients, we will get a peck of gradients of pixels which consist of magnitude and orientation. And each gradient contributes a weighted vote for orientation based on the orientation of gradient itself. The orientation bins of cell will be accumulated by the weighted vote of its pixel gradients. The orientation bins can be over 0-180^{o} (“unsigned” gradient) or 0-360 (“signed” gradient). Moreover, in order to avoid bias, we use tri-linearly interpolation voting method which is regards orientation and position matters to vote to cell bins. This idea is illustrated by below figure 10.

The number of bin of histogram of cell is also the factor effect a lot to performance. The performance of $\beta=9$ is significant better than of $\beta<9$ (β is the number of bins of histogram in cell) [82]. However, performance will not increase much when β exceed 9.



- θ=85 degrees
- Distance to bin centers
  - Bin 70 -> 15 degrees
  - Bin 90 -> 5 degress
- Ratios: 5/20=1/4, 15/20=3/4

- Distance to bin centers
  - Left: 2, Right: 6
  - Top: 2, Bottom: 6
- Ratio Left-Right: 6/8, 2/8
- Ratio Top-Bottom: 6/8, 2/8
- Ratios:
  - 6/8*6/8 = 36/64 = 9/16
  - 6/8*2/8 = 12/64 = 3/16
  - 2/8*6/8 = 12/64 = 3/16
  - 2/8*2/8 = 4/64 = 1/16

Figure 10: Overview of trilinear interpolation

A gradient with angle θ= 85^0, it will affect 2 bins: bin 90 and bin 70. Because the distance from 85^0 to 90^0 is nearer than the distance from 85^0 to 70^0 so that bin 90 will be affected more than bin 70. And the ratios of affection are:$\frac{90-85}{20} = \frac{1}{4}$ (bin 70) and $\frac{85-70}{20} = \frac{3}{4}$ (bin 90)

A gradient of a pixel will affect to all nearest cells around it. Like the example above, a gradient of pixel affects 4 cells. The ratio of affection of a cell is larger if the distance from the pixel to this cell is smaller.

"Signed" or "Unsigned" gradient is also the matter put into concern. The natural object such as human, cat dog can be diverse in shape and contour. Hence, "signed" gradient is unsuitable to be used because this probably reduces the performance [82]. At this circumstance, "unsigned" gradient give best results. In return, “signed” gradient gives very good performance for objects which are man-made because their shapes are likely constant.

4.2.5. Block Normalization, Block Size, and Overlap

Block Normalization has a great effect on HOG descriptor. The fact that Gradients strengths vary from over a wide range due to local variations in illumination and foreground-background contrast. Hence, it is necessary to normalize block to get good performance. Block is a local group of cells, and each block is normalized separately.

We will evaluate four different block normalization types. Let **v** be the un-normalized block descriptor vector, and ε is a very small number employed to avoid division by zero. The four normalization types are:

- L_2 norm: $\mathbf{v} \leftarrow \mathrm{sqrt}(|\mathbf{v}|_2^2 + \varepsilon^2)$
- L_2-Hys: L_2-norm by clipping and renormalizing.
- L_1-norm: $\mathbf{v} \leftarrow \mathbf{v}/(|\mathbf{v}|_1 + \varepsilon)$
- L_1-sqrt: $v \leftarrow \mathrm{sqrt}(\mathbf{v}/(|\mathbf{v}|_1 + \varepsilon))$

For pedestrian detection, the performance of L_2-Hys and L_1-sqrt are equal, and they are outstanding to the rest. By experiment, using L_1-norm reduces performance 15%. And the value ε should be taken in range $1e^{-3} – 5e^{-2}$.

We are now investigating the effect of block size. For pedestrian detection, 3x3 cell blocks and 6x6 pixel cells gives best result with 11% miss rate at $1e^{-4}$ FPPW [83]. According to Dalal’s experiment, cell size varies form 6 x 6 to 8 x 8 and block size varies form 2 x 2 to 3 x 3 gives the best performance for all kind of objects. The most likely reason is that if the size of block or cell is too big or too small, the valuable spatial information will be lost.

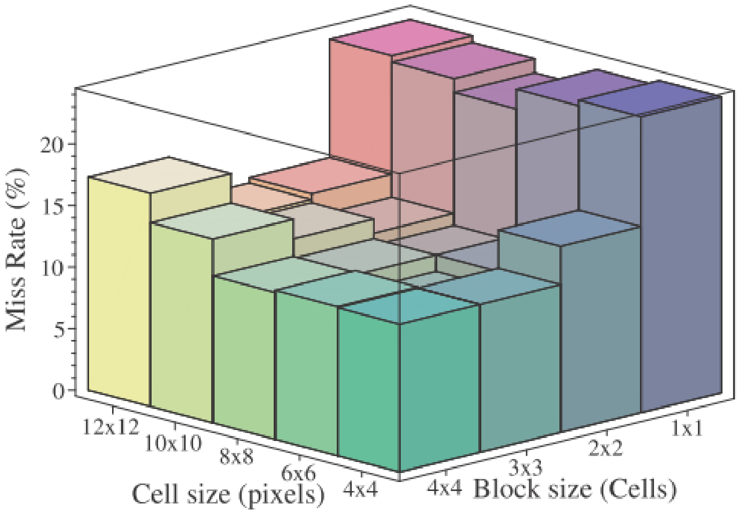


Figure 11: The effect of cell size and block size to the overall performance. From this figure, we see that cell size is 6x6 and block size is 3x3 give the best result

4.3. Visual Curve of Person Detection

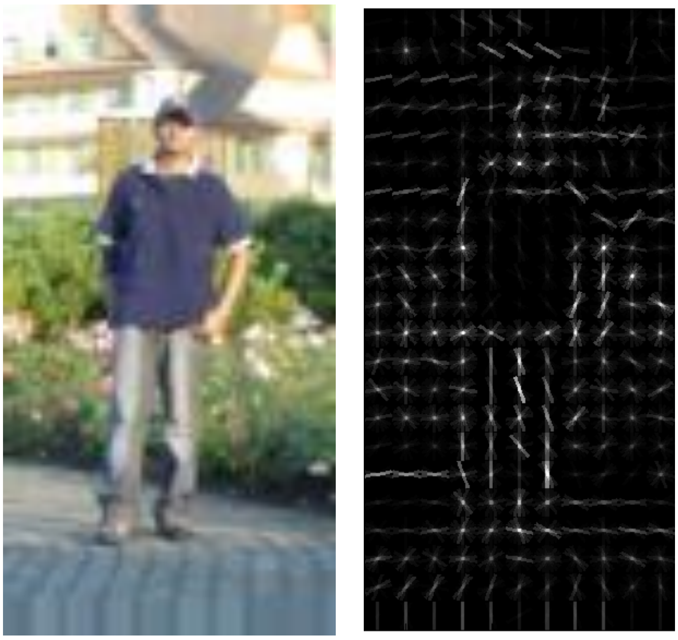


Figure 12: Visual curve of person. (a) test image. (b) image’s HOG descriptor.

Intuition about HOG characteristics is reassured with Figure 12. Fig. 12(b) suggests that person’s head, shoulders, and legs and the points of intersection between feet and ground are most likely to be relevant curve for classification.

4.4. HOG Based Models

In section 4.2, we see that the more overlap of blocks, the better performance is. However, the block overlapping accompanies with the size of feature vector. Hence, if we enhance performance by increasing blocks overlapping, we will also reduce the program performance because of the expanding of feature vector. Thus, we propose some models to reduce the length of HOG feature vector. In addition, we also introduce the method called “multi-level” to increase the performance of HOG descriptor.

4.4.1. Spatial-Selective Approach

This approach is only useful when the hypothesis which is the pedestrian is up-right shape and central alignment is hold. This assumption is assured in MIT and INRIA pedestrian dataset. We observe that there is a small region in the center of window which mostly contains chest and stomach is less informative. The reason is that this region usually falls into internal part of pedestrian’s body. Hence, this small region is often covered by colors of cloths without shape curve information.

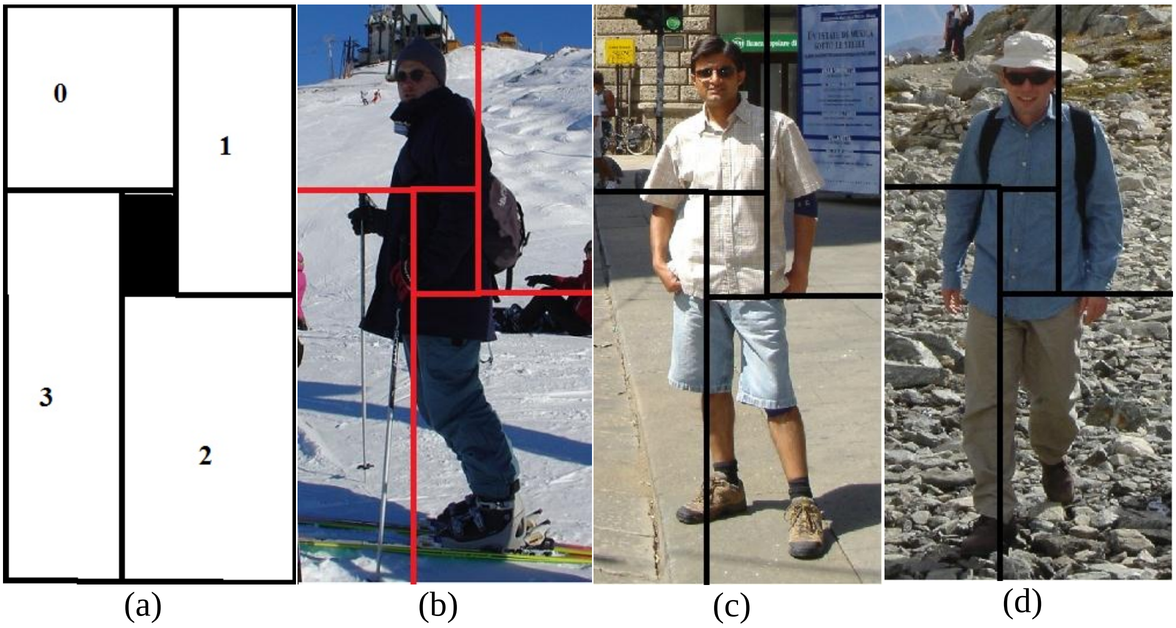


Figure 13:Visual idea of spatial selective approach. The fist figure is the mask which is applied to image windows. The black-center region is ignored because it is just uniform as clothing. The remaining four regions are independently extracted to feature vector, and then they are combined together. (a) the mask. (b,c,d) mask is applied to windows.

In this model, image is divided into five regions which are (0), (1), (2), (3), and center-black one (Fig. 13). Usually, region (0) is the most valuable because it contains the head and left shoulder. Region (1) occupies right shoulder, while region (2) has information of legs. The (3) one is unreliable because sometimes it does not have any object information. When we test these four parts independently, their performance of them is extremely low because they lack of whole object information. However, we observe that the performance of part/region (0) and (2) are moderate and much better than the rests. This is also the proof of the assumption which states that head, shoulder and legs are important parts.

By experiment, we observe that "black center" region should be less than 18.25% of window size (one and a half cell over eight cells) in order to maintain the same accuracy with original one. However, if we choose "black center" region too small, the size of feature vector of window will increase correspondently.

One more thing that significantly affects performance is the overlap of regions. The more these regions overlap to each other, the more accuracy it is. Nonetheless, percentages of overlap of regions accompanies with the size of feature vector.

The performance of this new one is approximate with the original one thought the length of new feature vector is reduced by 15-25%.

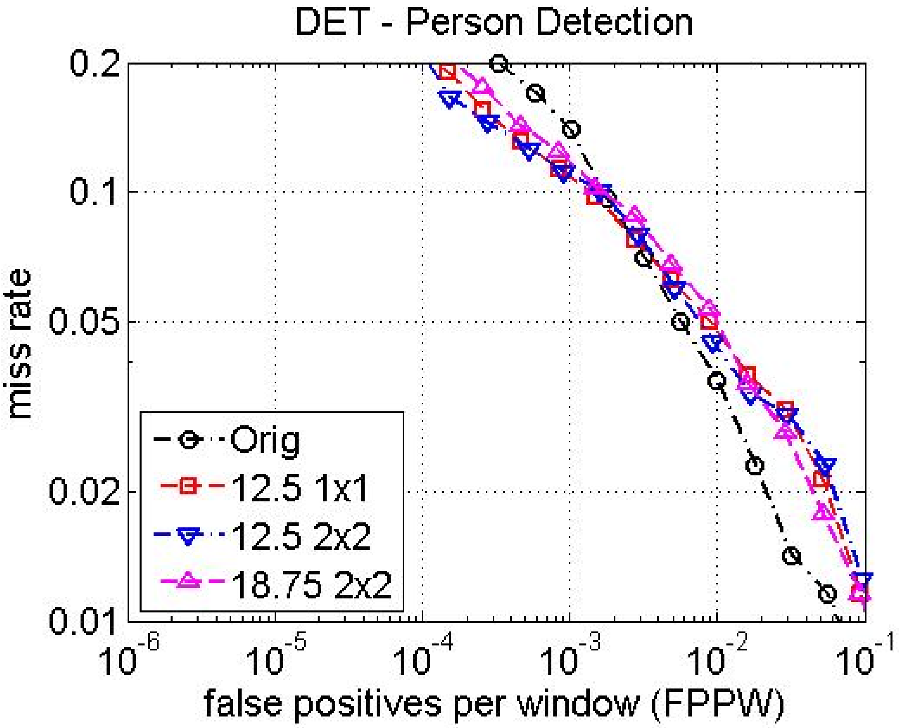


Figure 14: The performances of "four regions" based approaches are as good as the original HOG descriptor. The black is the original one, and the rests are different kinds of four regions based approaches.

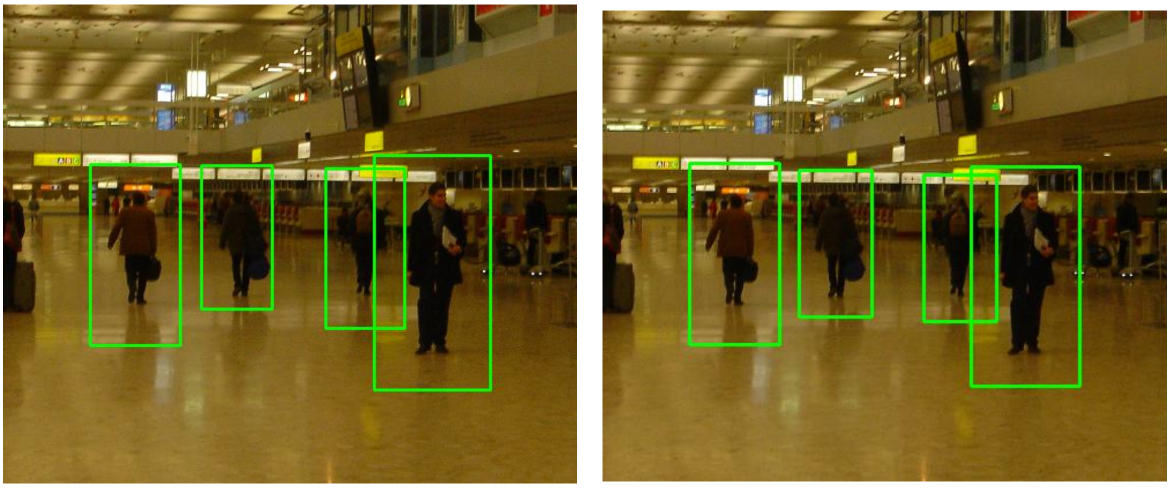


Figure 15:The performances of original & spatial-selective method are similar

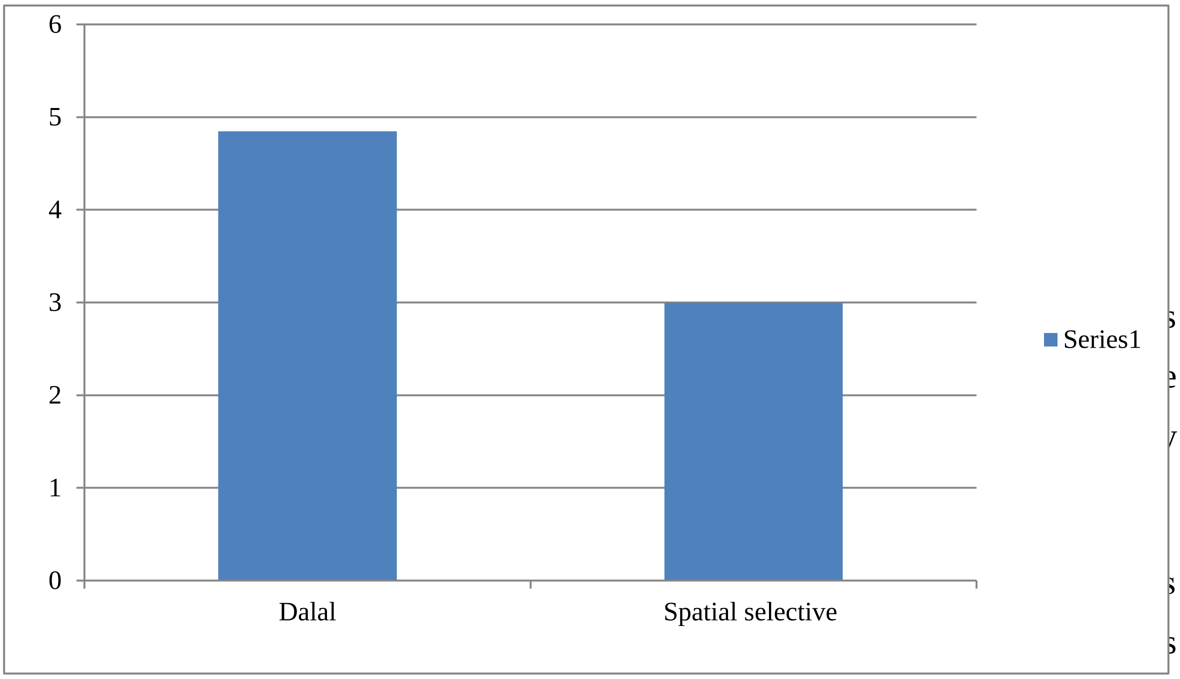


Figure 16: The consuming time it take to encode 2416 feature vector. The Oy coordinate is minutes. By experiments, when reduce the feature vector length, we also make the algorithm run faster

vector of this level is the concatenation of HOG feature of each cell.

Step 2: size of non-overlap cells is increased or reduced to form anther level. And it is similar to step 1 to calculate window feature at this level.

Step 3: Concatenate all these windows feature to create final window feature vector.

However, my concept “multi-level HOG” is a bit different from the above method. At each level, instead we calculate window level feature using whole technique of Dalal and Trigg. The following steps describes in details:

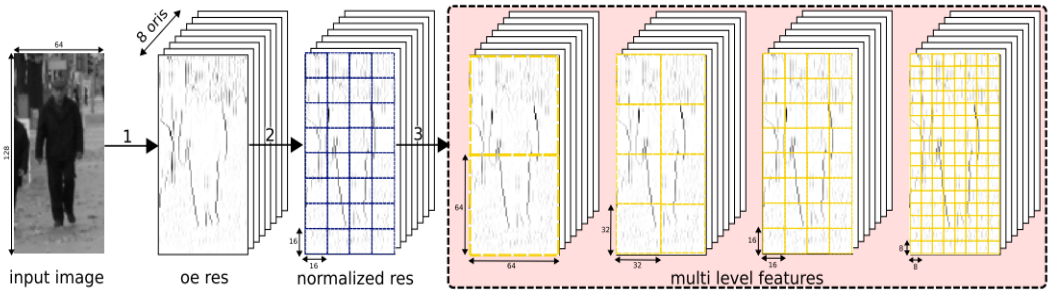


Figure 17: The multi-level approach model [75]

- Calculate the gradient of each pixel.
- At reach level, we create uniform grid of points as figure 17. Different levels have different grid as well as cell size.
- Calculate each level window using HOG.
- Concatenate all levels to form final window feature vector.

We try two different multi-level approaches in which they have three and five level layers respectively. Three-levels one consists of three levels of grid points which is 8x16, 4x8, and 2x4 (they look like the last three window in Fig. 17). The scale between two consecutive levels is 2 (or 1/2). Similarly, five-levels approach has five levels, and the scale is $\sqrt{2}$. Intuitively, we thought that the more levels it has, the better it is. However, this assumption is false according to experiments. The reason may be the saturation in getting object information. The additional levels of five-levels seem to not get any new information about object structure and shape.

In return, the disadvantage of these methods is that they increase the length of feature vector. This probably limits us to apply these methods to real-time applications.

Here are the results of multi-level approach compared with the original one. At 3 levels method, we enhance performance by 2%.

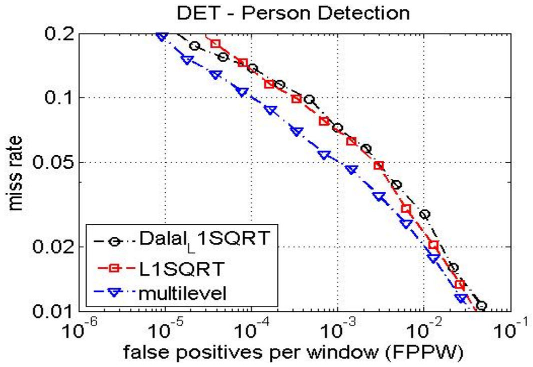


Figure 19: The performance of three level approach (blue curve) is better than the original one. .

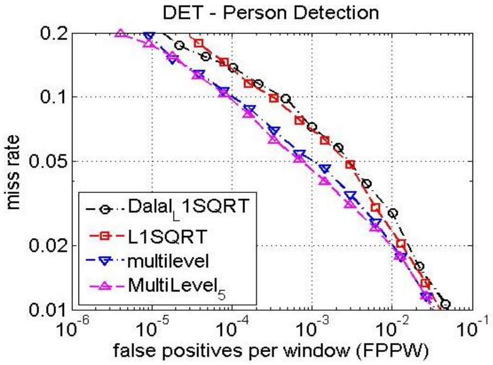


Figure 18: When the number of levels is increased form three to five, the performance is saturated. This is caused by five levels method does not get any more useful structure and shapes of object.

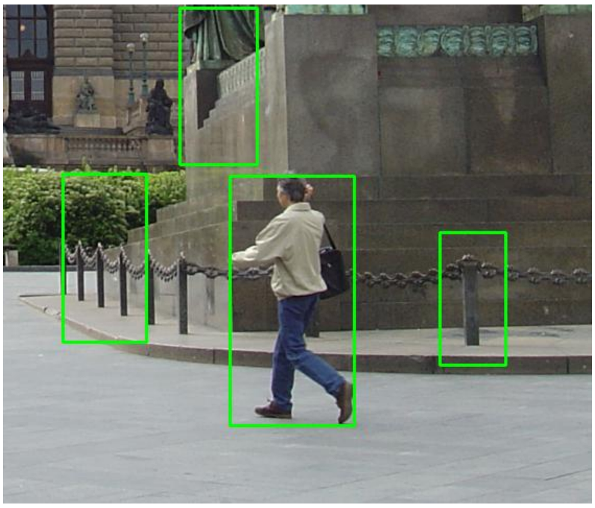


Figure 20: Original Method

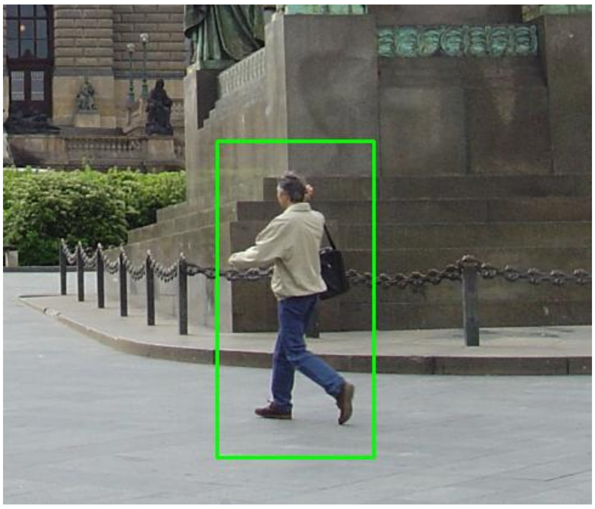


Figure 21: Multi-level Method

Fig. 20 is the result of original method of HOG; its result is adversely affected by the shape of object look likely body. By using multi-level approach, we can reduce the false positive rate (Fig. 21).

4.5 Discussions

This section presents and evaluates several key results. The advent of HOG descriptor has made a breakthrough in computer vision, and it is considered as one of three most influent milestones in feature extraction methods.

As common, people smooth or blur image as pre-processing stage before do anything, but it can be a mistake in HOG descriptor. In other to distinguish object and non-object, edge, contour and shape of image regions must be maintained.

By experiment, cell size of 6-8 pixels offers the best result. Histogram of cell is the basic ingredient to form final histogram vector. If cell size is too large or too small, it does not get the structure as well as shape of this image region. In addition, wide orientation bin and range of orientation bin are also two key largely effecting performances. For all object classes, 20^0 wide bins give good result. And the range of orientation bins (0^0-180^0 or 0^0-360^0) is dependent on each object. By observation, preserving gradient sign information does not seem to help for pedestrian detection because humans wear clothes of all colors, while it is helpful for man-made object classes and for consistently colored natural ones.

Another essence for good result is strong local contrast normalization. One might have thought that a single large many celled HOG descriptor covering the whole detection window would give the best performance, but the results show that a more local normalization policy improves the performance significantly. However it is still best to normalize over a finite spatial patch: normalizing over orientations alone (a HOG block with a single spatial cell) worsens performance. The way that normalization is done is also important.

We can improve the performance by normalizing each element such as edge, cell several times with respect to different local regions, and treating the results as independent signal. Indeed, in HOG algorithm, almost one sell appears in four different blocks (if a block has 2x2 cells). This may seem redundant as the only difference in their votes is the different normalization in the HOG blocks, but the performance will be improved from 84% to 90% at 10^{-4} FFPW if including this redundant information. detection from 84% to 89% at 10− 4 FPPW. Physiological studies also highlight the fact that mammalian visual system has overlapping cells in its primary cortex [8].

Input: image window at current scale

Output: the encoded feature vector

Initial steps:

- Gamma normalize each color channel of input window
- For each color channel, convolve with [-1 0 1] mask along *x* and [-1 0 1]’ mask along *y*. The channel with largest magnitude will be preserved.

Descriptor computation:

- Divide image window to a dense uniform grid of points, and for each point
- Divide ςη x ςη square pixel image region centered on the point into cells
- Create a ς × ς × β spatial and orientation histogram. For each pixel in block, we use trilinear interpolation (described in figure 10) to vote into the histogram using gradient magnitude.

Final steps:

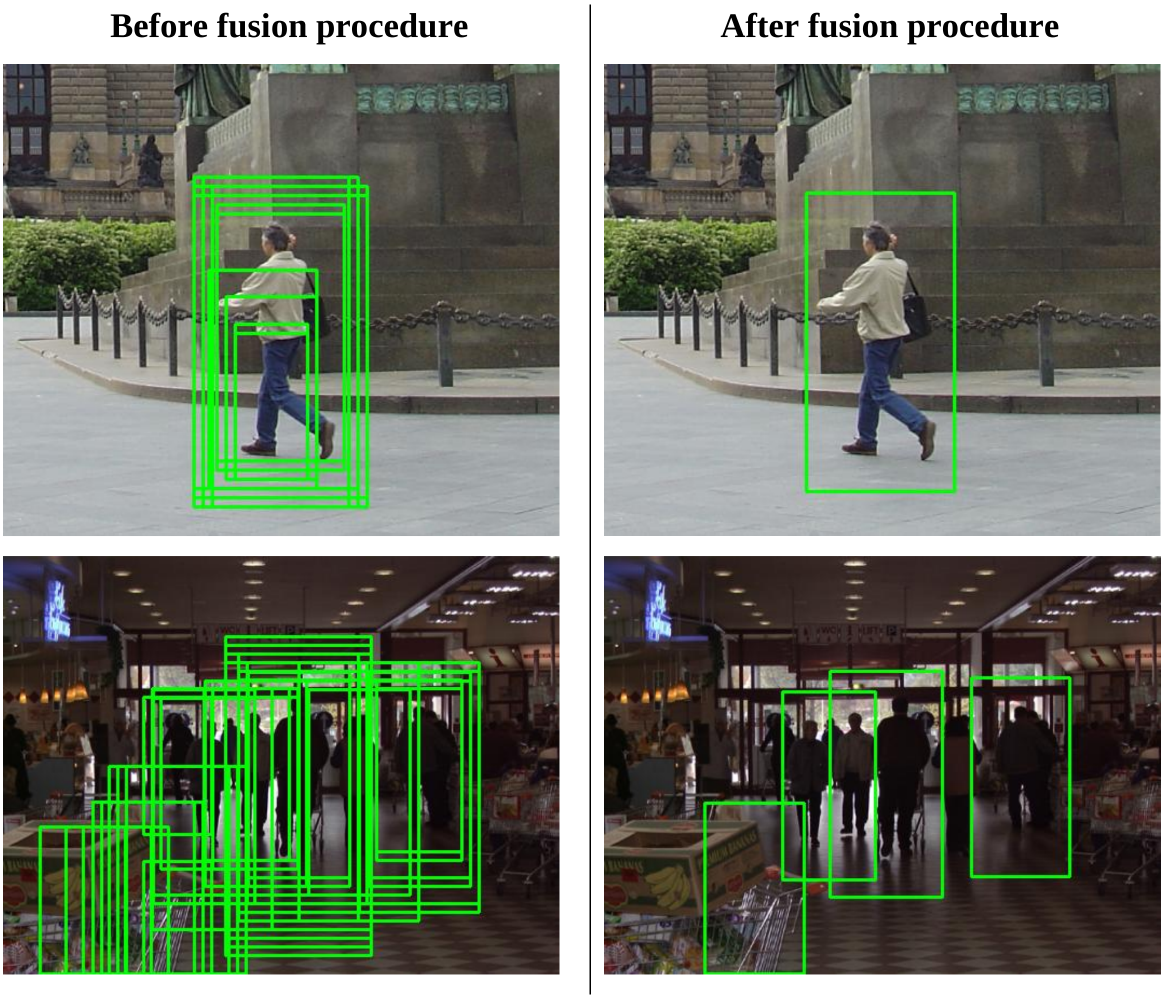
- Apply L2-Hys or L1-Sqrt normalization independently to each block.
- Collect all HOGs of all blocks in the window into one big descriptor vector

Table 2: HOG descriptor extraction algorithm

Chapter 5. Multi scale object localization

Chapter 4 gives details to build up a detector to object/non-object decision. We densely scan through image at all positions and scales. This leads to several detections overlap each other. So, the fusion stage is necessary to yield the final object detections.

This section proposes a solution for fusion of multiple overlapping detections. The overall of this method has been described briefly in chapter 3. In this chapter, we will go through the detail of the method of fusing multiple overlapping detections.


Before fusion procedure

After fusion procedure

Figure 22: Some test image before and after fusion algorithm

4.5. Binary Classifier for Object Localization

Object detection and localization which base on scanning detection windows requires the fusion of overlapping detections. Our fusion method is held if these following assumptions are true:

- If the detector is robust, it should give a strong positive (though not maximum) response even if the detection window is slightly off-center or off-scale on the object.
- A reliable detector will not fire with same frequency and confidence for non-object image windows.

The first hypothesis assumes that the detector response degrades gradually under small changes in object position or scale, but that the maximum response occurs only at the right position and scale. The second hypothesis implies that false positives are mainly due to accidental alignments, so that their probability of occurring consistently at several adjacent scale levels and positions is low.

In addition, the fusion method is based on these following characteristics:

- The higher the peak detection score, the higher the probability for the image region to be a true positive.
- The more overlapping detections there are in the neighborhood of an image region, the higher the probability for the image region to be a true positive.
- Nearby overlapping detections should be fused together, but overlaps occurring at very different scales or positive positions should not be fused.

The third characteristic is based on the observation that the windows used to learn binary classifiers can be larger than the object to allow some context. Thus there may be scenarios where detection windows overlap for nearby objects.

We now present the method called Mean Shift which enables us to clutter the distributed points to proper groups.

4.6. Mean Shift

4.6.1. Brief introduction to Mean Shift

Mean shift is a procedure for locating the maxima of a density function given discrete data sampled from that function. It is useful for detecting the modes of this density. This is an iterative method, and we start with an initial estimate *x*. Let a kernel function $\boldsymbol{K}(x_i - x)$ be given. This function determines the weight of nearby points for re-estimation of the mean. Typically we use the Gaussian kernel on the distance to the current estimate, $K(x_i - x) = e^{c||x_i - x||^2}$. The weighted mean of the density in the window determined by *K* is:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

Where $N(x)$ is the neighborhood of *x*, a set of points for which $K(x) \neq 0$. The mean-shift algorithm now sets x ← m(x) , and repeats the estimation until $m(x)$ converges to *x*

4.6.2. Pros and Cons of Mean Shift method

Pros:

It does not require the prior knowledge of the number of clusters, and does not constrain the shape of clusters and has good performance compared with other clustering algorithm.

Cons:

The only parameter in Mean Shift is the radius to determine the neighborhood. And an issue is arisen how we determine the radius parameter. There is a tradeoff between the accuracy and running time when we choose radius. If we choose large radius, the program will run very fast, but the outcome's performance cannot be guaranteed. On the other hand, if we take small radius, the performance is good, but it takes much time to run. Moreover, Mean Shift is more complex and slower than K-Mean.

5.2.3. Algorithm

In our thesis, we do not use directly the original Mean Shift method. More specific, we add one more parameter to take into account on the overlapping detection windows. From experiments, we observe that one instance of object is usually detected by several nearby windows. Moreover, one important point is that false positive windows are usually distributed randomly. Hence, any cluster which contains instance of object usually has several detections. So, the new parameter is the number of detections in one cluster. If the number of entity in one cluster is lower than some criteria, we will dismiss this cluster without leaving any adverse effect on performance.

The details of algorithm are described below:

Input: - Test image - Trained window classifier - Scale step, the radius of neighborhood, and the minimum number of detections in one cluster Output: Bounding boxes of object detections
HOG descriptor: - Detect windows at all locations and multiple scales - Use HOG descriptor to extract window to high dimensional vector. - Use classifier to take positive windows.
Mean Shift - Consider each window detection as a weighted 3-D point which dimension are two dimensions in image and scale. And the weight of each point is the score of itself. - At each point, we determine the neighborhood and use the equation (6.2) to calculate the mean. - Assign the mean back to the point - Iteratively for each point until it converges to the mode. - For each mode compute the bounding box from the final center point and scale.

Table 3: Fusion algorithm

Chapter 6. Conclusion & Future work

Our thesis provides the evaluation of HOG descriptor as well as its strength and weakness. This thesis also proposes some slight improvement to HOG original. Our ideas are based on the observation of computer vision, machine learning, and visual psychology. Our main contribution is the exploration the inside characteristic of HOG, and base on this observation, we enhance HOG's strength and ease its weakness. The original purpose of HOG descriptor is to build up the image feature set for object detection. Not only that, it has been used in variety tasks such as tracking and activity recognition [..], and context based object detection [..].

6.1. Key Contributions

HOG feature vector is well-known due to its outstanding performance and reliability. But it just does look like a black box; the reasoning beneath this method are vague. This is the motivation for us develop our contributions.

First of all, we spend a lot of time to re-construct HOG framework and evaluate its performance. Though it looks like we try to re-invent the wheel, it is worth to to that because we want to get some insights about characteristics of HOG feature vector. By experiment, we observe that HOG descriptor has very good performance in term of DET (Detection Error Tradeoff) curve. However, it is very sensitive to false positive windows. Hence, to avoid this problem, people usually take the criterion in which FPPW is low. But the tradeoff is that this will miss object in detection process.

After that, from the insights got from re-implementing HOG, we improve HOG descriptor in term of two following aspects. Firstly, original HOG descriptor based on the assumption which is all location in image window are equally important. Hence, by taking all location in image window, the final encoding feature vector is very high, and it slows down the detector. To overcome this stuck; we propose a method called four regions based approach which ignores the small region of image center. By that one, we can shrink feature vector length by 15-25%, while the performance is maintained. Secondly, we see that the more we get information about the structure and shape of object, the better performance is. By this observation, we provide a method called multi-level in which we independently divide image window into different grid of points (each division of grid points called a level). The performance of this one is improved by 2-3%, while the length of feature vector is longer than the original one.

6.2. Limitations

Though HOG descriptor provides good result, there exist some limitations of this approach. These limitations are classified into two main catalogues which are intrinsic and extrinsic.

For one thing, the good performance of HOG method is heavily based on the rigid shape and well aligned image dataset. It requires a lot of effort to annotate dataset in order to make object locate in the center of window. Moreover, the common share about shape of objects in one object class is also required. One more important thing is that training phase needs large number of image in dataset because HOG descriptor is statistical approach.

One intrinsic fragile of statistical approach such as HOG is that it cannot handle the variation of object shape. It the shape and structure of object is not rigid, HOG is failed to detect it. Thinking of human detector, as like figure 23, human can be in diverse of shape; and partial human can be invisible.

In the framework of HOG detector, the most time consuming is the process of calculating HOG feature. According to our experiment, HOG feature calculation costs more than 80% of all computational time. Many resolutions has been proposed to overcome this issue. Zhu et al [73] in 2006 used AdaBoost to speed up algorithm, and allow it to apply in real-time implementation.

Finally, HOG method has one weak point which is that there are a lot of parameters. And the combinations of these parameters make us impossible to evaluate all case to find out the optimal one.

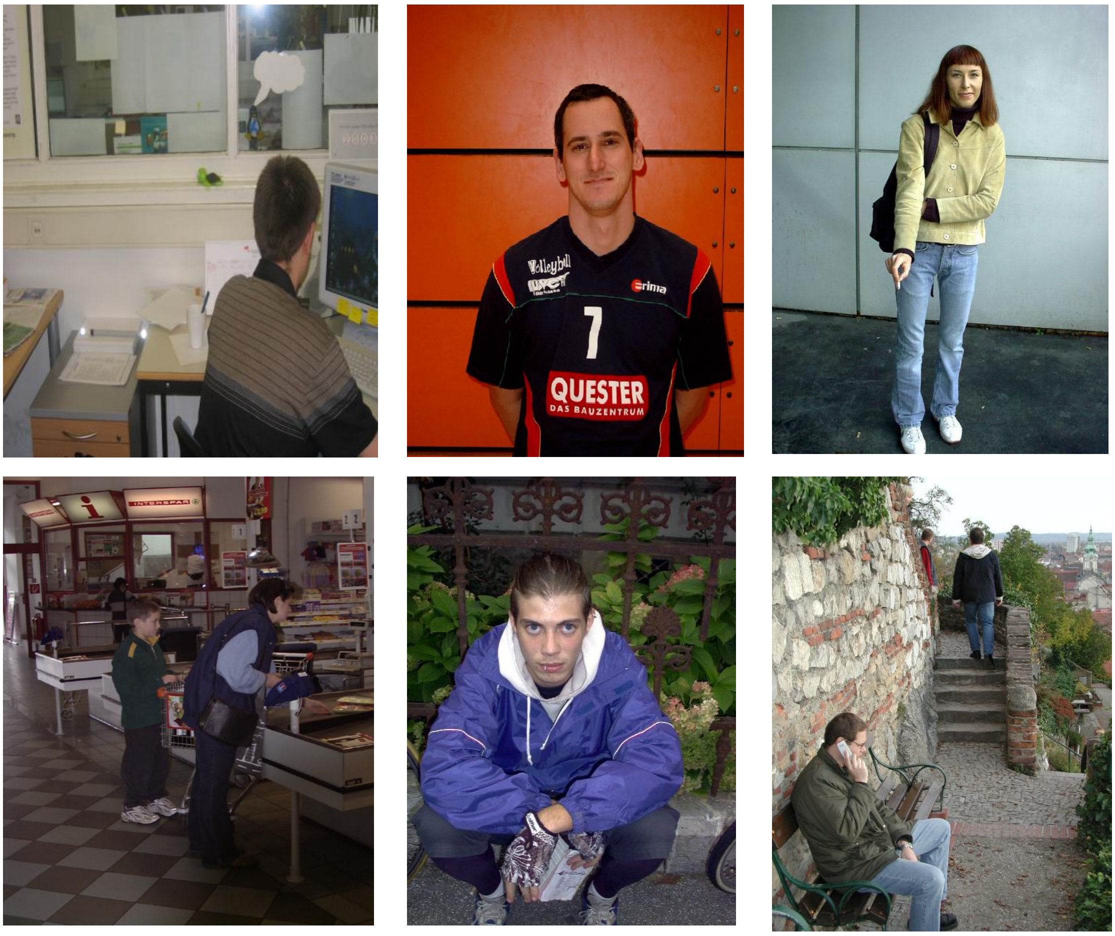


Figure 23: Images from VOC Pascall 2005. People in these images come from diverse shape.

6.3. Future Work

HOG descriptor feature set has made a breakthrough in computer vision in proposing a robust and reliable feature set. However, there are still some aspects to improve its performance.

Inspiring by the approach called “four regions” which is described above, the assumption, that there is some region less informative than the others, should be considered. Depend on each object and dataset, we can decide which region is informative in image window (see figure 24). This approach employs non-uniform grid of points perspective which concentrates more point into informative regions, and not concentrate on less informative ones. This does not mean that we totally ignore the background surroundings object, but we just make it less important than the regions contain object's contour and shape.

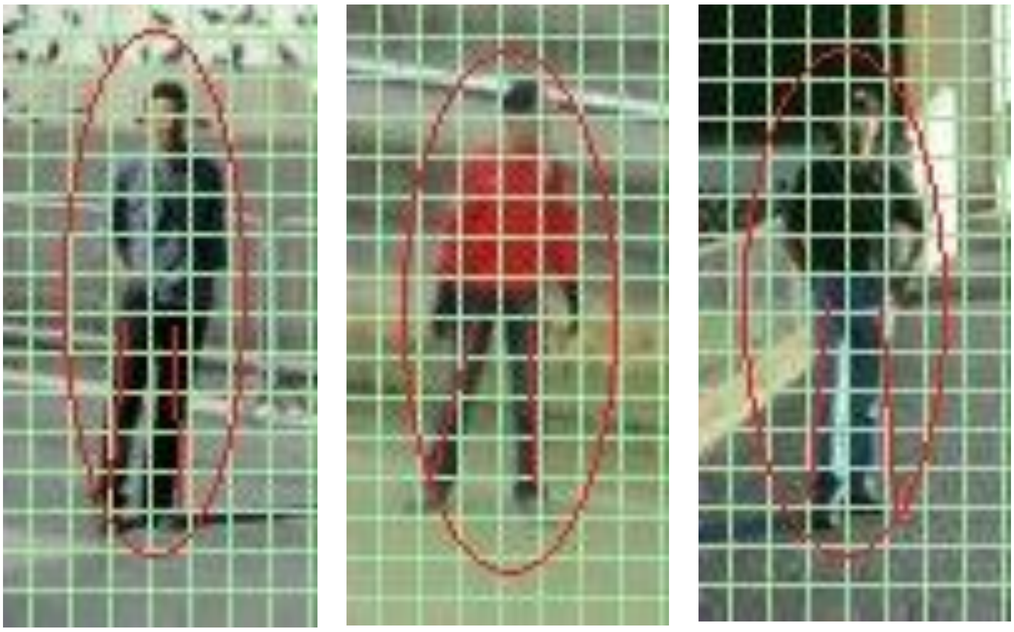


Figure 24: Some training image from pedestrian INRIA dataset. The grid points along the red curve and line is assumedly more important than the others, especially the grid points in corner of image window.

Secondly, fusion algorithm used to cluster detection windows is also the one should be improved. In our thesis, we use mean shift as an algorithm to suppress the non-maxima detection windows. And its performance heavily depends on the given bandwidth. In crowded pedestrians image, we need small bandwidth; while in spare pedestrian image, we need large bandwidth to avoid false positive detection. Hence, the task of building up a system that can automatically select proper bandwidth is very essential. Dorin Comaniciu et al 2003 [78] has proposed a method to try to resolve this one.

REFERENCE

[1] Richard Szeliski, book "Computer Vision: Algorithms and Applications", section 1.1, page 3, line 4-7.

[2] CHIARI, Y., WANG, B., RUSHMEIER, H. and CACCONE, A. (2008), Using digital images to reconstruct three-dimensional biological forms: a new tool for morphological studies. Biological Journal of the Linnean Society, 95: 425–436.

[3] David King et al. 1996. Is the human mind a Turing machine? Synthese, Springer Netherlands. Subject: Humanities, Social Sciences and Law, pages: 379-389, Volumn 108.

[4] Lin Yang. 2009. Robust segmentation and object classification in natural and medical images IEEE. PhD thesis abstraction. The State University of New Jersey.

[5] C. Zetzsche, K. Schill, H. Deubel, G. Krieger, E. Umkehrer, and S. Beinlich. Investigation of a sensorimotor system for saccadic scene analysis: an integrated approach. In From animals to animats, Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior, volume 5, pages 120–126, 1998.

[6] E. Barth, C. Zetzsche, and I. Rentschler. Intrinsic two-dimensional features as textons. Journal of the Optical Society of America – Optics, Image Science, and Vision, 15(7):1723–1732, 1998.

[7] C. Carson et al, 1999. Blobworld: A system for Region Based Image Indexing and Retrieval, Volumn 1615, pp 660-660.

[8] B. Fischer. Overlap of receptive field centers and representation of the visual field in the cat's optic tract. Vision Research, 13:2113–2120, 1973.

[9] D.H. Hubel and T.N. Wiesel. Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor. J. of Computational Neurology, 158(3):295– 305, 1974.

[10] D. Hubel. Eye, Brain, and Vision. W.H. Freeman & Company, 1995.

[11] L. Fei-Fei, R. Fergus, and A. Torralba. "Recognizing and Learning Object Categories, CVPR 2007 short course".

[12] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part Based Models. Pattern Analysis and Machine Intelligence, IEEE, 32: 1627 – 1645.

[13] V. Vapnik. The nature of statistical learning theory. Springer-Verlag, 1995.

[14] N. Cristianini and J. Shawe-Taylor. Support Vector Machines. Cambridge University Press, 2000.

[15] B. Sch¨olkopf and A. Smola. Learning with Kernels. The MIT Press, Cambridge, MA, USA, 2002.

[16] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[17] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In Intl. Workshop on Automatic Face- and Gesture-Recognition, IEEE Computer Society, Zurich, Switzerland, pages 296–301, June 1995.

[18] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In 2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA, pages 100–105, October 1996.

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

[20] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, pages 454–461, 2001.

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

[22] . D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA, pages 87–93, 1999.

[23] Alexander Neubeck and Luc Van Gool, Efficient Non-Maximum Suppression, ICPR 2006.

[24] Frederic Devernay , Programme Robotique , and Projet Robotvis, A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy

[25] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–151, 1988.

[26] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume I, pages 128–142, May 2002.

[27] T. Lindeberg. Feature detection with automatic scale selection. International Journal of Computer Vision, 30(2):79–116, 1998.

[28] D. G. Lowe. Local feature view clustering for 3D object recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, pages 682–688, December 2001.

[29] T. Kadir and M. Brady. Scale, saliency and image description. International Journal of Computer Vision, 45(2):83–105, 2001.

[30] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In 4th International Workshop on Visual Form, Capri, Italy, May 2001.

[31] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In Proceedings of the 9th International Conference on Computer Vision, Nice, France, pages 281–288, 2003.

[32] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume IV, pages 700–714, 2002.

[33] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, volume I, pages 69–81, 2004.

[34] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5):530–549, May 2004.

[35] B. Schiele and J.L. Crowley. Object recognition using multidimensional receptive field histograms. In Proceedings of the 4th European Conference on Computer Vision, Cambridge, England, pages 610–619, 1996a.

[36] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5):530–534, May 1997.

[37] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, pages 18–32, 2000.

[38] D. G. Lowe. Local feature view clustering for 3D object recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, pages 682-688, December 2001.

[39] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume IV, pages 113–127, 2002.

[40] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, volume II, pages 264–271, 2003.

[41] G. Dork´o and C. Schmid. Selection of scale-invariant parts for object class recognition. In Proceedings of the 9th International Conference on Computer Vision, Nice, France, volume 1, pages 634–640, 2003.

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

[43] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, volume II, pages 71–84, 2004.

[44] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA, pages 876–885, June 2005.

[45] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In Proceedings of the 10th International Conference on Computer Vision, Bejing, China, volume 2, pages 1792–1799, 2005.

[46] W. F¨orstner and A. Pertl. Photogrammetric standard methods and digital image matching techniques for high precision surface measurements. In E.S. Gelsema and L.N. Kanal, editors, Pattern Recognition in Practice II, pages 57–72. Elsevier Science Publishers B.V., 1986.

[47] W. F¨orstner and E. G ¨ulch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, pages 281–305, June 1987.

[48] N. Dalal, Ph.D thesis “Finding People in Images and Videos”, 2006. Appendix C, page 113-113.

[49] B. Leibe. Informative features for profile-view pedestrians. Personal Communication, March 2006.

[50] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. International Journal of Computer Vision, 60(1):63–86, 2004.

[51] Saal, H. et al., 2006. Salient image regions as guide for useful visual features. *IEEE Advances in Cybernetic Systems*.

[52] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume IV, pages 700–714, 2002.

[53] C. Papageorgiou and T. Poggio. A trainable system for object detection. International Journal of Computer Vision, 38(1):15–33, 2000.

[54] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, pages 66–75, 2000.

[55] S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. International Journal of Computer Vision, 43(1):45–68, 2001b.

[56] Heusch G., Rodriguez Y. and Marcel S. (2006), Local Binary Patterns as an Image Processing for Face Authentication, Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition.

[57] Liao, S., Law M.W.K. and Chung A.C.S. (2009), Dominant Local Binary Patterns for Texture Classification. IEEE Trans. Image Processing 18(5):1107-1118.

[58] Tan X. and Triggs B. (2007), Fusing Gabor and LBP feature Sets for Kernel-based Face Recognition, Proceedings of the IEEE International Workshop on Analysis and Modeling of Face and Gesture, pp.235-249.

[59] Wang J.-G., Yau W.-Y. and Wang H. L. (2009), Age Categorization via ECOC with Fused Gabor and LBP Features. Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV), pp.313-318.

[60] Zhao, G. and Pietikäinen, M. (2007), Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. IEEE Trans. Pattern Analysis and Machine Intelligence 29(6:915-928.

[61] Zhang, W., Shan, S., Gao, W., Chen, X. and Zhang H. (2005), Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-statistical Model for Face Representation and Recognition. In Proc. Tenth IEEE International Conference on Computer Vision, 1:786-791.

[62] Mäenpää, T. and Pietikäinen, M. (2005) Texture Analysis with Local Binary Patterns. In: Chen, C.H. and Wang, P.S.P. (eds.) Handbook of Pattern Recognition and Computer Vision, 3rd ed.. World Scientific, pp. 197-216

[63] Raj Gupta, Harshal Patil and AnuragMittal. Robust Order-basedMethods for Feature Description. CVPR 2010.

[64] XiaoyuWang et al. An HOG-LBP Human Detector with Partial Occlusion Handling. ICCV 2009.

[65] Yinan Yu et al. Object Detection by Context and Boosted HOG-LBP. Pattern Recognition 2010.

[66] AI Magazine Volumn 11, Number 3, AAAI 1990.

[67] Hichem Sahbi, Jean-Yves Audibert and Renaud Keriven. Context-Dependent Kernels for Object Classification. PAMI 2010.

[68] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(4):349–361, April 2001.

[69] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In Proceedings of the 6th International Conference on Computer Vision, Bombay, India, pages 555–562, 1998.

[70] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, volume I, pages 511–518, 2001

[71] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In Proceedings of the 9th International Conference on Computer Vision, Nice, France, volume 1, pages 734–741, 2003

[72] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. International Journal of Computer Vision, 56(3):151–177, 2004.

[73] Q. Zhu, S. Avidan, M. Ye, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 2006. Accepted for publication.

[74] One Response to *The big bang of computer vision apps, Part 2.* Date: 4/2/2011, 5:00 PM, http://linkapic.com/blog/2011/01/the-big-bang-of-computer-vision-apps-part-2/

[75] Subhransu Maji et al. Classification using Intersection Kernel Support Vector Machines is Efficient. IEEE Computer Vision and Pattern Recognition 2008.

[76] Leow Wee Kheng, "Mean Shift Tracking" lecture on Course CS4243 Computer Vision and Pattern Recognition, Department of Computer Science School of Computing, National University of Singapore.

[77] Dorin Comaniciu and Peter Meer, *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 2 (1999), pp. 1197-1203 vol.2.

[78] Dorin Comaniciu et al. An Algorithm for Data-Driven Bandwidth Selection. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 2, FEBRUARY 2003.

[79] S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, Vol. 31 (2007), pp. 249-268.

[80] A.W.Moore, K-Mean and Hierarchical Clustering, lecture notes. School of Computer Science, Carnegie Mallon University. Website: http://www.autonlab.org/tutorials/kmeans09.pdf

[81] N. Dalal, Ph.D thesis "Finding People in Images and Videos", 2006, pages 43-43.

[82] N. Dalal, Ph.D thesis "Finding People in Images and Videos", 2006, pages 38-38.

[82] N. Dalal, Ph.D thesis "Finding People in Images and Videos", 2006, pages 39-39.

APPENDIX

Dataset

The data set which we used is “INRIA static” person data set. It can be get freely from http://pascal.inrialpes.fr/data/human . Positive training images have unique sizes, and the people in these images are usually standing at the center so that it’s very suitable for training pedestrians. In negative images, there are indoor, outdoor, beach, mountain, and city scenes. They have no people but have other objects like vehicles, furniture, animals… . The data set is split into two small sets: a set for training and a set for testing. The positive training set contains 1208 images and the positive test set has 566 images. The negative training set contains 1218 images and in the negative test set there’re 453 images.