**UNIVERSITY OF SCIENCE**

**ADVANCED PROGRAM IN COMPUTER SCIENCE**

**<FULLNAME OF ALL STUDENTS IN A SINGLE LINE>**

**<THESIS’S FULL TITLE>**

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**HO CHI MINH CITY, <20XX****>**

**UNIVERSITY OF SCIENCE**

**ADVANCED PROGRAM IN COMPUTER SCIENCE**

**<STUDENT1’S NAME> <0xxxxxx>**

**<STUDENT2’S NAME> <0xxxxxx>**

**<THESIS’S FULL TITLE>**

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**THESIS ADVISOR**

**<FULLNAME OF THESIS ADVISOR(S)>**

**HO CHI MINH CITY, <20XX>**

# Acknowledgements

I greatly want to express my gratitude to my thesis directors, Professor Le Hoai Bac and Vo Dinh Phong, who have helped and guided us a lot when we start to do this thesis. They helped me get the background and have the overall view of computer vision. They also give me some very incredible ideas and advises, as well as encouragement when we got stuck.

Moreover, we express my thanks to Tran Ngoc Trung and Pham Hoai Vu for their objective argument and assistances to our thesis.

Table of Contents

[Acknowledgements i](#_Toc289182274)

Table of contentsii

[Abstract vi](#_Toc289182275)

[Chapter 1. Introduction 1](#_Toc289182276)

[1.1. Briefly introduction about Computer Vision 1](#_Toc289182277)

[1.2. Thesis introduction 1](#_Toc289182278)

[1.3. Goal and Applications 3](#_Toc289182279)

[1.3.1. The goal 3](#_Toc289182280)

[1.3.2. Applications 4](#_Toc289182281)

[1.4. Challenge 5](#_Toc289182282)

[1.5. Some background of object detect 6](#_Toc289182283)

[1.6. Overview of our approach 8](#_Toc289182284)

[1.7. Overview of extraction algorithm 11](#_Toc289182285)

[1.8. Summary of Contributions 12](#_Toc289182286)

[1.9. Outline 12](#_Toc289182287)

[Chapter 2. Related work 15](#_Toc289182288)

[2.1. Image features 15](#_Toc289182289)

[2.1.1. Spare Local Representations 17](#_Toc289182290)

[2.1.2. Dense Representation of Image Regions 21](#_Toc289182291)

[**Regions and Fragments Based on Image Intensity** 21](#_Toc289182292)

[**Edge and Gradient Based Detectors** 22](#_Toc289182293)

[**Wavelet Based Detector** 22](#_Toc289182294)

[**Order of Pixel Detector** 23](#_Toc289182295)

[2.2. Classification Methods 24](#_Toc289182296)

[2.2.1. Discriminative Approaches 24](#_Toc289182297)

[2.2.2. Bayesian and Graphical Models 26](#_Toc289182298)

[2.3. Fusion of Multiple Detections Approaches 26](#_Toc289182299)

[2.4. Motivations of Employment HOG Methods 27](#_Toc289182300)

[Chapter 3. Overview of Detection Methodology and results 29](#_Toc289182301)

[3.1. Overall framework 29](#_Toc289182302)

[3.1.1. Training phase 29](#_Toc289182303)

[3.1.2. Detection phase 33](#_Toc289182304)

[3.2. Overview of feature sets 34](#_Toc289182305)

[3.3. Fusion of multiple detections 38](#_Toc289182306)

[Chapter 4. HOG 40](#_Toc289182307)

[4.1. Static HOG Descriptor 40](#_Toc289182308)

[4.2. Implement and Performance Study 41](#_Toc289182309)

[4.2.1. Color channel 41](#_Toc289182310)

[4.2.2. Color/Gamma Normalization 41](#_Toc289182311)

[4.2.3. Gradient Computation 42](#_Toc289182312)

[4.2.4. Spatial/Orientation Binning 43](#_Toc289182313)

[4.2.5. Block Normalization, Block Size, and Overlap 44](#_Toc289182314)

[4.2.6. Classifier 46](#_Toc289182315)

[4.3. Hard Samples 48](#_Toc289182316)

[4.4. 4.4 Overall Results 51](#_Toc289182317)

[4.5 Discussion 51](#_Toc289182318)

[Chapter 5. Some modified HOG 55](#_Toc289182319)

[5.1. Four regions based approach 55](#_Toc289182320)

[5.2. Non-uniform based approach 58](#_Toc289182321)

[5.3. Multi-level based approach 59](#_Toc289182322)

[Chapter 6. Multi scale object localization 62](#_Toc289182323)

[6.1. Binary Classifier for Object Localization 62](#_Toc289182324)

[6.2. Mean Shift 63](#_Toc289182325)

[6.2.1. Brief introduction to Mean Shift 63](#_Toc289182326)

[6.2.2. Pros and Cons of Mean Shift method 64](#_Toc289182327)

[6.2.3. Algorithm 64](#_Toc289182328)

[Chapter 7. Dataset 67](#_Toc289182329)

[Chapter 8. Conclusion 68](#_Toc289182330)

[Chapter 9. Future work 69](#_Toc289182331)

# Abstract

This thesis’ goal is to build up the automatic system which enable to detection object in static image, particularly pedestrian. Our main concern is on exploring robust feature extraction algorithms which encode image regions to high dimensional feature vector 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.

# Introduction

* 1. Briefly introduction about Computer Vision

As human, we perceive the three dimension structure of the word surroundings with apparent easy. For instance, think of how vivid three dimensions perceive is when you look at the flowers in garden. You can tell the shape and translucency of each petal through the subtle patterns of light and shading that play across its surface and effortlessly segment each flower from the background of the scene. However, the way of computer perceives the world is totally different from human. Hence, perceptual psychologists and computer scientists have spent decades trying to understand how the visual system works, and they try to devise a method to link the gap of human and computers perspectives.

Recent years, researchers in computer vision and mathematicians have developed some reliable models and formulas to help computer have abilities of mimicking human. For example, by gathering several images from different point of view of one object, we can accurately reconstruct three dimension model of this object. Moreover, with the advanced techniques and facilities, tracking object in complex surroundings is also possible. However, despite of all above achievements, inventing program which has ability to interpret an image at the same level as a two-year old is just a dreaming, and it is much more difficult than writing an algorithm to defeat International Chess Champion.

* 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. 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. 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 shape. 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. Goal and Applications

### 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 create object detection from the view of point of computer, in which detector scan all the given image 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 shape, 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.

### 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.

* 1. Challenge

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

* 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.
* 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.
* 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 good performance detectors is very difficult.
  1. Some background of object detect

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 [Zetzsche et al. 1988, Barth et al, 1988.], 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 [Fischer 1973, Hubel and Wiesel 1974, Hubel 1995].

One important thing is that these two approaches may not be as different as they seem because the detection of salient regions also requires a dense scan whole image. Indeed, the criterion used to scan image for salient regions and the information encoded in the final feature vector can be different. For instance, in SIFT descriptor, the interest point based approach, blob structures are used due to its valuable informative region, while the final feature vector is computed by gradient or contour information (or shape content [Belongie et al. 2001]). In contrast, the detector in spare approach only receives the filtered response of the salient region detector; it has no awareness how to choose these salient regions in the first stage.

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” 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.

Both generative and discriminative methods can be used in detection phase. Typically, generative approaches use Bayesian graphical models Expectation-Maximisation (EM) to characterise these parts and to model their co-occurrences. Discriminative approaches use machine learning techniques to classify each feature vector as belonging object or not.

* 1. 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 point as far as possible.

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). 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. 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 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 to find only one window most likely contains object of a class. In this thesis, Mean Shift is used as a suppression algorithm in this process.

.



Pedestrian/ non-pedestrian classifier

Figure : Overview of detection method

1. 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.
   1. Overview of extraction algorithm

* Normalize gamma, using equalizeHist function in OpenCv in each channel color.
* At each color channel, compute gradient of each pixel using centered mask, the info of gradient is the weight and orientation. And at each pixel, we choose color with greatest magnitude of gradient.
* Divide window into many equal squared cells.
* In each pixel, we calculate how it contributes to histogram of cell which contain it, as well as the neighborhood cells. (spatial and orientation histogram). (See figure 2)
* In the window, blocks are created by group of cells. Blocks can overlap each other.
* Concatenate histograms of blocks to form feature vector of window.
  1. 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:

* We re-implement the work of HOG and have the same 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. 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 will examine the effects of enormous parameter in HOG descriptor. The results show that uniform grids of HOG descriptors significantly outperform several existing feature sets for human detection. Furthermore, we study the effect of each stage in computing feature sets, with the conclusion is that fine-scale image gradients, fine orientation binning, relatively coarse spatial binning, and high quality contrast normalization are all important for good result. We also give the observation of how to choose parameters which appropriate for particular object.
* Chapter 5: presents some modified HOGs. 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. In the second one, we assume that the regions which contain the contour and shape are more informative than the others, so I put more grid points to these regions. However, the performance of this method is lower than the original one. 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 6: 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.

# 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.
* The detection framework which is built over these descriptors.

The following sections will present some related works which work on these categories. Section …

* 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 represenration 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: the pioneer researchers are:
  + Harris and Stephens 1988, in paper “A combined corner and edge detector”, Alvey Vision Conference, pages 147–151, 1988
  + Mikolajczyk and Schmid 2002, in paper “An affine invariant interest point detector”, Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume I, pages 128–142, May 2002.
* Blobs: pioneer researchers are:
  + Laplacian of Gaussian (Lindeberg 1998) in paper “Feature detection with automatic scale selection”, International Journal of Computer Vision, 30(2):79–116, 1998.
  + Difference of Gaussian (Lowe 2001) in paper “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.
* Intensities: pioneer researchers are:
  + Kadir and Brady 2001, with paper “Scale, saliency and image description”, International Journal of Computer Vision, 45(2):83–105, 2001.
  + Ullman et all 2001, with paper “A fragment-based approach to object representation and classification”, in 4th International Workshop on Visual Form, Capri, Italy, May 2001.
  + Vidal-Naquet and Ullman 2003, with paper “Object recognition with informative features and linear classification”, in Proceedings of the 9th International Conference on Computer Vision, Nice, France, pages 281–288, 2003.
* Gradients: pioneer researchers are:
  + Ronfard et al 2002, with paper “Learning to parse pictures of people”, in Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, volume IV, pages 700–714, 2002.
  + Mikolajczyk et al 2004, with paper “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.
* Color, texture, or combinations of several or all of these: the pioneer researcher is:
  + Martin et al 2004, with paper “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.

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.

### 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. This approach is simulated by the observation process of human beings. 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: papers in this one are:
  + W. Forstner 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.
  + W. F¨orstner and E. G ¨ulch. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, pages 281–305, June 1987.
  + C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–151, 1988.
* Laplacian: this method is described in paper:
  + T. Lindeberg. Feature detection with automatic scale selection. International Journal of Computer Vision, 30(2):79–116, 1998.
* Difference of Gaussians (DoG): reference paper:
  + D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.
* Scale Invariant Harris-Laplace:
  + K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. International Journal of Computer Vision, 60(1):63–86, 2004.

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:

* Compactness of representation is one 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.
* Robust to scale and rotation of objects.
* Point detectors are based on a very strong hypothesis which is salient image regions are most valuable and repeatable one.

However, there are some inherent drawbacks which are:

* 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.
* 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.

One or many key point detectors are combined to find the interest points due the complication of particular object class. For instance, Fergus et al [2003] used the entropy based region detector of Kadir and Brady [2001] for object representation.

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) and *shape context*. 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 ultilizes log-polar grids.

#### Part Detectors

Part based detector/model has widely been used in object detection and recognition [Forsyth and Fleck 1997, Ioffe and Forsyth 1999, Schneiderman and Kanade 2000, Ronfard et al. 2002, Ramanan and Forsyth 2003, Sigal et al. 2003, Schneiderman and Kanade 2004]. For example, Forsyth and Fleck [1997], Ioffe and Forsyth [1999, 2001a,b] and Ramanan and Forsyth [2003] use explicit human body segments (forearm, upper arm, upper leg, lower leg, torso, etc.) which are assumed to be well represented by cylinders. Pedro F. Felzenszwalb ultilize parts based model on HOG descriptor won VOC Passcal Challenges as the first order Parallel edge detectors are then used to detect the corresponding image segments, and body-geometry based detectors are built by using articulation constraints or graphical models to constrain the relative geometry of the limbs. 3D limb detectors have also been used, c.f . Sigal et al. [2003]. One problem with these approaches is that the assumption that limbs can be represented by parallel lines is rather simplistic and its scalability to real world examples is questionable. This may explain the lack of extensive testing on real world images in these works.

### Dense Representation of Image Regions

In stead 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.

**Regions and Fragments Based on Image Intensity**

Two decades ago, Sirovitch and Kirby [1987], Turk and Pentland [1991] has used image intensities as a eigen-faces to approach the problem. In this simple approach, the pixels of fixed-resolution face images are rearranged to build a large feature vector, and use PCA (Principal Component Analysis) to draw the main variations of the ensemble of face vectors. Another one utilizing image intensities is face detection system of Rowley [1998] which locally correct the lighting of image and passing them to Neural Network Classifier [Bishop 2005] for face/non-face detections. Ullman et al [2001], Vidal-Naquet and Ullman [2003] use object fragments based on image intensity patterns. Fragments are generated randomly and the most relevant are selected by maximizing the mutual information between fragment and class. The authors declare the improvement relative to wavelet based descriptor, but it is just true for inflexible objects such as cars, faces.

**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 [1999] 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 [2002] and Mikolajczyk et al [2004] to build an automatic body detector. Similarly, Felzenszwalb and Huttenlocher [2000], Ioffe and Forsyth [2001], and Mikolajczyk et al [2004] design human detection system which propose 7 parts detectors: head, upper body and legs.

**Wavelet Based Detector**

In this kind of detector, some well-known approaches are developed by Papageorgiou and Poggio [2000], Mohan et al [2001], Viola and Jones [2001]. Their computations of image regions share some similarity to Haar wacelets.

Papageorgiou and Poggio [2000] use absolute value of Haar wavelet coefficients at different orientations and scales as their local descriptors. Images that are mapped from pixel space to an over complete dictionary of Haar wavelets that is rich enough to describe the pattern. Three different kinds of wavelet patterns which are horizontal, vertical, diagonal have been used. In order to achieve over-complete basis, the wavelets are computed with overlapping supports. The advantages of Haar wavelet are that it can be calculated efficiently, and possibility of encoding significant patterns. The vector descriptors are then used in kernelized Support Vector Machine. Papageorgiou and Poggio [2000] method has been extensively tested in car, face, and pedestrian.

Mohan et al [2001] design parts based model whose descriptor is of Papageorgiou and Poggio [2000] works. Four part detectors are built for pedestrians: head, left arm, right arm, and legs detectors. The response of part detectors are checked for a proper geometric configuration, and the final classification is performed by using SVM on their outputs.

Viola and Jones [2001] and Viola et al [2003] have made a breakthrough in classification field by devising a method called cascade model which efficiently rejects negative cases as much as possible. As a sequence, most obvious negative one will be eliminated at the first or second stage of cascade model, while remaining positive ones. Hence, it speeds up detector a lot and enables to apply to real time detector.

**Order of Pixel Detector**

Order of pixel is an alternative approach of extracting local image regions to feature vectors. And Local Binary Pattern (LBP) 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].

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.

* 1. 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. 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.

### 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.

#### 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. Moreover, Hichem Sahbi has designed a context dependent kernel which is based on characters of particular object to enhance the classification process.

#### 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 [2001], Viola et al [2003] use AdaBoost to train cascades of weak classifiers for face and pedestrian detection. Opelt et al [2004] use AdaBoost framework for interest point based weak classifiers. Schneiderman and Kanade [2000, 2004] 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 [2006] 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.

### Bayesian and Graphical Models

Ullman et al. [2001] and Vidal-Naquet and Ullman [2003] show that image fragments selected by maximizing the mutual information between the fragment and the class label provide an informative and independent representation. They use Na¨ıve Bayes classification and show that using tree based Bayesian networks over the same fragment set does not give a noticeable improvement in classification results.

Weber et al. [2000] use Bayesian generative models learned with EM to characterize classes, and use likelihood ratios for classification. Fergus et al. [2003] also use likelihood ratios, but with a more elaborate model of conditional probabilities that includes the position and scale of the features as well as their appearance.

* 1. Fusion of Multiple Detections Approaches

Fusion algorithm of overlapping detections is needed because it is possible to have multiple positive windows over same instance of object class. Rowley et al. [1998] proposed a heuristic method for fusing overlapping detections. The number of detections within a specified neighborhood is computed and if it is greater than a threshold, the centroid of these detections is taken as the location of the detection result. Centroids are computed in 3-D position and scale space. The number of detections gives the detection score. The scale value of the centroid defines the bounding region for the detected object. Each centroid is examined to check whether its bounding region overlaps with other centroids. Overlapping centroids with lower score are removed and the remaining centroids constitute the final result. Viola and Jones [2004] proposed a simpler method. The set of detections is partitioned into disjoint subsets, and each partition corresponds to a single final detection. Two detections are taken to be in the same subset if their bounding regions overlap. The final detection and location region is the average of all of the detections in the set.

* 1. Motivations of Employment HOG Methods

Firstly, I am interested in methods using SIFT to exact feature vector. However, SIFT is very slow compared with modern methods like HOG, LBP, and its performance in pedestrians object is not high, around 80%.

Secondly, I did not choose LBP because of some reasons. Despite of the simplification of naïve LBP method, its accuracy is not high. Moreover, LBP based method which is carefully implement gives good results, but it is a bit complicated.

Finally, I choose HOG based method because it is not too hard to implement. In addition, papers using HOG as a method for object detection has many citations. And the other reason is that there is a thesis which describes HOG method very carefully.

Moreover, I suggest several modified HOG approaches with the purposes are to reduce the size of feature vector as well as increase the performance. The details of these approaches will be described latter.

# 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.

* 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. The performance of final step is mostly depended on the reliability and robustness of classifier.

### Training phase

Figure : Training phase overview.

*A given test image is densely uniform scanned at all locations and multi-scale. Each sliding window is extracted to create high dimensional feature vector. The extraction algorithm here is Histogram of Oriented Gradients. The feature vector is then put to linear SVM classifier to have object/non-object decision.*

Input image

Scan image at all position and multi scale

Extract feature over windows

Put feature vectors into linear SVM

Detection window

Normalize gamma

Compute Gradients

Weighted vote in Spatial & orientation cells

Contrast normalize over overlapping spatial cells

Collect HOGs over detection window

HOG

The first stage of training process is the preparation of training data. 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. The positive window is the one ideally contains only one centered object, 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 a result, the classifier will be very sensitive with negative sample. For example, let take of one image, we can generate 10,000 windows. And we assume that the false positive rate (false positive is the item indicates that window is actual negative, but the classifier says it as positive) is small as 0.1%, it can generate 10 false positive windows which can severely effect on fusing all detections to create the final one.

In this stage, we use a method called Histogram of Oriented Gradients to map each training window to high dimensional feature vector. After that we use some Machine Learning technique to build up a classifier. 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.

We have to take into account that the number of negative windows is much larger than number of positive windows. However, it is impossible for us to put all positive and negative training windows into training set at one time because the shortage of memory of computer. That is the reason we have to train dataset multiple times. We will briefly the method which is used.

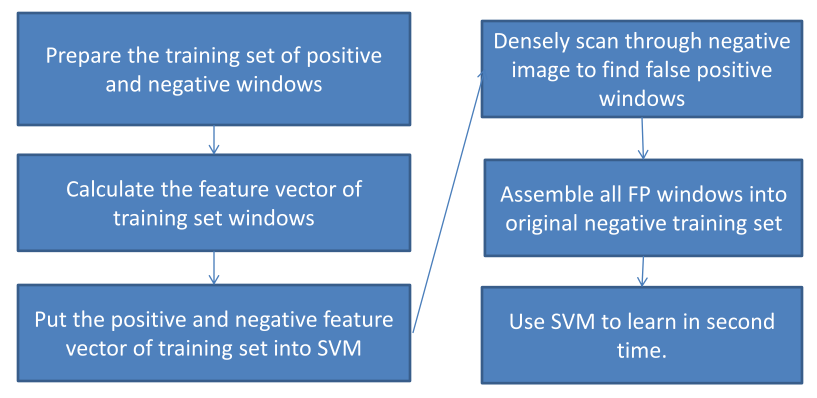


Figure : Hard sample generation overview.

*In other to make performance less sensitive to false positive rate, hard samples generation is employed. After create the initial linear SVM, all negative images are scanned to find out false positive windows.*

* Firstly, we gather whole positive training windows and some negative windows (the number of negative ones is as many as 5-7 times number of positive ones). Then we put them into SVM as a training data.
* Secondly, when SVM learn this dataset, it will output a classifier. This classifier cannot be used immediately because it will increase the false positive rate.
* Thirdly, we densely scan through whole set of negative images to find false positive windows.
* Fourthly, we concatenate all false positive windows derived from third step to the negative sample in training set. This stage is usually called “find hard negative sample”. And we use linear SVM to train this new training set to create new classifier.
* Finally, repeat third step if you wish to find more false positive windows. By experiment, the process of finding hard negative sample is employed one or two times will give good performance. Overview of feature sets

### Detection phase

Figure : Fusion algorithm overview.

*Many overlapping detection windows are fused together to finalize the bounding boxes.*

Scan image at all scale and location

Extract feature over windows

Run linear SVM classifier at all locations

Fuse multiple detections at all position and scale space

Object detection with boundary box

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. 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.

* 1. 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. 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:

* 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.
* Thirdly, 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 section 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.

Figure : Overview extraction algorithm Histogram of Oriented Gradient

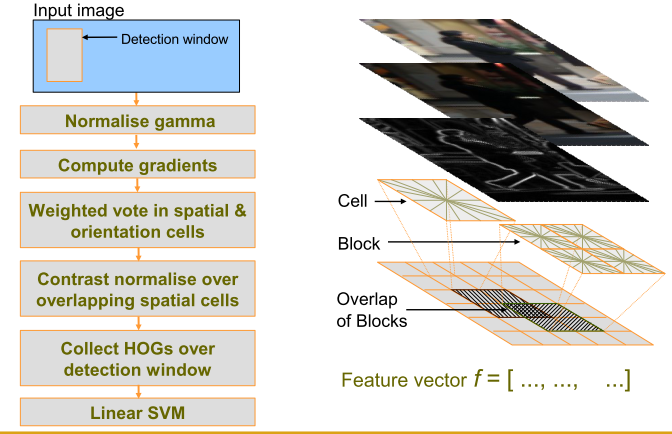


Figure : Workflow of HOG descriptor

1. *Normalize gamma: to reduce illumination effect. (2) Compute gradients: to extract contour and shape of object. (3) Weighted vote in spatial & orientation cells: extract information and store in histogram. (4)*

The HOG descriptor has several advantages which are:

* Capture 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.
* 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.
* Illumination invariant is assured by gamma normalization and contrast normalization.
* The overlapping blocks has a benefit that it allows little information can be missed during the encoding process.
  1. 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 and mean shift. And then, we decide to use Mean Shift because the number of object in image is unknown.

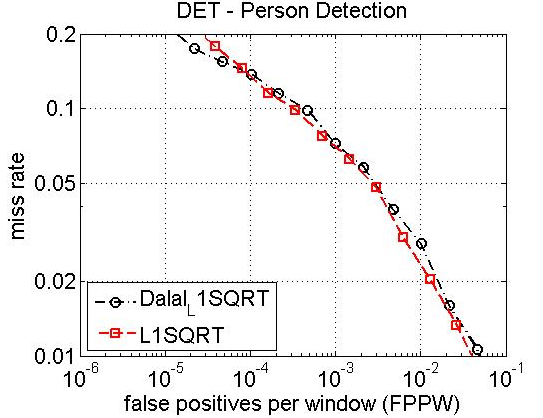
3.4 Overview of results

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 Dala’s work.

At first, we re-implement HOG detector of Dalal, and its performance is approximate 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.

(2 bieu do o day)



# HOG

In this section, we are going to introduce in details about 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. 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.

* 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:

* Rectangle HOG(R-HOG): it 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 ϛ x ϛ, η x η, β which are number of cells in one block, number of pixels in one cell, and number of bins respectively.
* Circular HOG(C-HOG): it 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.
* Bar HOG: 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.
* 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. To overcome this issue, in Center-Surround HOG, every cell is normalized just only one time. So it speeds up computation.
  1. 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.

### Color channel

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

### 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.

### 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: Sx = (vẽ 2 hình minh họa, )



Sy =

-1

1

Corresponding mask:

(2 mask)



Two sided: Sx =



Sy =

-1

0

1

Corresponding mask:

Note: ‘h’ is usually taken as 1.

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



* + Magnitude:



* + Orientation:

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.

### 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 0-180o (“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.

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

“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. 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.



Figure : Example of Spatial & orientation histogram

### Block Normalization, Block Size, and Overlap

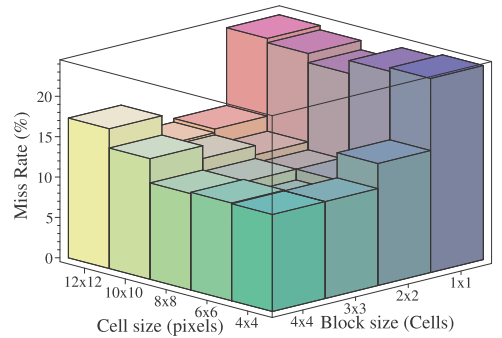
Block Normalization has a great effect on HOG descriptor. The fact that Gradients strengths vary from over a wide range due to local variations in illumination and foreground-background contrast. Hence, it is necessary to normalize block to get good performance. Block is a local group of cells, and each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

* L2 norm: **v** 🡨 sqrt(|**v**|22 + ε2)
* L­2-Hys: L2-norm by clipping and renormalizing.
* L1-norm: **v** 🡨 **v**/(|**v**|1 + ε)
* L1-sqrt: v 🡨 sqrt(**v**/(|**v**|1+ ε))

For pedestrian detection, the performance of L­2-Hys and L1-sqrt are equal, and they are outstanding to the rest. By experiment, using L1-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. 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.

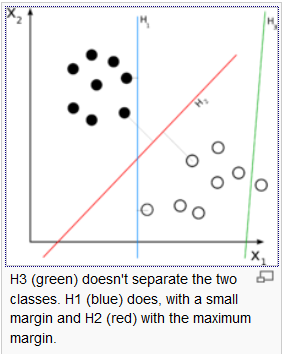


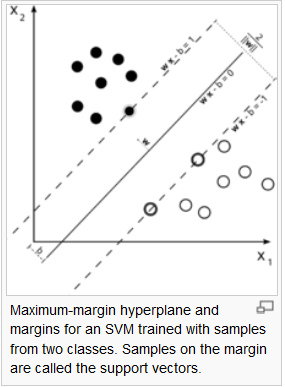
### Classifier

Recently, classifier is the very active filed drawing a lot of researchers’ attention. Building the robust classifier is the other main key in any detector because detector is nothing if we had good descriptor but bad classifier. Among many classifiers, SVM and adaboost have very outstanding performance. We choose linear SVM as our classifier because it is simple and we have work a lot on it. Although linear SVM is simple, it runs really fast and has good performance.

**Support vector machines** (**SVMs**) 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.





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.

* 1. Hard Samples

In HOG descriptor, a given image is densely and uniformly scanned at all locations and scales to detect positive windows. Because the size of detect windows are very small compared with the image, there are a lot of detect windows in one image. By experiments, we find out that an average image (450 x 600) has approximately 10,000 detect windows. Among these windows, just a few ones are positive, and almost are negative windows. Hence, HOG detector is very sensitive with false positive windows. As mentioned above, even a very small false positive per windows also adversely affects the performance of detector. Therefore, reducing the false positive rate is the vital thing to do.

Remind that HOG descriptor based on the well-normalized densely histogram of oriented gradients. HOG feature sets contain a group of very high dimensional vectors which occupy a lot of memory. When using SVM to learn, every feature vector has to be loaded into RAM to process. As consequent, we cannot take negative samples as many as we can because of limitation of hardware.

To overcome this obstacle, we employ a method called “Generating hard samples”. The flow work of it is illustrated below steps:

Positive samples

Negative samples

Training samples

Classifier

Training negative image set

False positive windows

* Firstly, training set is collected as the combination of positive and negative samples. Each of samples is a HOG description feature vector.
* Secondly, we use linear SVM to train the training set. The result is a classifier.
* Thirdly, we randomly pick up some points in negative image in negative sample set. After that we get a group of detection windows at all scales taken this random point as a center.
* Next, we use the classifier in second step to score the detection windows from third step. In this step, we are just interested in the window having negative score.
* Finally, we get all the false positive windows (window has negative score) and put them into the original negative samples. And we start the process again until we do not get new false positive ones, or the number of false positive ones is under some criteria.

Detector’s performance is boosted significantly during the process of finding out negative hard samples, approximately by 15% at 10-4 FPPW. We all know that the number of positive samples is limited because it takes a lot of effort to collect and annotate. On the other side, the negative samples are huge and definitely larger than positive ones. Negative samples can be collected from a lot of scenarios including indoor, outdoor, natural scene, street, and so on. Hence, high false positive per windows rate can severely affect detector’s performance.

Overall, the process of finding the hard negative samples (false positive samples) is essential for good performance detector. It can enhance a lot the performance of DET cure (Detect Error Tradeoff cure). 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. 4.4 Overall Results

Our main training and testing dataset is INRIA Pedestrian dataset. The performance will be evaluated through Detector Error Tradeoff (DET) curve.

4.5 Discussion

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, 200 wide bins give good result. And the range of orientation bins (00-1800 or 00-3600) 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 [Fischer 1973, Hubel and Wiesel 1974].

|  |
| --- |
| 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 ) 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 : HOG descriptor extraction algorithm

|  |
| --- |
| 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 the 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. |

# Some modified HOG

In this previous section, 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 methods 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.

Because of shortage of time and limitation of computer, we just only test these new approaches on pedestrian object which is people have up-right shape and full visible. We cannot guarantee whether these approaches can be applied to the other objects or not.

* 1. Four regions based 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.

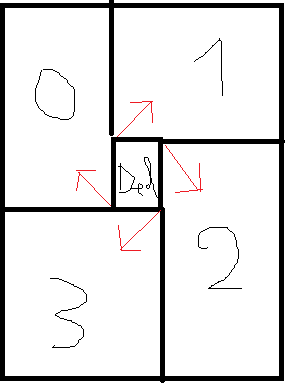


Figure : model of four regions based method

“Del” region is less informative than the others. We separately calculate histogram of these four regions, and concatenate them together.

By experiment, we observe that “Del” 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 “Del” 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%.

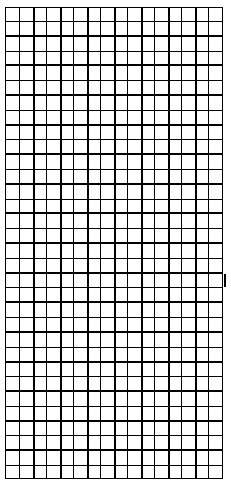


Figure : effect of size of "Del" region and overlap of regions on performance

* 1. Non-uniform based approach

The idea is similar to the four regions based approach, this approach also try to take advantages of less informative center region in the training image. However, instead of divide image into four region and ignore the central one, we use the distribution of grid points to catch this idea.

In the original HOG descriptor, image is divided into uniform grid of points. As a result, the author of HOG, Dalal, has assumed that every part in this image is equally important. However, in fact, the information about the background is usually less important than the body curve shape of human. Thus, from this observation, we devise a method call “non-uniform based approach” which is to concentrate grid points on the much informative regions and to loosely distribute grid points to less informative regions.



105 point

84 point

Unfortunately, the result of this approach from the experiment is below the original one. More specific, this performance is 7% lower than of the original. The reason probably is that out intuitive about less informative regions is wrong. Because of the shortage of time, we do not intensively evaluate this method on other case of grid points scatter.

* 1. Multi-level based approach

The original HOG method of Dalal is just one level. The drawback of this one is that it is less informative about structure of object. Consequently, I propose a new concept called “multi-level HOG”. The concept multi-level is not new. Many researchers use this one to enhance performance of certain extraction algorithm.

In paper “Classification using Intersection Kernel Support Vector Machines is Efficient”, author has proposed a method named multi-level which is calculated as following steps:

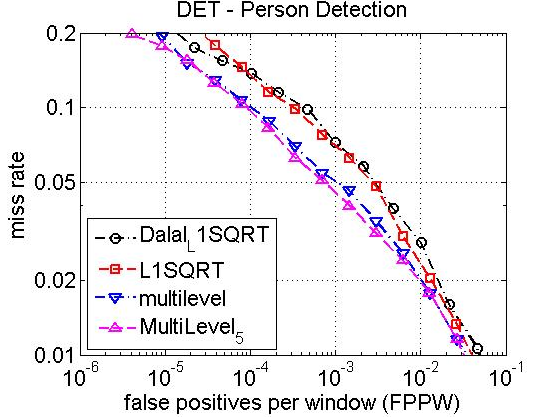
* Step 1: At level one, window is divided into non-overlap cells. And the feature vector of this level is the concatenation of HOG feature of each cell.
* Step 2: size of non-overlap cells is increased or reduced to form anther level. And it is similar to step 1 to calculate window feature at this level.
* 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:



* Calculate the gradient of each pixel.
* At reach level, we create uniform grid of points as above figure. 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.

Here are the results of multi-level approach compared with the original one. At 3 levels method, we enhance performance by 2%. Moreover, we intuitively think that we will get better performance as far as we increase the number of levels, but it is not hold. The performance of 5 levels method is approximately the same as 3 levels one.

# 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.

* 1. 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.

* 1. Mean Shift

### Brief introduction to Mean Shift

 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)}  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 ***K***(*xi* − *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:

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

### Pros and Cons of Mean Shift method

Pros:

* Does not require the prior knowledge of the number of clusters, and does not constrain the shape of clusters.
* 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.
* Mean Shift is more complex and slower than K-Mean.

### 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. |
|  |

# 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.

# Conclusion

# Future work