**Name: Đặng Trương Khánh Linh**

**ID: 0612743**

**Problem Statement**

My goal is to build a system that automatically detect and localize pedestrians in static image. One constraint is that pedestrians have to be full people, and the size is not less than 64 x 128.

***Input***

Static image.

***Output***

The boundary boxes contain pedestrians.

**Some popular approaches to this problem**

1. Raw intensities

Around more than 10 years ago, raw intensities based approaches has been used in object detection field. The most advantage of this approach is that it can be robust to noise of background. In this approach, there are two main popular extraction methods named SIFT and HOG.

Popular papers in this approach:

SIFT:

* + - Distinctive Image Features from Scale-Invariant Key-points, David G. Lowe, January 5, 2004.
    - SIFT: Scale Invariant Feature Transform, David G. Lowe 1999.

HOG:

* + Histograms of Oriented Gradients for Human Detection, Navneet Dalal and Bill Triggs.
  + Human Detection Using Oriented Histograms of Flow and Appearance, Navneet Dalal, Bill Triggs, and Cordelia Schmid
  + Fast Human Detection Using a Cascade of Histograms of Oriented Gradients, Qiang Zhu, Shai Avidan, Mei-Chen Yeh, Kwang-Ting Cheng

1. Oder of pixels

Instead of using gradient like above methods, this one use the order of pixels. Recently, this approach has been widely used for object detection and tracking algorithm. The algorithm which is used to extract feature in this approach called local binary pattern (LBP). In recent months, someone has defined another notion called local ternary pattern (LTP) and show that LTP is better than LBP.

1. Hybrid model

In hybrid model, they concatenate two histograms of two different extraction algorithms. Before doing concatenate, they prune some useless item in each histograms. The first advantage of this is that the program will run faster due to the short of dimension. The second advantage is that it enhances the accuracy of detection rate.

Challenges:

Image is just a matter of pixel, it lacks of motion knowledge like in video. Moreover, the variety of background and osculation can severely effect on the detection rate. Some approaches based on edge detection has been fail to resolve this problem.

In addition, pedestrians probably wear diversity of clothes and stand in different stances. This causes a lot of problems when we extract feature to build object/non-object classifier.

**Most popular approaches**

1. Histogram of Orientation Gradient

This method is introduced by Naveet Dalal in 2005, in paper “Histogram of Oriented Gradient for Human Detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)”. Later, Zhu has proposed a speed up version of this method, detail in “Fast Human Detection Using a Cascade of Histogram of Oriented Gradients, 2006”.

Some popular papers in this field:

* + Histograms of Oriented Gradients for Human Detection, Navneet Dalal and Bill Triggs.
  + Human Detection Using Oriented Histograms of Flow and Appearance, Navneet Dalal, Bill Triggs, and Cordelia Schmid
  + Fast Human Detection Using a Cascade of Histograms of Oriented Gradients, Qiang Zhu, Shai Avidan, Mei-Chen Yeh, Kwang-Ting Cheng

1. Local Binary Pattern

This is one method of approach based order of pixels. It is used to extract local feature of object which is robust to Gaussian noise.

Some popular papers in this field:

* + Ahonen, T., Hadid, A. and Pietikäinen, M. (2006), Face Description with Local Binary Patterns: Application to Face Recognition
  + Heikkilä, M., Pietikäinen, M. and Schmid, C. (2009), Description of Interest Regions with Local Binary Patterns. Pattern Recognition
  + 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.
  + 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.
  + 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
  + 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.
  + 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.
  + 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
  + 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
  + Discriminative Local Binary Patterns for Human Detection in Personal Album Yadong Mu1, Shuicheng Yan2, Yi Liu1, Thomas Huang3, Bingfeng Zhou1

1Peking University, 2National University of Singapore, 3University of Illinois at Urbana-Champaign

1. Hybrid system

In recent years, many researchers in computer vision try to combine two separate histogram form different extraction methods. The results of this approach seem to outperform other single extraction method. However, before concatenating two separate histograms, it is a need to carefully prune some useless features.

Some popular papers in this approach:

* An HOG-LBP Human Detector with Partial Occlusion Handling", Xiaoyu Wang, Tony X. Han, Shuicheng Yan, ICCV 2009
* Latent Hierarchical Structural Learning for Object Detection, Long Zhu.

**My approach**

My approach of building automatic system of detection pedestrians in static image is mostly method based Histogram of Oriented Gradients. The reasons are that this method is not too complicated and has high performance.

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 methodology of HOG

In an image, we use sliding window to densely scan at all position and different scales. In each window, we get its score, and we decide this window contains object or not is dependent on the threshold. After densely scanning image, we filter windows which exceed the threshold. We assume that our extract algorithm is robust which means that the score of window is still large (not maximum) even if this window slightly off center of object. And then, we use non maximal suppression (such as mean shift) to find the mode.

In process of extract a slide window to feature vector, we use robust extraction algorithm to avoid the noise effects of background. And HOG (histogram of gradient) seems to be the best choice because it is invariance with variety of background. In realistic, HOG is declared as the state of the art in extract feature vector to distinguish object or non-object.

INRIA pedestrian dataset is intensively used in training phase and testing phase. In training phase, there are 2416 positive windows and 1220 negative images; and there are 1126 positive windows and 453 negative images.

Figure 1: 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.



Pedestrian/ non-pedestrian classifier

**Overview of extraction algorithm**

* Normalize gamma (using square root or logarithm formula) in each channel color. In experiment, using square root in gamma correction process is the best.
* 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.

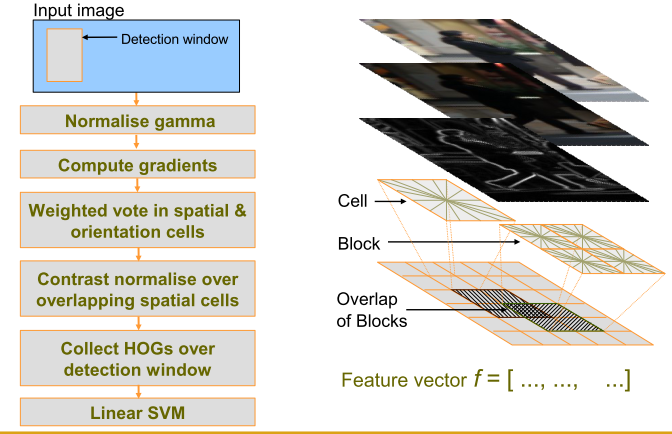


Figure 2: Overview extraction algorithm Histogram of Oriented Gradient

Figure 3: Example of Spatial & orientation histogram

Result

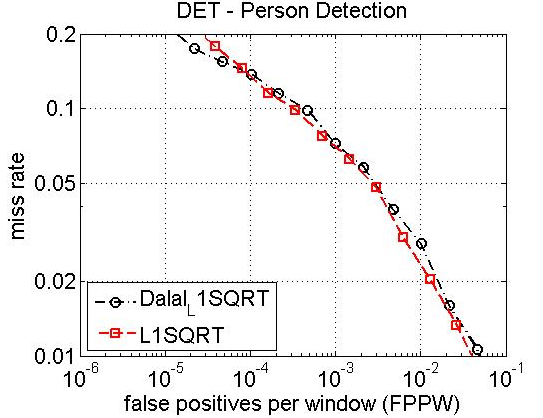


Figure : The red line is my own implementation of HOG. It approximate Dalal’s one (black line).

Details in my approach

Training phase

In training phase, the training dataset consists of 2416 positive windows and 1220 negative images. And there are 1126 positive windows and 453 negative images to validate. The size of window in positive and negative is 64 x 128 pixels. Negative windows are created by the following step:

* + - In negative image, we randomly choose some points.
    - Form those points, we take a widow which size is 64 x 128 pixels and have this point as a center.
    - We take several multi-scale windows which has the same center.

At each window, we use HOG algorithm to extract feature vector. The following diagram describes

Detection phase

Figure 5: Training phase of HOG algorithm

Input image

Scan image at any position

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

This phase is processed after we finish the training phase. The input data is an image, and the output is the image which has the bounding boxes containing pedestrian. Some sample output images are below.

Figure : Overview of detection phase

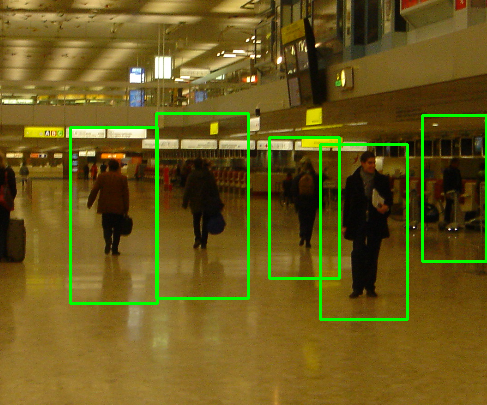
Scan image at all scale and location

Extract feature over windows

Run linear SVM classifier at all locations

Fuse multiple detections at all position and scale space

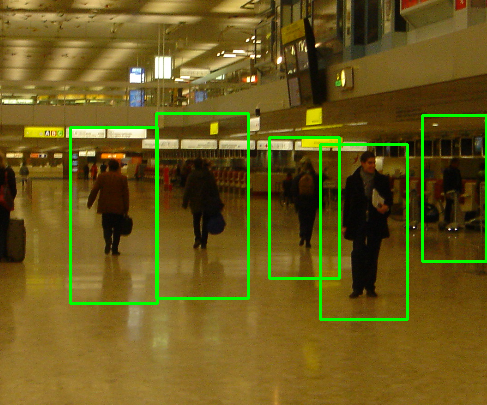
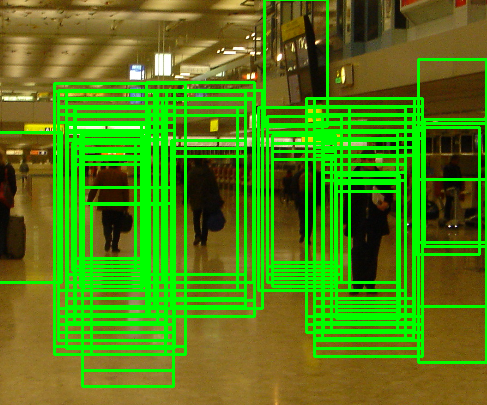
Object detection with boundary box

Generally, in detection phase, we densely scan whole the image to find the pedestrians. On the image, we build up a grid of points which are uniformly dense in this image. The gap between two adjacent points is equal to cell size (8 pixels). At each point in the grid, we get rectangle which has this point as a center. The initial rectangle we take has size 64 x 128. After that, we create several rectangles by enlarging the initial rectangle until it reaches the edge of image. These new rectangles have to have the same center as initial one. The multi-scale parameter in enlarging process is taken from 1.05 to 1.2 according to users.

After densely scanning through the image, we probably get a lot of positive windows. The number of positive windows may be larger than the number of pedestrians that image contains. The reason is that a window would probably be positive although it is slightly center off pedestrian.

As a result, one problem arises here is that how we can keep positive windows which have pedestrians at center, and eliminate the windows off center pedestrian. In order to resolve this problem, mean shift is the best candidate. Mean shift is the algorithm used to find the modes of scatter of points. Here is some example of using mean shift to get the bounding boxes.



Some slight contributions

1. Four regions based approach

In the first one, we assume that pedestrians in training or testing windows are central alignment (i.e people are in the center of windows). This condition is hold in MIT and INRIA pedestrian datasets. 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 curve information.

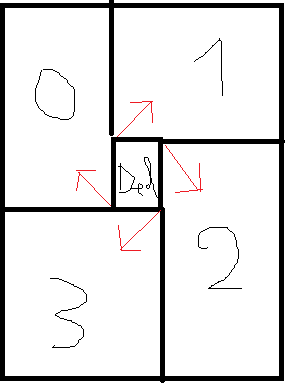


Figure 7: model of four regions based method

“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.



Figure 8: effect of size of "Del" region and overlap of regions on performance

1. Non-uniform based approach
2. 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. 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.

Result



Figure : Performance of multi-level HOG compare with original HOG

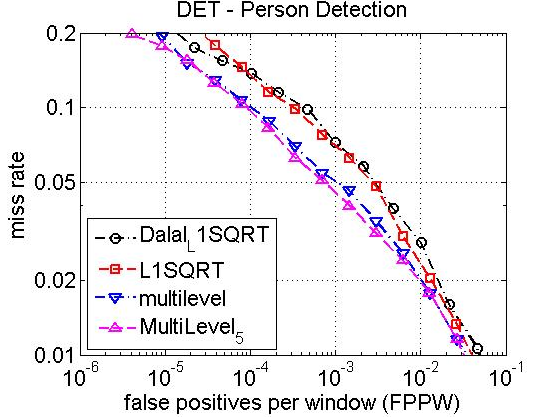


Figure : Two types of multi-level HOG: the blue one is three level, and the pink one is 5 level