Real-Time Bottle Detection Using Histogram of Oriented Gradients

Mahesh Jangid(✉), Sumit Srivastava, and Vivek Kumar Verma

SCIT, Manipal University Jaipur, Jaipur, India
mahesh_seelak@yahoo.co.in

1 Introduction

The object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos. The involvement of the computer vision is rapidly increased around us and also used for many purposes like pedestrian detection [3–5], vehicle detection [6], traffic signal recognition [7], fire detection [8], etc. The computer vision technology [9] is also being used with the robots to sense the environment and perform tasks accordingly. We are working on to develop a robot to serve a water bottle at the desk of person in the office to minimize the human affords that is basically object detection problem [10]. The bottle is to be detected and classified from a video feed from a nonstationary camera mounted on the top of the robot.

Owing to its age, this problem has a lot of literature published. Two of the most important approaches include using HAAR wavelet descriptors [11, 12] as input parameters or using part-based method containing detectors for various objects. An extension of this algorithm can be seen as the skeleton modeling done by Kinect using RGBD images. The approach we propose to implement is much simpler than the abovementioned methods and is proven to provide significantly higher performance in the real world.

The basic idea behind this approach is capturing the object appearance and shape by characterizing it using local intensity gradients and edge directions. The image is densely divided into small special regions called cells (Fig. 1).

For each cell, a 1-D histogram of gradient directions/edge directions is computed and later all cell data is combined to give a complete HOG descriptor of the window. The variety of colors and illumination in the surrounding make normalization inevitable. We further describe the normalization technique as a part of our approach later in the report. In their work, we make our own dataset for training purpose, which has a sufficiently large negative set by sampling out patches from bottle-free images. Figure 2 shows the direction information in each cell and next picture shows the feature descriptor information per block.

S. Bhattacharyya et al. (eds.), *Advanced Computational and Communication Paradigms*, Lecture Notes in Electrical Engineering 475,
https://doi.org/10.1007/978-981-10-8240-5_13

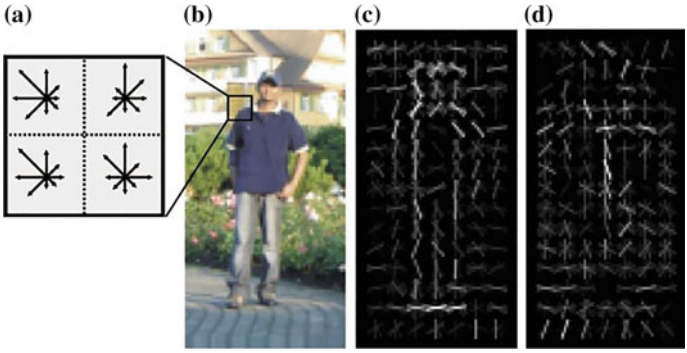

Fig. 1. Histogram of oriented gradients

2 Problem Statement

We aim to implement an object detection system for detecting and marking one or more bottles in a scene. This project is to serve the purpose of bottle detection and classification in a video feed captured by a robot in the office to serve the needs of person. This will help in automating the servant work and reduce human involvement as well as dependency.

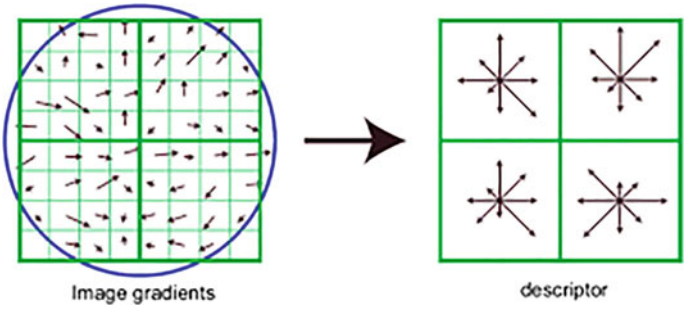

Fig. 2. Image gradients and spatial orientation binning

3 System Overview

3.1 Database

There is no standard dataset for bottles. The dataset has been prepared by capturing the images in the office. The positive and negative images (samples) were captured at the same time, which includes 500 positive and 500 negative pictures. Entire pictures have been normalized in 64 × 128 dimensions.

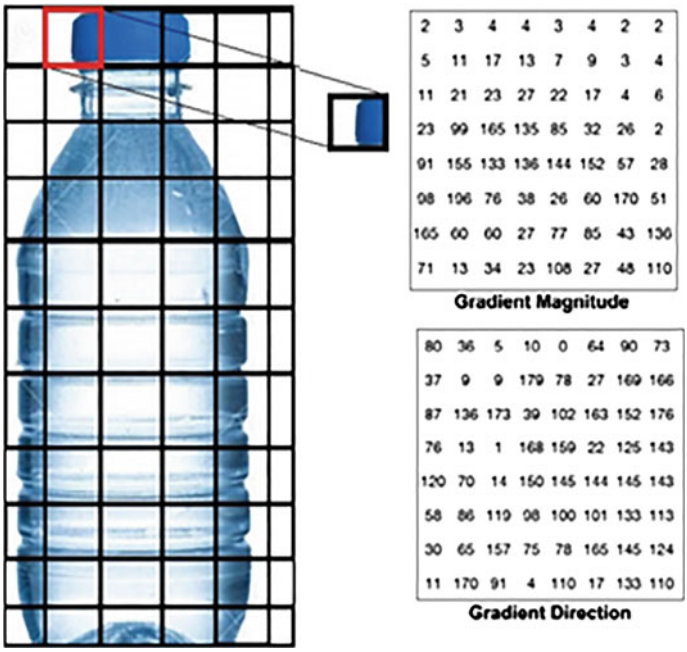

Fig. 3. Calculating the gradient direction and magnitude of the image and storing them in a vector

3.2 Preprocessing

We preprocessed the images in the grayscale space as the paper by Dalal and Triggs [3] gives no distinct advantage of using the RGB or LAB color spaces. Apart from that, we apply gamma normalization to improve the intensity of the image. This has been done as images clicked from camera devices that have low illumination.

3.3 HOG Feature Extraction

The following sections describe the HOG feature extraction procedure from scratch as implementation given by Dalal and Triggs [3]:

A. Gradient Computation
To compute the gradient of the image, we simply apply the point discrete derivative mask in both horizontal and vertical directions. This method requires filtering the intensity data of the image with the kernels [−1 01] and its transpose in both horizontal and vertical directions, respectively.

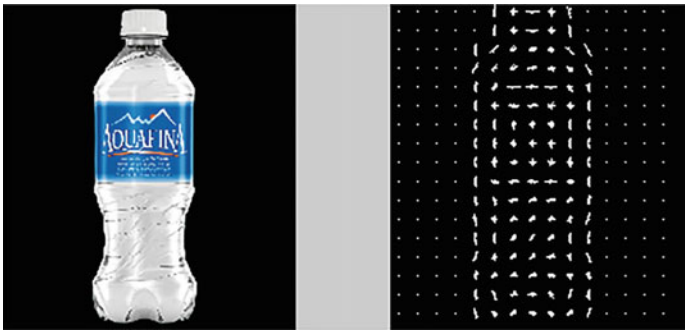

Fig. 4. HOG visualization of a bottle

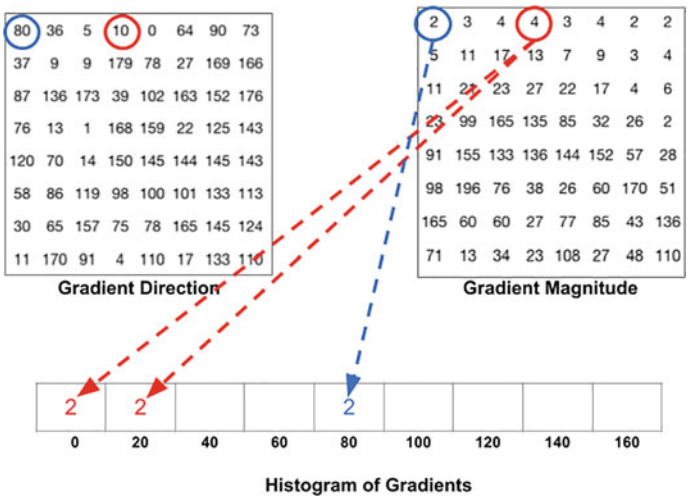

Fig. 5. Histograms calculation of each block

B. Orientation Binning

A cell histogram is created by weighted quantization of the orientation of each pixel of the cell into predefined orientation-based bins. The cells are usually square in shape (for convenience we will stick with rectangular) but they can be rectangular or circular. The weighting of the orientations can be either using the gradient magnitude itself. To calculate gradient magnitude and direction, we use

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x},$$

where g_x and g_y are the Sobel directional vectors. The gradient magnitude and direction are calculated by the given formulas and then stored in different vectors. Figure 3 shows the magnitude and direction information (Fig. 4).

C. Block Division and Normalization

The cells must be grouped together in order to factor in the changes in illumination and contrast. The complete HOG descriptor is then the vector of the components of the normalized cell histograms from all of the block regions as shown in Fig. 5. These blocks are normalized by four prominent methods: L1 norm, L1 norm square root, L2 norm, and L2 norm followed by clipping (L2 Hys). We experimented and choose the one that works best. Figure 4 shows the HOG visualization of a bottle.

The following formula is used for normalization of the blocks of the image:

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}},$$

where "v" be the non-normalized vector containing all histograms in a given block, is its *k*-norm for k = 1, 2, and e is a small constant.

D. Calculation of HOG Feature Vector

The final step collects the HOG descriptors from all blocks of a dense overlapping grid of blocks covering the detection window into a combined feature vector for use in the window classifier. We calculate the final feature vector for the entire image patch, and the 36 × 1 vectors are concatenated into one giant vector. There are 7 horizontals and 15 vertical positions of the 16 × 16 blocks making a total of 7 × 15 = 105 positions. Each 16 × 16 block is represented by a 36 × 1 vector. So when we concatenate them all into one giant vector, we obtain a 36 × 105 = 3780 dimensional vector. Training the classifier: Based on the literature survey done, we chose a linear kernel SVM for the classification purpose. SVM is one of the best classifiers used for the computer vision area.

E. Feature Dimension Reduction

HOG method produced a high-dimensional feature vector which needs more memory and computational power. So we reduced the feature using principal component analysis (PCA) which has been widely used for the feature reduction. PCA helps us to reduce features 3.7–1 K.

3.4 Sliding Window

To detect the bottle in a given image, we applied a sliding window approach to predict the presence of a bottle in a window which kept sliding over the complete image as shown in Fig. 6. The window has been shifted with a step length equal to the length and width of the block size, respectively, in the vertical and horizontal directions. This process is computationally heavy, given that the gradient magnitudes and orientations for each patch needs to be computed for each window during sliding. To speed up the process, we divide the whole image into the blocks of the given block size, and compute gradient histograms over them before applying the sliding window. Thus, after computing these histograms beforehand, now while applying sliding window,

we just need to consider the subset of blocks which belong to the window and concatenate their histograms to get our feature vector (Fig. 7).

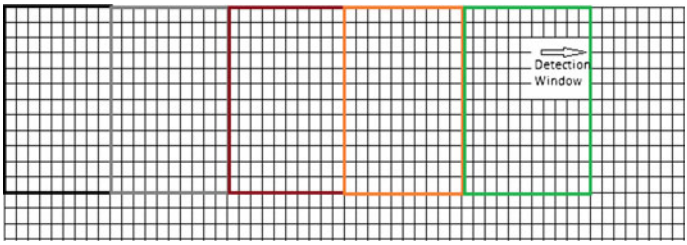

Fig. 6. Sliding window over the image

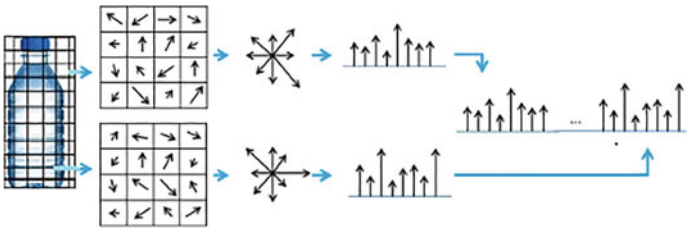

Fig. 7. HOG descriptor representation

4 Experimental Results

Our target is to detect bottles from an image taken from a camera. 100 experimental images were taken for the testing of the system. The images were processed on Intel(R) i5 processor at a clock speed of 1.60 GHz. During the calculation of HOG features, we tried several variations in its parameters.

First, we varied the number of bins used in the HOG descriptor. We used the values 5, 7, and 9 for the number of bins. With the bin number 5, the 180° gradient orientation range was divided into five segments of around 36° each. Out of the 100 test images, 85 successfully detected the bottle. The 7-bin system divided the 180° gradient orientation range into equal segments of around 25° each. This had a very good detection rate at 93 out of 100 bottles. The 9-bin system was tested. It divided the range into 9 segments of 20° each and detected 91 out of 100 bottles as shown in Table 1.

Table 1. Variation of accuracy with the variation of the bins

Bins	Block size	Cell Size	Overlap (%)	Accuracy (%)
5	2 × 2	8 × 8	50	85
7	2 × 2	8 × 8	50	**93**
9	2 × 2	8 × 8	50	91

Next, we changed the block overlap to 0%, which meant the blocks did not have any cells in common. The observation of this variation was very poor as many objects lay partly in multiple blocks, and they did not get detected. As shown in Table 2, the detection rate was 75 out of 100, whereas with 50% the detection rate was 93 out of 100. Lastly, we varied the cell size. We changed the cell sizes as 2×2, 8×8, 16×16, and 32×32 and calculated the HOG feature with 50% overlap. The results are shown in Table 3. The HOG value of each block is accumulated into a single value; here, HOG value of each block was too muddled to detect the bottles accurately. The 8×8 cell size provided a much better detection rate as compared to other cell size.

Table 2. Variation of accuracy with the variation of block overlap

Bins	Block Size	Cell Size	Overlap (%)	Accuracy (%)
7	2×2	8×8	50	**93**
7	2×2	8×8	0	75

Table 3. Variation of accuracy with the variation of the cell size

Bins	Block size	Cell size	Overlap (%)	Accuracy (%)
7	2×2	4×4	50	88
7	2×2	8×8	50	**93**
7	2×2	16×16	50	90
7	2×2	32×32	50	77

5 Conclusions

The computer vision is rapidly involving every sector owing to the security and the vision power as human being. This paper primarily focused on the water bottle detection to reduce the human involvement and dependency on him. We used the HOG features for this purpose and got the satisfactory results. We also found that the HOG feature works well with the cell overlap and performed badly without it. We considered the water bottles of different colors, sizes, and shapes that why the accuracy is around 93. Our future work will be in the same direction to improve the detection rate and also toward the experiment with other object detection approaches to make an independent hardware for the robot.

References

1. Mohan A, Papageorgiou C, Poggio T (2001) Example-based object detection in images by components. PAMI
2. Lowe DG (2004) Distinctive image features from scale-invariant key points. IJCV 60(2):91–110

3. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005 (CVPR 2005), vol 1. IEEE, pp 886–893
4. Ye Q, Jiao J, Zhang B (2010) Fast pedestrian detection with multi-scale orientation features and two-stage classifiers. In: Proceedings of the IEEE international conference on image processing
5. Suard F, Rakotomamonjy A, Bensrhair A, Broggi A (2006) Pedestrian detection using infrared images and histograms of oriented gradients. IEEE
6. Gavrila DM, Philomin V (1999) Real-time object detection for smart vehicles. In: Conference on computer vision and pattern recognition (CVPR)
7. Kassani PH, Teoh ABJ (2017) A new sparse model for traffic sign classification using soft histogram of oriented gradients. Appl Soft Comput 52:231–246
8. Chen T-H, Wu P-H, Chiou Y-C (2004) An early fire-detection method based on image processing. In: 2004 international conference on image processing, 2004 (ICIP'04), vol 3. IEEE, pp 1707–1710
9. Ren X, Ramanan D (2013) Histograms of sparse codes for object detection. In: Proceedings of the IEEE international conference on computer vision and pattern recognition
10. Papageorgiou C, Poggio T (2000) A trainable system for object detection. IJCV 38(1):15–33
11. Amit Y (2002) 2D object detection and recognition: models, algorithms and networks. MIT Press, Cambridge, MA
12. Viola P, Jones MJ (2004) Robust real time face detection. Int J Comput Vis