Real Time Pedestrian Detection Using CENTRIST Feature with Distance Estimation

Kaushal Joshi, R. Kavitha and Madhu S. Nair

Abstract Pedestrian Detection (PD) is an active research area for improving road safety. Most of the existing PD system does not meet the demanded performance. This paper presents a working PD system which improves performance. The system uses CENTRIST feature extractor and the linear Support Vector Machine (SVM) for training and detection of pedestrian. CENTRIST is very easy to compute without any preprocessing and normalization that makes it suitable for on-board system. During the training procedure, we exhaustively searched for negative samples. Detection results on INRIA dataset are more accurate compared to benchmark method HOG. We used monocular camera to estimate pedestrian distance which is fairly accurate. We apply our detector on real-time video without region of interest (ROI) selection and could achieve 7 fps detection speed.

Keywords ADAS · Pedestrian detection · CENTRIST · SVM · Distance estimation · Monocular camera

1 Introduction

Due to tremendous growth of automobile industry over the last century, road accidents have become an important cause of fatalities. In 2012, 4743 pedestrians were killed and an estimated 76,000 were injured in traffic crashes in the United States. On an average, a pedestrian was killed every 2 h and injured every 7 min in

K. Joshi · M.S. Nair (✉)
Department of Computer Science, University of Kerala, Kariavattom,
Thiruvananthapuram 695581, Kerala, India
e-mail: madhu_s_nair2001@yahoo.com

K. Joshi
e-mail: kaushalknack7210@gmail.com

R. Kavitha
Tata Elxsi Limited, ITPB Road Whitefield, Bangalore 560048, India
e-mail: kavitha.r@tataelxsi.co.in

R.K. Choudhary et al. (eds.), *Advanced Computing and Communication Technologies*,
Advances in Intelligent Systems and Computing 452,
DOI 10.1007/978-981-10-1023-1_23

traffic crashes [1]. So we must have a reliable safety system which help to avoid the road accidents. Society also expect more intelligent vehicles, which are capable of assisting the driver in driving process. Such intelligent vehicles are equipped with Advanced Driver Assistance System (ADAS). ADAS is a system which control vehicle to improve road safety. While driving it assist driver by visual or audio or vibrate alarm. ADAS is a collection of subsystems like Pedestrian detection, Collision avoidance, Traffic sign recognition, Lane departure assistance, Adaptive cruise control and Parking assistance.

Pedestrian detection for ADAS is still a very challenging task. Pedestrian appears very different in different conditions like change in pose, different clothing style, carrying some object, having different size, weather conditions and cluttered background. So, developing a working system which overcome all of these challenges is not trivial. In the last decade, there has been a significant progress within pedestrian detection. The quality of pedestrian detection system depends on the features extracted, classifiers and datasets used. In 2003, Viola et al. [2] presented a method which uses intensity and motion information as features and trained a detector using AdaBoost. Dalal and Triggs [3] presented Histograms of Oriented Gradients (HOGs) feature and trained a liner SVM classifier. HOG became most popular feature for pedestrian detection. After HOG-SVM, other authors showed more detection accuracy using different features and classifier combination in [4–6]. Some recent works have got good pedestrian detection accuracy like in Zhang et al. [7], Benenson et al. [8] and Costea and Nedevschi [9], where [7] is based on informed haar feature and AdaBoost classifier with maximum detection accuracy on every pedestrian dataset. In [8] detector is based on HOG and color based feature using linear SVM classifier [7, 9] needs GPU for fast pedestrian detection. These detectors are still far from desired performance for on-board pedestrian detection system.

There is a lack of complete working PD system with balanced accuracy and speed of detection. After detection of pedestrian, we must give distance information to system to take appropriate action (i.e. give alarm to driver or decrease speed automatically). Combination of multiple features makes system complex and slow during detection. So we present here a working PD system which uses a single feature and achieve fast detection speed. We showed that proper training leads to better detection accuracy, and using monocular camera we can get almost correct distance.

2 Proposed Work

The paper describes a real-time PD scheme that uses CENTRIST feature proposed by Wu and Rehg [10]. To classify the extracted CENTRIST features, the method uses linear SVM [11]. The system uses Efficient Sub-window Search (ESS) by Lampert [12] which help to improve performance. It is also uses Non Maximum Suppression (NMS) algorithm to eliminate multiple detection. This system uses

Fig. 1 **a** Sample image dataset from INRIA and **b** sample image dataset from MIT

camera geometry to estimate distance between camera and pedestrian using monocular camera. The system detect pedestrians and also estimate the distance in each video frame or image.

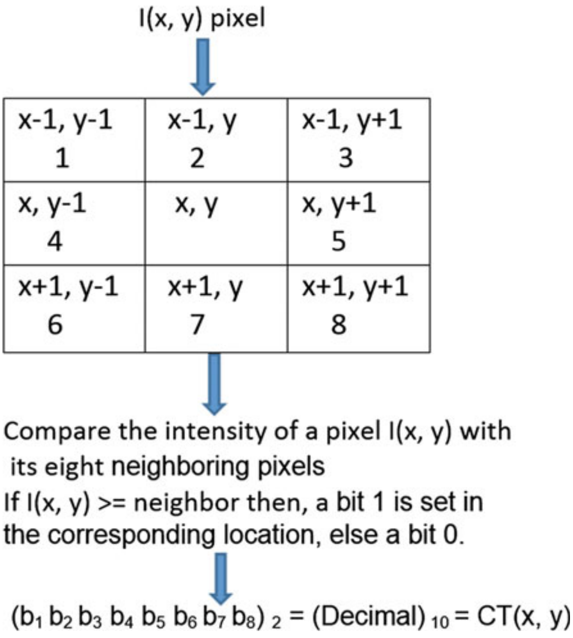
2.1 Training Phase

Training phase has two modules: feature extraction and classifier learning. The training samples are collected from INRIA [13] and MIT datasets [14], and some of the sample images are shown in Fig. 1.

This system used CENTRIST (CENsus TRansform hISTogram) [8] feature vector for training the linear SVM classifier. Earlier, CENTRIST was developed as a visual descriptor for recognizing topological places or scene categories. CENTRIST mainly encodes the structural properties within an image and suppresses detailed textural information. To get the Census Transform (CT) value of a pixel, compare the intensity with its eight neighboring pixels. If the center pixel is greater than or equal to one of its neighbors, a bit 1 is set in the corresponding location. Otherwise a bit 0 is set. The generated eight bit can be put together, which is consequently converted to a base-10 number (CT value). Figure 2 shows the conversion of a pixel value to CT value. Convert all pixel intensity values to CT values in the image and the resulting image is called a CT image.

For training, the method used small gray-scale image patches with 108 × 36 resolution and converted these image patches to CT images. The CT image is then divided into 9 × 4 blocks. Assuming 2 × 2 neighbor block as a super-block, extract 256 bin histogram of CT value from every super-blocks. Super-block is 50 % overlapped to its neighboring super-blocks. Concatenate all histograms and form a 6144 (24 × 256) dimension feature vector, which h represent an image patch. Generate 6144 dimension feature vectors for all training samples. Figure 3 shows an arrangement of blocks and super-blocks for an image patch.

Fig. 2 Convert a pixel value to CT value



I(x, y) pixel

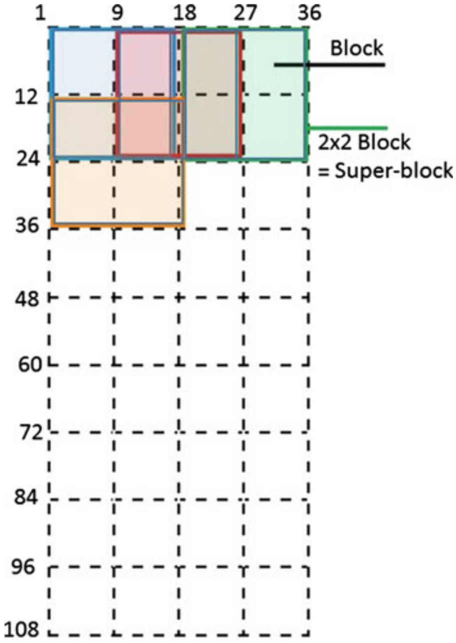
x-1, y-1 1	x-1, y 2	x-1, y+1 3
x, y-1 4	x, y	x, y+1 5
x+1, y-1 6	x+1, y 7	x+1, y+1 8

Compare the intensity of a pixel I(x, y) with its eight neighboring pixels

If I(x, y) >= neighbor then, a bit 1 is set in the corresponding location, else a bit 0.

$(b_1\,b_2\,b_3\,b_4\,b_5\,b_6\,b_7\,b_8)_2 = (\text{Decimal})_{10} = CT(x, y)$

Fig. 3 Divide 108 × 36 pixel CT image into 9 × 4 blocks = 24(8 × 3) super-blocks



2.2 Detection Phase

The detection phase is comprised of five modules (i) image scaling, (ii) feature extraction, (iii) classification, (iv) NMS and (v) distance estimation as explained below.

Image Scaling To perform a full image detection, we slide a window over the whole image and resize the input image to detect pedestrians of different scales. Pedestrian close to the vehicle appear with more pixel height compared to the pedestrian far from the vehicle in the image. To detect pedestrians who are close to the vehicle, down-scale the input image and then search for pedestrians.

$$Scale(I, S) = I_1; (I_1, S) = I_2; \ldots (I_{n-1}, S) = I_n, \text{ where } I_n > \text{ window size} \quad (1)$$

Scale (I, S) is a scaling function which resize the image, I. S = (S_x, S_y) is a scaling factor along the horizontal axis (S_x) and vertical axis (S_y). One criterion to be satisfied here is that the size of the scaled image (In) must be greater than the detection window size.

Feature Extraction In the proposed method, we used CENTRIST feature for pedestrian detection, as it is easy to compute, fast and there is no need of any pre-processing. If we replace all bit 0 to 1 and 1 to 0 in Fig. 2 (8 bit binary sequence) then the intermediate result is Local Binary Pattern (LBP) [15]. The superiority of CENTRIST over well established conventional feature such as LBP is that, CENTRIST encode global structure of pedestrian better than LBP.

After rescaling the image, convert whole grayscale image into a CT image by modifying all pixel values to CT values as explained in Sect. 2.1 and Fig. 2. Now take the CT image patch of size 108 × 36 as search window, and extract CENTRIST feature from the search window. Here search window size is same as the training image patches shown in Fig. 3.

Classification Let C represent the scaled CT image corresponding to the input image I. Using a search window, extract 6144 dimension feature vector (i.e. f $\in$ R^{6144}). If we have already trained a linear classifier w $\in$ R^{6144}, then a search window is classified as an object of interest if and only if $w^T f \geq 0$. Inspired by Efficient Sub-window Search (ESS), Wu et al. [16] proposed an algorithm to compute $w^T f$ using a fixed number of machine instruction, without generating feature vector *f*. Generate an auxiliary image *A* by replacing the values of the CT image by its corresponding weight value w_i (obtained from trained classifier). Then the *score* is calculated as the sum of all the values of *A*, which is equivalent to $w^T f$. If the score is greater than 0, then a pedestrian is detected in the search window. So based on the coordinate of search window, draw a bounding box on the image, which surrounds the detected pedestrian.

Non Maximum Suppression (NMS) The system uses NMS to avoid multiple detections of same pedestrian. Multiple detection may occur because of (i) multi-scale search and (ii) sliding window approach, which leads to more than 90 % overlap to its neighbor. The method uses intersection based multiple detection removal and keep only a bounding box having highest detection score among all intersecting bounding boxes. Suppose we have 2 detection box A and B, where coordinate of box is (top, left, bottom, right = x_1, y_1, x_2, y_2) then:

$$\begin{aligned}\text{Intersection area}, I = {} & \max(0, \min(A \cdot x_2, B \cdot x_2) - \max(A \cdot x_1, B \cdot x_1)) \\ & \times \max(0, \min(A \cdot y_2, B \cdot y_2) - \max(A \cdot y_1, B \cdot y_1))\end{aligned} \tag{2}$$

Distance Estimation Distance estimation is a crucial step in PD systems as it is needed to take appropriate action in time (i.e. giving alarm to driver, automatically decrease the speed of vehicle etc.). The method uses monocular camera for pedestrian detection and distance estimation. Using camera similar triangle property in Eq. (3), we can find the distance from camera to pedestrian based on the following mathematical relationship:

$$\frac{\text{Pixel height of pedestrian}}{\text{Focal length of camera}} = \frac{\text{Pedestrian height in real world}}{\text{Distance from camera to pedestrian}} \tag{3}$$

Focal length is a camera specific parameter. To find the focal length of the experimental camera, we captured an object of known size at a known distance. In Fig. 4, the object (paper) of length 30 cm is placed at a distance of 100 cm from the camera. The pixel length of the object can be found out from the captured image (say x), then the focal length = (x × 100)/30.

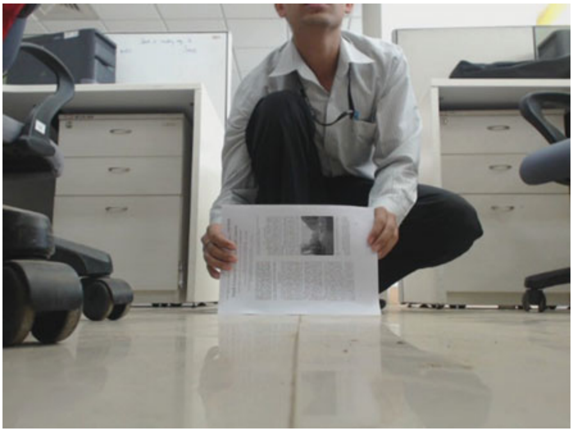


Fig. 4 Captured object at known distance to find the focal length of the camera

3 Experimental Analysis

To measure the performance of the proposed method, we used the ground truth of INRIA dataset and the matching criteria specified in [17]. A detection window R_d and a ground truth window R_g is considered as similar if:

$$\frac{Area\left(R_g \cap R_d\right)}{Area\left(R_g \cup R_d\right)} \geq 0.5 \tag{4}$$

For performance evaluation we used the standard metrics such as (i) False Positives Per Image (FPPI) and (ii) Miss Rate (MR). FPPI indicates the average number of false windows present in one image. Miss rate is the ratio of missed pedestrians and total pedestrians in the test dataset. MR and FPPI can be defined as:

$$MR = \frac{Total\,Missed\,Pedestrians}{Total\,Pedestrians} \tag{5}$$

$$FPPI = \frac{Total\,False\,Positives}{Total\,Number\,of\,Images} \tag{6}$$

The proposed method has been compared with the standard HOG based detector based on the quantitative metrics mentioned above. To compare the detectors we plot miss rate against false positives per image (using log-log plots) by varying the threshold on detection score, as shown in Fig. 5. It is evident from the figure that the proposed method have lesser miss rate (0.18) compared to HOG (0.23).

To check the detection speed we applied our method on a real video with 640 × 480 resolution. The quantitative analysis of standard HOG and the proposed method using different performance metrics are shown in Table 1. The proposed method achieved a detection speed of 7 fps and better detection accuracy rate of

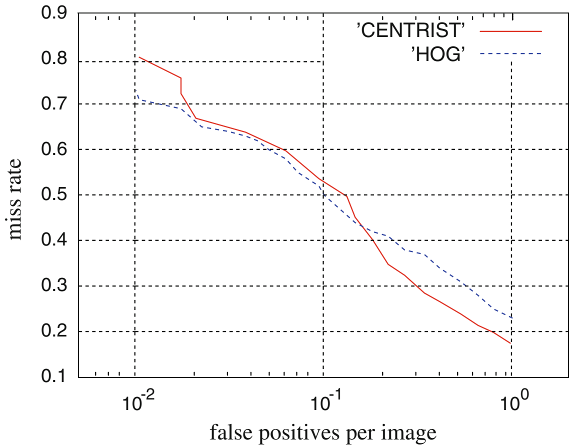
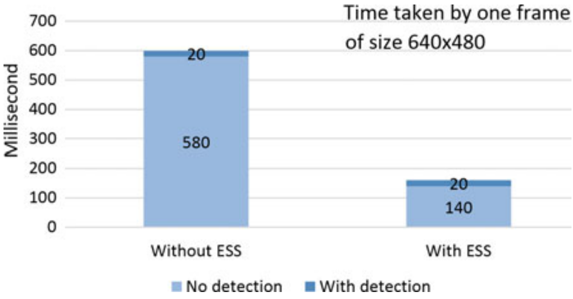


Fig. 5 Performance on the INRIA dataset

Table 1 Detection on 640 × 480 video frames

Methods	Miss rate	Detection accuracy	Average detection time (fps)
HOG	0.23	77	0.239
Proposed method	0.18	82	6.8

Fig. 6 Time to process one frame

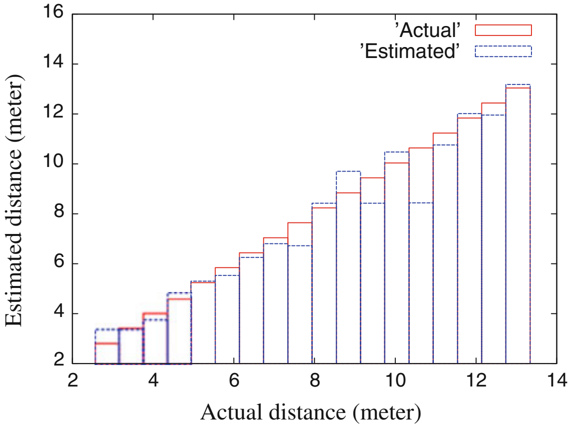


82 % compared to HOG based system with a speed of 0.239 fps and a detection accuracy rate of 77 %, for 640 × 480 resolution frames.

We trained a linear SVM classifier with 3341 positive samples from INRIA (2416) and MIT (925) datasets and an initial negative set of 12,180 patches. We used training methodology explained in [3]. During detection we used ESS technique which speed-up the detection process by 3–4 times as shown in Fig. 6.

To check the accuracy of the distance estimation, we found the average distance error, by comparing the estimated distance and the actual distance, which is approximately 0.4994 m. Figure 7 shows the difference between actual and estimated distances of some experimental samples. Detection result obtained based on

Fig. 7 Plot of actual and estimated distance



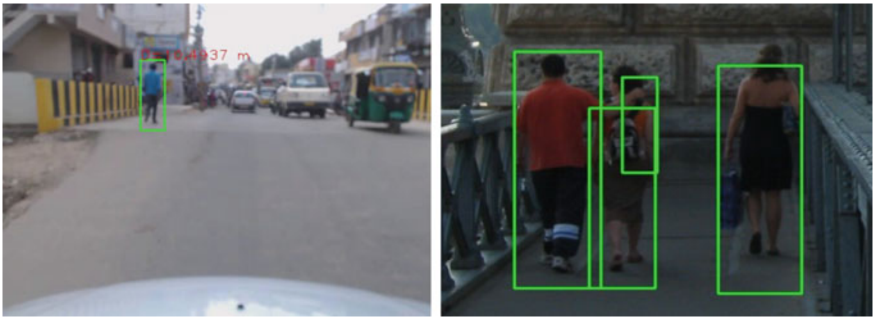


Fig. 8 Detection on real video frame and INRIA dataset image with low illumination

real video and INRIA dataset, shown in Fig. 8, proves that the system works fine in different illumination and blurring conditions.

4 Conclusion

In this paper a working PD system using CENTRIST feature is proposed, which is fast and doesn't need any pre-processing. Through experimental analysis it has been found that the proposed system is robust to illumination changes and blurring conditions. Combination of CENTRIST and linear SVM is sufficient enough to discriminate between pedestrian and non-pedestrian, if training is proper. Through experimental analysis it is proved that the proposed method works well in situation where the camera may lost its focus because of shake resulting in blurred video being recorded. As a future work, we are planning to restrict the search window to ROI so that the detection speed can be enhanced to 3–4 times faster, compared to the current detection speed.

References

1. Traffic Safety Facts. http://www-nrd.nhtsa.dot.gov/Pubs/811888.pdf (2015). Accessed on 05 Aug 2015
2. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. CVPR **2**, 734–741 (2003)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR **1**, 886–893 (2005)
4. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: BMVC, pp. 91.1–91.11 (2009)
5. Dollár, P., Belongie, S., Perona, P.: The fastest pedestrian detector in the west. In: BMVC, pp. 68.1–68.11 (2010)

6. Benenson, R., Mathias, M., Timofte, R., Van Gool, L.: Pedestrian detection at 100 frames per second. In: CVPR, pp. 2903–2910 (2012)
7. Zhang, S., Bauckhage, C., Cremers, A.B.: Informed haar-like features improve pedestrian detection. In: CVPR, pp. 947–954 (2014)
8. Benenson, R., Mathias, M., Tuytelaars, T., Van Gool, L.: Seeking the strongest rigid detector. In: CVPR, pp. 3666–3673 (2013)
9. Costea, A.D., Nedevschi, S.: Word channel based multiscale pedestrian detection without image resizing and using only one classifier. In: CVPR, pp. 2393–2400. (2014)
10. Wu, J., Rehg, J.M.: CENTRIST: A visual descriptor for scene categorization. PAMI **33**(8), 1489–1501 (2011)
11. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning, vol. 20, no. 3, pp. 273–297. Springer (1995)
12. Lampert, C.H., et al.: Efficient subwindow search: a branch and bound framework for object localization. PAMI **31**(12), 2129–2142 (2009)
13. INRIA Person Dataset. http://pascal.inrialpes.fr/data/human/ (2015). 05 Aug 2015
14. MIT Pedestrian Data. http://cbcl.mit.edu/software-datasets/PedestrianData.html (2015). Accessed 05 Aug 2015
15. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI **24**(7), 971–987 (2002)
16. Wu, J., Liu, N., Geyer, C., Rehg, J.M.: A real-time object detection frame work. IP. IEEE Trans. **22**(10), 4096–4107 (2013)
17. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. PAMI **34**(4), 743–761 (2012)