# DETECTION OF SUDDEN PEDESTRIAN CROSSINGS FOR DRIVING ASSISTANCE SYSTEMS

* Y.Ramesh Reddy[1], Dr.R.V.Krishnaiah[2]
[1]PG Student (M.Tech VLSI&ES), Dept. of ECE, DRK Institute of Science & Technology, Hyderabad, AP, India
[2]Professor, Dept. of ECE, DRK Institute of Science & Technology, Hyderabad, AP, India

**Abstract:** The main aim of this paper is to design "DETECTION OF SUDDEN PEDESTRIAN CROSSINGS FOR DRIVING ASSISTANCE SYSTEMS". This application has two major requirements: to detect crossing pedestrians as early as possible just as they enter the view of the car-mounted camera and to maintain alarm System to alert the Driver .This system takes capture image by means of web camera connected to ARM microcontroller through USB and the image is processed by using image processing technique. In this paper we also use S3C2440 based microcontroller, which is the current dominant microcontroller in mobile based products.

**Keywords:** ARM, S3C2440.

## 1. INTRODUCTION

We are aware of the problem of detecting sudden pedestrian crossings to assist drivers in avoiding accidents. ARM 32 bit Microcontroller has feature of image/video processing by using various features and classification algorithms have been proposed for pedestrian detection. It overcomes the performance in terms of sensors and hardware cost is also too high. So, our design Embedded system that detects partially visible pedestrians just as they enter the camera view, with low false alarm rate and high speed. When any pedestrian is detected it alerts driver by providing alarm sound and also it stops vehicle automatically. The display unit in vehicle provides clear details at position it detects pedestrian either right or left.

Our work is motivated by two factors. One is that the proposed problem has great social meaning and application value. According to the traffic safety data from the National Highway Traffic Safety Administration and the EU , many people are killed/injured each year in pedestrian–motor collisions; most of which occur when pedestrians attempt a road crossing at nonintersections. Second, the proposed problem has special requirements that make it different from existing related research. Drivers must be alerted to crossing pedestrians as early as possible for evasive maneuvers to be most effective. For this, crossing pedestrians should be identified even before they come into full view. The need for a combination of high processing speed, detection of partially visible pedestrians as they enter the scene, rejection of static or exiting pedestrians, and handling of unconstrained camera motion distinguishes this application from related work on event/action detection, generic (image/video based) pedestrian detection, and even current methods in intelligent vehicle systems.

## 2. PROPOSED METHOD

To detect pedestrian crossing events, we propose a three-level coarse-to-fine approach based on sliding windows, the levels are defined as the following: 1) sparse sliding window sampling with a motion filter based on a new LBP difference feature at the local level; 2) coarse detection and rough localization at the frame level; and 3) spatiotemporal refinement and fine localization at the video level.

### Local Level

At the local level, we search for pedestrians using a bundle of sliding windows at different scales. For efficient yet accurate operation of sliding windows, we take advantage of three properties of sudden pedestrian crossings. Since a crossing pedestrian need not be detected in every frame of the event in order to issue an alert to the driver, a sparse sliding window scanning strategy is used to improve processing speed. In addition to this, only sliding windows that contain significant motion according to an LBP difference feature need to be considered since crossing pedestrians must be moving. Third, our system examines windows only at the left or right edges of each frame, where sudden crossings of nearby pedestrians are presumed to begin.

In this paper, a new LBP difference feature is used to detect regions with significant motion. For each frame, the LBP8,1 transformation is performed on each pixel in the ROI that covers all of the sliding windows, and then, the LBP histogram of each of these windows is efficiently calculated using integral images, which are computed concurrently with the LBP transformation and is normalized by its L1-norm. The histogram is subtracted from a cached LBP histogram of a previous frame to obtain the LBP difference. If the magnitude of the LBP difference in a sliding window is larger than a threshold f, which is empirically set in our experiments, this sliding window will be tagged for further processing, and the cached LBP histogram of this sliding window is replaced by the present one. The cached histogram is also replaced by the current one if its corresponding window was not tagged in the previous θf frames (six in our implementation).

### Frame Level

The sliding windows that pass the motion filter are then passed to a cascade of classifiers for coarse detection and

International Journal of Instrumentation Electrical Electronics Engineering
p- ISSN: 2321 – 0621 e- ISSN: 2321- 063X

rough localization. First, a HOG+LinSVM classifier is used because HOG and HOG-like features have demonstrated great success in various object detection problems, especially in pedestrian detection. For HOG feature extraction, the gradient of each pixel in the windows is computed, and then, the gradient magnitude is inserted into one of the nine histogram bins that span a 180◦ range. In these histograms, an 8 × 8 cell size is used, and 2 × 2 cells form a block. Each block half overlaps each of its neighbors and is normalized using the L2-norm. The final HOG vector is composed of all normalized block histograms, with a total dimension of 1620 for the 128 × 32 sliding windows employed in our work. Since the Y -axis shift of our sliding window sampling strategy is exactly one block size, the computation of HOG features can be expedited by saving and reusing the feature computation and normalization among overlapping blocks.

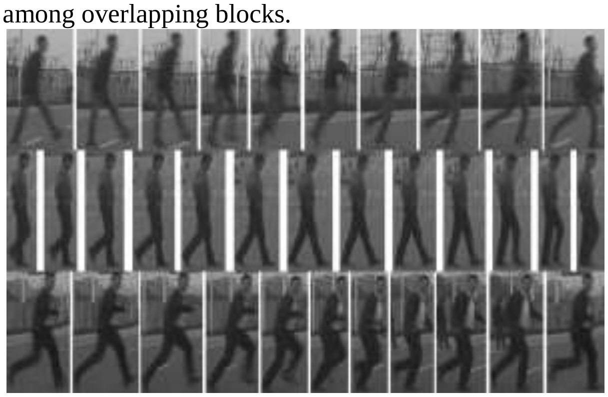


**Figure 1 Examples of various pedestrian entering styles. The first row illustrates**

Since HOG features represent only edge information, we additionally utilize a texture feature to obtain lower false positive rates. As reported in, the combination of HOG and Haar wavelets achieves its best performance with the help of Adaboost and bootstrapping.

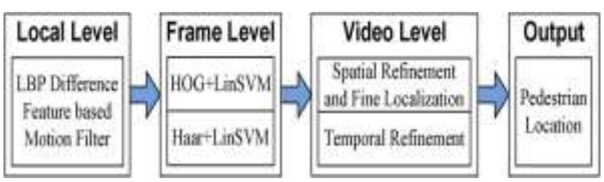



**Figure 2 Flowchart of the proposed three-level framework.**

However, since Haar wavelets are densely sampled, they require much computation time. For efficiency, we only use global Haar wavelet features with a dimension of 128. The Haar wavelet classifier is cascaded after the HOG-based classifier for the following reasons: 1) concatenating the 1620-D HOG feature with the 128-D Haar feature to form a new longer feature will make the Haar feature insignificant due to its relatively much lower dimensionality, and

2) the 128-D Haar feature requires double the computation of a 1620-D HOG, so it should come second for speed considerations. For both the HOG and Haar wavelet classifiers, the linear SVMs are implemented with the libLinear toolbox.

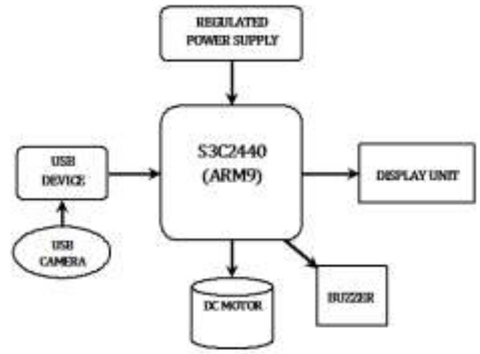



**Figure 3 Block Diagram**

## 3. EXPERIMENTAL RESULTS

Using the acquired data sets and presented evaluation criteria, we report the performance of our method on this problem. The experiments were performed on a 2.67-GHz PC with 1.5-GB DDRII RAM using a single thread, with implementations of HOG classification and optical flow that were revised from OpenCV2.0. It is shown that the three-level framework obtains solid detection performance at a high detection speed. Our experiments examine cases with only pedestrians that enter from the left side. To also handle pedestrians entering from the right, each video frame can be flipped about the Y -axis to double the number of input frames and to reduce the processing speed by half. The processing speeds reported in the Abstract and reflect this two-sided configuration, while the speeds given in this section assume left-entering pedestrians only.

### Parameters

A total of nine parameters, listed in Table I, are used in our algorithm. Four of them are related to event definition and evaluation and do not affect pedestrian detection performance. Four of the remaining five parameters have fixed values throughout all experimentation, while one parameter requires tuning. The following parameters affect system performance.

1) The threshold ω is used in NMS for merging overlapped windows with positive detections. Since it is employed with the commonly used intersection-over-union evaluation metric, we set it to the typical value of 0.5.

2) The p and μ in the temporal refinement step are parameters used in examining the support of a positive detection among previous frames. Here, we set the parameters to require at least one of the preceding three frames to also have a positive response.

3) In the LBP difference-based local level detection, f and θf need to be set to balance detection accuracy and speed. We empirically fix θf to 6 and tune f according to vehicle speed. In our experiments, f can take one of two values, depending on whether the camera is fixed or

International Journal of Instrumentation Electrical Electronics Engineering
p- ISSN: 2321 – 0621 e- ISSN: 2321- 063X

moving. In this way, only one parameter needs to be tuned in this paper.

**Performance of Each Level**

In the coarse-to-fine framework, each level contributes to detection performance and accelerating the detection process. The effectiveness of each level in terms of candidate window reduction, false positive reduction, and execution time is shown in Table II for the event definition parameters αe = 0.25 and αl = 1.5 and true positive evaluation threshold θo = 0.5. In the table, AIW# denotes the average number of input windows to that level, and AOW# refers to the output windows. The corresponding ROC curves and scored curves show that each level improves detection accuracy.

**Table 1Parameter settings**

| Category | | Notation | Value/Range |
|---|---|---|---|
| Performance Independent | Event Definition | $\alpha_e$<br>$\alpha_l$ | 0.25<br>{0.75,1.5} |
| | Evaluation Metric | $\theta_o$<br>$b$ | {0.25,0.5}<br>5 |
| Performance Related | Fixed value | $\omega$<br>$p$<br>$\mu$ | 0.5<br>3<br>2 |
| | To be tuned | $f$<br>$\theta_f$ | [0,1]<br>6 |

**Table 2 Performance of each level**

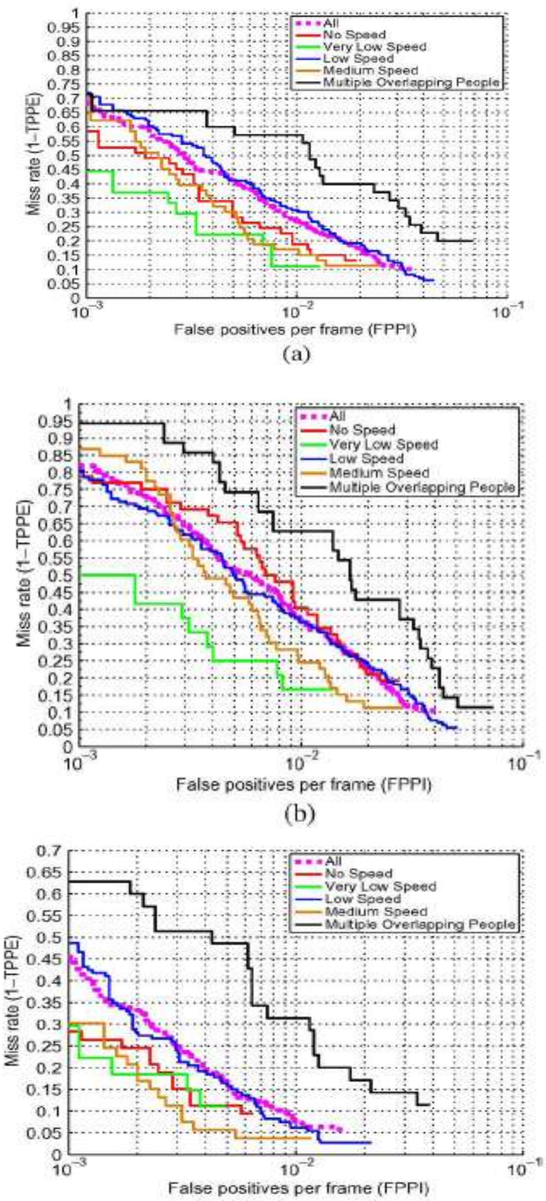
| Level # | AIW# | AOW# | TPPE | FFPI | Time/frame |
|---|---|---|---|---|---|
| 1. Local | 184 | 8.056 | 1.000 | 7.716 | 5.797ms |
| 2. Frame | 8.056 | 0.137 | 0.971 | 0.077 | 2.280ms |
| 3. Video | 0.137 | 0.061 | 0.911 | 0.035 | 0.980ms |

In Table II, the benefit of the sparse sliding window strategy is evident, with only 184 windows in each frame that need to be processed. Moreover, the LBP difference-based motion filter rejects most of the negative windows while retaining 100% of the true positives for later processing.

**Performance With Different Vehicle Speeds and Pedestrian Occlusions**

In this section, we investigate the performance of the proposed approach under different vehicle speeds and also different occlusions due to pedestrian overlap. We asked the pedestrians to enter with different movement styles and speeds. Vehicle speed was purposely kept at one of four levels. We did not control the entering of other people, vehicles, and bicycles in the scene.

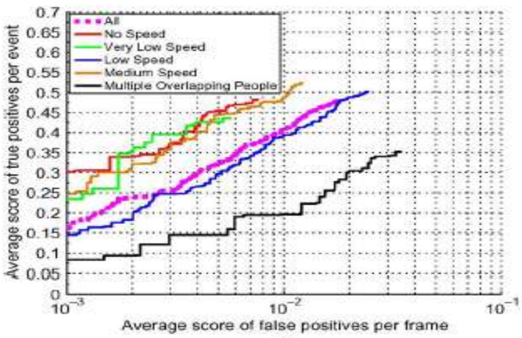
From the ROC curves shown in Fig. 2, one can make a conclusion that the proposed system exhibits reduced performance the case of multiple overlapping pedestrians in comparison to single individuals. This is as expected since the multiple overlapping pedestrian case is particularly difficult. The occluded person usually cannot be correctly detected since only a very small part is visible and may be included in the bounding box of another visible person. We note, however, that acceptable results can be obtained in practice if the person in front is correctly detected.



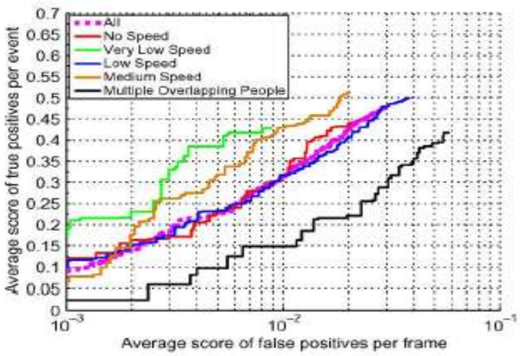

(a)

(b)

(c)




(d)




(e)

**Figure 4**
**(a) ROC curves ,**
**(b) ROC curves**
**(c) ROC curves for different vehicle speeds with parameters αl = 1.5 and θ0 = 0.25**
**(d) scored ROC curves of different vehicle speeds with parameters αl = 1.5 and θ0 = 0.5. scored ROC**
**(e) scored ROC curves of different vehicle speeds with parameters αl = 0.75 and θ0 = 0.5.**

Except for this difficult case, we can observe that the system performance is no worse than the reported average performance; however, the effect of vehicle speed remains unclear. In Fig. 2(a) and (b), the performance is comparable for different vehicle speeds, except for the very low speed case. With our proposed scored criteria, in Fig. 2(d), the performance for different vehicle speeds is comparable, except for the low speed case; in Fig. 2(e), only the very low speed case has higher performance, especially at low false positive rates (e.g., 0.001FPPI). In Fig. 2(c), when the threshold θo is relaxed to 0.25, the performance is almost the same at a low false positive rate of 0.001FPPI with different vehicle speeds, except for the very low speed case. We believe that the slight differences in performance may be the result of different background and pedestrian appearances among the videos at different speeds.

## 4. CONCLUSIONS

This paper "Detection of Sudden Pedestrian Crossings for Driving Assistance SYSTEMS" has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced ARM9 board and with the help of growing technology the project has been successfully implemented.

In future work, we plan to elevate the performance in four ways. 1)We plan to include sensor (camera) characteristics into the detection algorithm and establish the distance relationships between virtual reality and physical reality. 2) We want to improve the performance in the frame-level processing, such as by investigating new static features. 3) We wish to introduce kinematics knowledge into the detection algorithm for better utilization of motion information. 4) We plan to improve classification accuracy by adopting new algorithms based on likelihoods. We also would like to extend this system to handle a broader set of obstacles, such as other vehicles.

***Acknowledgements***

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

## References

[1] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[2] I. P. Alonso, D. F. Llorca, M. A. Sotelo, L. M. Bergasa, P. R. de Toro, J. Nuevo, M. Ocana, and M. A. G. Garrido, "Combination of feature extraction methods for SVM pedestrian detection," IEEE Trans. Intell. Transp. Syst., vol. 8, no. 2, pp. 292–307, Jun. 2007.
[3] N. Bellotto and H. S. Hu, "Multisensor-based human detection and tracking for mobile service robots," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 1, pp. 167–181, Feb. 2009.
[4] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in Proc. IEEE Conf. CVPR, 2009, pp. 304–311.
[5] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," IEEE Trans. Pattern Anal.Mach. Intell., vol. 31, no. 12, pp. 2179–2195, Dec. 2009.
[6] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," Int. J. Comput. Vis. (IJCV), vol. 63, no. 2, pp. 153–161, 2005.
[7] N. Suresh Kumar et al., "Modern Computer Graphics Technologies Used at Educational Programs and Some Graphical output screens", IJCSIS, Vol. 8, No. 3, June 2010, PP 172-174

## Biographies

**Y.Ramesh Reddy** is a PG Student. He got his M.Tech degree in Electronics and Communication Engineering from DRK Institute of Science&Technology, Hyderabad. He is specialized in VLSI&ES.

**Dr.R.V.Krishnaiah** received his B.Tech degree in ECE from Bapatla Engineering College. He received his M.Tech degree in Computer Science Engineering from JNTU and also M.Tech-(EIE) from NIT,Warangal. He did Ph.D (MIE, MIETE, MISTE) from JNTU Ananthapur.