Monocular Vision based Road Marking Recognition for Driver Assistance and Safety

Mohak Sukhwani[1] Suriya Singh[1] Anirudh Goyal[1] Aseem Behl[1] Pritish Mohapatra[1]
Brijendra Kumar Bharti[2] C. V. Jawahar[1]
[1]CVIT, IIIT Hyderabad, India [2] RNTBCI, Chennai, India

Abstract— **In this paper, we present a solution to generate semantically richer descriptions and instructions for driver assistance and safety. Our solution builds upon a set of computer vision and machine learning modules. We start with low-level image processing and finally generate high-level descriptions. We do this by combining the results of the image pattern recognition module with the prior knowledge on traffic rules and larger context present in the video sequence. For recognition of road markings, we use a SVM based classifier and HOG based classifier. We test our method on real data captured in urban settings, and report impressive performance. Qualitative and quantitative performance of various modules are presented.**

Key Words: Computer Vision, Road Marking Recognition, Driver Assistance, Vehicular Safety.

I. Introduction and Related Work

Semantic understanding of the scene is one of the ultimate goals of computer vision. In case of natural outdoor environment this problem becomes extremely complex. When employing computer vision for designing safety systems or for driver assistance, there has been a persistent need for generating semantically rich descriptions and instructions. This work is a step in this direction. Our solution enhances the state of the art in recognition, and attempts to bridge the semantic gap of "recognition as labelling" and "human friendly understanding".

There have been many works in the past that recognize isolated road signs for various applications including safety and autonomous navigation [1]–[11]. However, the recognition capabilities of these methods are often limited to assign a label to the symbol, and possibly localize it in the image. In this work, we extend the recognition to richer semantic understanding that exploits the larger context of the road markings in time and space. Exploiting the context for better understanding of visual data has emerged as a major direction of research in the recent past [17], [18]. In our setting, context includes the presence or absence of certain other symbols in present and previous frames. This high level reasoning starting from the isolated recognition is carried out by integrating the top down and bottom up cues in the navigation.

Our objective is to build human-like understanding from the visual cues using video stream captured from a moving car. Our solution is characterised by the following aspects: (i) We process the video captured using single monocular camera and design a low cost solution (with minimal hardware requirements) for the safety and driver assistance. (ii) We start with recognition of isolated symbols or markings (such as zebra crossing) on the road and move to semantic descriptions (such as "You may anticipate a zebra crossing soon and slow down") (iii) Our system learns the co-occurrence patterns of the symbols (such as "zebra crossing" and "diamond"), and uses it for generating descriptions which are richer, predictive and semantic (iv) we use ideas from machine learning so that the solution is easily adaptable to a new situation (a new city or a new imaging conditions) with minimal supervision.

The problem of autonomous navigation and driver assistance using computer vision techniques isn't new and has been studied well in recent past. Most of the work in this area can be broadly classified into two categories: (i) Computer vision based analysis of road/traffic scenes and (ii) Computer vision based vehicle safety systems. Road/traffic scene understanding can either use static cameras [12], [13] or can involve cameras mounted on moving vehicles [2]–[11]. Modern vehicles are getting equipped with various sensors for advanced functionalities. This has led to a surge in research activities for vehicle safety and navigation involving vehicle mounted sensors. Vision is an important component of modern smart vehicles. Many modern vehicles are equipped with safety systems that warn drivers and take control over the drive in case of an emergency. For example, [5] provides solutions for monitoring driver's condition using a camera facing towards driver. [1]–[4], propose methods for monitoring the road for lane departure, nearby vehicles and obstacles. Works like [7]–[10] try understanding traffic scenes by detecting lanes and traffic signs using vehicle mounted cameras.

We use monocular camera based vision in a highly constrained setting. Our objective is to design modules that can provide semantic instructions (for safety and guidance) for humans and machines. Our problem is similar to that of [11]. However, our approach is more principled and robust. We design and implement a set of computer vision modules that begins with low level image processing and eventually leads to richer semantic descriptions that has larger context and utility, for both man and machine. We validate our method on real life data captured in challenging natural urban traffic situations. We consider 19 road signs and report results on our dataset that is three times larger than the dataset used

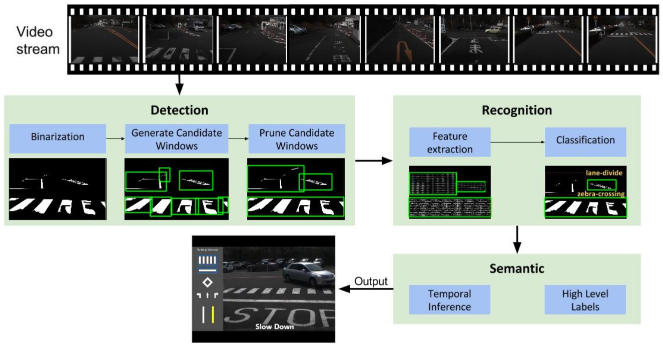


Fig. 1: Overview of our system.

by [11] and includes variations in illumination, degree of erosion and occlusion.

II. Overview

Our system takes visual data in the form of a video stream as input and provides human friendly semantic descriptions and suggestions about the road-traffic scene as output. While the input video stream is captured using a camera mounted on the top of the vehicle, the semantic descriptions and suggestions are presented to the driver through the 'Drive Assist System'.

Figure1 presents an overview of our system. For each frame of the video, we begin by segmenting it into foreground and background regions using binarization. The goal of this step is to push most of the irrelevant parts into the background and segment out only the salient parts of the road such as the road signs and lane markers as the foreground. Second, we identify candidate bounding boxes for road signs from segments of the foreground. Third, for each candidate bounding box, we run a classifier to determine the presence or absence of a road sign inside the bounding box. Finally, we use the position and label of each sign for semantic inference. Our algorithm keeps track of road signs detected for previous frames and builds a semantic network over it. The system then translates the results into a human friendly format which can be easily grasped by the driver.

III. Technical Details

A. Detection

The first module in our system detects road sign bounding boxes on the road irrespective of the category they belong to. This is challenging task because of the high degree of degradation of road signs present in a large number of frames and also due of the lack of decent lighting conditions in some of the frames. We describe the different stages in our detection module in the subsections below.

1) Binarization: We first binarize an image so that we have segments like road signs and lane markers as the foreground and and try to push everything else to the background. Binarization becomes challenging for images in which the road signs are eroded or there are light variations due to presence of other objects blocking or reflecting normal sunlight. Here we describe two methods that we tried for binarizing the frames.

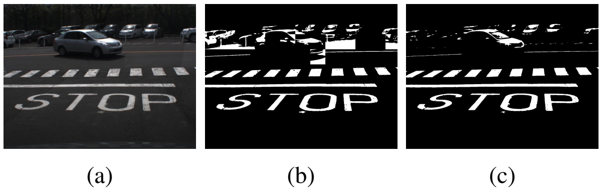


Fig. 2: Binarization results for a typical frame; (a) input frame (b) using Otsu's method and (c) using GMM.

Binarization using Otsu's method: This algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram (e.g. foreground and background); it computes the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. Instead of using a global threshold, we divide images into blocks, each of size 100×100 pixels and a threshold is calculated for each block using Otsu's method. It has been verified through experiments that block level threshold, though takes more time, is more robust than global threshold. This algorithm is a simple and effective one, however, it involves calculations on all possible threshold values.

Binarization using GMM: This algorithm assumes that the distribution of intensities taken by pixels corresponding to road signs can be modelled using a Gaussian Mixture Model (GMM). The GMM model consists of a fixed number (typically between 3 and 7) of weighted Gaussian distributions. Any pixel which is highly unlikely to have been generated by this model is considered irrelevant and hence discarded as background. The GMM model is initialized with binarization output using Otsu's method for few initial frames and is updated regularly as more frames come by. This algorithm is fast and for a frame size of 1280×960 pixel, runs at 5 *fps* on a single core system. The algorithm can be easily extended to run parallely on a multi-core architectures like multi-core CPUs and GPUs. The GMM based binarization results in lesser false positives as compared to binarization using Otsu's method. Figure2 gives a comparison between Otsu's method and GMM based binarization.

2) Generate Candidate Windows: Next, we identify bounding boxes on image which can possibly contain a road sign. In case of an ideal binarization, every connected component would correspond to a candidate bounding box. However, our dataset consists of images from natural urban roads and as expected, we encounter plenty of road signs which are degraded and lead to fragmented segments on binarization. We need to merge these smaller fragments into a single bounding box which can then be passed on to the classifier. In addition to this, we also have road signs like zebra crossing which naturally consists of several isolated components and need to be merged into a single bounding box. Merging several components in binarized image to generate useful candidate windows is not a trivial task.

Here, we try to increase the system's recall to get candidate bounding boxes for all road signs present while tolerating false detections. For this, we consider only segments that are bigger than a certain size and then generate bounding boxes for all possible combinations of these segments.

3) Prune Candidate Windows: As previous step results in a large number of false candidate windows, in order to prune them out we learn a linear SVM to differentiate between that contain a road sign and one that does not. We train this SVM using a few hand labelled examples.

B. Recognition and Classification

In this module, we take the candidate bounding boxes generated by the previous module and predict what road sign a bounding box probably has.

1) Feature Extraction: We extract HOG features [15] for each of the candidate bounding boxes. HOG descriptors capture the distribution of orientation of intensity gradients inside the bounding box. These features are robust towards variation in scale, lighting and moderate degradations. Figure3 gives a visualization of the HOG features for some typical bounding boxes.

2) Classification: We model the problem as a multi-class classification problem that is built by training a 1-vs-rest linear SVM for each of the 19 road sign categories. For each candidate bounding box, we compute the score using all the 19 linear SVMs and the road sign category with the highest score is assigned to the candidate bounding box. We use the publicly available *liblinear* library [14] for training the 1-vs-rest classifiers.

In order to train the SVM classifiers, we divide our dataset into a training and a validation set. We annotate the candidate windows with their respective road sign category labels. We train the linear SVM for each of the road sign category over the HOG features of the candidate bounding boxes of the training set with positive label for candidate windows containing the particular road sign and negative label for every other candidate window.

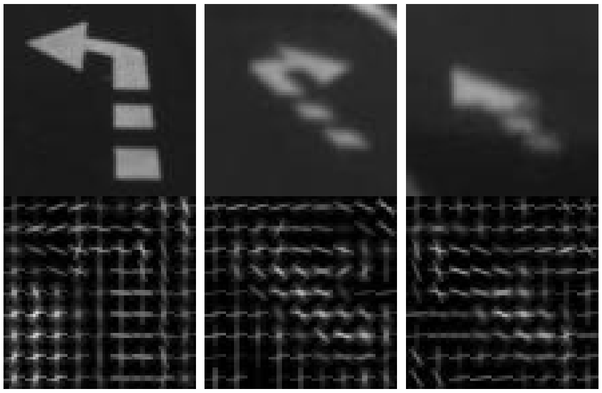


Fig. 3: Visualization of HOG features for example candidate windows (a) Left-turn (b) Right-turn and (c) Straight

C. Semantic Inference

In this module we process individual labels of detected road signs and try to present the relevant information in a human friendly format to driver in the form of scene descriptions and suggestions. The final output is presented in audio and visual format on a panel as shown in Figure8.

The challenge is in presenting the information in a format which the user could understand and make use of with minimum additional effort. For achieving this, we first start with a temporal inference. Our algorithm detects and recognizes symbols in every frame. We keep track of previous output labels and build a semantic description over it. Doing inference that is spread of several frames, makes the results more reliable. As new frames are processed we get new recognition outputs in the form of detected road signs and corresponding scores. And the new inferences are made by integrating these individual outputs with results from the previous frames.

Afterwards, we convert the labels into more meaningful descriptions. The final displayed results are in the form of human understandable sentences, phrases or comments (which is later converted to audio outputs) as well as the detected symbols are highlighted in the pane on left side of the output screen as can be seen in Figure8. The different steps followed in this module are described below in detail.

1) Temporal Inference: Similar to HMMs, the output of present frame in our case depends on the previous frames and therefore we have modelled our system as an order-n Markov. The detected symbol counts are the ones that govern the relationship between the previous frames. With increase in order of system, the overall complexity grows making our system slow without any significant improvements in results. Our system performs best in case of n is 5.

Since we work with video as an input to our system, a missed or an incorrect detection at level of frames may build to the system error. To make system robust to minor miss classifications we keep track of counts of detected symbols in 'n' previous frames; if some how we fail to classify a symbol in particular frame previous frame detections come to our rescue. The symbol is considered as detected only if its count is above certain threshold ($count_{thresh}$) in present and previous frames.

2) High Level Label: We can have multiple road signs in each frame and thus to display all at given instance we have designed a display system that highlights all the detected road-signs on 'most frequent marking' pane. We even display the text related to most significant detection at the center of frame. Most significant detections are determined by set of pre-determined priority of symbols (configurable to requirements).

Our semantic generation module is at heart of the safety system. The rich descriptions add to driving assistance and audio clues in cases of high priority symbols (configurable) warns drivers to be extra cautious. This module adds to ability of the system to make one a 'safer driver'.

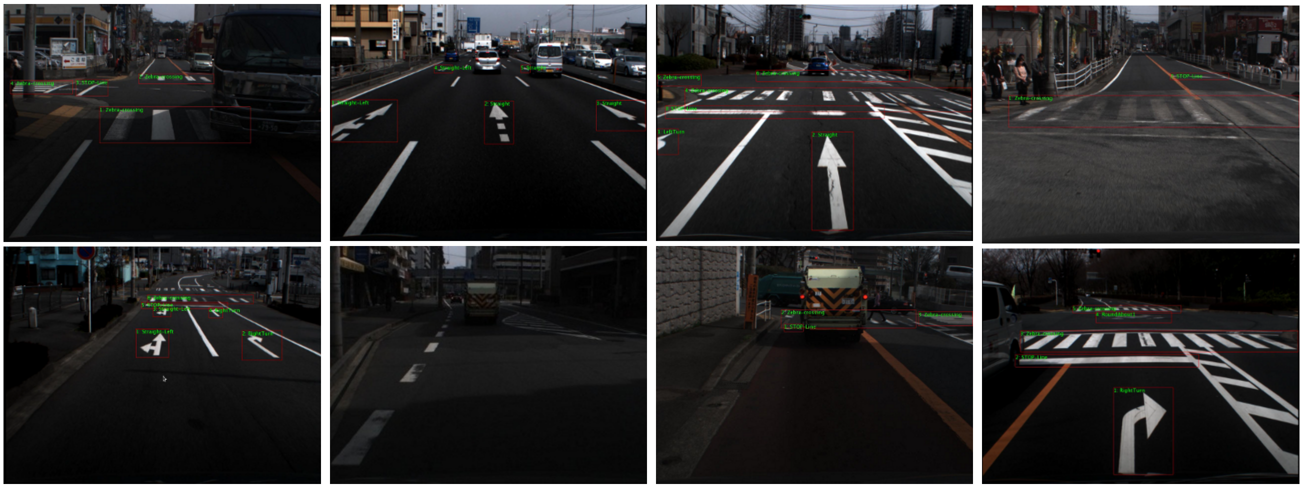


Fig. 4: Examples of Images from Our Data Set.

IV. Experiments and Validation

A. Dataset and Annotation

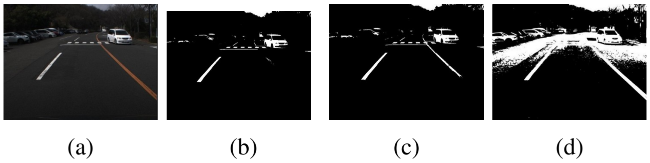
The dataset used in this work is created by the camera mounted on the car capturing front view of the travelling direction. The vehicle was driven across the urban roads of Hon-Atsugi Area, Kanagawa, Japan and captures over an hour of drive. The camera records the video of road at a resolution of 1280 X 960 and captures scenarios such as bright sunny day, tunnel passing, dim sunlight, shadow of vehicles on road signs, vehicles obstacles over symbols etc. To best of our knowledge we don't have any public repository capturing such variations. Our dataset comprises 65k frames with 100K annotated road signs categorized into 19 classes.

Annotating such voluminous dataset has been no mean task; to minimize the effort we devised a semi-supervised method to annotate the dataset by annotating every fifth frame and then propagated the results to remaining frames. In all we manually annotated 15k frames and propagated the result on 50k frames. Minor adjustments if any (because of propagation) to the annotations were done by visiting and verifying annotations of each frame.

We capture both subjective (e.g. occlusion, brightness) and objective(eg. Driving direction) attributes for each road sign. The details captured in annotations are extensive in nature and following attributes are defined for each road-sign:

1) Occlusion: no-occlusion, medium-occlusion, high-occlusion.
2) Brightness: low-brightness, medium-brightness, high-brightness.
3) Erosion: no-erosion, medium-erosion, high-erosion.
4) Lane: not-applicable, current-lane, different-lane.
5) Driving Direction:not-applicable, driving-direction, opposite-direction.

The annotated road-sign identifies 19 variants of road signs including speed limit signs '30', '40' and '50', 'Zebra-crossing', 'STOP', 'STOP-Line', 'Diagonal', 'Diamond', 'chevron', 'RightTurn', 'Straight', 'No-U-Turn', 'U-Turn', 'Straight-Left', 'Cycle-crossing', 'RoundAbout1', 'Straight-Right', 'RoundAbout2', 'LeftTurn'.



(a) (b) (c) (d)

Fig. 5: Figure explains the effect of number of components (K) in the GMM based binarization. (a) input image (b) $K = 2$ (c) $K = 5$ (d) $K = 11$.

B. Experiments and Evaluation

We implement our solution on top of the existing open source and popular libraries. In the present form, we uncompress and work on individual frames. In addition to the use of low level modules from OpenCV and matlab, we also use parts of other implementations [19]. This helps in making the solution efficient, better modular and robust. Line detection uses RANSAC based spline fitting to detect lanes on the street.

a) Segmentation: Our segmentation uses GMM to model the intensity distribution of the the road signs. The strength of this scheme is the ease in adapting the Gaussians and the thresholds with minimal computations. Experimentaly, we find that 3 to 7 components are sufficient for modelling the intensity distribution. Fig. 5 shows the effect of number of Gaussian components in the GMM based binarization. This method is fast and run at 5 *fps* when implemented serially (frame size is 1280×960 pixels) on 2.0 GHz CPU. The algorithm can be easily extended to parallel implementation.

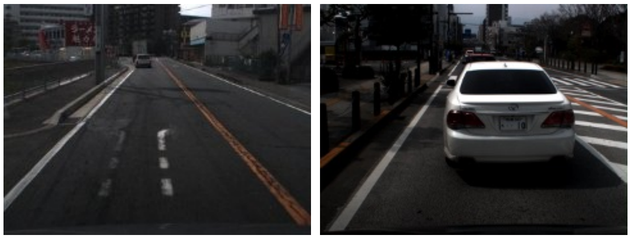
b) Detection: The challenges in detection of road signs includes a large number of degraded and/or fragmented signs. They are either completely broken or the breaks are introduced due to a highly variable colour distribution. Another key challenge is occlusions by other objects or due to limited field of view of the camera. Shapes which are occluded by the vehicles ahead are practically impossible to detect. These require special care in building a complete robust solution. See Fig. 6 for challenging cases. In this work, we have focussed on the robustness in the recognition module.

c) Classification: We use a SVM based classifier with HOG as feature for efficient and accurrate classification. In order to compute HOG features, we divide the gray scaled version of window in 8×8 pixels non-overlapping cells. 1D histogram of gradients is accumulated over cell pixels for each cell. We use 9 orientation bin to discretize the gradients at each pixel. These parameters were found optimal to capture local shape properties of windows and are considerably robust to small deformations.

We have carried out multiple types of evaluations. This include the evaluation of the individual one vs all classifiers using mean average precision and the evaluation of the overall accuracy using a fused multi class classifer model. We report an accuracy of 92% over the test set with training and the further improvement with hard mining.

The dataset specifies ground truth bounding boxes for each road sign and we have over 95K road-sign windows in our dataset. We train our SVM based classifiers (1 vs all) on randomly selected 22K windows. For each window we compute classifier scores for all category classifiers and assign the category label with the highest classifier score. In order to learn the SVM hyper-parameters we perform 5-fold cross-validation on training dataset. Using the hyper-parameters learned during cross-validation, we re-train the classifiers for each road-sign category on the complete training dataset.

In order to simulate the conditions for noisy detections, we add random noise to our ground-truth bounding boxes. We do this by shifting the location of ground-truth bounding boxes in a random manner around the original location. This makes the classifier more robust to noise. We show accuracy and average precision (for four popular classes) in Table 1.

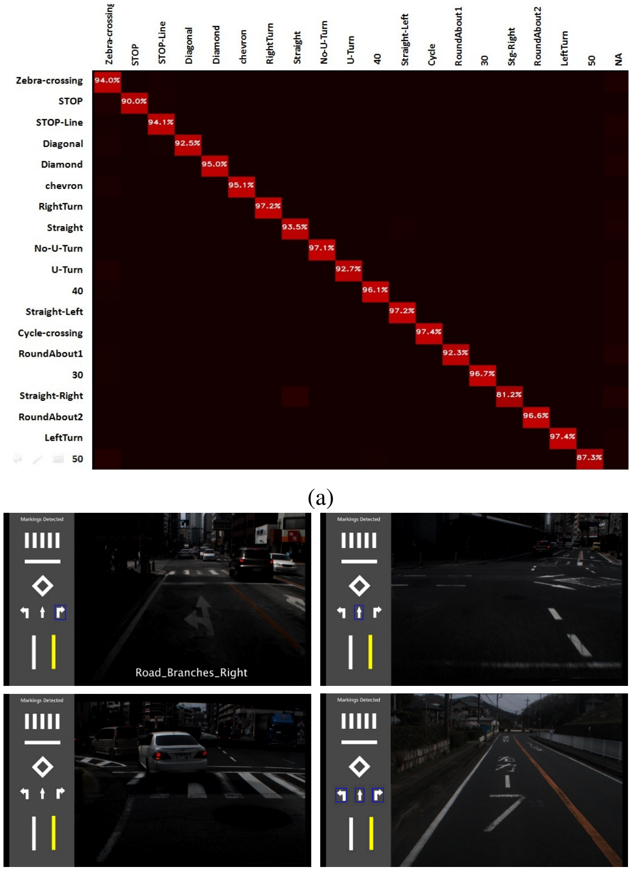


(a) (b)

Fig. 6: Challenging situations for detection in isolated frames due to (a) degradations for 'left-turn' and 'right-turn' signs and (ii) occlusions of 'zebra-crossing' sign due to traffic.

Symbol	Accuracy(%)	AP
Diamond	98.3	0.971
Left	99.1	0.986
Right	99.1	0.994
Straight	98.5	0.981

TABLE I: Accuracy and AP for four popular classes



(a)

(b)

Fig. 7: (a) Confusion matrix for a 5 class classification (b) Examples of some of the windows that got mis classifed.

Furthermore, we evaluate the performance of the recognition system by evaluating accuracy of the multi-class classifier on the test dataset. In our experiments multi-class accuracy was found to be 91.7%. .Fig 7. summarizes the results in form of confusion matrix of all detected road signs. The corresponding i^{th} row and j^{th} column entry signify the percentage of symbol 'i'classified as symbol 'j'. The symbols detected with confidence scores of less than 0.8 are discarded and are labelled as 'NA'in the confusion matrix. The system performs equally good in cases of partially occluded, eroded and low illuminated symbols.

d) Generation of Semantic Descriptions: Qualitatively our semantic generation module performs best in case when we consider detected symbols in previous five frames (n). We tested our system with values of n ranging from 3 to 10. $count_{thresh}$ value is set as 0.6, i.e. if $n = 5$ the detected symbol count should atleast be $0.6 * n = 3$ for being considered as detected.

V. Conclusions and Future Work

In this work, we use computer vision techniques for detection and recognition of road signs. For fast and robust sign detection, we extract simple features from sliding window and use a linear binary SVM for rejecting windows that do

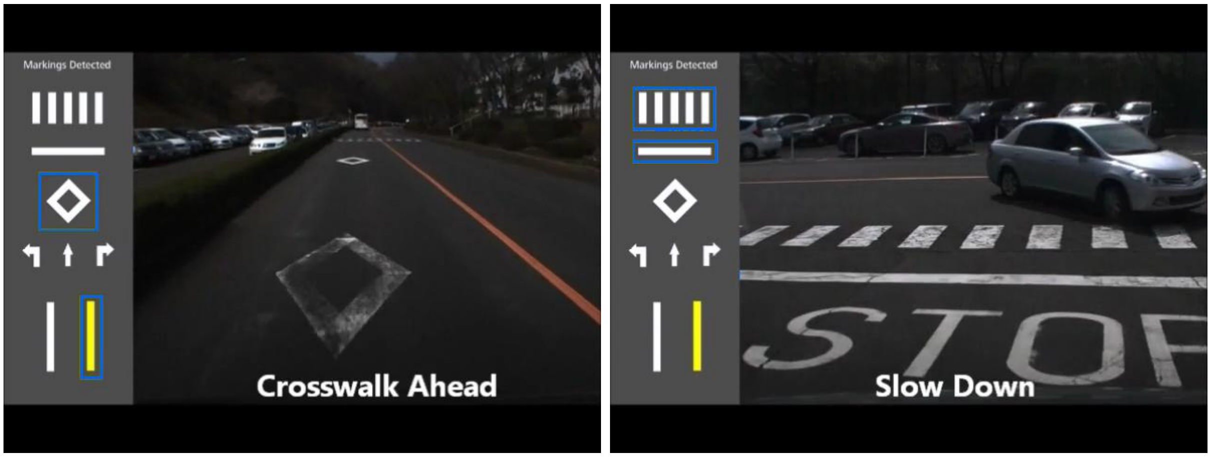


Fig. 8: Sample output screen.

not contain any sign. Later, we extract HOG features and use multiclass SVM in one-versus-all fashion for classification. The classification results are used to generate semantically meaningful text and are displayed as output along with the sign detected.

We have been successful in developing a technique that is capable of detection and recognition of various road signs. Our technique is robust in handling partial occlusion, motion blur, changes in illumination and partially degraded signs. However, special care needs to be taken for highly degraded signs. Sliding window based detection takes longest time in the pipeline. One promising direction for future work is to learn the associated parameters in an unsupervised manner taking cues from previous frames and use it for the present frame. Such techniques would make solution easy to adapt for different cities and weather conditions with minimal manual intervention.

References

[1] Mohan Manubhai Trivedi, Tarak Gandhi, and Joel McCall. Looking-In and Looking-Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety. IEEE Transactions on Intelligent Transportation Systems, 2007.

[2] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Automatic Change Detection of Driving Environments in a Vision-Based Driver Assistance System. IEEE Transactions on Neural Network, 2003.

[3] Jing-Fu Liu , Yi-Feng Su, Ming-Kuan Ko, Pen-Ning Yu. Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning Functions. DICTA, 2008.

[4] Donguk Seo, Hansung Park, Kanghyun Jo, Kangik Eom, Sungmin Yang and Taeho Kim. Omnidirectional Stereo Vision based Vehicle Detection and Distance Measurement for Driver Assistance System. IECON, 2013.

[5] Ashish Tawari, Sujitha Martin, and Mohan Manubhai Trivedi. Continuous Head Movement Estimator for Driver Assistance: Issues, Algorithms, and On-Road Evaluations. IEEE Transactions on Intelligent Transportation Systems, 2014.

[6] Andreas Møgelmose, Mohan Manubhai Trivedi, and Thomas B. Moeslund. Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. IEEE Transactions on Intelligent Transportation Systems, 2012.

[7] Gangyi Wang, Guanghui Ren, Zhilu Wu, Yaqin Zhao, and Lihui Jiang. A robust, coarse-to-fine traffic sign detection method. IJCNN, 2013.

[8] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. IJCNN, 2013.

[9] Yingying Zhu, Xinggang Wang, Cong Yao, Xiang Bai. Traffic sign classification using two-layer image representation. ICIP, 2013.

[10] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic Sign Recognition How far are we from the solution? IJCNN, 2013.

[11] Tao Wu and Ananth Ranganathan. A Practical System for Road Marking Detection and Recognition. IEEE Intelligent Vehicles Symposium, 2012.

[12] Yang Yang, Jingen Liu, and Mubarak Shah. Video Scene Understanding Using Multi-scale Analysis. ICCV, 2009.

[13] Satyam Srivastava and Edward J. Delp. Standoff video analysis for the detection of security anomalies in vehicles. Applied Imagery Pattern Recognition Workshop, 2010.

[14] Rong-En Fan and Kai-Wei Chang and Cho-Jui Hsieh and Xiang-Rui Wang and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. JMLR, 2008.

[15] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. CVPR, 2005.

[16] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man and Cybernetics, 1979.

[17] Anand Mishra, Karteek Alahari, and C.V. Jawahar. Top-down and Bottom-up cues for Scene Text Recognition. CVPR, 2012.

[18] Ankush Gupta, Yashaswi Verma, and C. V. Jawahar. Choosing Linguistics over Vision to Describe Images. AAAI, 2012.

[19] Mohamed Aly. Real Time Detection of Lane Markers in Urban Streets. IEEE Intelligent Vehicles Symposium, 2008.