Looking at Vehicles in the Night: Detection and Dynamics of Rear Lights

Ravi Kumar Satzoda, *Member, IEEE*, and Mohan Manubhai Trivedi, *Fellow, IEEE*

***Abstract*— Existing nighttime vehicle detection methods use color as the primary cue for detecting vehicles. However, complex road and ambient lighting conditions, and camera configurations can influence the effectiveness of such explicit rule and threshold-based methods. In this paper, there are three main contributions. First, we present a novel method to detect vehicles during nighttime involving both learned classifiers and explicit rules, which can operate in the presence of varying ambient lighting conditions. The proposed method that is titled as Vehicle Detection using Active-learning during Nighttime (VeDANt) employs a modified form of active learning for training Adaboost classifiers with Haar-like features using gray-level input images. The hypothesis windows are then further verified using the proposed techniques involving perspective geometries and color information of the taillights. Second, VeDANt is extended to analyze the dynamics of the vehicles during nighttime by detecting three taillight activities—braking, turning left, and turning right. Third, we release three new and fully annotated Laboratory for Intelligent and Safe Automobiles-Night data sets with over 5000 frames for evaluation and benchmarking, which capture a variety of complex traffic and lighting conditions. Such comprehensively annotated and complex public data sets are a first in the area of nighttime vehicle detection. We show that VeDANt is able to detect vehicles during nighttime with over 98% accuracy and less than 1% false detections.**

***Index Terms*— Taillight activity analysis, machine vision, machine learning, vehicle behavior, active safety.**

I. Introduction

DETECTION of on-road vehicles using on-board cameras plays a vital role in a variety of tasks related to both driver assistance and intelligent vehicles [1]–[4]. Although camera-based vehicle detection during daytime is widely researched and evaluated ([1], [2], [5]–[8]), a survey of existing literature shows that vision-based techniques for vehicle detection during nighttime is less comprehensively studied [9], [10]. However, there is far higher accident-per-trip ratio during nighttime compared to daytime [11], [12].

Most existing techniques for nighttime rely on detecting and pairing vehicle taillights or headlights [9], [10], [13], [14]. Given that the taillights are usually red in color, the input images are usually transformed into different color spaces. The dominant color components are then thresholded to extract blobs for the taillights [9], [13], [15]. In addition to explicit rules and models on the detected light blobs, classifiers have also been used as post processing techniques to verify candidate taillight blobs as vehicles [14], [16]. However, the robustness of such color based methods is dependent on ambient lighting conditions and exposure settings of the sensing cameras [9], [13]. More importantly, although detection of vehicles has been studied in literature, the dynamics of the vehicles during nighttime is less addressed. Reference [10] is one of the first and recent works that analyzes taillight activity during nighttime.

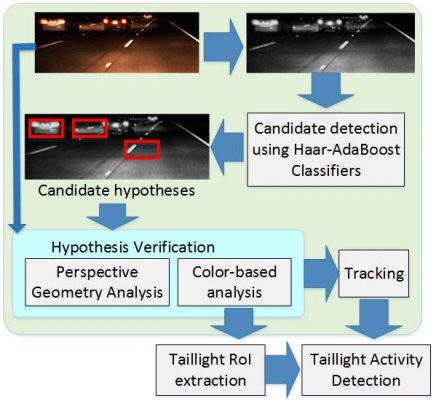


Fig. 1. Proposed vehicle detection and behavior analysis during nighttime.

In this paper, we propose a data-driven approach for vehicle dynamics detection and analysis using monocular cameras during nighttime. We call this method VeDANt which stands for **Ve**hicle **D**etection using **A**ctive-learning during **N**ight**t**ime. The main contributions of this paper can be summarized as follows: (a) Unlike existing methods that use color as the primary cue to generate hypotheses, VeDANt uses *learned Haar-based Adaboost classifiers* on *gray-level* input images to generate hypothesis windows. In order to reduce false positives further, the color components of the hypothesized candidate windows and geometric constraints are used to verify the presence of a vehicle; (b) In addition to detection of vehicles, we propose techniques to analyze the dynamics of the vehicles. This is done by detecting taillight activities such as braking and turn indicators; (c) Given that there are no publicly available benchmarking datasets for evaluating vehicle detection during nighttime, we release three fully annotated LISA-Night datasets, which capture varying levels of complexity and driving conditions during nighttime. Detailed evaluation of VeDANt on LISA-Night datasets is presented in this paper. Fig. 1 shows the proposed detection and analysis method,

Manuscript received August 12, 2015; revised December 14, 2015; accepted January 6, 2016. The Associate Editor for this paper was R. I. Hammoud.

The authors are with the University of California at San Diego, La Jolla, CA 92093 USA (e-mail: rsatzoda@eng.ucsd.edu; mtrivedi@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2614545

TABLE I
NIGHTTIME VEHICLE DETECTION: SELECTED RESEARCH STUDIES

Name	Type[1]	Approach	Evaluation
O'Malley 2010 [9]	R	Color space transformation, blob detection, symmetry and perspective transformations, catered for low-exposure settings	Proprietary dataset, 97% detection rate with 2.58% false detection rate
Alcantarilla 2011 [16]	R-L	Bright object segmentation using adaptive thresholding, SVMs to classify light blobs	1182 frames of proprietary testing data, 96% accuracy with 6% error rate
Chen 2012 [17]	R	Color intensity image thresholding using Nakagami imaging	Proprietary images, 71% on urban roads and 85% of highways
Eum 2013 [13]	R	Multi-camera setup with lane integration, light blob detection, Distance based thresholding function, primarily for low exposure images	24 proprietary sequences, 90% true positive rate for 3 false positives per image
Tehrani 2014 [18]	L	Discriminative parts models (DPM) on infrared (IR) camera imaging	8000+ IR images, Precision of 90% for a recall rate of 80%
Almag. 2015 [10]	R	Color based thresholding, candidate pairing using blob analysis, symmetry and color histograms, taillight activity recognition	Proprietary dataset[2], 97% mean detection rate, no information about false detection rate
Kosaka 2015 [14]	R-L	LoG based filtering for blob detection, SVM for blob classification	Proprietary dataset[3], 95% mean detection rate for vehicle taillight blobs, 11% nuisance light blob detections (false positives)
This Work - VeDANt	L-R	Modified active learning based detection on *gray-level* images, Color based hypothesis verification, Taillight analysis for brake and turn signal detection	New public datasets with complex lighting conditions for benchmarking, 98.37% accuracy rate with 0.9% false detection rate on complex datasets

R - rule based, L - learning based, R-L - rule based followed by learned classifiers
[2],[3] Number of "annotated" frames is unclear in the datasets

which is designed to detect the vehicles as well as analyze the behavior in terms of taillight activity.

II. RELATED RESEARCH

In this section, we survey some of the recent notable works that address nighttime vehicle detection using a forward looking camera, which is mounted inside the ego-vehicle. Table I lists some of the recent works. We discuss the techniques and challenges in this section.

A. Techniques

Existing techniques include taillight blob detection as the first step to generate hypothesis windows. The hypothesis windows are then processed further by analyzing the blobs either by using explicit rules such as symmetry and shape analysis, or by using classifiers to detect valid blobs. Most existing methods [9], [10], [15], [17] employ color image processing to detect vehicle taillights in the night. In all these methods, different color channels are thresholded to extract blobs that could possibly correspond to vehicle taillights (which are assumed to be predominantly red in color). Therefore, binarization of the different color channels and gray level images is a critical step in most methods. Thresholding is performed using either fixed thresholds [9], [10], or adaptively based on the input image [16]. Imaging properties such as Nakagami imaging [17] are also used to determine the appropriate thresholds for binarization.

The resulting blobs are usually processed further to verify they are indeed part of a vehicle. Symmetry between light blobs is one of the common parameters that is used in multiple works [9], [10]. Shape related properties are explored in [16] to eliminate blobs that do not belong to vehicle taillights. Additionally, [16] also employs an SVM classifier at this stage by extracting features using shape properties of the blobs such as the centroid, distance between centroids etc. A similar SVM is used in [14] after the blobs are detected using Laplacian of Gaussian (LoG) operation. Also, specific constraints on lighting and exposure are laid out in some works such as [9] and [13] for the techniques to operate, such as low exposure conditions.

In Table I, we also list the type of technique as either rule based (R), or learning based (L) or rule based followed by learned classifiers (R-L). It can be seen that all the methods except [18] employ explicit rules for detecting the taillights using the color information. The rules form a series of thresholds on the color of the taillights and the characteristics of the blobs. Although [14] and [16] are indicated as R-L, the first step in both these methods use explicit thresholds for detecting possible blob candidates, which are then classified using learned SVM classifiers. We have included the IR imaging based technique in [18], which uses discriminative parts models (DPM) [19] for detecting vehicles. However, IR images have their own constraints in terms of the technology itself, which is out of scope of this paper.

B. Challenges

The techniques listed in Table I address nighttime vehicle detection to varying degrees of robustness and accuracy. It can be seen from Table I that most of the techniques are rule based methods. Therefore, the effectiveness of these techniques depends on the thresholding and binarization process. This inherently poses challenges because color and intensity variations are directly influenced by camera settings, calibration, ambient lighting, and incident lighting from other light sources (such as trailing cars, lamps etc.). This is the reason why [9], [13] clearly mention that they operate under low exposure settings more effectively. In the case of urban scenarios and vehicles that are closer to the ego-vehicle, there is more incident lighting on the vehicles in front of the ego-vehicle. Color and blob analysis based techniques lead to inaccurate detections. Such challenges need to be addressed by ensuring that techniques are applicable under auto exposure settings [10], [14].

The next open item in this body of work is the lack of benchmarking datasets. The different works listed in Table I evaluate their techniques using proprietary datasets. Therefore, there are fewer evaluations between different methods. A well-annotated dataset for evaluation and benchmarking needs to be designed for future comparisons. Considering that nighttime scenario could include a variety of lighting conditions, the benchmarking datasets must include such complexities in order to evaluate the effectiveness of the different techniques.

In the forthcoming sections, we address above challenges using the proposed method and new datasets for nighttime vehicle detection using monocular cameras.

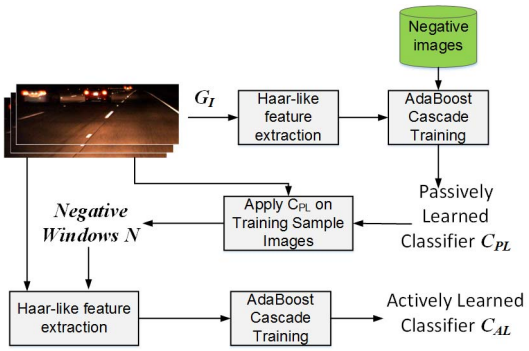


Fig. 2. Modified active learning flow for training Haar-AdaBoost classifiers.

III. VeDANt: Proposed Method

In this section, we will elaborate the proposed method for vehicle detection during nighttime. It will be referred to as VeDANt (an acronym for Vehicle Detection using Active-learning during Nighttime). It comprises of two steps: (1) Using learned classifiers for detecting possible hypotheses windows, and (2) Using color and geometric properties for confirming vehicles. We describe the two steps in detail.

A. Step 1: Hypothesis Windows Using Learned Classifiers

In VeDANt, hypothesis generation is performed in a data-driven manner. Instead of using rules and thresholds to extract blobs, VeDANt uses learned classifiers to detect possible hypothesis windows.

The first step in VeDANt employs AdaBoost cascade classifiers for detecting hypothesis windows using *gray-scale* images only. A modified version of the active learning framework [5], [20] is used to train an AdaBoost cascade classifier for detecting rear-views of vehicles during nighttime. The modified active learning framework is described as follows using Fig. 2. A training dataset consisting of 8000 annotated vehicles from nighttime driving conditions is first generated. The annotated vehicles are extracted from gray-scale images only, i.e., no color information is used. Let G_I be the set of annotated windows for a given training image I. A window $r_{G_I} \in G_I$ is defined by $r_{G_I} = [x\ \ y\ \ w\ \ h]$, where (x, y) indicate the top left corner of the window, and w and h indicate the width and height of the window r_{G_I}. Samples are particularly chosen from auto-exposure cameras such that there are varying intensities of light incident on the vehicles. Negatives are randomly taken from images that do not contain any vehicles. Haar-like features are extracted from these gray-level training samples, and they are used to train a 10-stage AdaBoost cascade classifier. This classifier is called passively learned (PL) classifier.

The PL classifier is then applied on the training dataset to extract a set of windows S_I in every gray-scale frame. These are the true positive (TP) windows according to the PL classifier. For every window $r_{S_I} \in S_I$ for a given image I, the overlap with the set of windows from G_I is computed. This overlap is defined by the standard PASCAL object detection metric, i.e. $\omega = \frac{(A_{r_S} \cap A_{r_G})}{(A_{r_S} \cup A_{r_G})}$, where A_r indicates the area of window r. The overlap criteria ω is used in the following way to generate a set of windows S'_I for a given training image such that:

$$S'_I = S_I - G_I = \{r_{S'_I} | r_{S'_I} \in S_I \ \ and \ \ r_{S'_I} \notin G_I\} \qquad (1)$$

ω is used to determine if the window $r_{S'_I} \in S_I$ and does not belong to G_I. An overlap ratio of 30% is used in our studies in this paper. This set S'_I is a set of false positives that are detected by the PL classifier for the image I.

A new set of negative images N is generated such that $N = \bigcup(\mathscr{C}(S'_I))$. N comprises of cropped images (denoted by $\mathscr{C}(\cdot)$ above) from the negative windows that are collected from each S'_I. The annotated positive samples and the newly obtained negative windows in N are then used to train another 10-stage AdaBoost cascade classifier resulting in the new classifier $\mathbb{C}_{AL}$ for vehicle detection during nighttime.

Given a new test image, Haar-like features are computed using multiple scales and the classifier $\mathbb{C}_{AL}$ is used to generate a set of hypothesis windows denoted by H. The windows $r_H \in H$ are further analyzed in the post processing step of VeDANt to verify true vehicle detections.

Both the stages of the classifiers are trained using MATLAB computer vision toolbox. In order to do this, we used 10,000 positive and 10,000 negative samples for training the first step classifiers (passively learned classifiers). A new set of 10,000 hard negatives are generated from the modified active learning stage which are used for training the second step classifiers (actively learned classifiers). Each classifier is a 10-stage cascade. Standard Haar-like features with four basic types of filters (horizontal and vertical) are used to generate the Haar features. The minimum size of the vehicles for training the cascades is set to 25×47 (height-by-width). The cascades are trained to meet a maximum false alarm rate of 0.5^{10} and minimum true positive rate of 0.995^{10}.

B. Step 2: Hypothesis Verification

After obtaining the hypothesis windows H by applying the AL classifier, the proposed method includes the second step of hypothesis verification.

1) Ground Plane Calibration Constraint: A hypothesis window $r_H \in H$ includes $r_H = [x\ \ y\ \ w\ \ h]$. The first operation in the proposed hypothesis verification step is to remove windows that do not align with the camera calibration. Given a camera calibration, the sizes of the vehicles are expected to be in particular aspect ratios, i.e., a vehicle cannot be smaller than a particular size when it is close to the ego-vehicle, and vice versa. Also, vehicles axle widths fall in a defined window. In order to eliminate such false windows in r_H that do not fit general vehicle dimensions, we employ the following ground plane homography using the camera calibration.

Fig. 3 illustrates the computation of homography matrix between the ground plane and the image domain. Given an image I in the image domain (or perspective view) as shown in Fig. 3, we choose four points on the lanes. We then create a virtual inverse perspective image as shown in Fig. 3. This is virtual because we do not actually generate any inverse perspective image. This is used only to compute the homography matrix Γ, which will then be used in the hypothesis verification step. The four points P_1 to P_4 in

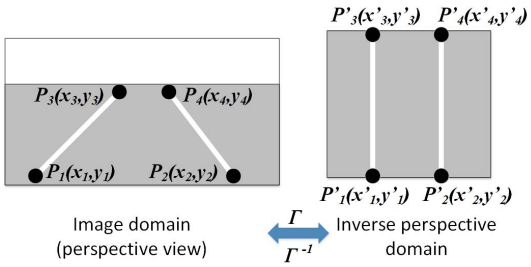


Fig. 3. Illustration on computing homography and physical constraint on vehicle width.

the image domain are mapped to four points P_1' to P_4' in the inverse perspective domain. The two sets of coordinates are related in the following way using a 3×3 homography matrix Γ, which is computed using linear algebra:

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & y_2' & y_3' & y_4' \\ 1 & 1 & 1 & 1 \end{bmatrix} = \Gamma \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2}$$

With Γ computed as a one-time ground plane calibration process, the hypothesis windows in H are now evaluated to check their validity in terms of physical constraints. Assuming that the axle lengths of vehicles (motor cars, trucks, vans and buses) have a fixed limited range, the widths of the hypothesis windows are computed in the inverse perspective domain. Therefore, if $r_H = [x \ y \ w \ h]$ is a hypothesis window, the following two coordinates are computed: $P_1 = (x, y+h)$ and $P_2 = (x+w, y+h)$, which correspond to the bottom left and right coordinates of the hypothesis window in the image domain. Assuming that they approximately lie on ground plane, we compute P_1^r and P_2^r using Γ, i.e.,

$$P_1^{r'} = \Gamma P_1^r \text{ and } P_2^{r'} = \Gamma P_2^r \tag{3}$$

These two points in the inverse perspective domain are used to find the width of the window (i.e. the vehicle) in the bird's eye view, i.e. $\Delta = x_2^{r'} - x_1^{r'}$. Δ is then checked against a range between Δ_{min} and Δ_{max}. This range is taken from 30 to 50 pixels in our studies. A wide range is considered because it accommodates the variations in the ground plane. The above derivation of Γ assumes that the ground plane is flat. If it is curving up or down, the tolerance in Δ will accommodate such changes in ground plane.

Fig. 4 shows the effect of applying the ground plane calibration on the hypothesis windows generated by the classifier. The classifiers gives all the three windows shown in Fig. 4 but the homography constraint eliminates the blue hypothesis windows in Fig. 4.

2) Taillight Verification: The classifier detection coupled with the proposed ground plane constraint results in high accuracy rates, which will be discussed in the next section. However unlike low exposure conditions, the presence of ambient light results in some false detections. In order to remove such false positives, we propose a taillight verification technique that is applied to the detected hypothesis windows. In the methods that were reviewed in Section II, binarization

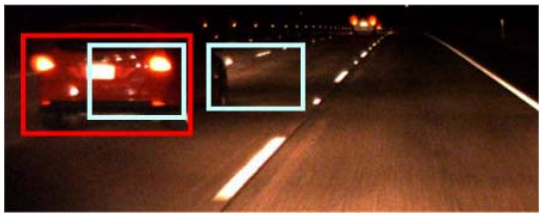


Fig. 4. Hypothesis windows shown in blue are eliminated using the geometric constraints.

is the first step to extract light blobs. However, in VeDANt, this is one of the verification steps.

Given a hypothesis window $r_H = [x \ y \ w \ h]$ in a test image I, we first segment two sub-images I_L and I_R from the top left and right corners of the window r_H. The widths and heights of I_L and I_R are $\frac{1}{3}w$ and $\frac{1}{3}h$ respectively. These windows *could* contain the taillight in the window r_H if r_H indeed contains a vehicle. Next, the red channel of the sub-images are used to generate four vectors $\mathbf{p_L^H}$, $\mathbf{p_R^H}$, $\mathbf{p_L^V}$, $\mathbf{p_R^V}$ which are defined as follows:

$$\mathbf{p_L^H}(i) = \sum_j I_L(i,j) \quad \mathbf{p_R^H}(i) = \sum_j I_R(i,j) \tag{4}$$

$$\mathbf{p_L^V}(j) = \sum_i I_L(i,j) \quad \mathbf{p_R^V}(j) = \sum_i I_R(i,j) \tag{5}$$

Indices i and j in the above equations are constrained within $[0, \lceil \frac{w}{3} \rceil - 1]$ and $[0, \lceil \frac{h}{3} \rceil - 1]$. In the above equations, $\mathbf{p_L^H}$ and $\mathbf{p_R^H}$ refer to the summation of red channel intensities along the vertical (y-direction) axis in I_L and I_R respectively. Similarly $\mathbf{p_L^V}$ and $\mathbf{p_R^V}$ include the summation of intensities along the horizontal direction.

$\mathbf{p_L^H}$, $\mathbf{p_R^H}$, $\mathbf{p_L^V}$ and $\mathbf{p_R^V}$ are normalized, and then are used to compute two correlation scores using the following equations:

$$\beta_C^H = \frac{\sum_i \overline{\mathbf{p_L^H}}(i) \cdot \overline{\mathbf{p_R^H}}(i)}{\lceil w/3 \rceil} \quad \beta_C^V = \frac{\sum_j \overline{\mathbf{p_L^V}}(j) \cdot \overline{\mathbf{p_R^V}}(j)}{\lceil h/3 \rceil} \tag{6}$$

In the above equations, $\overline{\mathbf{p}}$ refers to the normalized vector.

Next, if β_C^H and β_C^V are both above a set of thresholds $T_{\beta H}$ and $T_{\beta V}$, then we consider the next step (we will derive both these thresholds using the learning data in the forthcoming sub-sections). If either of the thresholds are not met, then the hypothesis window r_H is rejected and no further processing is done.

If β_C^H and β_C^V satisfy $T_{\beta H}$ and $T_{\beta V}$ respectively, we segment possible taillights. This is done by checking the normalized vectors $\overline{\mathbf{p_L^H}}$, $\overline{\mathbf{p_R^H}}$, $\overline{\mathbf{p_L^V}}$ and $\overline{\mathbf{p_R^V}}$. We describe the method for the left taillight using $\overline{\mathbf{p_L^H}}$ and $\overline{\mathbf{p_L^V}}$, which are used to find the horizontal and vertical bounds of the taillight in I_L in the following way:

$$\begin{aligned} i_{Lmin} &= \min\left(\arg\left(\overline{\mathbf{p_L^H}} > T_m\right)\right) \\ i_{Lmax} &= \max\left(\arg\left(\overline{\mathbf{p_L^H}} > T_m\right)\right) \end{aligned} \tag{7}$$

$$\begin{aligned} j_{Lmin} &= \min\left(\arg\left(\overline{\mathbf{p_L^V}} > T_m\right)\right) \\ j_{Lmax} &= \max\left(\arg\left(\overline{\mathbf{p_L^V}} > T_m\right)\right) \end{aligned} \tag{8}$$

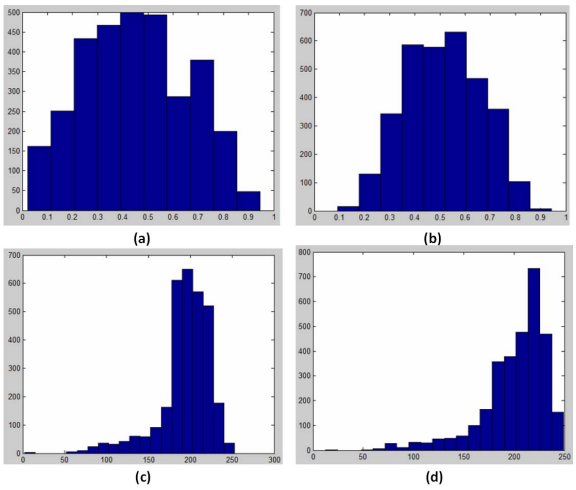


Fig. 5. Distributions are generated using the training data for: (a) β_C^H, (b) β_C^V, (c) $\mu_{L_{tail}}$, and (d) $\mu_{R_{tail}}$.

In words, the horizontal and vertical bounds of the taillight in I_L are obtained by checking where $\mathbf{p_L^H}$ and $\mathbf{p_L^V}$ are above $T_m = 0.5$. The left taillight is therefore segmented from I_L using the bounds in (7) and (8) resulting in the image sub-window $I_{L_{tail}}$. Similarly $I_{R_{tail}}$ is extracted using the normalized vectors and bounds for the right image sub-window I_R. The mean red intensities $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$ for the two taillight windows are computed and checked against T_{μ_L} and T_{μ_R} respectively. If these conditions are satisfied, then the hypothesis window is considered to have a vehicle, otherwise it is rejected.

3) Determining Verification Parameters: There are four main thresholds that are needed in the hypothesis verification step. They are two thresholds for the correlation measures (i.e. T_{β^H} and T_{β^V}), and two thresholds for the mean intensity values for the left and right taillights (i.e. T_{μ_L} and T_{μ_R}). In order to determine these parameters, we use the training data that was previously used for training the classifiers. Fig. 5 (a)-(d) shows four distributions for β_C^H, β_C^V, $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$ respectively using the 8000 training samples. These distributions are used to find the mean and standard deviations for each of the four parameters. For the correlation metrics, the thresholds are set one standard deviation below the mean, i.e., $T_\beta = \mu_{f(\beta)} - \sigma_{f(\beta)}$. In the case of mean intensities of the taillights, we use the following: $T_\mu = max(100, \mu_{f(\mu_I)} - 3\sigma_{f(\mu_I)})$.

C. Tracking Using Kalman Filters

The main focus of this paper is primarily on detection rather than tracking. We have however used tracking to highlight the efficiency of VeDANt's detection operation, and also for comparing with other methods that employ tracking. We employ an extended Kalman filter (EKF) for tracking the detection windows in VeDANt. This step is similar to other works that use EKF for tracking objects such as pedestrians and vehicles [2], [21]. Therefore, we keep this description short. The following are the state space equations for tracking the vehicles in VeDANt using EKF:

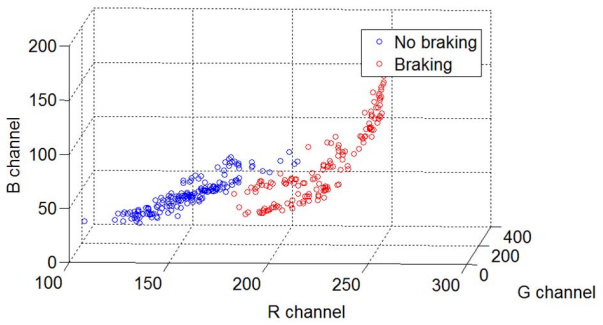


Fig. 6. Scatter plot of RGB channels when taillights are braking and not braking.

$$X^{t+1|t} = AX^{t|t}, \quad X = \begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where the vector X represents the properties of the detection window.

IV. VeDANt for Brake & Turn Light Indicator Detection

The generation of detection windows followed by tracking in VeDANt is used further to analyze the dynamics of the leading vehicles. VeDANt analyzes the tracked detection windows in a temporal manner to determine the taillight activity of the vehicles.

A. Brake Light Activity Detection

In order to detect braking event, the taillight regions are segmented from the detection windows using the techniques presented in Section III-B.2 and the mean red channel values are computed for the left and right taillights, i.e., $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$. These values were previously compared with T_{μ_L} and T_{μ_R} respectively to verify the presence of the vehicle. In addition to the above check, if $\mu_{L_{tail}} > T_H$ and $\mu_{R_{tail}} > T_H$, then the detected vehicle is considered as braking. T_H is the high intensity threshold, which indicates that the taillights are lit up further due to braking. T_H is determined from the training data. Fig. 6 shows the plot of RGB channels for two sets of taillights - no braking activity and braking activity. It can be seen from Fig. 6 that the two classes are separable along the red channel. $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$ also correspond to the red channel. T_H is set to 200 based the data plotted in Fig. 6. Therefore, if $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$ are both found to be greater than 200, the vehicle is said to be braking.

B. Turn Light Indicator Detection

In order to determine if a vehicle has put its turn indicator on, we check $\mu_{L_{tail}}$ and $\mu_{R_{tail}}$ temporally. The Kalman tracker enables tracking using the parameters of the detection windows as described in Section III-C. It enables continuous detection of vehicles and removes miss-detections. However, it does not directly add the IDs to the track, i.e., it does not say

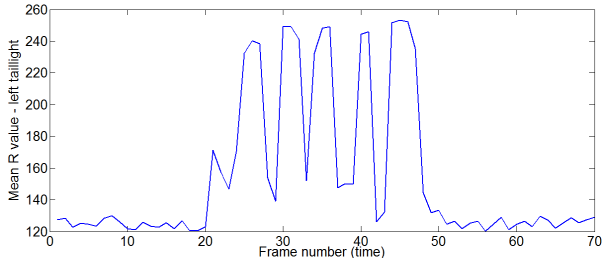


Fig. 7. Turning activity indicated by R channel of left taillight.

whether the vehicle being detection is part of a particular track. An additional computation is performed to determine the track ID. Given two detection windows r^t and r^{t-1} from current frame t and previous frame $t-1$ respectively, $\delta_x^t = |x^t - x^{t-1}|$ and $\delta_y^t = |y^t - y^{t-1}|$ are computed. If δ_x^t and δ_y^t are both less than T_δ, then the same track ID is assigned to both r^t and r^{t-1}. This process enables generating tracks of vehicles. The tracks are generated for N_f frames from the current time instance (or frame).

Once the track ID is assigned to a vehicle, we consider a maximum of previous N_f frames and determine α_L which is given by:

$$\alpha_L = \frac{\text{Number of frames with } \mu_{L_{tail}} > T_H}{N_f} \tag{10}$$

Similarly α_R is computed for the right taillight sub-window. If $\alpha_L > 0.4$ and $\alpha_R < 0.4$, then the vehicle is considered to have switched on the left turn light. In the same way, the right turn indicator is detected. Fig. 7 shows the mean red channel value for the left taillight over a series of frames. The taillight activity of left turn is indicated by the varying mean R value of the left taillight region as indicated in Fig. 7.

V. Datasets & Evaluation

In this section, we first present new datasets for nighttime vehicle detection, followed by a detailed performance evaluation of VeDANt using the new datasets.

A. LISA-Night Datasets

A survey of literature shows that except for the iROADs dataset in [22], all other datasets in [9], [10], [13], and [14] are proprietary night vehicle datasets and are not publicly available. Fig. 8 shows sample test images that are copied from the respective papers. It can be seen that most of these test images usually involve input image frames with low density traffic and/or catered for low exposure conditions.

In this paper, we introduce three new public datasets for evaluating and benchmarking nighttime vehicle detection, which are particularly intended to capture complex driving conditions during nighttime. These datasets are henceforth named as *LISA-Night* datasets, which include the following three different types of datasets: (1) Multilane-Free-Flowing dataset, (2) Multilane-Stop-Go dataset, and (3) Low-Density dataset. The three datasets are generated from the right camera of a stereo vision capture system that is placed on top of

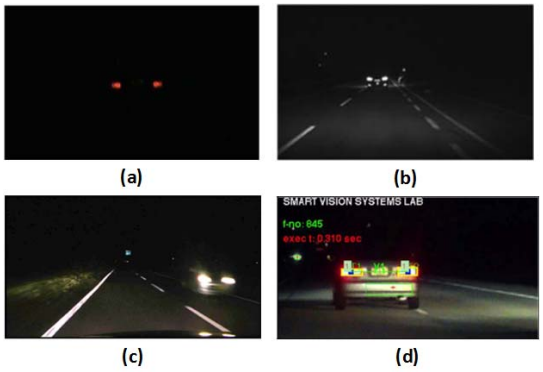


Fig. 8. Sample test images from other datasets that published in the respective papers: (a) Low exposure nighttime dataset from [9], (b) Dataset from [13] that caters to low exposure conditions, (c) Light blob detection dataset from [14], and (d) Night dataset from [10].

the testbeds in Laboratory for Intelligent and Safe Automobiles (LISA), UCSD. The three datasets are collected under different exposure settings of the lens such that they satisfy the minimum luminosity value which will be described in the next sub-section. Therefore, the datasets present three different lighting conditions during nighttime driving. The datasets were collected at 15 frames per second.

1) Lighting Conditions in LISA-Night Datasets: Similar to daytime conditions, varying ambient lighting is a challenge for nighttime vehicle detection also. Unlike works like [9], VeDANt operates under conditions where there are sources of ambient lighting. In order to determine the possible conditions when VeDANt operates, the following experiment was conducted using two different vehicles.

Each vehicle was parked on a road (tar surface) where there was no source of ambient lighting, i.e. street lamps, other vehicles etc. Additionally, a no moon day was chosen to conduct the experiment so that the ambient lighting due to moonlight is also avoided. The vehicle headlights were turned on and a series of images were captured using twelve different exposure settings of the camera. In addition, a marker was placed at distance of 10 meters from the front of the vehicle.

After collecting the images using the setup describe above, a 50×50 pixels patch is selected near the marker (which corresponds to 10 meters from the vehicle) from each image. The mean luminosity value L for each patch (i.e. for each exposure setting) is computed. This is repeated for the second vehicle also.

Fig. 9 shows the luminosity plots for the two vehicles. It can be seen that as the exposure setting of the camera is increased, the luminosity of the patch at 10 meters from the ego-vehicle (with the headlights switched on) shows similar trend for both vehicles.

In order to create the LISA-Night datasets, we set a minimum threshold on the luminosity value of 400 units, which must be satisfied while capturing the data during nighttime. VeDANt is designed to operate on input image sequences that are captured in such ambient lighting conditions.

2) Datasets: The LISA-datasets which we propose as part of this paper, are particularly designed to capture complex

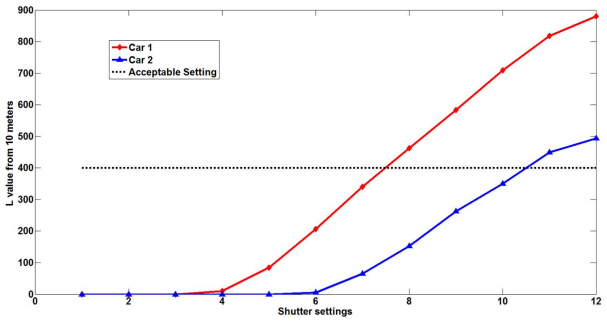


Fig. 9. Incident luminance settings that are captured in LISA-Night datasets.

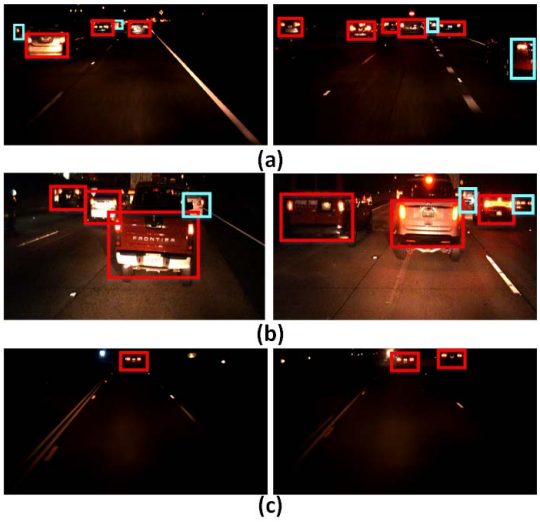


Fig. 10. LISA-Night datasets with annotations (red windows are fully visible rear-views and blue are partially visible rear-views (don't care windows for evaluation). (a) Multilane-Free-Flowing dataset: faster shutter speeds (medium exposure conditions) with multiple vehicles in a free flowing traffic; (b) Multilane-Stop-Go dataset: auto exposure conditions with varying lighting conditions, and high traffic conditions in a slow moving traffic; (c) Low-Density dataset: low exposure conditions with low traffic conditions and taillight activity of the leading vehicle.

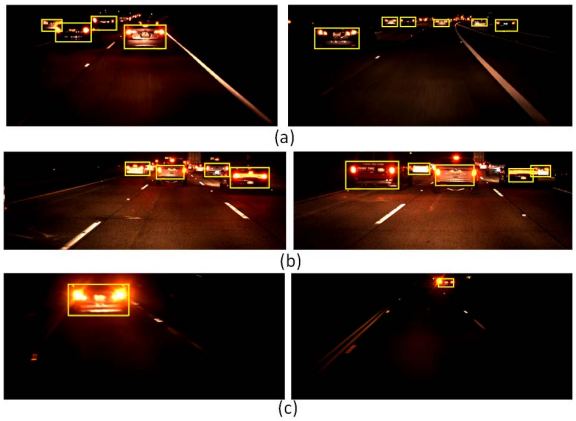


Fig. 11. Sample results from evaluation on LISA-Night datasets: (a) Multilane-Free-Flowing, (b) Multilane-Stop-Go, (c) Low-Density. In (a), the complexity is due to lower exposure conditions and multiple vehicles in each frame. In (b), the auto exposure settings add complexity in terms of lighting due to high traffic scenarios. Also, there are multiple types of vehicle taillights.

TABLE II
DETAILS OF LISA-NIGHT (LISA-N) DATASETS

Dataset	# frames	# ve-hi-cles	Camera Config.	Remarks
Multilane-Free-Flowing	1005	4500	Shutter speed = 32ms, Gain = 8	Highway driving conditions, multiple vehicles in multiple lanes, free flowing high traffic, high complexity due to multiple lanes and vehicles
Multilane-Stop-Go	1200	4500	Auto exposure, Auto gain	Highway driving conditions, multiple vehicles, slow moving traffic, taillight activity, very high complexity due to multiple lanes, vehicles and varying light conditions
Low-Density	3000	3050	Shutter speed = 4ms, Gain = 8	Usually one vehicle per frame, urban/city limit condition, tail-light activity of leading vehicle
Total	5205	12050		

driving conditions. Fig. 10 shows sample images from the three different LISA-Night datasets. Multilane-Free-Flowing dataset captures vehicles on a free-flowing multi-lane highway during nighttime. This dataset is generated from a real-world driving trip that was conducted on freeways in California, USA. It can be seen from Fig. 10(a) that Multilane-Free-Flowing includes input image frames with multiple vehicles (sometimes seven vehicles per frame). This increases the complexity in detecting vehicles during nighttime because existing methods that depend on taillight detection need to associate the correct pairs of taillights.

The second dataset which is titled as Multilane-Stop-Go comprises of 1200 frames. This dataset is collected from a drive during heavy traffic conditions on a multi-lane highway. There are multiple vehicles in multiple lanes, each creating a different lighting effect due to their braking/normal/turning light activities. Additionally, this dataset is generated under auto exposure conditions, which is usually the case when generic cameras are used with mobile computing devices [10]. The auto exposure condition increases the complexity further because it could introduce a bright lighting effect on the surface of the road and on the nearby vehicles especially in high traffic scenarios (see Fig. 10(b)).

The last dataset - Low-Density is particularly catered for detecting vehicle taillight activity during nighttime, while driving in city limit roads. Each of the 3000 frames in this dataset primarily consists of one vehicle per frame (most of the time), which is performing a series of turns and lane changes. Therefore, there is a taillight activity, i.e. blinking of taillights during lane changes and turns, and also due to intermittent braking. Sample images are shown in Fig. 10(c).

Table II summarizes the different properties of each of the dataset which include the total number of frames, the number of vehicles in each dataset, the exposure settings (shutter speed) and gain of the sensor. LISA-Night (LISA-N) datasets have a total of 5200 frames with over 12000 vehicles for testing and evaluation.

3) Annotations: All the frames in LISA-Night datasets are annotated as shown in Fig. 10. It is to be noted that the complete rear-views of the vehicles are annotated (denoted by red windows in Fig. 10). Since partial views are not in

TABLE III
ACCURACY EVALUATION OF VeDANt

	Without Tracking			With Tracking		
	Gray-level + Color		Gray-level only	Gray-level + Color		Gray-level only
	TPR	FDR	FDR	TPR	FDR	FDR
Multilane-Free-Flowing	97.4	1.11	4.5	99.31	0.27	2.35
Multilane-Stop-Go	92.3	4.5	6.2	95.8	2.5	4.25
Low-Density	98.5	2.36	3.56	100	0.1	3.56

the scope of this paper, annotations of the fully visible rear-views are discussed here. However, partial views are marked as 'don't care' detections (blue windows in Fig. 10). Also, vehicles within 90 meters and of height greater than 25 pixels from ego-vehicle are considered for evaluation. The vehicles beyond this distance or less than 25 pixels in height are considered as don't care detections.

Additionally, Low-Density dataset is also annotated with taillight activity. The following three taillight activities are annotated in this dataset: (1) Lane change blinker activity, (2) Turn indicator activity, (3) Braking activity. It is to be noted that the LISA-N datasets are the most comprehensively annotated 'publicly' available datasets for nighttime vehicle detection, which include complex driving conditions (based on a detailed literature search in public forums). In the forthcoming sections, the three datasets are used to evaluate the proposed method in different ways.

B. Performance Evaluation

VeDANt is evaluated for accuracy and performance using the new LISA-Night datasets. The results for the different datasets are presented and discussed in detail in this section. In order to perform the evaluation, we use the following two commonly used metrics for vehicle detection [5]. True Positive Rate (TPR) is computed as follows: $TPR = \frac{\text{Total detected vehicles}}{\text{Total number of vehicles}}$. TPR is also called hit rate or recall, and the miss rate can be computed using TPR using $Miss\ rate = 1 - TPR$. False Detection Rate (FDR) takes into account the total number of false positives or false detections, and is computed using $FDR = \frac{\text{False positives}}{\text{Detected vehicles + False positives}}$. Similar to TPR being equivalent to recall, FDR quantifies precision in the following way $FDR = 1 - precision$.

1) Detection Analysis: Table III shows the results of applying VeDANt on the three LISA-Night datasets. Two different sets of results are listed in Table III. The first part of Table III lists TPR and FDR for detection results that are obtained by applying VeDANt without enabling tracking. Furthermore the results are organized under two columns: Gray-level+Color and Gray-level only. The first column refers to the results obtained from applying the entire VeDANt algorithm, i.e. the classifiers which are not dependent on color followed by the hypothesis verification steps including color based processing (as described in Section III-B). Under the 'Gray-level only' column, only FDR is reported because the TPR remains the same. This is because the detections obtained from the classifiers (applied on gray level images) are further processed using color information in VeDANt. Therefore, application of color based processing in VeDANt aids in reducing the false positive windows only. It can be seen that the detection rates of VeDANt are over 97% for Multilane-Free-Flowing and Low-Density datasets, and 92% for Multilane-Stop-Go dataset. The lower percentage in the Multilane-Stop-Go dataset is because of the significantly higher complexity in the dataset with varying lighting and taillight conditions as shown in Fig. 10(c). It can also be seen that the actively learned classifiers are able to eliminate a large number of false positives yielding FDRs of less than 5% in the case of Multilane-Free-Flowing and Low-Density datasets. This percentage is slightly higher for complex dataset at 6.2%. It is to be noted that such low false detection rates are obtained by using gray-scale images only. Also, on adding the hypothesis verification step using the color information and perspective geometry, the FDRs have reduced by more than 50%.

It is to be noted that the above results are obtained without applying any tracking. The second part of Table III shows the results that are obtained after applying tracking to the detection windows in VeDANt. It can be seen that overall detection rates have improved to nearly 99% for Highway dataset and 100% for Low-Density dataset. The Complex dataset also shows higher detection rate of 95.8%. Additionally, the false detection rates have reduced to less than 3%.

Fig. 11 shows sample detection results from the three datasets. In Fig. 11(a), the detection results for Multilane-Free-Flowing dataset are shown. It can be seen that VeDANt is able to detect the vehicles to nearly 80m from the ego-vehicle. More details on distance based evaluation is also presented in this section. Fig. 11(b) shows the detection results on Multilane-Stop-Go dataset. It can be seen that the driving conditions which are captured in the Multilane-Stop-Go dataset are more complex as compared to the Multilane-Free-Flowing and Low-Density datasets. VeDANt is able to detect the vehicles that are posing multiple challenges such as varying taillight activities, ambient lighting conditions and varying types of taillights (e.g. the second row in Fig. 11(b)). Finally, in the last row (Fig. 11(c)), the detection results from Low-Density dataset are shown. As mentioned previously, Low-Density dataset is particularly aimed at capturing the challenges due to taillight activity. Two different cases of taillight activity are captured in Fig. 11(c): braking and left turn indicator. VeDANt is able to detect the different cases of taillight activity in the Low-Density dataset. More quantifiable results on activity based analysis are presented later in this section.

In addition to the three LISA-Night datasets, VeDANt was also applied to naturalistic driving data [23], [24] where long video sequences were analyzed for vehicles during nighttime. We applied VeDANt on a video sequence with 20000 frames, which was captured during a naturalistic drive on a freeway during night. VeDANt shows TPR of 97% and an FDR of 2.6% on this naturalistic driving data.

2) Taillight Activity Evaluation: In addition to vehicle detection during nighttime, LISA-Night datasets are also aimed at benchmarking taillight activity during nighttime.

Fig. 12. Sequence in Multilane-Stop-Go dataset with multiple braking events. Each color of the detection window represents a track that is identified as the same vehicle. A dashed window denotes braking activity.

TABLE IV
EVALUATION OF TURN & BRAKE ACTIVITY DETECTION BY VEDANT

Activity	TP/FP/Total	TPR/FDR	Det. Seq./Total Seq.	Mean Δ_T
Braking	155/10/160	96.8/6.06	14/14	-0.1
Left turn	70/5/74	94.59/6.66	8/8	-0.87
Right turn	85/5/88	96.59/5.54	11/11	-0.85

Low-Density and Multilane-Stop-Go datasets involve over 3000 frames where there are braking and turning activities by the leading vehicles. In Low-Density dataset, there is one vehicle in an urban road which performs a series of taillight activities. In Multilane-Stop-Go dataset, there are multiple vehicles in each frame, where one or more vehicles are either braking or changing lanes.

We evaluate the taillight light activity in three different ways. First, we evaluate the taillight activities using individual vehicles that are either braking or turning. Table IV lists the number of true positives (TPs), false positives (FPs) and total braking/turning vehicles. TPR and FDR are computed and listed in Table IV. It can be seen that VeDANt shows over 95% accuracy in detecting all the different activities with a false alarm rate of around 6%.

Next, we first report the number of sequences that are correctly detected. Unlike the number of braking/turning vehicles, a sequence is a set of frames when the braking/turning occurs. The number of instances that are annotated and the number of sequences that are detected are indicated in Table IV. It can be seen that VeDANt is able to detect all the instances of the taillight activity correctly. It is to be noted that this evaluation is done in both the Low-Density dataset and the complex Multilane-Stop-Go dataset. In the Multilane-Stop-Go dataset, there are multiple vehicles that are braking. VeDANt is able to detect all such events. Fig. 12 shows one such sequence where braking event of multiple vehicles is tracked by VeDANt. Each color of the detection window denotes a track, i.e., the same color is assigned to the same vehicle that is being tracked for braking and turning activity. A dashed box denotes detection of a braking event in Fig. 12.

However, mere detection of the activity is not sufficient. Therefore, we introduce another metric termed as 'Δ_T', which is defined as:

$$\Delta_T = T_{GT} - T_{ALGO} \tag{11}$$

where T_{GT} refers to the time instance when the taillight activity begins according to ground truth (GT), and T_{ALGO} indicates the time instance when the algorithm (ALGO) detects the taillight activity. Both T_{GT} and T_{ALGO} are referenced with respect to start of the video sequence (Low-Density). Therefore, for a given taillight activity, Δ_T shows how early does the algorithm (VeDANt in this paper) detect the activity with respect to the ground truth. Table IV lists the *mean* Δ_T for each activity, where the mean refers to the average of the Δ_T's that are obtained from the different activities under each category. It can be seen that VeDANt detects braking activity as early as 0.15 seconds when compared to the ground truth. The negative sign indicates that VeDANt detects the activity after the actual activity starts (as annotated in the ground truth). VeDANt takes a little longer to detect the turn activities giving mean Δ_T's around -0.85 seconds.

3) Computational Performance: VeDANt is evaluated for computational performance in this section. It is currently implemented in MATLAB, and the algorithm is evaluated on a personal computer with Intel Xeon 3.10GHz quadcore processor and 8GB RAM. VeDANt takes an average of 0.071 seconds per frame, which is equivalent 14 frames per second. This includes classification using the actively learned Adaboost classifiers followed by hypothesis verification steps and tracking. It is to be noted that the current implementation does not involve any optimization with regards to speed, and a more optimized OpenCV implementation using C/C++ usually gives a higher performance than MATLAB-based implementation.

C. Comparison With Existing Methods

In the survey presented in Section II, it is shown that most existing methods use proprietary datasets for evaluations. Most color segmentation based methods such as [9] and [10] use the color property of taillights being explicitly visible and separated from the background (as seen in sample images in Fig. 8). VeDANt is designed for different types of *complex* lighting conditions, wherein there can be different kinds of ambient lighting. This is particularly the case with natural driving conditions on highways, urban/city limit conditions and high traffic conditions. Therefore, directly comparing VeDANt with existing methods such as [9] and [14] is not fair.

However, we list some quantitative results from the different works and show that under the complex lighting conditions that are captured in LISA-Night datasets, VeDANt gives similar or better detection results than existing works. Table V lists the different detection results. One of the most notable works in nighttime vehicle detection is by O'Malley *et al.* [9]. A detection rate of 97.26% with a false detection rate of 2.58% is obtained by applying tracking on the detections obtained using the method described in [9]. A more recent work is [14] which is particularly designed for light blob detection. The detection rates are particularly meant for finding light blobs. A valid light blob detection rate of 92.5% is

TABLE V

COMPARISON RESULTS OF EXISTING WORKS ON *LOW* COMPLEXITY DATASETS VERSUS VEDANT ON *HIGH* DENSITY AND COMPLEX LISA-N DATASETS

Method	**Test Dataset**	**Detection rate (%)**	**False detection rate (%)**
O'Malley 2010 [9]	Low exposure, *low* density, *low* complexity	97.26	2.58
Alcantarilla 2011 [16]	Low exposure, *low* density, *low* complexity	96	6
Chen 2012 [17]	Auto exposure	89	3
Kosaka 2015 [14]	*Low* density, *low* complexity	92.5	11
Almagambetov 2015 [10]	*Low* to medium density	97.0	Not reported
This work - VeDANt 2015 on LISA-N-Multilane-Free-Flowing	*Very high* complexity, multiple-vehicles & medium exposure	97.4/99.31*	1.11/0.27*
This work - VeDANt 2015 on LISA-N-Multilane-Stop-Go	*Very high* density, & auto exposure, *very high* complexity	92.3/95.8*	4.5/2.5*
This work - VeDANt 2015 on LISA-N-Low-Density	Low exposure & varying taillight activities	98.5/100*	2.36/0.10*

* before tracking/after tracking

TABLE VI

COMPARISON BETWEEN COLOR+GEOMETRY BASED METHOD AND VEDANT USING LISA-N DATASETS

	LISA-Low-Density		LISA-Multilane-Freeflow	
	TPR	FDR	TPR	FDR
[9] + geometric constraints)	55.2	42.2	57	70.1
VeDANt (This Work)	98.5	2.36	97.4	1.11
VeDANt with Tracking (This Work)	100	0.1	99.31	0.27

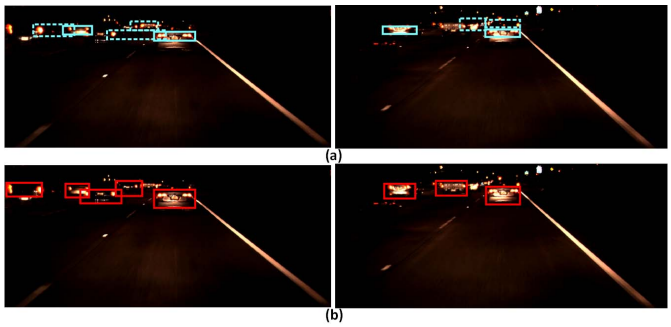


Fig. 13. (a) Detection results using rule based method. Dotted boxes are false positives. (b) Detection results using VeDANt on the same frames.

obtained with a 'nuisance' (false positive) light blob detection rate of 11%. The most recent work on nighttime vehicle detection is by Almagambetov *et al.* [10] which reports a detection rate of 97% (false detection rates are not reported).

We would like to highlight the datasets which are used for evaluation. Most works are evaluated on low density traffic (as shown in sample images in Fig. 8). Lower density also implies lower complexity in robustly detecting the vehicles. In contrast, VeDANt is evaluated using highly complex LISA-N datasets that have multiple vehicles (upto as high as 7 vehicles per frame) and varying ambient lighting conditions. In Table V, we show two values for TPR and FDR for VeDANt (we present the rates before and after tracking). It can be seen that VeDANt either outperforms or achieves similar detection rates on more complex LISA-N datasets as compared to existing methods on lower complexity datasets.

Additionally, we implemented a rule based method and evaluated it on the new LISA-N datasets. This method is largely based on the color based method in [9] which is one of the most cited vehicle detection methods for nighttime. In addition color based thresholding and shape matching method in [9], we added geometric constraints based on IPM as described previously in Section III-B. We denote this method as rule based method in Table VI and compare it against VeDANt by evaluating using the two LISA-N datasets. It can be seen that the rule-based method has TPRs of around 55% with false detection rates as high as 70%. Such poor results are because of low true positives and high number of false positives and negatives. In the case of the Urban dataset, there is only one vehicle under low exposure conditions. However, there is taillight activity. The rule-based method successfully detects the vehicle when there is no taillight activity but fails otherwise because shape matching technique (according to [9]) fails whenever there is brake light activity. Therefore, there are high number of false negatives resulting in lower TPR. In the case of more complex Multilane dataset, the rule-based method results in high number of false positives because of multiple blobs that occur due to presence of multiple vehicles in the test images. The presence of such large number of light blobs mis-associates the blobs leading to high false positives. This is shown in Fig. 13(a), where false positives are shown by dotted boxes. Therefore, TPR is low and FDR is higher compared to the Urban dataset. Therefore, it can be seen that rule-based methods that are dependent on color channel thresholding and geometric constraints do not perform as well as VeDANt which introduces learned classifiers as the first step to get hypothesis windows followed by rules.

VI. CONCLUSIONS

In this paper, we have shown that introducing learned classifiers to generate hypothesis windows followed by a post processing step involving color and geometry based rules results in more robust detection of vehicles during nighttime than most existing rule-based methods. Unlike existing methods, the actively learned classifiers in VeDANt operate on gray level images which are used for extracting proposal windows. Additionally, tracking the detection windows followed by color based processing also enables a more efficient way of analyzing the dynamics of the vehicles rather than tracking individual taillight blobs. This paper also introduces LISA-Night datasets that capture a variety of complex driving conditions involving multiple lanes, multiple vehicles, varying lighting and traffic conditions. The proposed LISA-Night datasets are first of its kind in terms of the complexity of the dataset and the types of annotations in the context of nighttime vehicle detection.

Although VeDANt achieves high detection accuracy, it is to be noted that VeDANt is particularly designed for detecting rear-views of vehicles that are fully visible in the frame. The main advantage of VeDANt is the presence of learned classifiers that generate reliable hypothesis windows. The classifier accuracy will be affected if it were to be trained for partial views of the vehicles. This is because there are fewer visible regions during nighttime and partial views will not give enough features for training the classifiers reliably. Therefore, newer features need to be defined for VeDANt to function on partial views of vehicles. Future work includes detecting partial views of vehicles during nighttime with high levels of accuracy. Additionally, detection itself is not the only objective; tracking and analyzing the behavior of the vehicles is more critical than just detection of the vehicles. This also presents possible future opportunities in this research area.

ACKNOWLEDGMENTS

The authors would also like to thank their colleague Sean Lee, Frankie Lu, Mark Philipsen, Morten Jensen, Miklas Kristoffersen and Jacob Dueholm from Laboratory for Intelligent and Safe Automobiles for their efforts in collecting and annotating valuable data during the course of this research.

REFERENCES

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006.

[2] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[3] A. Doshi, B. T. Morris, and M. M. Trivedi, "On-road prediction of driver's intent with multimodal sensory cues," *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 22–34, Jul. 2011.

[4] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 90–104, Mar. 2016.

[5] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, Jun. 2010.

[6] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, Jun. 2013.

[7] R. K. Satzoda and M. M. Trivedi, "Efficient lane and vehicle detection with integrated synergies (ELVIS)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 708–713.

[8] R. K. Satzoda and M. M. Trivedi, "Multipart vehicle detection using symmetry-derived analysis and active learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 926–937, Apr. 2016.

[9] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 453–462, Jun. 2010.

[10] A. Almagambetov, S. Velipasalar, and M. Casares, "Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3732–3741, Jun. 2015.

[11] C. Hughes, R. O'Malley, D. O'Cualain, M. Glavin, and E. Jones, "Trends towards automotive electronic vision systems for mitigation of accidents in safety critical situations," in *New Trends and Developments in Automotive System Engineering*, M. Chiaberge, Ed. Intech., Jan. 2011, pp. 494–512.

[12] (May 2010). *Fatality Analysis Reporting System: Fatal Crashes by Weather Condition and Light Condition*. [Online]. Available: http://www-fars.nhtsa.dot.gov/Crashes/CrashesTime.aspx

[13] S. Eum and H. G. Jung, "Enhancing light blob detection for intelligent headlight control using lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 1003–1011, Jun. 2013.

[14] N. Kosaka and G. Ohashi, "Vision-based nighttime vehicle detection using CenSurE and SVM," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2599–2608, Oct. 2015.

[15] J. H. Connell, B. W. Herta, S. Pankanti, H. Hess, and S. Pliefke, "A fast and robust intelligent headlight controller for vehicles," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 703–708.

[16] P. F. Alcantarilla *et al.*, "Automatic lightbeam controller for driver assistance," *Mach. Vis. Appl.*, vol. 22, no. 5, pp. 819–835, Mar. 2011.

[17] D. Y. Chen, Y. H. Lin, and Y. J. Peng, "Nighttime brake-light detection by Nakagami imaging," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1627–1637, Dec. 2012.

[18] H. Tehrani, T. Kawano, and S. Mita, "Car detection at night using latent filters," in *Proc. Intell. Veh. Symp.*, Jun. 2014, pp. 839–844.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[20] S. Dasgupta, "Analysis of a greedy active learning strategy," in *Proc. NIPS*, Jan. 2005, pp. 1–15.

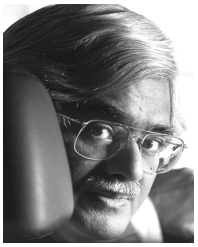
[21] S. Y. Chen, "Kalman Filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.

[22] M. Rezaei and M. Terauchi, "Vehicle detection based on multi-feature clues and Dempster-Shafer fusion theory," in *Proc. 6th Pacific-Rim Symp. Image Video Technol.*, 2014, pp. 60–72.

[23] R. K. Satzoda and M. M. Trivedi, "Drive analysis using vehicle dynamics and vision-based lane semantics," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 9–18, Feb. 2015.

[24] R. K. Satzoda, P. Gunaratne, and M. M. Trivedi, "Drive analysis using lane semantics for data reduction in naturalistic driving studies," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 293–298.

Ravi Kumar Satzoda (M'13) received the B.Eng. (Hons.), M.Eng., and Ph.D. degrees from Nanyang Technological University, Singapore, in 2004, 2007, and 2013, respectively. He is currently a Post-Doctoral Fellow with the Laboratory for Intelligent and Safe Automobiles, University of California at San Diego. His research interests include computer vision, embedded vision systems, and intelligent vehicles.



Mohan Manubhai Trivedi (F'08) received the B.E. degree (Hons.) in electronics from the Birla Institute of Technology and Science, Pilani, India, in 1974, and the M.S. and Ph.D. degrees in electrical engineering from Utah State University, Logan, UT, USA, in 1976 and 1979, respectively. He is currently a Professor of Electrical and Computer Engineering. He has also established the Laboratory for Intelligent and Safe Automobiles, and Computer Vision and the Robotics Research Laboratory, University of California at San Diego, where he and his team are currently pursuing research in machine and human perception, machine learning, intelligent transportation, driver assistance, active safety systems, and naturalistic driving study analytics. He received the IEEE Intelligent Transportation Systems Society's Highest Honor, Outstanding Research Award in 2013.