Physical Features and Deep Learning-based Appearance Features for Vehicle Classification from Rear View Videos

Rajkumar Theagarajan, *Student Member, IEEE*, Ninad S. Thakoor, *Member, IEEE*, and Bir Bhanu, *Life Fellow, IEEE*

***Abstract*— Currently, there are many approaches for vehicle classification, but there is no specific study on automated, rear view, and video-based robust vehicle classification. The rear view is important for intelligent transportation systems since not all states in the United States require a frontal license plate on a vehicle. The classification of vehicles, from their rear views, is challenging since vehicles have only subtle appearance differences and there are changing illumination conditions and the presence of moving shadows. In this paper, we present a novel multi-class vehicle classification system that classifies a vehicle into one of four possible classes (sedan, minivan, SUV, and a pickup truck) from its rear view video, using physical and visual features. For a given geometric setup of the camera on highways, we make physical measurements on a vehicle. These measurements include visual rear ground clearance, the height of the vehicle, and the distance between the license plate and the rear bumper. We call these distances as the physical features. The visual features, also called appearance-based features, are extracted using convolutional neural networks from the input images. We achieve a classification accuracy of 93.22% and 91.52% using physical and visual features, respectively. Furthermore, we achieve a higher classification accuracy of 94.81% by fusing both the features together. The results are shown on a dataset consisting of 1831 rear view videos of vehicles and they are compared with various approaches, including deep learning techniques.**

***Index Terms*— Convolutional neural networks, feature level fusion, multi-frame analysis, physical features, visual features, vehicle classification on highways/freeways.**

I. Introduction

THE growth of population and economic prosperity has led to a huge increase in the number of vehicles around the world. This requires an increasing need for automated and efficient classification techniques for different vehicle categories for diverse applications such as automatic toll collection, efficient use of parking spaces, etc. Vehicle classification systems should be robust to changes due to illumination, shadows, occlusions, viewpoint changes, and other distortions etc. Most approaches use computer vision and pattern recognition based systems, which use feature extraction to detect and classify a vehicle in still images and video streams. These systems are the best solution for vehicle classification as they are easy to install, relatively cheap, and provide direct visual feedback and flexibility in mounting. However, this is not a trivial task, the main problem being the selection of a feature set that is discriminative and provides the best possible classification of a vehicle type.

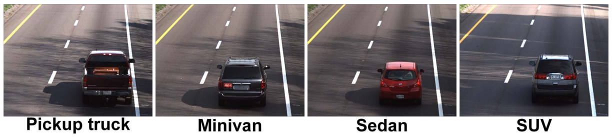


Fig. 1. Examples of the direct rear view of moving vehicles.

On an average 86.2 million cars are manufactured every year with an average increase of 2.75% every year since 2010 [1]. This leads to an increase in the aesthetic similarity among different vehicle models. A human subject can identify the class of a vehicle with a quick glance of digital data but to accomplish this with a computer is not as straight forward. Several problems such as handling shadows and occlusion, robust tracking of moving vehicles, lack of color invariance, etc. must be carefully considered in order to design an effective and robust automatic vehicle classification system which can work in real-world conditions. Fig. 1 shows rear view images of vehicles from our datasets.

The methods for vehicle classification can be broadly split into two categories: discriminative and generative. Discriminative classifiers learn the decision boundary between different classes, whereas generative classifiers learn the underlying distribution of a given class. In this paper, we propose a discriminative multi-class vehicle classification system that classifies a vehicle using physical and visual features, given its direct rear view, into one of four classes: Sedan,

Manuscript received March 22, 2018; revised October 17, 2018 and February 15, 2019; accepted February 16, 2019. This work was supported by a Grant from NSF 1330110 and PIPS/3M. The Associate Editor for this paper was S. C. Wong. *(Corresponding author: Rajkumar Theagarajan.)*

R. Theagarajan and B. Bhanu are with the Center for Research in Intelligent Systems, University of California at Riverside, Riverside, CA 92521 USA (e-mail: rthea001@ucr.edu; bhanu@cris.ucr.edu).

N. S. Thakoor was with the Center for Research in Intelligent Systems, University of California at Riverside, Riverside, CA 92521 USA. He is now with Cognex Corporation, Natick, MA 01760 USA (e-mail: nthakoor@ece.ucr.edu).

Digital Object Identifier 10.1109/TITS.2019.2902312

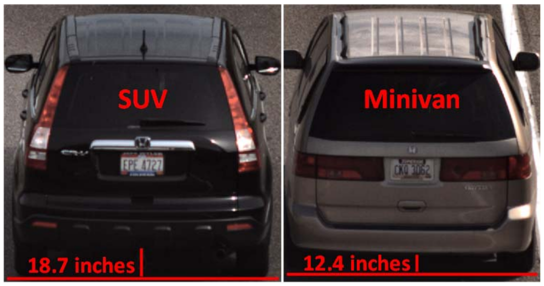


Fig. 2. Comparison of VRGC between SUV and Minivan.

Pickup truck, SUV and Minivan. As of 2012, these four classes constituted 233.76 million registered vehicles in USA, which is approximately 93.07% of the total registered vehicles in USA [2].

Physical measurements such as the computed height, the **visual rear ground clearance (VRGC)** of a vehicle and distance between the bottom edges of the license plate and the rear bumper are used. These distances are estimated in inches and we call them as the physical features. Generally, ground clearance of a vehicle is defined as the distance between the lowest mechanical part of a vehicle and the flat ground surface. However, for the given geometrical setup of a camera in our application, the camera cannot see the bottom most mechanical part of a vehicle. Therefore, we rename the traditional ground clearance as **Visual Rear Ground Clearance (VRGC)** in this paper. It is defined as the distance between the lowest visible rear part of a vehicle **(bottom edge of the rear bumper)** and the ground surface as viewed by a camera in a given geometric setup installed above the road/freeway/highway along which a vehicle is travelling.

The visual features are obtained by training a Convolutional Neural Network (CNN) to classify vehicles from the rear view. After training the CNN, we extract features from the final fully connected layer and call these as visual or appearance-based features. In our approach, we automatically compute the physical and visual features from a video and fuse them using SVM for classifying a vehicle. The fused feature vector consists of the VRGC, height of a vehicle, the distance between the license plate and rear bumper and features extracted by the CNN.

It is to be noted that VRGC is a novel and important feature. Following are the important characteristics of VRGC:

- Visual differences between a SUV and minivan from the rear view are quite subtle. VRGC is robust in differentiating a SUV from a minivan in rear view videos of a vehicle as shown in Fig. 2.
- Majority of the freeways/highways in the United States have cameras that look at the rear view of vehicles, because frontal license plates are not required in all states. Therefore, classification techniques that use the frontal/side profiles of a vehicle will not be applicable.
- VRGC is scale invariant and it is effective in performing rear view based classification with already existing freeways/highways cameras, thus eliminating the need for significantly new hardware.

In summary, this paper develops a multi-class rear view video vehicle classification system which classifies vehicles into four classes namely: sedan, pickup truck, SUV and minivan. It introduces a novel feature called Visual Rear Ground Clearance. It computes physical and deep learning-based visual features and fuse them together for classification. It validates the results with 1,831 real-world vehicle rear view videos.

This paper is organized as follows. Section II highlights the related work and the contributions of this paper. The technical approach is explained in Section III. Experimental results are shown and discussed in detail in Section IV. Finally, Section V concludes the paper.

II. Related Work and our Contributions

Automated booths on freeways are widely used for toll collection. These booths generally rely on reading the license plate of a vehicle and retrieving the vehicle and driver information from the Department of Motor Vehicles (DMV) records. This method is prone to a failure for vehicle classification, in the case of a license plate theft, if the detected license plate is for a different class of vehicle. In 2014, the total number of stolen motor vehicles was 689,527 and approximately one out of every five stolen vehicles was due to license plate theft [3]. The average value of a stolen vehicle in 2014 was $6,537. Integrating the rear view vehicle classification along with reading the license plate can be used to improve the accuracy of automated tollbooths and also help detect vehicle theft.

A. Related Work

In the past, majority of the vehicle classification techniques utilized the side profile information of a vehicle. The limitation of this technique is that side profiles of vehicles are prone to occlusions on multi-lane roads. Additionally, surveillance cameras on freeways capture either the frontal or the rear view of vehicles. This limits the use of side view classification in real-world applications. A summary of the related works is shown in Table I.

To the best of our knowledge, the only approaches focused on rear view based classification were suggested by Bhanu and his associates [18]– [21]. Kafai and Bhanu [18] used the spatial information between landmarks of a vehicle (e.g. tail lights and license plates) and a dynamic Bayesian network for classification. Thakoor and Bhanu [19] used the variation in structural signature as vehicles moved forward to classify them. Thakoor and Bhanu [20] used a scale matching approach for estimating the height of vehicles. They related the standard dimension of the license plate with the number of pixels in an image to perform scale matching. This approach requires very accurate camera calibration and even the presence of small amount of noise can cause large variations in the scale. A key limitation of the approaches [18]– [20] is that they could not distinguish between SUV and minivans because they look visually very similar from the rear view.

Theagarajan *et al*. [21] were able to solve the problem of distinguishing between SUV and minivans from the rear view. The authors estimated the Visual Rear Ground

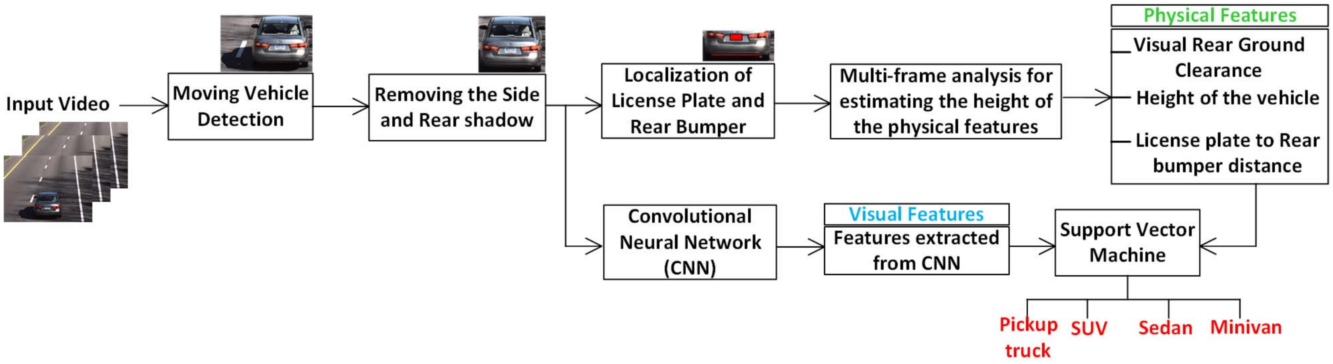

Input Video
Moving Vehicle Detection
Removing the Side and Rear shadow
Localization of License Plate and Rear Bumper
Multi-frame analysis for estimating the height of the physical features
Physical Features
Visual Rear Ground Clearance
Height of the vehicle
License plate to Rear bumper distance
Convolutional Neural Network (CNN)
Visual Features
Features extracted from CNN
Support Vector Machine
Pickup truck
SUV
Sedan
Minivan

Fig. 3. Overall architecture of our approach.

TABLE I
SUMMARY OF THE RELATED WORKS. * INDICATES THAT THE APPROACH IS NOT FULLY AUTOMATIC

Authors	View point	Comments
shan *et al.* [4]	Side view	Edge-based classification
Ma *et al.* [5]	Side view	Constellation model
Gupte *et al.* [6]	Side view	Dynamic patch modelling
Chen *et al.* [7]*	Side view	Manual segmentation
Psyllos *et al.* [8]	Frontal view	Neural network
Pearce *et al.* [9]	Frontal view	Harris corner strength
hsieh *et al.* [10]	Frontal view	Lane based normalization
Zhang *et al.* [11]	Frontal view	Vector Quantization
Dong *et al.* [12]	Frontal view	Laplacian filter learning
Cretu *et al.* [13]	Multi-view	Visual attention features
Ohn-bar *et al.* [14]	Multi-view	Clustering of features
Theagarajan *et al.* [15]	Multi-view	Ensemble of CNNs
Yu *et al.* [16]	Multi-view	CNN, Bayesian network
Zhuo *et al.* [17]	Multi view	CNN and pre-training
Kafai *et al.* [18]*	Rear view	Dynamic Bayesian network
Thakoor *et al.* [19]	Rear view	Structural signatures
Thakoor *et al.* [20]	Rear view	Scale matching
Theagarajan *et al.* [21]	Rear view	Low/ high VRGC
This Paper	**Rear view**	**Fusion of physical and visual features**

Clearance (**VRGC**) for binary classification as either a Low VRGC vehicle (sedan and minivan) or a High VRGC vehicle (SUV and pickup truck).

B. Contributions of This Paper

Unlike the previous work as shown in Table I, the contributions of this paper are:

(a) We propose a multi-class rear view vehicle classification system given the vehicles direct rear view as shown in Fig. 1. The main reasons for choosing the rear view are: *First*, vehicle classification using the side view is prone to occlusion from multi-lane traffic. *Second*, 19 states in the United States require only the rear license plate, hence frontal view classification is not suitable. *Third*, most of the cameras on freeways capture the rear view of moving vehicles, hence side/frontal view based classification techniques are not going to be widely applicable in the real-world.

(b) We use the physical height of features along with visual features that are computed using deep learning architectures for classifying a vehicle. This kind of fusion provides improved performance and it has have never been used before. The approach keeps the physical interpretation of height of features in the image of a vehicle and it is not lost by carrying out principal component or discriminant analysis for data dimensionality reduction.

(c) The experimental results have been validated on three datasets consisting of 1,831 rear view videos for four classes of vehicles namely: sedan, minivan, SUV and pickup truck. The deep learning network has been pre-trained on a database of 650,000 images of vehicles and then fine-tuned on rear view images. These results are shown with physical features and visual features alone and after the fusion of these features.

This paper is an extension of the preliminary paper by Theagarajan *et al.* [21] with significant advances in both theory and experimentation and overlap of less than 20%. In this paper, in addition, to the VRGC we estimate the height of the vehicle and spatial distance between the bottom edges of the license plate and rear bumper and fuse these features along with the visual features from deep learning to classify vehicles. Our approach does not require full camera calibration and requires only the height at which the camera is installed, depression angle and focal length of the camera as inputs in order to estimate the height of the physical features.

III. TECHNICAL APPROACH

This section describes the technical approach on how a vehicle is classified from the rear view video. First, the Region-of-Interest (ROI) for a moving vehicle is extracted and the ROI is processed to remove shadows followed by localization of the license plate and rear bumper. Next, the height of physical features is estimated using multi-frames.

The visual features are obtained by passing the processed ROI through CNN and extracting the features from the final fully connected layer. Finally, the extracted physical and visual features are fused using SVM to classify vehicles into one of the four classes. The complete proposed system is shown in Fig. 3. The components of the system are described below.

A. Localization and Analysis of ROI

This subsection explains the steps for detecting the ROI of a moving vehicle, processing the ROI to remove shadows and localizing the license plate and rear bumper.

1) Moving Vehicle Detection: In this paper a mixture of Gaussian models is used for moving vehicle detection. The R, G and B channels of the input color image are individually modeled as Gaussian distributions. If a given pixel in the current frame is within three standard deviations in the R, G and B color planes, then it is considered to be a background pixel, otherwise, it is a foreground pixel [22]. After scanning all the pixels, the pixels that belong to the foreground are cropped out. This results in an image containing the moving vehicle and its associated shadow.

The shadow associated with the moving vehicle consists of two separate parts: rear shadow and side shadow. The shadow immediately below the rear bumper of a vehicle is called the rear shadow, whereas the shadow on either the right or left side of the vehicle is called the side shadow. We need to separate these shadows [23] associated with the moving vehicle for two main reasons. *First*, the rear shadow can be misidentified as the rear bumper which will result in the incorrect estimation of the VRGC. *Second*, we need to track the side shadow in order to estimate the velocity at which the vehicle is travelling.

2) Removing the Side Shadow: All vehicles are designed such that they are bilaterally symmetric around their central vertical axis (reflection symmetry) from their rear view. We exploit this symmetry property to separate the moving vehicle from the side shadow.

We assume the axis of symmetry to be vertical based on the orientation of the ROI. From this assumption, we can conclude that the axis of symmetry corresponds to one of the columns of the ROI. To estimate the axis of symmetry, we first estimate edge magnitudes and orientation of edges using Gabor filters given by:

$$g(x;\, y;\, \theta;\, \lambda;\, \psi;\, \sigma;\, \gamma) = exp(\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) cos(2\pi \frac{x'}{\lambda} + \psi) \quad (1)$$

where, $x' = xcos\theta + ysin\theta$ and $y' = -xsin\theta + ycos\theta$. We use a bank of eight Gabor filters with orientation of $\theta = 0°, 22.5°, \ldots, 157.5°$. In equation (1) x and y are the locations of the pixels in the image, λ is the wavelength, θ is the orientation, γ is the aspect ratio and σ is the area of the receptive field of the Gabor filter [24]. The values of the parameters are given in Table II in section IV.

We apply surround suppression to the response of the Gabor filter to remove the background texture in the ROI. The surround suppression process checks if a small window W_1 around the edge *(x, y)* of the Gabor filters response reappears in a larger window W_2 around the edge *(x, y)*. If W_1 reappears in W_2, then the edge *(x, y)* is assumed to be a background texture and it is suppressed. Finally, we perform non-maximal suppression to get the final set of edges.

Next, we use a voting scheme to estimate the axis of symmetry. For any given column y of the image I with edge magnitude H and orientation θ, the votes G are counted as

$$\sum_{\forall x, y^-, y^+ : (x:y^+) \in I, (x:y^-) \in I} G(x, y^-, y^+) \quad (2)$$

$$G(x, y^-, y^+) = \begin{cases} min(H_{x,y^-}, H_{x,y^+}), & if \ \ \theta_{x,y^-} = \theta'_{x,y^+} \\ 0, & otherwise \end{cases} \quad (3)$$

Algorithm 1 Estimating the Axis of Symmetry

```
Input: Grayscale image of ROI
Output: Axis of symmetry (Column)
for y = 1 : width of the ROI
Initialize vote (y) = 0
   for $\varepsilon$ = 1 : min(y, width (ROI) - y)
      for x = 1 : length of the ROI
          $y^+ = y + \varepsilon$
          $y^- = y - \varepsilon$
          $\theta'_{x, y^+} = \pi - \theta_{x, y^-}$
          if ($\theta'_{x, y^+} = \theta_{x, y^-}$)
          vote (y) = vote (y) + min ($H_{x,y^-}$,$H_{x,y^+}$)
      end
   end
end
return vote (y)
```

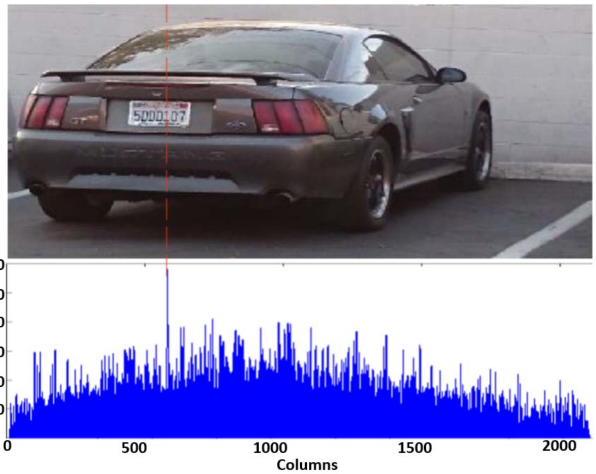
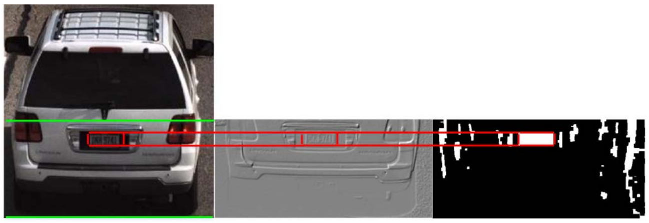


Fig. 4. Vehicle axis of symmetry from skewed rear view.

where, $y^+ = y + \varepsilon, y^- = y - \varepsilon, \theta'_{x,y} = \pi - \theta_{x,y}$ and ε takes values from 1 to min(*y*, *width (I)* - *y*). The column with the maximum number of votes is assumed to be the axis of symmetry. The algorithm for obtaining the axis of symmetry is given below.

After estimating the axis of symmetry, we then select the farthest pair of symmetrical edges as horizontal extremes of the bounding box and crop the ROI. Anything protruding out of this bounding box is assumed to be the part of the side shadow. The output of this step is the vehicle with its side shadow removed and rear shadow present below the rear bumper. In order to evaluate the robustness of symmetry detection algorithm, we tested it on images where the rear view of the vehicle was skewed a little to the side. In Fig. 4 the rear view was skewed towards the right side and hence the axis of symmetry shifted more towards the left side and it is correctly detected.

3) License Plate Localization: We identify the region of the license plate by performing morphological operations on the integral image of the ROI. For a given location (x, y) in the image, the integral image is computed as the sum of the values above and to the left of (x, y). It is easy to detect square



(a) (b) (c)

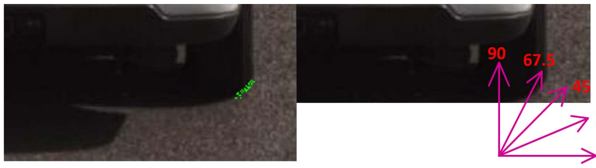
Fig. 5. (a) Vehicle with no shadow, (b) Integral image for bottom half of the vehicle, (c) localized license plate.

and rectangular objects in an integral image, making it easy to detect the license plate which is rectangular in shape. Based on the ROI image, we can safely say that the license plate is always located in the bottom half of the ROI, so for the following steps, we will be using only the bottom half of the ROI to localize the license plate.

After obtaining the integral image, we use Haar wavelets [25] to detect the horizontal and vertical edges in the integral image. We suppressed all the edges that have a magnitude lesser than 0.5. This threshold was obtained after experimentation. After obtaining the binary image, we dilate the image using a straight horizontal line whose length is 3% of the total width of the integral image. We further dilate it with a square mask of length 5 pixels followed by erosion with a square mask of 7 pixels. This gives us regions of blobs and based on the properties of the license plate we filter out the blobs. The blob with aspect ratio within the limits of 1.3 to 1.6, area greater than 2000 pixels and orientation within the limits of +5° to -5° with respect to the x- axis is identified as the region corresponding to the license plate. After identifying the blob, we fill in the empty area of the blob to make it a perfect rectangle. An example of the license plate detection is shown in Fig. 5. In Fig. 5 the green lines indicate the bottom half of the vehicle, Fig. 5(b) is the integral image for the bottom half of the vehicle and Fig. 5(c) is the localized license plate.

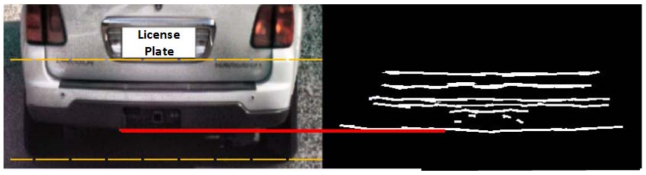
4) Removing the Rear Shadow: The symmetry detection algorithm described above does not remove the non-symmetrical features present between vertical extremes. In order to remove the rear shadow we observed that the outer curvature of the tire on the side opposite to the side shadow can be used to isolate the vehicle and the rear shadow. This is achieved by analyzing the responses of Gabor filters by locally scanning the area around the tire. We can safely assume that the tires of all vehicles are present in the bottom half of the ROI. Thus, we need to consider only the bottom half of the ROI. We divide the bottom half of the ROI into 4 quadrants. This results in the left tire being present in the third quadrant and the right tire in the fourth quadrant.

Based on the location of side shadow, we scan the quadrant on the opposite side of the side shadow. For example, if the side shadow was on the left side, we scan the fourth quadrant and *vice-versa*. We chose the orientation of Gabor filter to be N x 22.5° where N = 5 if we scan the third quadrant and N = 3 if we scan the fourth quadrant as shown in Fig. 6(b).



(a) (b)

Fig. 6. (a) Detected curvature of the right tire, (b) orientation of the Gabor filter bank.



(a) (b)

Fig. 7. Detected bottom edge of the rear bumper.

We chose the value of N by analyzing the curvature of the tire for all the vehicles in our dataset. Finally, after detecting the curvature, we crop the rear shadow that is present below the curvature from the ROI. Fig. 6 shows an example of the detected curvature of the right tire.

In Fig. 6 the side shadow was cropped from the left side of the vehicle, hence we detect the curvature of the right tire in the fourth quadrant as shown in Fig. 6(a). A drawback of this approach is when both the tires of the vehicle are surrounded by the rear shadow, which occludes the curvature of both the tires. This situation occurs only when the source of light is *exactly* in front of the vehicle and it occurs only for a very short duration of time and hence it is neglected.

5) Identifying the Bottom Edge of the Rear Bumper: Looking at the rear view of a vehicle, we observe that the bottom edge of the rear bumper is always below the bottom edge of the license plate and above the tires. By identifying the license plate and curvature of the tire from the previous discussion, we conclude that the bottom edge of the rear bumper lies in the area between the bottom edge of the license plate and the curvature of the tire. This observation helps us to narrow down the search area for identifying the bottom edge of the rear bumper. We perform histogram equalization to enhance the contrast of the image and detect all the horizontal edges within this search area using a Gabor filter at an orientation of 180° as shown in Fig. 7. In Fig. 7(a), the yellow dashed lines indicates the search area within which we need to detect the dominant horizontal edges. Next, we use a straight-line mask whose length is 3% of the total width of the ROI to dilate this binary image, which results in closing all the small gaps lying on the same row. Finally, the first longest line from the bottom whose length is more than 75% of the total width of the ROI is selected as the bottom edge of the rear bumper. In Fig. 7, the red line is the identified bottom edge of the rear bumper for the vehicle.

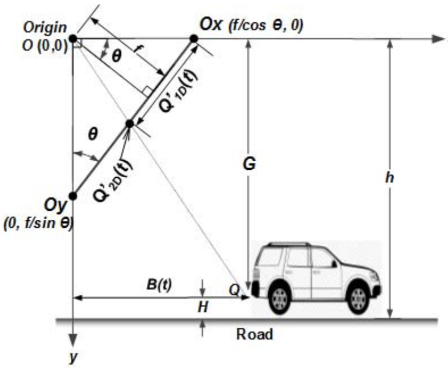


Fig. 8. Parallel projection scene geometry.

B. Multi-frame Analysis for Computing Physical Features

In our approach, the physical features is the feature set that consists of the height of the vehicle, VRGC and distance between the bottom edges of the license plate and rear bumper. In the following, we explain the procedure for estimating the height of these features using a multi-frame approach. Since vehicles are travelling down the traffic lane, for a small duration of time we can safely assume that the velocity of the vehicle is constant along the lane it is travelling and negligible in the direction perpendicular to the lane. The geometry of this scene is characterized as a parallel projection where the projection plane is orthogonal to the camera image plane and the road plane. This results in the road and camera image plane to appear as lines. Fig. 8 shows the parallel projection scene geometry.

The origin of the 2D camera coordinate system is located at O, with the x-axis parallel to the road and the y-axis perpendicular to the road. The camera is installed at a height h, with focal length f and depression angle as θ with respect to the x-axis. The image plane of the camera intersects with x-axis and y-axis at $Ox \equiv \left(\frac{f}{cos\theta}, 0\right)$ and $Oy \equiv \left(0, \frac{f}{sin\theta}\right)$, respectively. Let Q be a point on a vehicle as seen from the camera. The coordinates of the point Q at time t is $(B(t), G)$. Q is projected to $Q'_{\mathrm{2D(t)}}$ on the line $OxOy$. Solving for $Q'_{\mathrm{2D(t)}}$ results in,

$$Q'_{\mathrm{2D(t)}} = \left(\frac{fB(t)}{B(t)cos\theta + Gsin\theta}, \frac{Gf}{B(t)cos\theta + Gsin\theta}\right) \tag{4}$$

The projection of $Q'_{2D(t)}$ at infinity can be shown as,

$$\lim_{B(t)\to\infty} Q'_{\mathrm{2D(t)}} \equiv \left(\frac{f}{cos\theta}, 0\right) = Ox \tag{5}$$

With the origin as Ox, the 1-D coordinate system of the above projection is the distance between Ox and $Q'_{\mathrm{2D(t)}}$, i.e.

$$Q'_{\mathrm{1D(t)}} = \frac{Gf}{cos\theta(B(t)cos\theta + Gsin\theta)} \tag{6}$$

Differentiating the inverse of the projection with respect to time t results in,

$$\frac{d}{dt}\frac{1}{Q'_{1D(t)}} = \frac{cos^2\theta}{Gf}\frac{d}{dt}B(t) \tag{7}$$

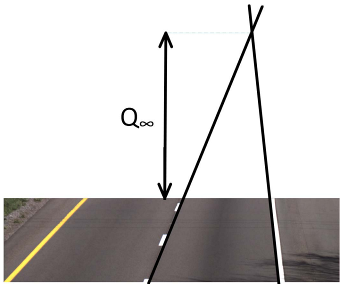


Fig. 9. Finding Q_∞ using lane marking and vanishing point.

Assuming the vehicle is travelling at a constant velocity V, we can write the above equation as a constant C.

$$\frac{d}{dt}\frac{1}{Q'_{1D(t)}} = \frac{Vcos^2\theta}{Gf} = C \tag{8}$$

It is observed from equation (8), the inverse of $Q'_{\mathrm{1D(t)}}$ varies with time in a linear fashion. In order to obtain the 1-D coordinate system from the 2-D coordinate system of the image, we must: First align the 1-D image coordinate along the y-axis of the 2-D coordinate system. Second, the origin of the 1-D coordinate system Ox is at infinity for the projection of the line as shown in Fig. 9.

$Q'_{\mathrm{1D(t)}} = Q'_{\mathrm{2D(t)(y)}}$ - Q_∞, where Q_∞ is the y-coordinate of the image projection of the line at the infinity. Fig. 9 shows an illustration of how Q_∞ can be found using the vanishing point of the parallel lane markings.

Based on equation (8), we define a constant $C_{vehicle}$ which is assumed not to change for a given video as $C_{vehicle} = \frac{Vcos^2\theta}{f}$. Equation (8) can be written in discrete form as,

$$\frac{1}{j-i}\left(\frac{1}{T_j} - \frac{1}{T_i}\right) = \frac{C_{vehicle}}{G} \tag{9}$$

In equation (9) $\mathrm{T_i}$ and $\mathrm{T_j}$ are the tracked locations for a given feature in frames i and j. We can estimate the height G of a feature by solving the above equation. It should be noted that, although we know the camera focal length f, depression angle θ and height h are constants, the velocity V of each vehicle is different. We solve this problem by tracking the side shadow cast by the vehicle. We assume that the side shadow and vehicle travel with the same velocity V.

Let the tracked locations of the side shadow be Z_i and Z_j in frames i and j, respectively. The shadow is assumed to be present on the ground plane and hence the distance between the camera and any point on the side shadow should be equal to the height at which the camera is placed, i.e. $G = h$. So equation (9) becomes,

$$\frac{h}{j-i}\left(\frac{1}{Z_j} - \frac{1}{Z_i}\right) = \frac{Vcos^2\theta}{f} \tag{10}$$

After solving equation (10) for the velocity V, the height of the tracked features on the vehicle can be computed. Let the location of the tracked vehicle features be X_i and X_j in frames i and j, respectively. Solving for the unknown G, the height of the feature H can be estimated as,

$$H = h - G = (h - C_{vehicle})\left(\frac{1}{X_j} - \frac{1}{X_i}\right)^{-1} \quad (11)$$

- ***Physical Feature Extraction**:* We estimate the height of the features with respect to the ground plane by detecting the locations of these features in successive frames ($X_i \; and \; X_j$) and substituting these locations into equation (11). After obtaining the heights, we compute the distance between the bottom edge of the license plate and rear bumper and we call it as the license plate-to-rear bumper distance. Similarly, we estimate the Visual Rear Ground Clearance (VRGC) which is the height of the bottom edge of the rear bumper from the ground plane. The height of the vehicle is found by estimating the height of the top most dominant horizontal edge of the ROI.

C. Visual Feature Extraction Using Deep Learning

We extract the visual features using Convolutional Neural Networks (CNN). In the recent years, CNNs have showed promising results for the task of vehicle classification [26]– [28]. We train the recent CNN architectures proposed by [29]– [32] to classify the vehicles from their rear view images into the four classes namely: Sedan, Minivan, SUV and Pickup truck.

The network takes as input the ROI after removing the rear and side shadows and we resize the ROI to a size of 224x224. We did random hyper parameter search to obtain the best hyper parameters for each network. The networks were trained using a mini batch size of 128 and learning rate annealing is done if the validation accuracy does not increase after 3 epochs. To prevent the networks from over-fitting we performed early stopping. Batch normalization is performed to get faster training convergence. Rectified Linear units (ReLu) are used as non-linearities and the network was trained to minimize the cross entropy loss function. We used a Softmax after the final fully connected layer. After training the network, the Visual/Appearance features are extracted from the fully connected layer before the softmax layer.

- ***Data Augmentation and Transfer Learning:*** In order to augment the size of the dataset, we randomly horizontally flipped the images and performed color jittering [29] during the training. Furthermore, since the size of our dataset is very limited, we performed transfer learning where all the networks were initialized by pre-training the networks on the MIO-TCD dataset [33]. The MIO-TCD dataset consists of approximately 650,000 images of vehicles distributed into 11 different classes taken from real-world traffic surveillance cameras across the USA and Canada.

The dataset, hyper-parameters and classification results for the individual networks are explained in detail in Section IV.

IV. Experimental Results

The proposed approach was evaluated on a dataset consisting of 1,831 rear view videos of vehicles from three different datasets. The vehicle classes in the datasets are sedan, minivan, SUV and pickup truck. We evaluated the extracted physical and visual features both individually and after fusing them together. The dataset was collected during the day time with good illumination. To the best of our knowledge the only other datasets that have night time images are the BIT-Vehicle [12] and MIO-TCD [33] datasets. The BIT-Vehicle dataset has 9,850 frontal view images out of which only 10% are night time images. The MIO-TCD dataset consists of 650,000 multi-view images of vehicles distributed into 11 different classes taken form real-world traffic surveillance cameras. Both of these datasets have night time images with very high illumination and contrast. Since neither of these datasets have the geometric parameters of the cameras, they cannot be directly used in our approach. Moreover, since the MIO-TCD dataset is very diverse with images of vehicles captured from different angles, we use this dataset to pre-train the CNNs and then fine tune the CNNs on our dataset.

A. Dataset, Ground-truth and Parameters

Dataset 1 consists of 876 vehicles (309 sedans, 259 SUVs, 119 minivans and 189 pickup trucks), *Dataset 2* consists of 896 vehicles (321 sedans, 266 SUVs, 130 minivans and 179 pickup trucks) and *Dataset 3* consists of 59 vehicles (35 sedans, 11 SUVs, 4 minivans and 9 pickup trucks). For Dataset 1 and Dataset 2, the camera was installed at a height of 22 feet, with a depression angle of 8° and focal length of 80mm. The images in Dataset 1 and Dataset 2 were captured during different times (morning to late evening) of the day on different days. To prove the robustness of our approach we installed a different camera at a height of 10 feet, with a depression angle of 14° and focal length of 50mm in front of the Bourns College of Engineering main building at UC-Riverside and the data was collected during 7:30 A.M - 9:00 A.M, 12:00 P.M - 1:00 P.M, 2:30 P.M - 4:00 P.M and 6:00 P.M - 6:30 P.M. It consists of 35 sedans, 11 SUVs, 4 minivans and 9 pickup trucks. All the videos in Dataset 1, Dataset 2 and Dataset 3 were recorded at 30 frames per second. It should be noted that the height and depression angle for the camera should be chosen such that, only the rear view of the vehicle is visible and not the bonnet/hood of the vehicle.

The ground-truth class labeling for each vehicle was done manually by two vehicle recognition researchers by examining the videos of all the vehicles. The parameters of the proposed system are shown in Table II. They were kept constant for all the datasets.

B. Performance Measures for Processing the ROI

We measure the performance of algorithms for predicting the axis of symmetry, license plate region, the curvature of the tire and the top most dominant horizontal edge. In order to measure the performance we manually annotated 100 randomly chosen vehicles (25 from each class).

1) Detecting the Moving Vehicle and Shadow: We detected the moving vehicle using the adaptive Gaussian background subtraction. The output of this is the binary mask of the vehicle along with its shadow. We evaluated the performance of our approach using the Intersection Over Union (IOU) between the ground-truth and predicted mask. We achieved an average IOU of **0.84 ± 0.13**.

TABLE II

PARAMETERS OF THE PHYSICAL FEATURES

Detection of moving objects	
Threshold for a pixel to be part of the moving ROI	3 standard deviations from the mean of the R, G and B plane
Removal of the side shadow	
Orientation of the Gabor filter bank θ	0°, 22.5°, 45°, ..., 157.5°
Aspect ratio of the Gabor filter bank γ	0.5
Area of the receptive field of the Gabor filter bank σ	3
Removal of the rear shadow	
Angle of curvature of the left tire	112.5°
Angle of curvature of the right tire	67.5°
Localization of the License plate	
Aspect ratio of the license plate	1.3 to 1.6
Area of the license plate	greater than 2000 pixels
Orientation of the license plate	+5° to -5° along the x- axis
Localization of the bottom edge of the rear bumper	
Length of the straight line mask	3% of the total width of the ROI
Length of the rear bumper	Greater than 75% of the width of the ROI
Multi-frame analysis for computing physical features	
Height of the camera	22 feet for Dataset 1 and Dataset 2, and 10 feet for dataset 3
Depression angle of the camera	8° for Dataset 1 and Dataset 2, and 14° for dataset 3
Focal length of the camera	80mm for Dataset 1 and Dataset 2, and 50mm for Dataset 3

2) Predicting the Axis of Symmetry: The output of the symmetry detection algorithm is the center column of the ROI. We measure the performance by computing the error as the difference in pixels between the ground-truth and the predicted column. We achieved an average error of **1.93 ± 2.14** pixels.

3) Localizing the License Plate: We measure the performance of the license plate detection by measuring the IOU between the ground-truth and the predicted region. We achieved an average IOU of **0.86 ± 0.15**. Moreover, our approach does not require the entire license plate region to be detected. Instead it requires the bottom edge of the license plate (row), in order to compute the distance between the bottom edge of the license plate and rear bumper. Based on this we measure the performance by computing the error in pixels between the ground-truth and predicted bottom edge. We achieved an average error of **3.41 ± 0.82** pixels.

4) Predicting the Curvature of the Tire: The output of this algorithm is the bottom most row of the tire that is touching the road plane. The performance of our approach is measured by computing the error as the difference between the ground-truth and the predicted row. We achieved an error of **2.82 ± 1.93** pixels.

5) Predicting the TOP MOST Dominant Horizontal Edge: In our approach we assume the top most dominant horizontal edge (row) of the ROI is guaranteed to represent the row corresponding to the height of the vehicle. We evaluate the performance by computing the error as the difference between the ground-truth and the predicted row. We achieved an error of **2.14 ± 1.02** pixels.

TABLE III

CONFUSION MATRIX FOR DATASET 1 AND DATASET 2

Class	SUV	Pickup truck	Sedan	Minivan
SUV	**476**	19	22	8
Pickup truck	2	**360**	5	1
Sedan	18	7	**597**	8
Minivan	22	1	13	**213**

TABLE IV

CONFUSION MATRIX FOR DATASET 3

Class	SUV	Pickup truck	Sedan	Minivan
SUV	**10**	0	1	0
Pickup truck	0	**9**	0	0
Sedan	2	0	**32**	1
Minivan	0	0	0	**4**

C. Experiments and Analysis for the Physical Features

The physical features (VRGC, height of the vehicle and license plate to rear bumper distance) are classified into the four classes using the C4.5 binary tree algorithm. This algorithm was found to be the best classifier for our system for the following reasons:

- There are only four output classes.
- The output classes are linearly separable.
- The dimensions of our feature space is three, which makes it easy to interpret the decision tree.

We trained and tested our binary tree classifier using the K- fold cross validation with K selected as 10.

1) Confusion Matrices: We used Dataset 1 and Dataset 2 for training and evaluating our approach. In order to show the robustness of our approach, we used Dataset 3 as an exclusive dataset for evaluating our approach which was collected at a different location and time compared to Dataset 1 and Dataset 2. Table III shows the confusion matrix for the 10 fold cross validation using Dataset 1 and Dataset 2.

From Table III, it can be seen that our rear view classification system using the physical features achieved an accuracy of 92.89% and false alarm of 3.68% for Dataset 1 and Dataset 2 combined. Out of 525 SUVs, 22 were misclassified as sedan because the height and VRGC features were underestimated and out of 630 sedans 18 were misclassified as SUV because the height and VRGC were over-estimated. The key reason for this was that the bottom of the rear bumper and the top most horizontal edge were misidentified because of poor illumination conditions. 19 SUV's were misclassified as pickup trucks because certain models of SUVs have their license plate located close to the bumper.

To evaluate our approach on Dataset 3, the correctly classified vehicles of Dataset 1 and Dataset 2 were combined together as the training set to build the model for classification and we used Dataset 3 as the unseen test data. The confusion matrix of the results is shown in Table IV. The system achieved a classification accuracy of 93.22% and false alarm of 3.39%. Only 1 SUV was misclassified as a sedan and 2 sedans were misclassified as SUVs and 1 sedan was misclassified as a minivan.

2) Consistency of the Results: To show that the estimated physical features are consistent, we tested our system

TABLE V

COMPARISON OF THE GROUND-TRUTH WITH THE COMPUTED PHYSICAL FEATURES

Vehicle model (class)	GT* VRGC (in.)	GT* height (in.)	GT* LP* to RB* distance (in.)	Computed VRGC Avg. (in.), %error	Computed height Avg. (in.), %error	Computed LP to RB distance Avg. (in.), %error
Honda CR-V (SUV)	17.7	64.7	21.6	18.78, 6.11%	71.21, 10.05%	24.53, 13.57%
Toyota Tundra (Pickup truck)	18.1	75.8	5.7	18.41, 1.71%	77.35, 1.98%	6.14, 7.72%
Chevrolet Impala (Sedan)	11.3	58.7	17.3	12.07, 6.81%	61.88, 5.28%	20.12, 16.30%
Honda Odyssey (Minivan)	13.2	68.4	26.4	13.58, 2.88%	70.75, 3.44%	24.23, 8.22%

***GT, *LP and *RB refer to Ground-Truth, License Plate and Rear Bumper, respectively.**

TABLE VI

STATISTICS OF MULTI-FRAME ANALYSIS USING DIFFERENT NUMBER OF FRAMES

Vehicle	Computed physical feature	ground-truth (in.)	Number of frames = 2	Number of frames = 3	Number of frames = 4	Number of frames = 5
Honda CR-V	Avg. VRGC (in.) ± S.D. (in.)	**17.7**	19.37 ± 0.91	19.64 ± 0.72	**18.78 ± 0.56**	19.04 ± 0.45
	Avg. Height (in.) ± S.D. (in.)	**64.7**	73.42 ± 1.61	71.97 ± 0.65	**71.21 ± 0.94**	72.40 ± 0.82
	Avg. LP* to RB* distance (in.) ± S.D. (in.)	**21.6**	26.76 ± 1.54	25.92 ± 0.87	**24.53 ± 0.75**	25.52 ± 0.82
Toyota Tundra	Avg. VRGC (in.) ± S.D. (in.)	**18.1**	17.74 ± 1.57	18.29 ± 0.81	18.41 ± 0.65	**17.83 ± 0.77**
	Avg. Height (in.) ± S.D. (in.)	**75.8**	77.80 ± 1.68	79.62 ± 1.22	**77.35 ± 0.58**	78.41 ± 0.61
	Avg. LP* to RB* distance (in.) ± S.D. (in.)	**5.7**	6.96 ± 0.62	6.23 ± 0.36	6.14 ± 0.28	**6.11 ± 0.41**
Chevrolet Impala	Avg. VRGC (in.) ± S.D. (in.)	**11.3**	12.47 ± 0.51	12.31 ± 0.42	**12.07 ± 0.77**	12.39 ± 0.87
	Avg. Height (in.) ± S.D. (in.)	**58.7**	61.30 ± 1.32	62.43 ± 0.54	61.88 ± 0.82	**60.44 ± 1.16**
	Avg. LP* to RB* distance (in.) ± S.D. (in.)	**17.3**	20.84 ± 0.76	20.51 ± 0.42	**20.12 ± 0.48**	20.18 ± 0.55
Honda Odyssey	Avg. VRGC (in.) ± S.D. (in.)	**13.2**	14.21 ± 0.60	14.42 ± 0.88	**13.58 ± 0.83**	14.17 ± 0.59
	Avg. Height (in.) ± S.D. (in.)	**68.4**	71.90 ± 1.01	73.23 ± 0.86	**70.75 ± 0.54**	71.46 ± 0.91
	Avg. LP* to RB* distance (in.) ± S.D. (in.)	**26.4**	22.57 ± 1.23	23.11 ± 0.56	**24.23 ± 0.69**	23.75 ± 0.94

***LP and *RB refer to License Plate and Rear Bumper, respectively.**

on 20 vehicles from four different vehicle models namely: Chevrolet Impala (Sedan), Honda CR-V (SUV), Toyota Tundra (Pickup truck) and Honda Odyssey (Minivan). Table V shows the comparison of the average VRGC, average height and average license plate to rear bumper distance (the average is taken over five different colored vehicles of each vehicle model) for the four vehicle models with their respective ground-truth. The ground-truth VRGC and ground-truth license plate to rear bumper distance for the 20 vehicles were obtained manually by measuring the distance by hand using an inch tape. The ground-truth height for the vehicle was obtained from the website edmunds.com.

3) Statistics of the Multi-frame Analysis: We tested the proposed multi-frame analysis algorithm for computing the physical features using different number of frames for the above mentioned 20 vehicles. Table VI shows the average and standard deviation of the physical features of the 20 vehicles using 2, 3, 4 and 5 frames for tracking. The overall results obtained by using 4 frames had lower error percentage and standard deviation with respect to the ground-truth. In Table VI, it can be observed that most of the predicted measurements were overestimated compared with the ground-truth. We observed that as the size of the ROI decreased from frame to frame, some of the tracked points in successive frames were predicted to be a few rows below the actual row. This means that the ground plane is estimated to be lower than the actual value leading to an over estimation in measurements. This is more evident in Table VI when the number of frames = 5, the measurements are even more over estimated compared to when number of frames = 4.

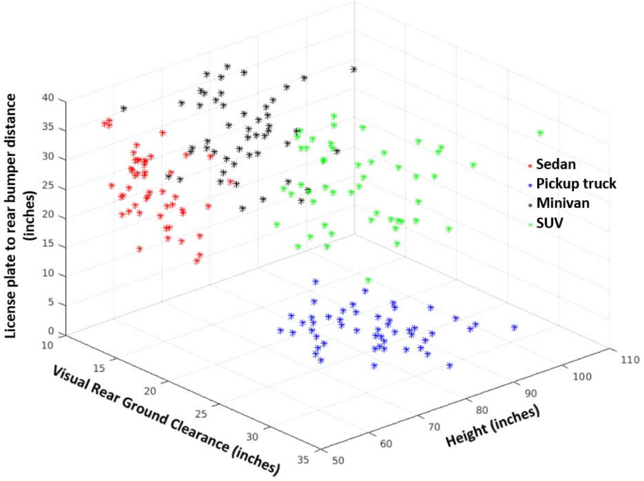


Fig. 10. Cloud point representation of the physical features.

4) Visualization of the Physical Features and their Statistics: Fig. 10 shows the cloud point representation of the physical features for 50 randomly chosen correctly classified vehicles of each class. Sedans are marked in red, minivans are marked in black, SUVs are marked in green and pickup trucks are marked in blue.

It can be observed from Fig. 10 that there is not much overlap in the feature space among the vehicle classes and they are linearly separable. From this we conclude that sedans and minivans have low VRGC, sedans have the lowest height compared to the other classes and pickup trucks have the least license plate to rear bumper distance.

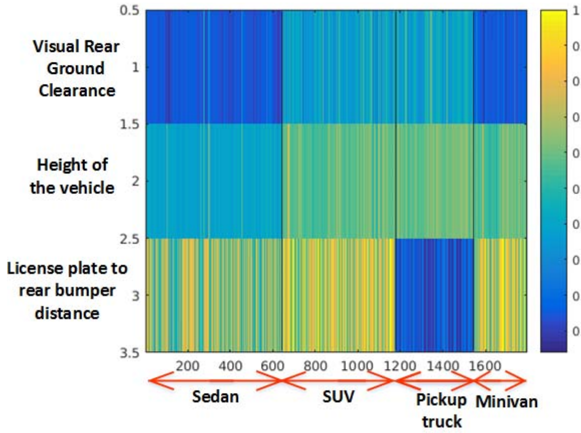


Fig. 11. Normalized pattern feature matrix. The scale is shown as color on the right side of the figure.

TABLE VII

STATISTICS OF THE COMPUTED PHYSICAL FEATURES

Class	VRGC (in.) **Avg. ± S.D.**	Height (in.) **Avg. ± S.D.**	LP* to RB* distance **Avg. ± S.D.**
SUV	22.08 ± 4.76	75.94 ± 7.58	31.78 ± 11.15
Pickup truck	23.43 ± 5.22	78.38 ± 6.76	9.31 ± 4.92
Sedan	13.87 ± 3.92	59.69 ± 6.48	24.55 ± 11.38
Minivan	14.88 ± 3.60	75.71 ± 7.10	33.15 ± 12.47

***LP and *RB refer to License Plate and Rear Bumper, respectively.**

Fig. 11 shows the normalized pattern feature matrix of the computed physical features for all the four classes. It can be observed from Fig. 11 that sedan and minivan have lower VRGC compared to SUV and pickup truck because the rear bumpers of sedan and minivan are much closer to the ground surface. Sedan has the minimum height compared to SUV, minivan and pickup truck. Pickup truck has the least license plate to rear bumper distance.

Table VII shows the computed average and standard deviation of the three features for all the 1,831 vehicles. The standard deviation of the license plate to rear bumper distance for pickup trucks was less compared to the standard deviation for SUV, sedan and minivan because the license plate for SUV, sedan and minivan can be located anywhere on the rear surface, whereas for pickup trucks the license plate is mostly located on the rear bumper.

D. Visual/Appearance Features

In this section we classify the vehicles based on their visual appearance. We used the state-of-the-art Convolutional Neural Network architectures namely: Alexnet [29], ResNet [30], VGG [31] and DenseNet [32]. The ROI of the vehicles after removing the side and rear shadow were used as input images for training the respective CNNs. We used Dataset 1 and Dataset 2 together consisting of 1,772 vehicles for training and evaluating the networks. The dataset was partitioned such that, from the combined data we used 60% of each class for training, 10% for validation and 30% for testing. We used Dataset 3 as an exclusive dataset for evaluating the networks, as this dataset was collected in a different environment with a different geometrical setup of the camera as compared with Dataset 1 and Dataset 2 which were obtained in two different states with cameras mounted on highways.

TABLE VIII

BEST HYPER-PARAMETERS FOR DIFFERENT CNNS

Network	**Learning rate**	**Momentum**	**Weight decay**
AlexNet [29]	3 x 10^{-3}	0.7	1 x 10^{-4}
ResNet18 [30]	6 x 10^{-3}	0.9	1 x 10^{-4}
ResNet34 [30]	6 x 10^{-3}	0.9	1 x 10^{-4}
ResNet50 [30]	6.5 x 10^{-3}	0.9	4 x 10^{-4}
VGG16 [31]	3 x 10^{-3}	0.8	2 x 10^{-4}
VGG19 [31]	4 x 10^{-3}	0.9	1 x 10^{-4}
DenseNet121 [32]	5 x 10^{-3}	0.9	6 x 10^{-4}

TABLE IX

AVERAGE CLASSIFICATION ACCURACY OF THE RESPECTIVE CNNS ON DATASET 1 AND DATASET 2 COMBINED

Network	**Avg. Classification accuracy % ± S.D.**
AlexNet [29]	86.32 ± 0.78
ResNet18 [30]	89.13 ± 0.52
ResNet34 [30]	89.87 ± 0.59
ResNet50 [30]	88.56 ± 0.83
VGG16 [31]	**91.34 ± 0.51**
VGG19 [31]	90.78 ± 0.67
DenseNet121 [32]	90.24 ± 0.41

We trained all the networks using a mini-batch size of 128. Learning rate annealing is done by a factor of 2, if the validation accuracy does not increase after 3 epochs. We used the stochastic gradient descent algorithm to minimize the weighted cross entropy loss function [29]. To prevent the network from over-fitting we performed early stopping. We stop the training, if the validation accuracy did not increase after 3 consecutive epochs. We used a Softmax after the fully connected layer. After training the respective networks, we extracted the features from the fully connected layer before the softmax. We call the extracted feature set from the CNN as the Visual features.

1) Hyper-parameters for Different CNNs: Table VIII shows the best hyper-parameters used for training the different CNNs. The best hyper-parameters were found by doing a random hyper-parameter search that gives us the best accuracy on the validation data after the first 3 epochs.

2) Performance of Different CNNs: We performed 3-fold cross validation using each CNN and the average accuracy is taken as the final classification accuracy. Table IX shows the average classification accuracy and standard deviation for each network. In Table IX, VGG16 achieved the highest average classification accuracy. Table X shows the confusion matrix obtained using VGG16.

From Table X, we can observe that 33 SUVs were misclassified as minivans and 24 minivans were misclassified as SUVs. This suggests that SUVs and minivans do have a similar visual appearance from the rear view. Comparing Table III and Table X, we observed that 3 SUVs were commonly misclassified as pickup trucks, and 4 minivans were commonly misclassified as SUVs. We further evaluated the

TABLE X

CONFUSION MATRIX FOR VGG16 EVALUATED ON DATASET 1 AND DATASET 2 COMBINED

Class	SUV	Pickup truck	Sedan	Minivan
SUV	**419**	14	6	33
Pickup truck	10	**307**	4	10
Sedan	7	6	**549**	5
Minivan	24	10	9	**181**

TABLE XI

EVALUATION OF THE NETWORKS ON DATASET 3

Network	Classification accuracy %
AlexNet [29]	81.54
ResNet18 [30]	83.05
ResNet34 [30]	86.44
ResNet50 [30]	84.75
VGG16 [31]	**91.52**
VGG19 [31]	89.83
DenseNet121 [32]	88.16

TABLE XII

CONFUSION MATRIX FOR VGG16 EVALUATED ON DATASET 3

Class	SUV	Pickup truck	Sedan	Minivan
SUV	**8**	0	1	2
Pickup truck	0	**9**	0	0
Sedan	1	0	**34**	0
Minivan	1	0	0	**3**

trained networks in Table IX on Dataset 3 which has never been seen by the respective networks. Table XI shows the classification accuracy of the networks on Dataset 3.

In Table XI, VGG16 achieved the highest classification accuracy on Dataset 3, which was collected in a different environment and at different intervals of the day compared to Dataset 1 and Dataset 2. This shows that VGG16 was the most generalizable out of all the networks. Table XII shows the confusion matrix for VGG16 evaluated on Dataset 3.

On comparing the evaluation of the physical and visual features on Dataset 3 from Table IV and Table XII respectively, it should be noted that physical features was able to classify SUVs and minivans better than the visual features, whereas the visual features were able to classify pickup trucks and sedans with a relatively higher classification accuracy. This shows that the CNN was able to learn some visual features that can be considered as complementary features to the physical features. We also observed that there was 1 SUV that was commonly misclassified as a sedan.

TABLE XIII

CROSS-PERSPECTIVE EVALUATION OF THE NETWORKS

Network	Classification accuracy %
AlexNet [29]	79.54
ResNet18 [30]	78.85
ResNet34 [30]	80.31
ResNet50 [30]	73.75
VGG16 [31]	76.37
VGG19 [31]	82.53
DenseNet121 [32]	75.91

TABLE XIV

AVERAGE CLASSIFICATION ACCURACY OF THE FUSED FEATURES ON DATASET 1 AND DATASET 2 COMBINED

Input feature	Feature dimension	Avg. Classification accuracy % ± S.D.
AlexNet + P.F.	4096 + 3	92.37 ± 0.55
ResNet18 + P.F.	1024 + 3	93.32 ± 0.47
ResNet34 + P.F.	1024 + 3	93.94 ± 0.63
ResNet50 + P.F.	2048 + 3	93.12 ± 0.71
VGG16 + P.F.	**4096 + 3**	**94.81 ± 0.44**
VGG19 + P.F.	4096 + 3	94.57 ± 0.58
DenseNet121 + P.F.	1024 + 3	94.16 ± 0.31

***P.F. refers to Physical Features**

TABLE XV

CONFUSION MATRIX OBTAINED BY FUSING VGG16 AND THE PHYSICAL FEATURES EVALUATED ON DATASET 3

Class	SUV	Pickup truck	Sedan	Minivan
SUV	**10**	0	0	1
Pickup truck	0	**9**	0	0
Sedan	0	0	**35**	0
Minivan	1	0	0	**3**

3) Rear View Perspective Versus Every Other Perspective: To further corroborate our claim that SUVs and minivans look visually similar from the rear view, we trained the networks mentioned above on images of SUV, minivan, pickup truck and sedan collected from all possible perspectives except the rear view. The images were collected from the Imagenet dataset [34] and MIO-TCD dataset resulting in 7,358 SUV, 9,564 sedan, 7,293 pickup truck and 7,098 minivan images. After training the networks, we evaluated them on our rear view dataset consisting of 1,831 vehicles. We call this as the *cross-perspective evaluation*. All the networks were initialized with the uniform Xavier initialization. Table XIII shows the classification accuracy of the individual networks for the *cross-perspective evaluation*. From Table XIII it can be observed that even with sufficiently large amount of images, the respective networks fell short in the *cross-perspective evaluation*. This suggests that some vehicles look visually similar from the rear view **(especially SUV and minivan)** and hence we would need complementary features such as the Visual Rear Ground Clearance, **VRGC**, to resolve this problem.

E. Fusing the Physical and Visual Features Using SVM

We evaluated both the physical and visual features by concatenating the individual feature vectors as input to train a SVM model. Table XIV shows the classification accuracy using 3-fold cross validation approach with Dataset 1 and Dataset 2 combined. It can be seen that the SVM model trained using VGG16 + physical features achieved the highest average classification accuracy of **94.81%**. Furthermore, we evaluated this SVM model on Dataset 3 and achieved an accuracy of **96.61%**. Table XV shows the confusion matrix obtained using visual features of VGG16 and the physical features.

F. Comparison With Other Rear View Classification Methods

Kafai and Bhanu [18] achieved a correct classification rate of 96.61% using a Hybrid Dynamic Bayesian Network (HDBN) with low-level features. It should be noted that

TABLE XVI
COMPARISON OF OUR APPROACH WITH OTHER REAR VIEW CLASSIFICATION METHODS

Method	**Results**	**Comments**
Thakoor and Bhanu [19]	86.06%	Used structural signatures for classification and could not distinguish between SUV and minivan
This paper	**94.81%**	**Automatically computed and fused the physical and visual features and classified SUV and minivan separately**

these authors evaluated their approach on a different dataset consisting of 177 rear view videos of vehicles travelling in front of the Engineering building at UC- Riverside. The drawbacks of their rear view classification system are:

- The features used for classification were hand-picked and not computed automatically.
- They could not classify between SUV and minivan. SUV and minivan was treated as a single category.
- They had a very small dataset of only 177 vehicles.

Furthermore, the authors did not evaluate their approach in different environments as has been done in this paper.

Thakoor and Bhanu [19] used the structural signatures of the moving vehicles and SVM classifier to achieve a correct classification rate of 86.06%. The authors used Dataset 1 and Dataset 2 that is used in this paper to perform the rear view classification but could not distinguish between SUV and minivan. Table. XVI shows a summary of the comparison between our approach and Thakoor and Bhanu.

All the approaches in Table. XVI were evaluated on Dataset 1 and Dataset 2 combined. We achieved average accuracy of **94.81%** on Dataset 1 and Dataset 2 (1,772 rear view videos) which were collected on two different highways and achieved an accuracy of **96.61%** on Dataset 3 (59 rear view videos) which was collected in front of the Bourns College of Engineering building at UC- Riverside.

V. CONCLUSION

We presented a vehicle classification system that can classify vehicles from their rear view using their physical and visual features. The proposed approach can extract physical features such as the height of the vehicle and Visual Rear Ground Clearance (VRGC) in the presence of a good light source(sunny and dry environmental conditions) on the highways. We showed the importance of the VRGC and how our approach overcame the problem of classifying between SUV and minivan, which have similar rear views, as compared to Kafai and Bhanu [18] and Thakoor and Bhanu [19]. It is accomplished by estimating the VRGC of a vehicle which is the most distinguishing feature for SUV and minivan. Our system achieved a correct classification rate of 93.22% and 91.52% using the physical and visual features, respectively. We also compared both feature sets and showed that each feature set can learn some features that are complementary to each other. We further evaluated our approach by fusing both the physical and visual features together and achieved a higher classification accuracy of 94.81%. In the future, we plan to use the physical and visual features for classification of other vehicle types.

VI. ACKNOWLEDGMENT

The contents of the information do not reflect the position or the policy of the US Government.

REFERENCES

[1] *Worldwide Automobile Production*. Accessed: Mar. 2018. [Online]. Available: https://www.statista.com/statistics/262747/worldwide-automobile-production-since-2000/

[2] *Passenger Vehicles in USA*. Accessed: Mar. 2018. [Online]. Available: https://en.wikipedia.org/wiki/Passenger_vehicles_in_the_United_States

[3] *Auto Theft*. Accessed: Mar. 2018. [Online]. Available: https://www.iii.org/fact-statistic/facts-statistics-auto-theft

[4] Y. Shan, H. S. Sawhney, and R. Kumar, "Unsupervised learning of discriminative edge measures for vehicle matching between nonoverlapping cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 700–711, Apr. 2008.

[5] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 1185–1192.

[6] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002.

[7] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle type categorization: A comparison of classification schemes," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2011, pp. 74–79.

[8] A. Psyllos, C. N. Anagnostopoulos, and E. Kayafas, "Vehicle model recognition from frontal view image measurements," *Comput. Standards Interfaces*, vol. 33, pp. 142–151, Feb. 2011.

[9] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," in *Proc. 8th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2011, pp. 373–378.

[10] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, and W.-F. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 175–187, Jun. 2006.

[11] B. Zhang, Y. Zhou, and H. Pan, "Vehicle classification with confidence by classified vector quantization," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 3, pp. 8–20, Jul. 2013.

[12] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015.

[13] A.-M. Cretu and P. Payeur, "Biologically-inspired visual attention features for a vehicle classification task," *Int. J. Smart Sens. Intell. Syst.*, vol. 4, no. 3, pp. 402–423, Sep. 2011.

[14] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2511–2521, Oct. 2015.

[15] R. Theagarajan, F. Pala, and B. Bhanu, "EDeN: Ensemble of deep networks for vehicle classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 906–913.

[16] S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, "A model for fine-grained vehicle classification based on deep learning," *Neurocomputing*, vol. 257, pp. 97–103, Sep. 2017.

[17] L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, "Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks," *Mach. Vis. Appl.*, vol. 28, no. 7, pp. 793–802, 2017.

[18] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.

[19] N. S. Thakoor and B. Bhanu, "Structural signatures for passenger vehicle classification in video," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1796–1805, Dec. 2013.

[20] N. Thakoor and B. Bhanu, "Method and system for vehicle classification," U.S. Patent 9 239 955, Jan. 2016.

[21] R. Theagarajan, N. S. Thakoor, and B. Bhanu, "Robust visual rear ground clearance estimation and classification of a passenger vehicle," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, Nov. 2016, pp. 2539–2544.

[22] N. Thakoor and J. Gao, "Automatic video object shape extraction and its classification with camera in motion," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, pp. 437–440.

[23] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1079–1087, Aug. 2004.

[24] W. Li, K. Mao, H. Zhang, and T. Chai, "Selection of Gabor filters for improved texture feature extraction," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 361–364.

[25] C. Mulcahy, "Image compression using the Haar wavelet transform," *Spelman Sci. Math. J.*, vol. 1, no. 1, pp. 22–31, 1997.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[27] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Proc. 2nd IAPR Asian Conf. Pattern Recognit.*, Nov. 2013, pp. 181–185.

[28] S. Wang, Z. Li, H. Zhang, Y. Ji, and Y. Li, "Classifying vehicles with convolutional neural network and feature encoding," in *Proc. IEEE 14th Int. Conf. Ind. Inform. (INDIN)*, Jul. 2016, pp. 784–787.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[31] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[32] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[33] Z. Luo *et al.*, "MIO-TCD: A new benchmark dataset for vehicle classification and localization," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5129–5141, Oct. 2018.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

Rajkumar Theagarajan (S'14) received the B.E degree in electronics and communication engineering from Anna University, Chennai, India, in 2014, and the M.S. degree in electrical and computer engineering from the University of California at Riverside, Riverside, CA, USA, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Center for Research in Intelligent Systems, University of California at Riverside. His research interests include computer vision, image processing, pattern recognition, and machine learning.

Ninad S. Thakoor (S'04–M'10) received the B.E. degree in electronics and telecommunication engineering from the University of Mumbai, Mumbai, India, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington, Arlington, TX, USA, in 2004 and 2009, respectively. He was a Postdoctoral Researcher with the Center for Research in Intelligent Systems, University of California at Riverside, Riverside, CA, USA. He is currently with Cognex Corporation. His research interests include vehicle recognition, stereo disparity segmentation, and structure-and-motion segmentation.

Bir Bhanu (M'82–F'95–LF'17) received the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, and the M.B.A. degree from the University of California at Irvine, Irvine, CA. He was the Founding Professor of electrical engineering with the University of California at Riverside (UCR), Riverside, CA, and served as its first Chair from 1991 to 1994. He has been the Cooperative Professor of computer science and engineering (since 1991), bioengineering (since 2006), and mechanical engineering (since 2008). He served as the Interim Chair of the Department of Bioengineering from 2014 to 2016. He also served as the Director of the National Science Foundation Graduate Research and Training Program in video bioinformatics with UCR. He is currently the Bourns Presidential Chair in engineering, the Distinguished Professor of electrical and computer engineering, and the Founding Director of the Interdisciplinary Center for Research in Intelligent Systems and the Visualization and Intelligent Systems Laboratory, UCR. He has published extensively and has 18 patents. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human-computer interactions, and biological, medical, military, and intelligence applications. In 1991, he was a Senior Honeywell Fellow with Honeywell Inc. He is a Fellow of AAAS, IAPR, SPIE, and AIMBE.