UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Robust Passenger Vehicle Classification Using Physical Measurements From Rear View

Permalink

https://escholarship.org/uc/item/2tm7d1c9

Author

Theagarajan, Rajkumar

Publication Date

2016

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NoDerivatives License, availalbe at https://creativecommons.org/licenses/by-nd/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Robust Passenger Vehicle Classification Using Physical Measurements From Rear View

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Electrical Engineering

by

Rajkumar Theagarajan

June 2016

Thesis Committee:
- Dr. Bir Bhanu, Chairperson
- Dr. Matthew Barth
- Dr. Yingbo Hua

Copyright by
Rajkumar Theagarajan
2016

The Thesis of Rajkumar Theagarajan is approved:

Committee Chairperson

University of California, Riverside

Table of Contents

List of Figures

List of Tables

I. INTRODUCTION

The growth of population and economic prosperity has led to a huge increase in the number of vehicles. This brings a growing need for automated and efficient classification techniques for different vehicle categories for a multitude of applications. For a vehicle classification system to be useful in a real-world application, it must be robust to illumination changes, shadows, partial detections, occlusion, camera viewpoint changes, etc. A common approach utilizes vision-based methods and different features to detect and classify a vehicle in still images and video streams. Vision systems are relatively cheap, easy to install and configure and offer direct visual feedback and flexibility in mounting. They are therefore an appropriate sensing solution for vehicle classification. However, the issue of vehicle classification from images is not trivial. Due to the ever increasing number of vehicle models and sizes and the aesthetic similarities between them, the main problem is the identification of a set of representative and discriminative features that allow for the best possible classification of the vehicle type.

A human being may be capable of identifying the class of a vehicle with a quick glance at the digital data but accomplishing that with a computer is not as straight forward. Several problems such as occlusion, tracking a moving object, shadows, lack of color invariance, and many more must be carefully considered in order to design an effective and robust automatic vehicle classification system which can work in real-world conditions.

Feature based methods are commonly used for object classification. As an example, Scale Invariant Feature Transform (SIFT) is a commonly used feature based method [1]. Classification methods for vehicle detection have followed the general trend and these classification methods 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 which classifies a vehicle given its direct rear view into one of four classes: Sedan, Pickup truck, SUV and Minivan based on physical measurements such as the height and Visual Rear Ground Clearance of the vehicle. Generally, ground clearance of a vehicle is defined as the distance between the lower most mechanical part of the vehicle to the flat ground surface. But in our case for the given geometrical setup of the camera we cannot see the bottom most mechanical part of the vehicle. So we rename the ground clearance as Visual Rear Ground Clearance and define it as the distance between the lowest visible part of a vehicle to the ground surface when viewed by a camera at a given geometric setup installed above the road along which the vehicle is travelling. The Visual Rear Ground Clearance is a very important feature for the following reasons:

1) SUV and minivan have very subtle appearance differences when viewed from the rear view. The Visual Rear Ground Clearance is the only distinguishing feature that differentiates between the minivan and SUV from the rear view.

2) Most of the freeways in USA have cameras looking at the rear view of vehicles and hence side view based classification techniques will not be applicable. This also applies to the dashboard cameras in intelligent vehicles which looks at the rear view of the vehicle in front of it.

3) Last, Visual Rear Ground Clearance along with other rear view features can be used to perform rear view based classification which can directly be interfaced with the freeways camera and other commercial applications.

The computation of these physical features is challenging as it is conducted in the outdoor environment with changing illumination and appearance, the presence of shadows, occlusion and

moving vehicles at freeway speed. This method eliminates the need of high resolution and close up images of the logo and license plate due to the use of simple low-level features. Figure 1 shows a few examples of rear view images of the vehicles from our dataset.

Figure 1. Example of the direct rear view of moving vehicles.

II. RELATED WORKS AND OUR CONTRIBUTION

A. Related works

Most of the work on vehicle classification has been done for the side view appearance of the vehicle. The problem with this is that a side view of the vehicle can be easily occluded on multilane roads. Additionally most of the cameras deployed along the road capture rear or frontal views of the vehicle, reducing the applicability of side view based techniques in real-world applications.

Shan *et al.* [2] presented an edge based method for vehicle matching for images from non-overlapping cameras. This method computed the probability of two vehicles from two cameras being similar. The authors define the vehicle matching problem as a 2-class classification problem, thereafter apply a weak classification algorithm to obtain labeled samples for each class. The main classifier is trained by an unsupervised learning algorithm built on Gibbs sampling and Fisher's linear discriminant using the labeled samples. A key limitation of this algorithm is that it can only

perform on images that contain similar vehicle pose and size across multiple cameras.

Ma and Grimson [3] used the oblique side view to classify sedans versus taxis and sedans versus minivans by building a constellation model from their edge based approach.

Wu *et al*. [4] used a parameterized model to describe vehicle features, and a multi-layer perceptron network for classification. The authors stated that their method falls short when performing on noisy and low quality images and that the range in which the vehicle appeared as small.

Vehicle make and model recognition from the frontal view was used in [5]-[8]. Psyllos *et al*. [5] used close up frontal view images and a neural network classifier to recognize the logo, manufacturer, and model of a vehicle. Initially the logo is segmented and used to recognize the manufacturer of the vehicle. The authors report 85% correct recognition rate for manufacturer classification and only 54% for model recognition. Such an approach entirely depends on the logo, therefore, it might fail if used for the rear view as the logo may not always be present.

Petrovic and Cootes [6] compare various appearance features for identifying the make and model of vehicles from their frontal views. Negri *et al*. [7] use oriented contour features of frontal views to classify vehicles. Pearce and Pears [8] use a recursive partitioning scheme with Harris corner features to identify the class of a vehicle. All of these approaches use appearance information, which can widely change under varying environmental conditions. Therefore, the applicability of these approaches in real world scenarios is limited.

Cretu *et al* [9] classified vehicles into sedan, sports car, SUV, wagon and pickup truck. The authors used images of the vehicle taken from 6 different viewpoints and identified salient visual attention features and classified these features with a series of binary support vector machines.

Wen *et al* [10] used haar-like feature selection for vehicle detection and classification. The authors used an improved normalization algorithm to reduce the intra-class variability while increasing the inter-class variability.

Ohn-Bar and Trivedi [11] detected vehicles by object subcategorization using clustering of 3D orientation, position, occlusion level, and geometrical shapes. They achieved a detection rate of 81.94% and correct orientation estimation of 80.92%.

Kafai and Bhanu [12] use the direct rear view to classify vehicles. They use features such as locations and dimensions of landmarks (e.g., license plates and tail lights) as well as their spatial relationships in the network and use a hybrid dynamic Bayesian network to classify rear views of vehicles. Detection of these landmarks is challenging under varying environmental conditions. Furthermore, the authors could not classify between SUV and Minivan.

Thakoor and Bhanu [13] use the direct rear view to classify vehicles. They use the structural signature of the vehicle from the rear view and Support Vector Machine with radial basis function kernel to classify the vehicles. The authors could not classify between SUV and Minivan.

A summary of the related work is shown in Table 1.

Table 1. Summary of related works.

Authors	Vehicle data	Comments
Shan *et al* [2]	Side view	Edge-based, binary classification, Fishers Linear Discriminants, Gibbs sampling, unsupervised learning.
Ma *et al* [3]	Side view	Constellation model, edge-based approach.
Wu *et al* [4]	Side view	Neural network classification, Parametric model.
Psyllos *et al* [5]	Frontal view	Neural network classification, Phase congruency calculation, SIFT fingerprinting.
Petrovic *et al* [6]	Frontal view	Extracting normalized structure samples.
Negri *et al* [7]	Frontal view	Oriented-contour points model, Euclidean edge distance.

Pearce *et al* [8]	Frontal view	Harris corner strength over recursively partitioned image regions.
Cretu *et al* [9]	6 different viewpoints	Salient visual attention features, binary Support Vector Machines.
Wen *et al* [10]	Multi view	Haar-like feature selection, improved normalization algorithm.
Ohn-bar *et al* [11]	Multi view	Object subcategorization using clustering of features.
Kafai *et al* [12]	Rear view	Spatial relationship between landmarks, Hybrid Dynamic Bayesian Network.
Thakoor *et al* [13]	Rear view	Structural signatures, Support Vector Machines.
This paper	**Rear view**	**Physical measurement of features, Binary tree classification.**

B. Contributions of this paper

Unlike previous works as shown in Table 1, 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 Figure 1. The main reasons for choosing the rear view are, first, most of the research done on vehicle classification utilizes only the side view and the rear view has not been investigated much. Second, 19 states in the USA require only the rear license plate and third, most of the cameras on the highways look at the rear view of the moving vehicles, hence side view based classification techniques are not widely applicable in the real-world.

(b) The need for high resolution close up images of the vehicle logo and license plate is eliminated as we use simple low level features which are computationally inexpensive.

(c) We classify the vehicles based on the physical height of the features which has never been used before.

(d) The experimental results for the proposed technique was validated on 3 data-sets consisting of 1831 videos of rear view videos for four classes of vehicles namely sedan, minivan, SUV and pickup truck.

III. TECHNICAL APPROACH

This section elaborates on how a vehicle is classified from the rear view video. The Region-of-Interest (ROI) is extracted and the shadow of the ROI is removed. Next, we identify the features and estimate their height in inches by a multi-frame approach. Based on the extracted features we then classify the vehicles into one of the four classes using the J48 binary tree classifier using the Weka machine learning software [21]. The complete proposed system is shown in Figure 2. All components are explained in detail in the following sections.

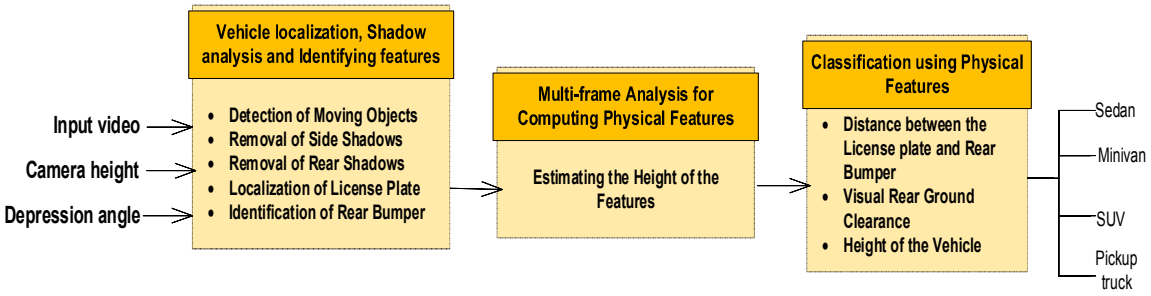


Figure 2. Our system for classifying the vehicle from rear view.

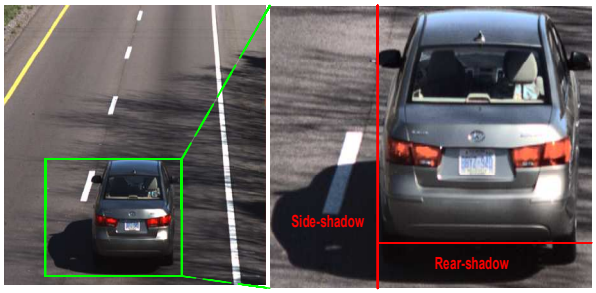
A. Vehicle Localization, Shadow Analysis and Identifying Features

This section explains about the steps for detecting the moving vehicle or Region-Of-Interest (ROI), processing the ROI to remove the shadows and identify the features.

1) Detection of Moving objects: Vehicle detection is an active research topic in transportation systems [14]-[17]. In this paper a Gaussian mixture model approach is used for moving vehicle detection. The Gaussian distributions are used to determine if a pixel is more likely to belong to the

background model or not. The Red, Green and Blue plane of the image are individually modelled as Gaussian models. An AND approach is used which determines a pixel as background only if it falls within three standard deviations for all the components in all three R, G, and B color channels [18]. The result of this process gives us the moving vehicle as well as the shadow associated with it. Figure 3 shows an example of detecting the moving vehicle. The moving vehicle and shadow shown in Figure 3 have pixel values that are significantly different than the background pixels (i.e., more than three standard deviations) and hence it was correctly detected as a moving vehicle.

We define the shadow associated with the moving vehicle to consist of two parts namely the Side-shadow and the Rear-shadow. The Side-shadow is the shadow present on either the left or right side of the vehicle. The Rear-shadow is the shadow present immediately below the rear bumper of the vehicle.



(a) (b)

Figure 3. (a) Input image from the video, (b) Extracted ROI.

We need to separate the shadow associated with the moving vehicle for two main reasons. First the Rear-shadow can be misidentified as the rear bumper which will result in a wrong estimation for the Visual Rear Ground Clearance. Second, we need to track the shadow in order to estimate the velocity at which the vehicle is travelling. We will explain the methods for removing these shadows in the next sub section.

2) Removal of Side-Shadows: In this sub section, we remove the Side-shadow associated with the ROI obtained from the previous step. When we look at a vehicle's rear view, it is bilaterally symmetric about the vertical axis. We exploit this symmetric property to separate the moving vehicle from the Side-shadow.

Given the orientation of ROI, the axis of symmetry is assumed to be vertical, i.e., it corresponds to one of the ROI columns. To estimate the axis of symmetry, we first estimate edge magnitudes and the orientation of the edges using Gabor filters. We used an 8 orientation Gabor filter with angles quantized between 0° to 180° (i.e., 0°, 22.5°, 45°, …, 180°) and wavelength of 5.35 pixels/cycle. To avoid texture edges, we apply surround suppression [19] to the Gabor response. We carry out non-maximal suppression to get the final set of edges. For a candidate axis location corresponding to the *j* th column of ROI *R* with edge magnitude *E* and its quantized orientation *O*, the votes are counted as

$$\sum_{\forall i,\, j^-,\, j^+ :\, (i,\, j^+) \in R,\, (i,\, j^-) \in R} v\left(i,\, j^-,\, j^+\right), \text{ where}$$

$$v\left(i,\, j^-,\, j^+\right) = \left\{ \begin{matrix} \min\left(E_{i,\, j^-}, E_{i,\, j^+}\right), & O_{i,\, j^-} = O'_{i,\, j^+} \\ 0, & \textit{otherwise} \end{matrix} \right\}$$

Additionally, $j^+ = j + \Delta$, $j^- = j - \Delta$, $O'_{i,j} = \pi - O_{i,j}$ for the candidate axis location j, Δ takes values from 1 to min(j, width(ROI) − j). The axis of symmetry is assigned to the column with the highest number of votes. Figure 4 shows the voting for the candidate axis location for the vehicle in Figure 3(b). The x-axis corresponds to the columns in the ROI and the y-axis is the number of votes assigned to the columns.

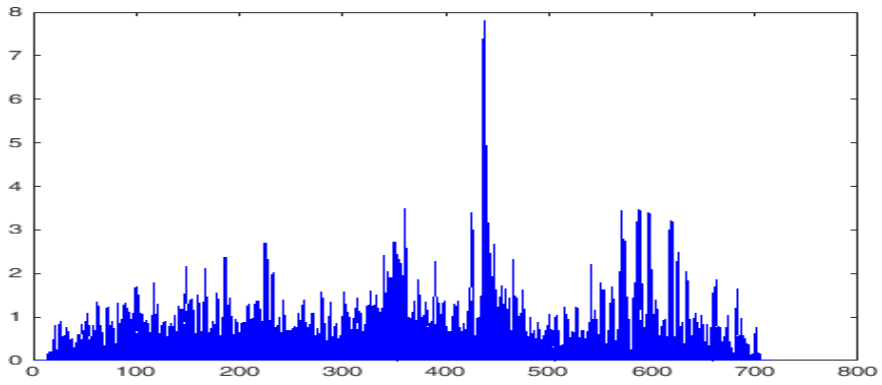


Figure 4. Voting of the candidate axis location.

The algorithm for obtaining the axis of symmetry is given below.

Algorithm 1. Estimating the axis of symmetry

Input: Grayscale image of ROI

Output: Axis of symmetry (Column)

for j = 1 : width of the ROI

Initialize vote(j) = 0

  for Δ = 1 : min(j, width(ROI) – j)

    for i = 1 : length of the ROI

      $j^{+} = j + \Delta$

      $j^{-} = j - \Delta$

      $O'_{i,j^{+}} = \pi - O_{i,j^{-}}$

      if ($O_{i,j^{-}} = O'_{i,j^{+}}$)

      $\text{vote}(j) = \text{vote}(j) + \min\left(E_{i,j^{-}}, E_{i,j^{+}}\right)$

    end

  end

end

return vote(j)

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 bounding box. Anything protruding out of

this bounding box is assumed to be part of the side shadow. The output of this step is the moving vehicle with the side shadow removed and rear shadow present below the rear bumper as shown in Figure 5.

(a) (b)

Figure 5. (a) Moving vehicle with Side-shadow and Rear-shadow, (b) Moving vehicle with Rear-shadow.

3) Removal of Rear-Shadows: This sub section explains a method for removing the shadow present below the bumper which we call as the Rear-shadow. To accomplish this we need a feature that would separate the moving vehicle from Rear-shadow. We observed that the outermost curvature of the tire (tire that is on the side opposite of the Side-shadow) that is almost touching the ground can be used to distinguish between the rear shadow and body of the vehicle. This feature will give us a row that would separate the rear shadow and the vehicle.

This is done by dividing the current ROI into 4 quadrants with the left tire in the third quadrant and right tire in the fourth quadrant. If the Side-shadow was cropped from the right, we need to scan the left tire or third quadrant. Similarly, if the Side-shadow was cropped from the left, we need to scan the right tire or fourth quadrant. We use Gabor filters at angles of 67.5° and 112.5° with 0° pointing towards the right side of the horizontal axis as shown in Figure 6.

In the previous sub section, we already identified from which side of the vehicle the Side-shadow was cropped out and based on that we scan for the tire that is on the opposite side of that shadow. After we have determined which tire to scan, we next determine the angle in degrees that the Gabor

filter should scan by N x 22.5°, where N is chosen depending on the location of the tire, N = 5 if we scan for the left tire and N = 3 for the right tire. After the curvature of the tire is detected, we assume everything below this line is part of the Rear-shadow and it is cropped out from the ROI. Figure 6 shows an example of a vehicle where the Side-shadow is on the left side of the car, hence we scan the fourth quadrant (right tire) with Gabor filters at 67.5°. The detected curvature of the tire is marked in green as shown in Figure 6(a) and the rear shadow below this curvature is cropped out as shown in Figure 6(b).

This approach fails when the curvature of both the tires are covered by the Rear-shadow. This happens only when the sun is present exactly in front of the vehicle. This orientation casts the shadow of the vehicle exactly behind it and does not cause any Side-shadow. This kind of scenario occurs only for a short duration of time in a whole year and hence we neglect this effect.

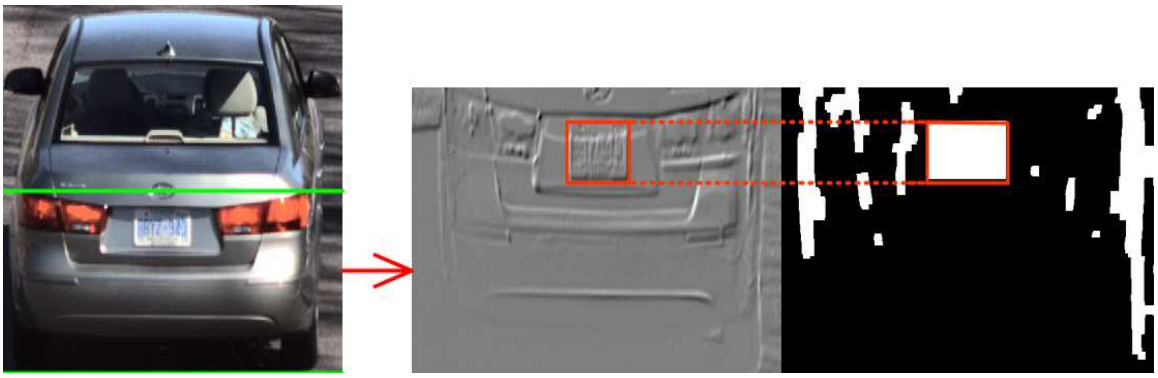
(a) (b)

Figure 6. (a) Curvature of the right tire, (b) Moving vehicle with rear shadow cropped out and orientation of Gabor filter.

4) Localization of License Plate: In this sub section 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 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 next following steps in this section we will be using only the bottom half of the ROI.

The integral image for the bottom half of the ROI is computed as in [20]. After obtaining the integral image, we use Haar-like wavelets to detect the horizontal and vertical edges in the image. Next we dilate the image consisting of only these edges using a straight line mask of length 3% of the total width of the integral image. Next 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 license plate such as aspect ratio of the license plate, area of the license plate and horizontal shape of the license plate we filter the blobs and identify the blob associated with 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 Figure 7. Figure 7(a) shows the detected ROI with no shadow, Figure 7(b) shows the Integral image of the bottom half of the ROI and Figure 7(c) shows the binary image of the extracted license plate.



(a) (b) (c)

Figure 7. (a) ROI with no shadow, (b) Integral image of the bottom half of the ROI, (c) Binary image of the extracted license plate.

5) Identification of the Rear Bumper: This section explains the method for identifying the bottom of the rear bumper. When we look the rear view of a passenger vehicle, it can be observed that the bottom of the license plate is always above the bottom of the rear bumper. By identifying the license plate and curvature of the tire from the previous sections, we can now conclude that the bottom of the rear bumper lies in between the region of the bottom of the license plate and curvature of the tire. This conclusion helps us reduce the search-area for identifying the bottom of the rear bumper.

We scan this search-area using a Gabor filter at an orientation of 180 degree. This results in detecting all horizontal edges in the search-area as shown in Figure 8. Next we dilate the image containing only these horizontal edges using a straight line mask of length 3% of the total width of the ROI. This is done to close any small gaps lying on the same row so that we have continuous horizontal lines.

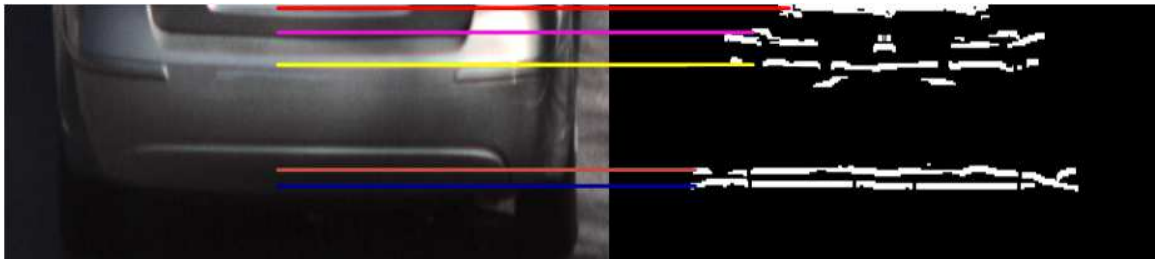


Figure 8. Detected horizontal edges in the search-area.

Next, we scan this binary image in reverse raster fashion. The first longest line from the bottom whose length is more than a threshold is selected as the bottom of the rear bumper. In our case the threshold was set to be 75% of the total width of the binary image. Figure 9 shows the identified bottom of the rear bumper of the vehicle in Figure 8.

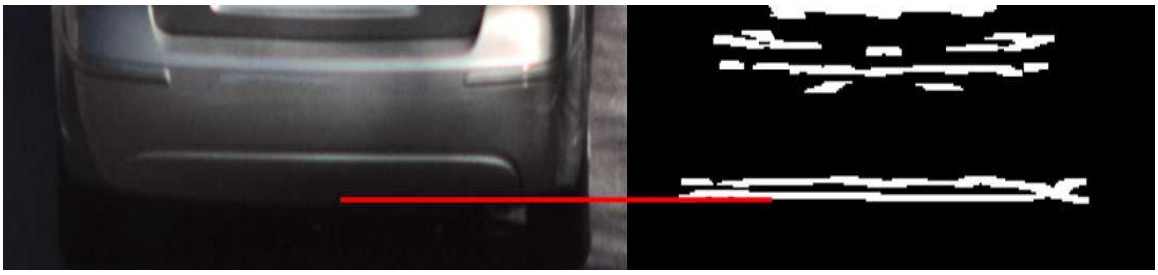


Figure 9. Identified bottom of the rear bumper.

B. Multi-frame Analysis for Computing Physical Features

In this section we estimate the height of the vehicle and height of the bottom the of rear bumper from the ground.

In most of the scenarios, we can assume that the velocity of a vehicle is negligible in the direction perpendicular to the lane and almost constant along the lane for a very short period of time when the video data are collected. We will characterize geometry of this scene in a parallel projection where the projection plane is orthogonal to the road plane as well as the camera image plane. Under this projection, the road and the image plane appear as lines. Figure 10 shows the scene geometry under the parallel projection.

The optical center of the camera is located at *O*. We treat this as the origin of the 2D camera coordinate system with the *x* axis pointing right and parallel to the road and the *y* axis pointing downwards. The height of the camera from the road is *h*. The focal length of the camera is *f* and the depression angle with respect to the *x* axis is θ.

The image plane of the camera intersects with *x*-axis and *y*-axis at $Ix \equiv \left(\frac{\mathrm{f}}{\cos\theta}, 0\right)$ and $Iy \equiv \left(0, \frac{\mathrm{f}}{\sin\theta}\right)$

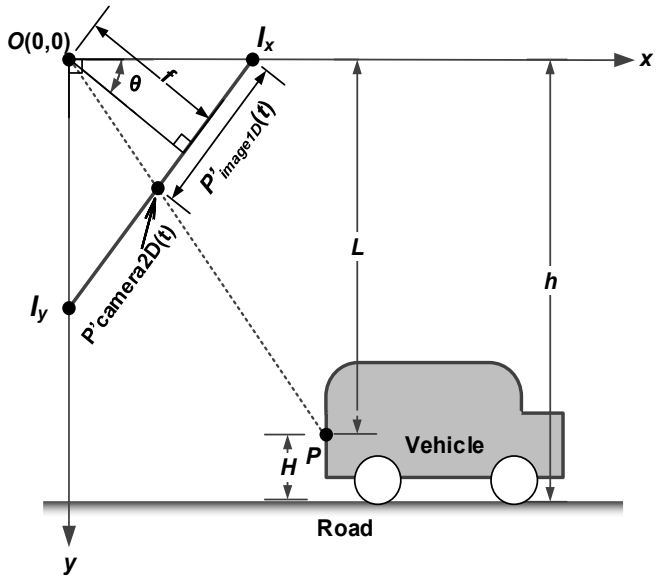


Figure 10. Scene geometry under parallel projection.

Consider point *P* on a vehicle visible from the camera. At time *t*, let the coordinates of *P* be *(D(t), L).* At time *t*, point *P* is projected at *P'(t),* which lies at the intersection of line *IxIy* and line *OP*.

Solving for this point yields, $P'_{camera2D(t)} = \left(\frac{fD(t)}{D(t)\cos\theta + L\sin\theta}, \frac{Lf}{D(t)\cos\theta + L\sin\theta}\right)$.

From the above, the projection of the line at infinity is, $\lim_{D(t)\to\infty} P'_{camera2D(t)} = \left(\frac{f}{\cos\theta}, 0\right) \equiv Ix$.

In the 1-D image co-ordinate system with *Ix* as the origin the projection is given by the distance between *Ix* and *P'(t)*, i.e., $P'_{image1D}(t) = \frac{Lf}{\cos\theta\left(D(t)\cos\theta + L\sin\theta\right)}$

Taking the inverse of the projection and differentiating with time t, we get,

$$\frac{d}{dt}\frac{1}{P'_{image1D}(t)} = \frac{cos^2\theta}{Lf}\frac{d}{dt}D(t).$$

For a vehicle travelling at a constant velocity V, the above expression turns into a constant K. Thus,

$$\frac{d}{dt}\frac{1}{P'_{image1D}(t)} = \frac{v cos^2\theta}{Lf} = K. \quad (1)$$

Thus, the inverse of $P'_{image1D}(t)$ varies linearly with time. The real 2-D image coordinates can be easily converted to the 1-D side-view image coordinates. First, the 1-D side-view image coordinate system is aligned along the y-axes of the 2-D image coordinates. Second, as shown in Figure 9, the origin Ix of the 1-D side-view image coordinate system is located on the projection of the line at the infinity. Thus, $P'_{image1D}(t) = P'_{image2D}(y) - P_\infty$, where P_∞ is the *y*-coordinate of the image projection of the line at the infinity.

Figure 11 illustrates how the parallel lane markings and their vanishing point can be used to find P_∞. Generally, P_∞ lies outside the image and has a negative value. In the rest of this section, we will use the 1-D side-view image coordinates.

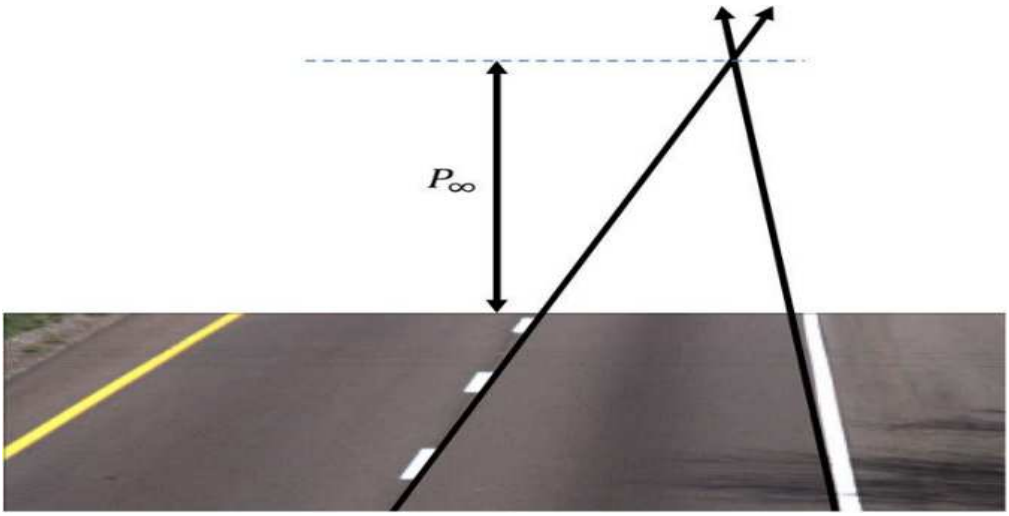


Figure 11. Finding P_∞ with the lane markings and vanishing point.

Based on Eq. (1), we define a constant $K_{vehicle}$ as $K_{vehicle} = \frac{v\cos^2\theta}{f}$ which is assumed not to change for a vehicle video. Eq. (1) can be written as $\frac{1}{j-i}\left(\frac{1}{Fj}-\frac{1}{Fi}\right)=\frac{K_{vehicle}}{L}$, where *Fi* and *Fj* are the tracked locations for a given feature in frames *i* and *j*. By knowing $K_{vehicle}$ the height of a feature *L* can be estimated by solving the above equation. However, even if the camera focal length, depression angle and height are known, the velocity *v* is different for each vehicle. We get around this by tracking the shadow cast by the vehicle which has same velocity as the vehicle.

Let the tracked locations of the shadow be S^i and S^j in frames *i* and *j* respectively. As for the shadow *L* = *h*, so Eq. (1) becomes $\frac{h}{j-i}.\left(\frac{1}{Sj}-\frac{1}{Si}\right)=\frac{v\cos^2\theta}{f}$.

Once the vehicle constant is known, the heights of vehicle features can be computed. For a feature with tracked locations F^i and F^j in frames *i* and *j* respectively. Solving for the unknown *L*, the height of the feature *H* can be estimated as $H = h - \mathrm{L} = \left(h - K_{vehicle}\right)\left(\frac{1}{Fj}-\frac{1}{Fi}\right)^{-1}$. (2)

C. Classification Using Physical Features

In this section we obtain the heights of the features required for the classification. We use a feature set of Visual Rear Ground Clearance, height of the vehicle and perpendicular distance between the bottom of the license plate and bottom of the rear bumper for classifying the vehicle. The term Visual Rear Ground Clearance is explained in the introduction section and the perpendicular distance between the bottom of the license plate and rear bumper is computed in pixels and normalized.

First we find the perpendicular distance between the bottom of the license plate and bottom of the rear bumper. We already found the locations of the license plate and bottom of the rear bumper in the previous sections. From this we calculate the perpendicular distance from the bottom row of

the license plate to the bottom of the rear bumper and normalize this distance with respect to the largest computed distance.

Next, we find the Visual Rear Ground Clearance. This is the height of the bottom of the rear bumper from the ground. This is found by obtaining the location of the bottom of the rear bumper in successive frames and substituting these locations into equation (2).

The height of the vehicle is found by substituting the location of the top most horizontal edge of the ROI in successive frames into equation (2).

After obtaining the height of the above mentioned features, we classify them using the J48 binary tree algorithm. J48 is an open source Java implementation of the C4.5 algorithm in the Weka machine learning software [21].

J48 builds decision trees from a set of labeled training data using the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. It examines the normalized information gain (difference in entropy) that results from choosing an attribute. It makes the decision by choosing the attribute with the highest normalized information gain and then it recurs on the smaller subsets. This procedure stops once all the instances in a subset belong to the same class. Then a leaf node is created in the decision tree telling us to choose that class.

IV. EXPERIMENTAL RESULTS

The proposed technique for classifying passenger vehicles from the rear view was implemented on Matlab and validated on video data from 3 datasets. The vehicle classes in the three datasets are sedan, minivan, SUV and pickup truck. We use the J48 binary tree algorithm for classification and k-fold cross validation with k=10 to evaluate our approach. The classification was done using the Weka machine learning software.

A. Datasets

The first dataset (called dataset 1) consists of 876 vehicles, the second dataset (called dataset 2) consists of 896 vehicles and the third dataset (called dataset 3) consists of 59 vehicles. For both dataset 1 and dataset 2, the camera was setup at two different freeways at a height of 22 feet and depression angle of 8 degree. The images in dataset 1 and dataset 2 were captured during different times (morning to late evening) of the day. The first dataset consists of 309 sedans, 259 SUV's, 119 minivans and 189 pickup trucks. The second dataset consists of 321 sedans, 266 SUV's, 130 minivans and 179 pickup trucks. For the third dataset, the camera was setup at a height of 10 feet and depression angle of 14 degree in front of the Bourns College of Engineering 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 SUV's, 4 minivans and 9 pickup trucks.

B. Performance Metrics

The confusion matrix for the classification results of dataset 1 and dataset 2 are shown below in Table 2 and Table 3. The ground-truth class labelling (correct class labelling) for each vehicle was done manually by looking at the video. We get a correct classification when the predicted class and the ground-truth class are the same. If the predicted class and the ground-truth class are not the same, then it is a misclassification.

Our rear view classification system achieved a correct classification rate of 92.81%, false alarm rate of 2.62% and precision rate of 92.64% for dataset 1. For dataset 2, our classification system achieved a correct classification rate of 92.85%, false alarm rate of 2.81% and precision rate of 92.9%. From Table 2 and Table 3, it can be seen that out of 525 SUV's 21 were misclassified as sedan because the height and Visual Rear Ground Clearance were under-estimated and out of 630 sedans 18 were misclassified as SUV because the height and Visual Rear Ground Clearance were

over-estimated. A key reason for this was that the bottom of the rear bumper and top most horizontal edge were misidentified because of poor illumination conditions. 19 SUV's were misclassified as pickup trucks because certain models of SUV have the license plate close to the rear bumper.

For evaluating dataset 3 we used the correctly classified vehicles of dataset 1 and dataset 2 combined together as the training set to build the model for classification and used dataset 3 as the testing set. The confusion matrix of the results is shown below in Table 4. The system achieved a correct classification rate of 94.91%, false alarm rate of 1.83% and precision rate of 92.79%. Only 1 SUV was misclassified as a sedan and 2 sedans were misclassified as SUV.

Table 2. Confusion matrix for dataset 1.

Predicted class ⟶	**SUV**	**Pickup truck**	**Sedan**	**Minivan**
SUV	233	10	10	6
Pickup truck	1	185	1	2
Sedan	8	1	294	6
Minivan	12	1	5	101

Table 3. Confusion matrix for dataset 2.

Predicted class ⟶	**SUV**	**Pickup truck**	**Sedan**	**Minivan**
SUV	243	9	11	3
Pickup truck	1	174	4	0
Sedan	10	7	302	2
Minivan	11	2	4	113

Table 4. Confusion matrix for dataset 3.

Predicted class ⟶	**SUV**	**Pickup truck**	**Sedan**	**Minivan**
SUV	10	0	1	0
Pickup truck	0	9	0	0
Sedan	2	0	33	0
Minivan	0	0	0	4

C. Analysis of Experiments

Figure 12 shows the ROC plot for all the 4 classes of vehicles from dataset 1, dataset 2 and dataset 3 combined.

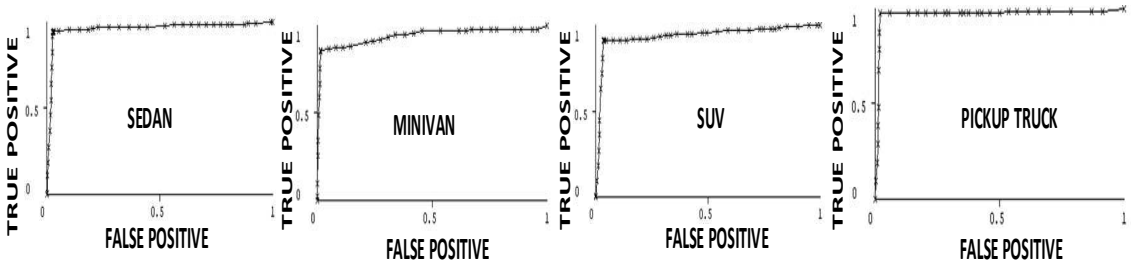


Figure 12. ROC plot for all the four classes in dataset 1, dataset 2 and dataset 3 combined.

Figure 13 shows the normalized pattern feature matrix for all the 1831 vehicles from dataset 1, dataset 2 and dataset 3 combined.

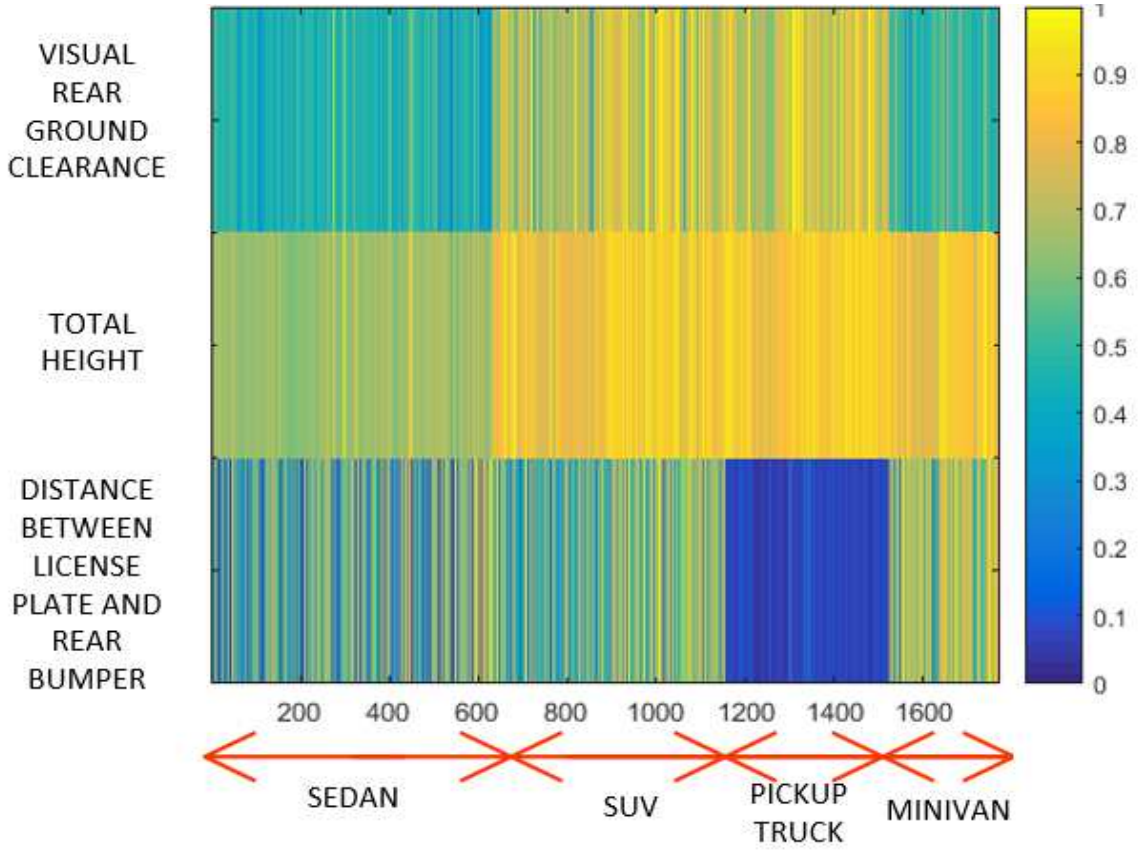


Figure 13. Normalized pattern feature matrix for 1831 vehicles from dataset 1, dataset 2 and dataset 3.

It can be observed from Figure 13 that Sedan and minivan have lower Visual Rear Ground Clearance compared to SUV and pickup truck because the rear bumper of sedan and minivan are much closer to the ground surface. Sedan have the shortest height compared to SUV, minivan and pickup truck. Pickup truck have the least perpendicular distance between the bottom of the license plate and bottom of the rear bumper.

Table 5 shows the shows the average and standard deviation for the 3 features for all 4 classes. The standard deviation of the Visual Rear Ground Clearance and height of the vehicle for all 4 classes did not deviate much, but for the normalized perpendicular distance between the bottom of the LP* and RB* the standard deviation for SUV, sedan and minivan were high because the license plate for these classes can be located anywhere on the rear surface, whereas for pickup trucks the license plate is mostly located on the rear bumper and hence the standard deviation for pickup truck was less compared to others.

Table 5. Statistic of computed features.

Class	**Visual Rear Ground Clearance** **Avg/S.D.**	**Height of the vehicle** **Avg/S.D.**	**Perpendicular distance between bottom of the license plate and bottom of the rear bumper** **Avg/S.D.**
SUV	22.08/4.76	75.94/7.58	0.474/0.198
Pickup truck	23.43/5.22	78.38/6.76	0.063/0.053
Sedan	13.87/3.92	59.69/6.48	0.398/0.232
Minivan	14.88/3.60	75.71/7.10	0.585/0.162

Table 6 shows the comparison of the average Visual Rear Ground Clearance and average height (average is taken over 5 different color vehicles of each vehicle model) for four vehicles with the ground truth. Chevrolet Impala (sedan) and Honda Odyssey (minivan) have lower Visual Rear Ground Clearance while Honda CR-V (SUV) and Toyota Tundra (pickup truck) have higher Visual Rear Ground clearance and Chevrolet Impala has the

least height. The results from Table 6 is consistent with the observations and inferences made from Table 5 and Figure 13.

Table 6. Comparison of ground truth with computed results.

Vehicle model (class)	Ground truth VRGC (in)	Ground truth Height (in)	Computed VRGC Avg. (in), % error	Computed Height Avg. (in), %error
Honda CR-V (SUV)	17.7	64.7	19.64, 10.73%	71.9, 11.13%
Toyota Tundra (Pickup truck)	18.1	75.8	18.29, 1.05%	79.6, 5.01%
Chevrolet Impala (Sedan)	11.3	58.7	12.31, 8.94%	62.4, 6.30%
Honda Odyssey (Minivan)	13.2	68.4	14.42, 9.10%	73.2, 7.02%

Figure 14 shows the rear view image of all the vehicles used in Table 6. The vehicles in the first row are different Chevrolet Impala's, followed by Honda CR-V, Toyota Tundra and Honda Odyssey in the following rows, respectively. Our Matlab implementation of the system runs on a computer with an Intel Celeron processing unit N2830 at 2.16GHz and the average time taken to classify the 25 vehicles in Figure 14 was 67.78 seconds per vehicle.

Figure 14. Rear view images of the vehicles used in Table 6.

Figure 15(a) shows an example of a Chevrolet Impala (sedan) with its physical features that are marked. The height of the Chevrolet Impala is estimated to be 60.3 inches which has an error percentage of 2.72% compared with the ground-truth and Visual Rear Ground Clearance was estimated to be 11.2 inches with an error percentage of 0.89% and the perpendicular distance between the bottom of the license plate and bottom of the rear bumper is 18 pixels. Figure 15(b) shows an example of a Honda Odyssey (minivan) with an estimated height of 70.1 inches with error percentage of 2.43% and Visual Rear Ground Clearance of 12.7inches with error percentage of 3.79% and the perpendicular distance between the bottom of the license plate and the bottom of the rear bumper is 93 pixels.

Figure 15. Example of a Chevrolet Impala and a Honda Odyssey with their rear view features.

Figure 16 shows examples of vehicles that were misclassified. In Figure 16(a) the SUV is misclassified as minivan because the bottom of the rear bumper was misidentified which resulted in under-estimating the Visual Rear Ground Clearance. In Figure 16(b) the height of the sedan was over-estimated due to tracking errors which resulted in misclassification as SUV. Figure 16(c) shows an example where the Visual Rear Ground Clearance was

over-estimated which resulted in the minivan being misclassified as SUV and in Figure 16(d) the height of the pickup truck was underestimated due to tracking errors and hence it was misclassified as a sedan.

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

Figure 16. Misclassified passenger vehicles.

Table 7, 8 and 9 shows the comparison of correct classification, false alarm and precision obtained by J48 and 3 well known classifiers: k- Nearest Neighbor (k-NN), Support Vector Machines (SVM) and Naïve Bayes [22]. We set the value of k to be 4 in the k-NN classifier and used a radial basis function as the kernel type for the SVM classifier. It can be observed from Table 7, 8 and 9 that the J48 classifier has the best overall performance in classification.

Table 7. Comparison of correct classification by J48, k-NN, SVM and Naïve Bayes.

Class	**k-NN k=4**	**SVM**	**Naïve Bayes**	**J48**
SUV	89.25%	78.31%	79.25%	**91.37%**
Pickup truck	95.41%	63.54%	**98.11%**	97.63%
Sedan	**94.57%**	94.46%	94.88%	94.00%
Minivan	80.95%	86.13%	77.78%	**84.58%**
Overall	91.26%	82.17%	88.53%	**92.59%**

Table 8. Comparison of False alarm by J48, k-NN, SVM and Naïve Bayes.

Class	k-NN k=4	SVM	Naïve Bayes	J48
SUV	5.02%	13.93%	**3.21%**	4.11%
Pickup truck	2.61%	7.17%	3.97%	**1.82%**
Sedan	3.28%	**2.01%**	5.40%	3.00%
Minivan	**1.37%**	1.46%	3.28%	2.84%
Overall	3.36%	6.56%	4.18%	**2.72%**

Table 9. Comparison of Precision by J48, k-NN, SVM and Naïve Bayes.

Class	k-NN k=4	SVM	Naïve Bayes	J48
SUV	88.27%	70.23%	**91.14%**	90.36%
Pickup truck	90.50%	70.17%	86.82%	**93.37%**
Sedan	94.33%	**96.41%**	90.95%	94.72%
Minivan	91.05%	90.82%	80.01%	**91.12%**
Overall	91.13%	82.55%	88.56%	**92.73%**

Figure 17 shows the of ROC plots of all the above classifiers. Although the ROC for J48, k-NN and Naïve Bayes look similar, J48 outperforms them.

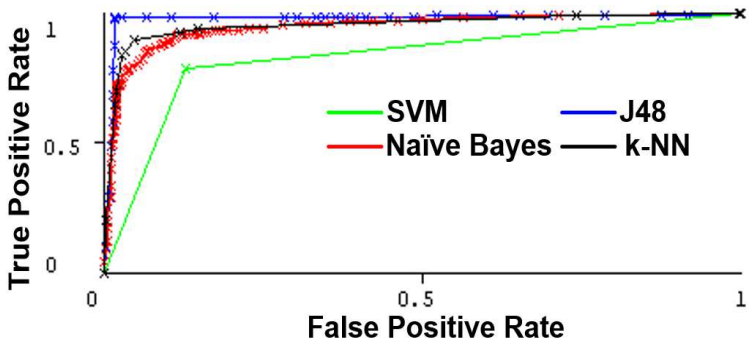


Figure 17. ROC plot for J48, SVM, Naïve Bayes and k-NN classifiers.

D. Comparison of Results

Table 10 shows the comparison of the results obtained by our system with other rear view classification systems.

Kafai and Bhanu [12] achieved a correct classification rate of 96.6% using a Hybrid Dynamic Bayesian Network (HDBN) with low level features. The authors collected 177 rear view videos of

vehicles travelling in front of the Bourns College of Engineering building at UC- Riverside for classification. The drawbacks of their rear view classification systems are:

- The features used for classification were handpicked.
- They could not classify between SUV and minivan.

Thakoor and Bhanu [13] used the structural signatures of the moving vehicles and SVM classifier to achieve a correct classification rate of 86%. The authors used dataset 1 and dataset 2 that are used in this paper to perform the rear view classification but could not classify between SUV and minivan.

Our system overcame the problem of classifying between SUV and minivan by estimating the Visual Rear Ground Clearance of a vehicle which is the most distinguishing feature for SUV and Minivan from the rear view. Using the Visual Rear Ground Clearance along with the height of the vehicle and normalized perpendicular distance between the bottom of the license plate and bottom of the rear bumper, our system achieved a correct classification rate of 92.59%. Although our correct classification rate is lesser compared to the authors of [12], overcoming the problem of classifying between SUV and minivan from the rear view overshadows this difference.

Table 10. Comparison of results.

Method	**Results**	**Advantages**	**Limitations**
Kafai and Bhanu [12]	96.61%	Used low level features & HBDN for classification	Handpicked features, could not classify between SUV and Minivan
Thakoor and Bhanu [13]	86.06%	Used structural signatures for classification	Could not classify between SUV and Minivan
This paper	**92.59%**	**Used physical measurement for classification and classified between SUV and Minivan**	**Requires a source of illumination**

V. CONCLUSION

We proposed a technique for classification of vehicle from the rear view using multiple video frames. First, we computed the Visual Rear Ground Clearance and height of the vehicle as physical measurements which has never been addressed before using video. Second, our approach is robust as it is less susceptible to illumination changes and independent of the color, size and type of vehicle. Moreover our approach does not need high resolution close up images because of our simple low-level features (e.g., height and normalized distance) making it possible to perform real-time classification. While sedans, pickup trucks, minivans, and SUVs form the majority of the vehicles on the road, including the heavy goods vehicles, buses, and motorcycles in the classifier would give a complete classification solution. Lastly, this approach is suitable only in the presence of daylight because as the illumination decreases certain features such as the rear bumper become less distinctive and we fail to identify them.

Future work will involve exploring the use of thermal images to perform classification under night conditions and incorporating this system into intelligent vehicles by attaching a dashboard camera which can view the rear view of the vehicle in front of it, compute the physical features and perform the classification in real-time. Additionally since vehicles have their headlights turned on when there is no daylight, making it possible to identify the bottom of the rear bumper and hence compute the Visual Rear Ground Clearance and other features for classification even in the absence of daylight.

REFERENCES

[1] [1] D.G. Lowe, "Object recognition from local scale-invariant features", *IEEE International Conference on Computer Vision", 2, pp. 1150-1157, 1999.*

[2] Y. Shan, H. Sawhney, and R. Kumar, "Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4), pp. 700–711, April 2008.

[3] X. Ma and W. Grimson, "Edge-based rich representation for vehicle classification," *IEEE International Conference on Computer Vision*, pp. 1185–1192, 3005.

[4] W. Wu, Z. QiSen, and W. Mingjun, "A method of vehicle classification using models and neural networks," *IEEE Vehicular Technology Conference*, 4, pp. 3022–3026, 2001.

[5] A. Psyllos, C. N. Anagnostopoulos, and E. Kayafas, "Vehicle model recognition from frontal view image measurements," *Computer Standard Interfaces*, 33, pp. 142–151, February 2011.

[6] V. S. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," *British Machine Vision Conference*, pp. 587–596, 2004.

[7] P. Negri, X. Clady, M. Milgram, and R. Poulenard, "An oriented-contour point based voting algorithm for vehicle type classification," *IEEE International Conference on Pattern Recognition*, pp. 574–577, 2006.

[8] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," *IEEE Advanced Video- and Signal based Surveillance,* pp. 373–378, 2011.

[9] A.-M. Cretu and P. Payeur "Biologically inspired visual attention features for a vehicle classification task" International Journal on Smart Sensing and Intelligent Systems. 4(3), September 2011.

[10] X. Wen, L. Shao, W. Fang and Y. Xue, “Efficient feature selection and classification for vehicle detection” *IEEE Transactions on Circuits and Systems for Video Technology,* 25(3), pp. 508-517, 2014.

[11] E. Ohn-Bar and M. M. Trivedi “Learning to detect vehicles by clustering appearance patterns” *IEEE Intelligent Transportation Systems*, 2015.

[12] M. Kafai and B. Bhanu, “Dynamic Bayesian networks for vehicle classification in video,” *IEEE Transactions on Industrial Informatics*, 8(1), pp 100-109, February 2012.

[13] N.S. Thakoor and B. Bhanu, "Structural signatures for passenger vehicle classification in Video," *IEEE Intelligent Transportation Systems,* 14(4), pp. 1796-1805, December 2013.

[14] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, “A real-time vision system for nighttime vehicle detection and traffic surveillance,” *IEEE Transactions on Industrial Electronics*, 58, pp. 2030–2044, May 2011.

[15] X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, “Vehicle detection and motion analysis in low-altitude airborne video under urban environment,” *IEEE Transactions on Circuits and Systems for Video Technology,* 21(10), pp. 1522–1533, October 2011.

[16] F. Alves, M. Ferreira, and C. Santos, “Vision based automatic traffic condition interpretation,” *IEEE International Conference on Industrial Informatics,* pp. 549–556, July 2010.

[17] M. Chacon and S. Gonzalez, “An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems,” *IEEE Transactions on Industrial Electronics,* 1, pp. 99, 2011.

[18] N. Thakoor, J. Gao, “Automatic Video Object Shape Extraction and its Classification with Camera in Motion”, *IEEE International Conference on Image Processing*, 3, pp. 437-440, September 2005.

[19] C. Grigorescu, N. Petkov, and M. Westenberg, “Contour detection based on nonclassical receptive field inhibition,” *IEEE Transactions on Image Processing*, 12(7), pp. 729–739, July 2003.

[20] P. Viola and M.J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, pp. 511–518, 2001.

[21] Weka machine learning software, version 3.6.13, The University of Waikato, New Zealand, 2015.

[22] I. Rish, “An empirical study of the naive Bayes classifier”, *International Joint Conference on Artificial Intelligence*, 3(22), pp. 41-46, August 2001.