Multiclass Vehicle Classification Across Different Environments

Aisha S. Azim[1], Ashraf Alkhairy[3]
King Abdulaziz City for Science and Technology
Riyadh, SA

Afshan Jafri[2]
College of Computer and Information Sciences
King Saud University, Riyadh, SA

***Abstract*—Vehicle detection and classification are necessary components in a variety of useful applications related to traffic, security, and autonomous driving systems. Many studies have focused on recognizing vehicles from the point of view of a single perspective, such as the rear of other cars from the driving seat, but not from all possible perspectives, including the aerial view. In addition, they are usually given prior knowledge of a specific kind of vehicle, such as the fact that it is a car, as opposed to being a bus, before deducing other information about it. One of the popular classification techniques used is boosting, where weak classifiers are combined to form a strong classifier. However, most boosting applications consider complex classification problems to be a combination of binary problems. This paper explores in detail the development of a multi-class classifier that recognizes vehicles of any type, from any view, without prior information, and without breaking the task into binary problems. Instead, a single multi-class application of the GentleBoost algorithm is used. This system is compared to a similar system built from a combination of separate classifiers that each classifies a single vehicle. The results show that a single, multi-class classifier clearly outperforms a combination of separate classifiers, and proves that a simple boosting classifier is sufficient for vehicle recognition, given any type of vehicle from any perspective of viewing, without the need of representing the problem as a complex 3D model.**

***Keywords*—Vehicle detection; vehicle recognition; multiclass learning; boosting; GentleBoost**

I. Introduction

The detection and classification of vehicles are essential steps in many important applications, including autonomous driving systems, traffic flow prediction for transport management, vehicular safety, criminal tracking, and intelligent transportation systems with implementations that range from security surveillance to traffic monitoring during the Hajj season, impacting millions of pilgrims at a time. Coupled with the fact that cameras and imaging technology have seen massive improvements in recent years, on-road vehicle detection has become an active research area with valuable progress for close to a decade.

A large number of vehicle detection studies concentrate on vehicles seen from a specific view or perspective, such as the rear view of vehicles as they appear from the ego vehicle's driving seat, or from a camera mounted on the ego vehicle. There are different forms of classification that can be used to detect vehicles: multi-view classification, which recognizes the same vehicle from different viewing perspectives or poses; and multi-class classification, which recognizes vehicles in spite of variations in their shapes and sizes, based on belonging to different classes, such as buses and cars. There are systems that have been designed to recognize objects from different views, but they are often generalized object detectors [1], or else they focus either on vehicle detection through multi-view classification [2], or else through multiclass classification [3], [4]. But so far there have not been many serious studies on vehicle detection that support both multi-view and multi-class applications.

This paper will explore the development of a system that recognizes vehicles both across views as well as across classes, using cascaded boosting.

First introduced by Viola and Jones [5] to detect human faces, the cascaded boosted classification is one of the popular techniques in use for vehicle detection and recognition. It reaches high levels of classification accuracy by using weak classifiers which individually have low accuracy, but which are combined together to produce a strong classifier.

Studies in the concept of boosted classification began in the 1990s [6], and have since been picked up by researchers and applied to a rich variety of problems across different fields. Breiman, an expert in machine learning, claimed that "Boosting is the best off-the-shelf classifier in the world." [7].

Many vehicle detection systems have been built using boosted classification as well. These are often developed with different variations in the features used for classification, the exact boosting algorithm implemented, or an efficient combination of the features and the boosting classifier. However, the vast majority of these systems remain confined to binary classification; hence they often address simple questions as well. Two such questions would be: "is this vehicle a so-and-so model?", or "is this object a car from the rear view?" Note that questions like these are either (a) answered with a-priori information, or (b) limited in scope. For example, the first question was already provided with information that the object was a vehicle, and the second asked whether an object was the rear-view of a car, but not whether it was a car given any view of it.

This paper will attempt to recognize vehicles in a real-world scenario, using multi-class boosted classification, with no a-priori knowledge. That is, our system must be able to answer the complex question of whether an object is a vehicle, irrespective of (a) what type of vehicle it is (car, truck, bus), and (b) what perspective it is viewed from. In order to do this,

we will extend the binary classification problem to m-ary classification.

The rest of this work is organized as follows. Related literature is presented in Section II, starting from an overview of the field of object detection, then focusing on vehicle detection in particular, and then briefly covering work on mutli-class classification. Section III highlights the contribution of this paper. The approach and system modeling of our study are described in detail in Section IV. The Boosting algorithm will first be introduced, followed by the formal modeling of our system, and then the two approaches that we take towards achieving Vehicle Classification. Section V describes the experimental setup, the tools, the data and the method in detail; and Section VI presents the experimental results and discussion. Section VII concludes this paper.

II. Related Work

A. Background

Object detection is a vast field within computer vision. With wide applications across robotics, control systems, security systems and automation, much research has been conducted in order to develop systems that can recognize vast arrays of objects from a single image. The problem is challenging, but progress has been made towards systems that can recognize limited numbers of objects.

Vehicle detection is one specific application of object detection that has seen notable progress in the past decade.

Methods of detecting vehicles in computer vision broadly fall under two categories: motion-based and appearance-based. Motion-based methods require an input of a stream of images, and they recognize the movement of vehicles against a stationary background. Whatever does not change – or changes slowly – over the image stream is taken to be the background, with the remaining objects being considered as moving objects. Motion-based methods are useful for applications such as driving assistance systems, where the vehicle is running live on the road and has access to a stream of input images, or for automated driving [8], [9], [33].

However, the drawback of motion-based methods is that they can work only given a stream of images, but not with individual, static images. This limits their application since there are many instances in traffic surveillance or crime tracking when a stream of images is not available as input. In addition, the majority of the motion-based approaches in the literature are useful from the point of view of the ego vehicle or a fixed camera.

Appearance-based methods are able to detect objects based on their appearance in a single, static image. Given efficient algorithms, they can also be used in real-time applications in the same way that motion-based methods can be used, utilizing continuity of motion to further enhance performance. While the literature has explored appearance-based methods also from the point of view of the ego vehicle, it has also been used extensively to detect vehicles from other angles.

In addition to traffic surveillance, criminal detection requires vehicle recognition too, and not only from the ground but often from high altitudes, and over different environments. While aerial view vehicle detection exists [10], it is specific to that application and not focused on accurate detection from the ground view.

This paper describes a system that is capable of recognizing different classes of vehicle, across different environments, and from different views -– aerial or ground.

B. Vehicle Detection

Under the appearance-based paradigm, different methods have been adopted for vehicle detection. We mention a few prominent ones below.

Behley et al [11] used a mixture model of bag-of-words representation of segments to classify segments from given input images. The system was specific for laser-based images and was particularly applicable to driving assistance for cars equipped for laser scan images.

Part-based models have also been used in vehicle detection, where the individual parts of a vehicle are used to detect the whole. This idea was used by Felzenszwalb et al [2] and Ye Li et al [12]. Felzenszwalb et al used latent Support Vector Machines to train mixtures of multi-scale, star-structured part-based models, relative to the “root” of the object. The parts were determined at a higher resolution using finer filters, while the root was detected using a coarse resolution. Scores were calculated to measure the relative distance of the parts from the root, using a feature pyramid representing the input image at different scales. While this method is capable of detecting vehicles despite variations including pose, it is limited to a range of angles, since deformable parts are not visible at all angles of a vehicle. For instance, the top or the side pose of a car are very different from the front view, and the detection of these views was not explored.

Similarly, Ye Li et al [12] used part-based models as well, using two-part vehicle models with a focus on tackling the occlusion challenge. The study focuses on urban environments and on vehicles in limited poses, while our work focuses on vehicles in multiple poses and across multiple environments.

Several other approaches have been used [13], [14], [15], [16], but one of the prominent approaches remains the boosted classification approach [6].

Boosted classification emerged as a powerful method of object detection after the instrumental work on face detection by Viola and Jones [5]. The Viola-Jones classifier gained its power and popularity by using classifiers which were individually only slightly more accurate than 50%, and hence did not require complex computations, but which together created a robust classifier when combined. Various boosting mechanisms have been used in vehicle detection as well, such as [17] and [18] that used online boosting, [19] and [34] which employed Adaboost including for active learning, and the various boosting studies in [20].

However, most of these methods have been focused on the detection of vehicles in limited poses and from limited perspectives, with the most common being vehicle rear-view detection from the perspective of the ego vehicle.

C. Boosted Classification for Object Detection

Aside from vehicle detection in particular, multi-class classification in the context of general object detection has been studied earlier. Torralba et al [21] trained images using JointBoost, which employs GentleBoost for training but with shared stumps among classes. The shared feature-learning was introduced to take advantage of the similarity of object features during multi-view classification, which reduces the space and time complexity for learning individual binary classifiers. On the other hand, using shared regression stumps reduced the precision of intra-class classification.

Shalev-Shwartz et al [22] followed a similar approach, but used different heuristics per boosting round in order to improve intra-class classification.

However, in both cases, the multi-class classification problem was still fundamentally treated as a combination of binary classifiers.

III. Contribution

This work explores the development of a comprehensive vehicle classification system. Its contributions are three-fold.

First, multi-view vehicle classification will be attempted for the first time using multi-class Gentle Boosting, where most other studies on vehicle detection have traditionally implemented boosted classification by dividing the problem into binary problems, rather than treating it as an m-ary problem.

Secondly, the system will detect vehicles across two major dimensions: vehicle class, where the classes consist of (i) cars, and (ii) big vehicles; and vehicle perspective, or view. This system considers 25 likely perspectives for each vehicle, starting with the horizontal rear view of the vehicle, and moving around the vehicle with different angles of inclination, until the final top view. Most other studies focus on classifying the view of cars only, or they focus on different vehicle classes but from a single viewpoint.

Thirdly, since the literature tends to study techniques intended to tackle the individual issues related to vehicle detection, such as detection in spite of occlusions, a paper that comprehensively describes the implementation of a vehicle classification system will be a valuable contribution to the field of vehicle detection at this point.

IV. Approach and System Model

A. Boosting

This paper describes the implementation of a multi-class boosting classifier for vehicle detection that treats the problem as inherently multi-class, rather than breaking it down into binary problems.

Boosting algorithms have been used for multi-class classification before. But before addressing multi-class classification, let us make a quick review of the basic boosting algorithm for binary problems.

Adaboost is one of the most basic boosting algorithms and was proposed by Freund and Schapire [6].

The crux of the algorithm is to use many weak learners, or classifiers with accuracy slightly better than 50%, and to combine them to build a strong classifier. The performance of weak classifiers are improved over a number of rounds on a given dataset, by noting which classifiers generated errors in previous rounds, and adjusting weights on misclassified training samples in order for the weak classifiers to "improve" classification in subsequent rounds.

AdaBoost, for a binary problem, is presented below as Algorithm 1 [23].

Algorithm 1 Discrete AdaBoost

1. Start with weights $w_i = 1/N$, $i=1,...,N$

2. Repeat for $m = 1,2,...,M$:

(a) Fit the classifier $f_m(x) \in \{-1,1\}$, using weights w_i on the training data.

(b) Compute $err_m = E_w[1_{(y \neq fm(x))}]$, $c_m = log((1-err_m)/err_m)$

(c) Set $w_i \leftarrow w_i \exp[c_m 1_{(y \neq fm(xi))}]$, $i=1,2,...,N$, and renormalize so that $\Sigma_i w_i = 1$.

3. Output the classifier $sign [\Sigma^M_{m=1} c_m f_m(x)]$.

In this algorithm, N represents the number of training samples, which are pairs of data points x_i and its corresponding true class y_i, which can be either -1 or 1. Training data is input as (x_1, y_1), ..., (x_N, y_N). M is the number of weak classifiers, $f_1(x)$, ..., $f_m(x)$, each of which can output either 1 or -1. Ew is the expectation of training data of weights $w=(w_1, ..., w_N)$, and $1\{(S)\}$ indicates the set S.

At the beginning of the algorithm, all training samples are given equal weights. Then, for each weak classifier $f_m(x)$, a constant, cm, is computed to generate a weight for each data point, based on the error of the classifier. New weights are then calculated for each training sample in such a way that those samples that were misclassified have their weights increased by a factor that depends on the weighted training error.

The strong classifier, $F(x)$ is defined as the sum of the products of cm and fm, a linear combination of all the weak classifiers. The final prediction is $sign(F(x))$.

However, because Adaboost concentrates weight exponentially on misclassified samples, it becomes sensitive to noise. In order to address this problem, GentleBoost was proposed [23]. It successfully overcomes the noise-sensitivity issue by updating the weak classifiers in bounded steps, rather than unbounded steps. GentleBoost classifiers are regression functions that return class probability estimates, which are then used in a factor for computing new weights to update the functions.

The GentleBoost algorithm is reproduced below in Algorithm 2.

Algorithm 2 Gentle AdaBoost

1. Start with weights $w_i = 1/N$, $i=1,...,N$, $F(x) = 0$.

2. Repeat for $m = 1,2,...,M$:

(a) Fit the regression function $f_m(x)$ by weighted least-squares of y_i to x_i with weights w_i.

(b) Update $F(x) \leftarrow F(x) + f_m(x)$.

(c) Update $w_i \leftarrow w_i \exp[-y_i f_m(x_i)]$, and renormalize.

3. Output the classifier $sign[F(x)] = \text{sign}[\Sigma^M_{m=1} f_m(x)]$.

Multi-class classification using boosting algorithms was traditionally implemented by breaking a single problem down into binary classifications of many problems. Then final class selection was then made using comparisons among the selections of all the different binary classifiers.

This could be done using three techniques:

1) One-versus-all appraoch. [24] This approach takes a single class as the base class which each of the other classes is paired up against to form a binary problem. After all the binary problems have made predictions, the prediction with the highest score is chosen.

2) All-versus-all appraoch. [24] In this case, given N classes, N(N-1) classifiers are built, with one classifier for each combination of binary pairs that the problem can be decomposed into. Note that classifiers need to be trained to distinguish the object they are built to classify, separately from objects not of that class. Therefore they are generally trained on sets of positive samples of data, and negative samples. In the case of a car-classifier, positive samples would comprise data or images that represent cars, while negative samples might comprise data related to bicycles, people, or animals. In the All-versus-All approach, if f_{ij} is taken as the classifier where class i consists of positive examples and class j samples are negative, then the final classified result is:

$f(x) = \text{argmax}_i(\Sigma i\ f_{ij}(x))$

3) Error-correcting codes. [25] This approach looks at the task as a decoding problem, where the correct output class is transmitted over a channel. A matrix representing the true prediction of each for each binary classifier is used as a reference of codewords against which the true class of the problem is then decoded.

Among the notable boosting algorithms for multiclass classification are:

1) Adaboost.MH. [26] – This is an implementation of the One-versus-All approach among several binary classifiers.

2) SAMME. [27] – This too is based on the original AdaBoost algorithm. However, it improves on Adaboost.MH by generically extending the algorithm to a multiclass problem without breaking down into binary problems.

3) GAMBLE [28] – "Gentle Adaptive Multiclass Boosting Learning". In the same way that SAMME generalizes AdaBoost.MH to the multiclass problem, GAMBLE is the generalization of GentleBoost.MH. It uses Quasi-Newton smoothing on the loss function.

4) GentleBoost.C. – This is also a natural multiclass extension to GentleBoost, but offers an improvement over GAMBLE because of the introduction of a new, smooth loss function, C-loss, which also incorporates conditional class probabilities. [29] Because of its greater robustness and insensitivity to noise, we use Gentleboost as our boosting framework, and in particular we implement Gentleboost.C.

B. Problem Formulation

We start by defining some important terms used in the rest of this paper:

- *Class:* The type of vehicle Car or Big Vehicle, such as a bus or truck.
- *View*: The view/perspective of vehicle. The total possible views explored in this paper are presented in Table I.
- *Environment:* The physical environment of the vehicle, i.e. "city" or "desert/mountain".
- *HoG:* "Histogram of Oriented Gradients" (HoGs); these are the features that our system uses to represent the vehicles, based on the changes in color intensity in the image. The HoG features of an image are computed by first dividing the image into equal blocks, and then computing the orientation of the gradients in each. This shows how color levels change in different locations within the image. The information from each block is then concatenated to form a feature vector of oriented gradients. HoG descriptors were introduced by Dalal and Triggs [30] for the detection of humans in images, and have since become one of the standard and oft-used features for object detection and classification.

From [29], we model our problem as a multiclass extension to the binary GentleBoost algorithm.

Let training data X consist of $x_{i...}x_n$ observations, where n is the number of training data samples. X represents the feature vectors of the observations. While any set of features could be used, in our implementation, we use HoG features. Each observation, xi, is provided with its response *y*, indicating its true class, which is a combination of what kind of vehicle it is, and from which view is it being seen. Two possible examples of what a true class might represent are:

- (vehicle: car, azimuth: 000, angle of inclination: 00)
- (vehicle: bus, azimuth: 045, angle of inclination: 30)

Let *m* be the total number of classes that the data can be classified into.

The multi-class classification task is modeled as a combination of linear, weighted regression problems, where each class represents one regression function. The regression parameters represent the features in each observation. The

regression weights are calculated using a multiclass C-loss, which is a smooth coherence loss function described in [31]. C-loss is superior to regular hinge loss or logit loss not only because of its statistically desirable properties but because it encapsulates conditional class probabilities.

In the spirit of boosting algorithms, the regression parameters are fine-tuned over a number of boosting rounds. Each round generates a weak classifier, which additively influences the classifiers of the next round, until we have completed H boosting rounds of our choice, to arrive at the final strong classifier.

The algorithm, GentleBoost.C, is listed at the end of Section V.

We implement GentleBoost.C for each of the two classification approaches that we adopt, explained below.

C. Two Approaches

Based on previous work by Viola and Jones [5], each view will require its own separate classifier. Training a single classifier with samples of all views of one object is likely to result in poor recognition [32].

Given our problem, we would like to see which level of detail is required to distinguish between views for accurate vehicle detection. We start with 25 views of the vehicle, and reduce the number of views until we reach the number that produces optimal results. We refer to this number as V.

The problem of complete detection can then be approached in two ways, illustrated in Fig. 1 and described below:

- Combine |C| vehicle class classifiers. Given an image, the system runs separate classifiers to identify the class of a vehicle, $c \in C$, which independently vote on the view of the assumed class. Take for instance the case of C = {car, bus, truck}, (so that |C| = 3) and V=25, so that there are 25 views per vehicle class. First the 25-view car classifier will generate confidence scores for each view given an image, followed by the bus classifier and then the truck classifier. With each classifier providing its own confidence measure of the possible view of the given vehicle, we normalize the scores from each classifier in order to make a final decision based on all the scores combined, and we select the view and class with the highest score.

- Build a single V × |C| multi-class classifier. In this case, a single V × |C|-class classifier is used to classify objects in a single step. So in the case of 3 vehicle classes and 25 views, this classifier would be built to distinguish between 25×3 = 75 possibilities, each possibility being a combination of vehicle class and view, plus one more possibility: not-a-vehicle. This particular case would therefore call for a 76-class classifier.

A third approach was considered, which first classifies the view of an image given |V| views, and then determines which of the |C| classes it belongs to. This approach was found not to be a viable option based on the HoG-based method employed (Section *D: Method*), which necessitates that objects of interest across different images must be somewhat similar in size and position, relative to each image's center. Because large vehicles such as trucks and buses occupy images very differently from cars, then a classifier trained on images of cars *and* trucks would perform poorly, despite all vehicles being of the same view.

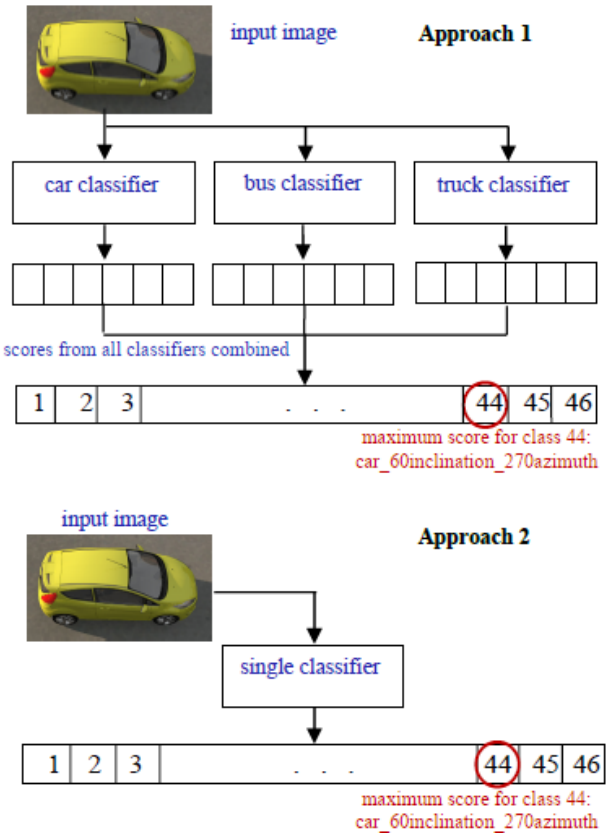


Fig. 1. Two Approaches to Multi-Class Vehicle Classification.

V. Contribution

A. Experimental Setup

Our training data consisted of equally sized images of cars, buses and trucks, in different environments, in all their different views. The testing dataset comprised of similar data not present in the training samples. (Details are presented in Section *C: Data Used*).

In the first phase of our work, we determined the number V, i.e. how many views would be optimal for the classifier. Different sets of views were proposed (such as all 25 views of a vehicle, or only 11 views, and so on). These were proposed based on eyeballing the similarity among different sets of the 25 views: in spite of different azimuths and angles of inclination, some different views appeared somewhat similar and could potentially be collapsed into a single view. A randomly chosen subset of the training data was used to train a different classifier for each set of views proposed.

The accuracy for each classifier was recorded in order to determine which set of views yielded the best results. This final number of views was then used to train the full system in both of its approaches.

The full system was then evaluated against the test dataset (details in Section *C: Data Used*), and its results were compared against a baseline system, built using AdaBoost. This baseline and the results of the comparison are discussed further in Section VI: Results.

B. Tools Used

Given that training data would be difficult to obtain from the real world, owing to our specific requirements of vehicle models and classes, we opted to create 3D simulations. Sketchup, a 3D modeling software from Trimble, was used to create several models of vehicles in city and desert environments. Vehicle models were obtained either from the 3D Sketchup Warehouse or the Podium Browser. Sketchup was extended using SU Podium to render photorealistic images of the vehicle models.

The multiclass classification algorithm was written in Python, using the following libraries: Numpy (for all array manipulation), statsmodels (for weighted least squares regression), scikit-image (for extracting HoG features), and scikit- learn and OpenCV (for some utility functions).

C. Data Used

As mentioned above in Section *B: Tools Used*, photorealistic 3D images were generated for our experiments. However, any dataset of real-world images could be used as well, if it covered as comprehensive a range of views, vehicles models and classes, as we propose in this paper. The raw images simulated for this paper were 1300×600 pixels each, of single vehicles. Vehicle models were randomly assigned three possible backgrounds: City1, City2, Desert1. There were initially three classes of vehicles (car, bus, truck), nine models per class (such as Honda Civic or Ford Fiesta for car models), and 25 views per model.

The 25 views are shown in Table I, where each image has its azimuth labeled below it.

This produced 675 images of vehicles. In addition, some images with vehicles that had low contrast against the background were duplicated on monochrome or transparent backgrounds. With duplicates included, the total of raw vehicle images for training was 707.

326 negative samples were created from various city streets and desert scenes taken from the internet.

Once obtained, all samples were sheared and rotated to create further samples in order to simulate more data. This resulted in approximately 1000 samples per view. Table II below lists the complete training data used.

For testing data, an additional car model was simulated through the seen 25 views to create 25 raw images, and was further sheared and rotated to form a total of 90 images. For trucks, buses and negative samples, however, some images were simulated in unseen views from existing models and some were taken from the internet, due to time limitations. This combination of simulated and real-world images proved important: we note, in Section VI, that there is a difference in the classification results on simulated images as opposed to the results on a combination of images that are simulated or taken from the internet. This difference gives us an insight into the performance of our system on testing data that is similar to the training data versus data that is not.

In total, 90 test images were produced for each vehicle class, resulting in a training to testing ratio of 1: 0.3 in terms of raw images.

Table III lists the number of unseen samples of cars, buses, trucks, and negatives in the test dataset.

TABLE I. 25 VEHICLE VIEWS: ANGLES OF INCLINATIONS AND AZIMUTHS

Inclination: 00  000	 045	 090	 135	 180	 225	 270	 315
Inclination: 30  000	 045	 090	 135	 180	 225	 270	 315
Inclination:60  000	 045	 090	 135	 180	 225	 270	 315
Inclination: 90  000							

TABLE II. TRAINING DATASET

	Class	Raw Images	Processed Totals
Positives	***Car***	243	25277
	Bus	239	24877
	Truck	225	23466
Negatives	***City/Desert***	326	979

TABLE III. TESTING DATASET

Cars	Buses	Trucks	Negatives
90	90	90	48

D. Method

The classification depends on extracting HoG features, which may produce feature vectors of different sizes depending on different HoG configurations, such as number of blocks that images were divided into, number of orientation bins, and so on. For calculation purposes, it was necessary to ensure that each feature vector extracted from an image was the same size. Hence, all training samples were cropped to an aspect ratio of 2:1 and were resized to 100×50 pixels. As suggested by Felzenszwalb and et al. [2], HoGs of 9 orientation bins, 8×8 pixels per cell with one cell per block were extracted. This generated a total of 648 features per image.

In order to build our classifiers, it was necessary to determine V, the optimal number of views to model. For this, a subset of the training data was used, with 2187 samples for cars, 2025 for trucks, 2151 for buses, and 978 negative samples.

A number of sets of views were proposed, shown in Table IV, and the classification accuracy on the 90-image test dataset for cars was recorded for each. The results, in Table V, showed that the selection of 15 views produced the highest accuracy.

The classification accuracy of trucks did not change, but the improvement for buses implied that 15 views was in fact a suitable choice. Hence, V was set to 15. Views were labeled from 0 to 15 in the order matching the angles and azimuths shown above in Section *C: Data Used*.

To ensure that this number improved accuracy across different types of vehicles, a subset of buses and trucks were also trained and classified on Sets 1 and 2.

Having concluded that the optimal number of views, V, was 15, three 15-view classifiers were trained, and one for each class of vehicle (car, bus, truck). This was for Approach 1, where separate classifiers were trained and then their combined scores compared. Each of these classifiers comprised 16 classes: 1 class to represent not-a-vehicle, and 15 to represent each of the different views of a vehicle.

For Approach 2, a single, 46-view classifier was trained. Again, one class was left for not-a-vehicle, and 45 classes were used for each of the different vehicles and their views. Table VI lists the results.

Classification accuracy refers to the classification of view (angle and azimuth). The Vehicle recognition accuracy refers to the classifier's ability to recognize the class of the vehicle, (i.e. that it was a car). The confusion matrices are in Table VII.

These results showed a clear trend in the accuracy of the classifiers. The cars' classifier performed best, followed by the buses' classifier, while the trucks' classifier had the poorest performance.

An analysis of the confusions revealed that the views of buses, to a certain extent, and trucks to a larger extent, were difficult to distinguish when the vehicles stood pointing to the left or to the right. Both classifiers had trouble in distinguishing the front of the big vehicle from its back.

The full test dataset was used to test Approach 1 (of separate classifiers) and compare it with Approach 2 (of a single classifier).

TABLE IV. CLASSIFICATION ACCURACY FOR DIFFERENT SETS OF VIEWS OF CARS

Views	Details	Accuracy
All 25 Views	None	91.1%
15 Views	0 degree inclination views 045, 090, 135 were collapsed into one view, "left". Corresponding views for the "right" direction were collapsed. Additionally, no distinction was made between angles of inclinations 30 and 60 for views 45, 90, 135, 225, 270 and 315.	97.3%
13 Views (1)	0 degree inclination views 045, 090, 135 were collapsed into one view, "left". Corresponding views for the "right" direction were collapsed.	93.3%
13 Views (2)	No distinction was made in a single view between angles of inclination 30 and 60. But for view 045, angles of inclination at 30 and 60 were collapsed into one view, and so on.	96.4%
11 Views	Views 045, 090, 135 were collapsed into one view, "left" for each angle of inclination (0, 30 and 60). Corresponding views for the "right" direction were collapsed.	95.6%

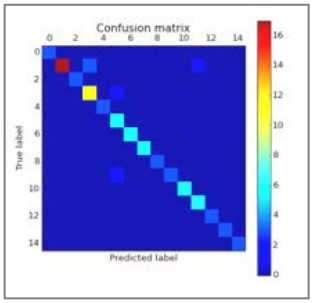
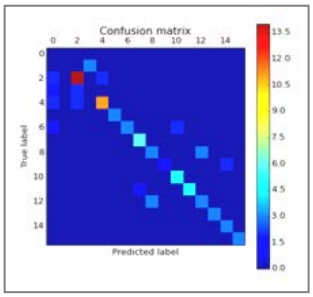
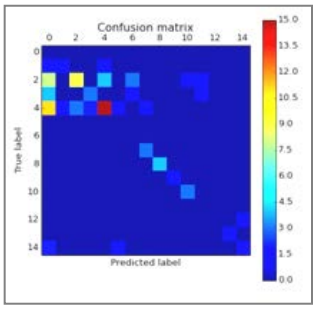
TABLE V. CLASSIFICATION ACCURACY FOR DIFFERENT SETS OF VIEWS OF BUSES AND TRUCKS

	Buses	Trucks
25 views	68.8%	80.3%
15 views	82.7%	80.3%

TABLE VI. PERCENTAGE ACCURACY OF INDIVIDUAL, 15-VIEW CLASSIFIERS

	View Classification Accuracy	Vehicle Recognition Accuracy
Cars classifier	92.2	100.0
Buses classifier	71.1	93.3
Trucks classifier	46.7	71.1

TABLE VII. CONFUSION MATRICES OF INDIVIDUAL, 15-VIEW

	Cars model (on cars' data)
	Buses model (on buses' data)
	Trucks model (on trucks' data)

For Approach 1, each classifier was given a vehicle image for which it generated confidence scores per view. Min-max normalization was then used to allow these scores to be compared across classifiers, and the highest score was selected to represent the final chosen class. The accuracy of this combined model was then compared with that of a single, 46-class classifier.

Tables VIII and IX compares the results of each approach, followed by the confusion matrix of the 46-class classifier in Fig. 2.

TABLE VIII. VEHICLE RECONGITION ACCURACY

	Cars	Buses	Trucks
46-class model	93.3	71.1	40.0
Combined models	85.6	42.2	42.2

TABLE IX. VIEW CLASSIFICATION ACCURACY. N-V REPRESENTS THE CLASS NOT-A-VEHICLE

	Cars	Buses	Trucks	N-V	Overall
46-class model	91.1	55.6	24.4	64.4	58.2
Combined models	81.1	34.4	66.7	68.8	51.6

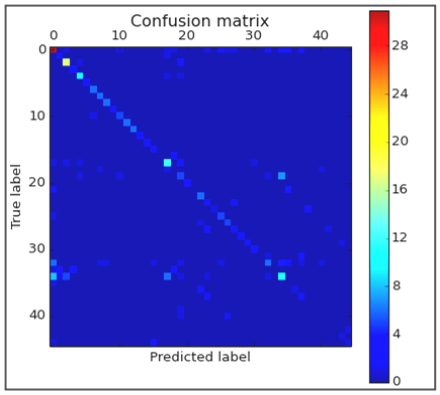


Fig. 2. 46-Class Model (on Full Test Data).

The accuracy drop of the classifiers when dealing with big vehicles was not only because of the trouble in distinguishing the front of a truck or bus from its back, but it was found that 33% of the errors in identifying trucks was caused by a confusion with identifying trucks as buses, and that 44% of the errors in identifying buses was caused by the converse. Examples of the confusions are shown in Fig. 3 and 4.

The greatest number of common confusions for both buses and trucks were in distinguishing whether they were facing towards the left or the right, i.e. at azimuths of 90 or 270.

Fig. 3. Confusion between Trucks in Similar Positions based on Difficulty in Distinguishing the Front of the Vehicle from the Back.

Fig. 4. Confusion between Trucks and Buses in Similar Positions.

Given the above results, buses and trucks would best be compressed into one class, Big Vehicles. Therefore the training data for buses and trucks was re-processed to produce the data displayed in Table X. The individual Cars' classifier retained 15 views, which were labeled from 1 to 15, with the 0 label used to refer to not-a-vehicle.

However, for the big vehicles, the views of vehicles standing facing left and right were compressed into one view. Therefore the Big Vehicles classifier was trained on 14 views, and one class for negatives.

For Approach 2, the single classifier was trained on 29 views. In this classifier, view 0 represented not-a-vehicle; views 1 to 15 represented the different views of a car, starting from a zero-angle of inclination and zero-azimuth; likewise, views 16 to 29 represented the views of a big vehicle. All the classifiers were tested on the same test data as before. The results are shown in Section VI.

TABLE X. UPDATED TRAINING SET

	Class	Raw Images	Processed Totals
Positives	Car	243	25277
	Big Vehicles	232	24177
Negatives	City/Desert	326	979

E. Limitations

A limitation of the method chosen is based on the selection of HoG features, which are dependent on image dimensions. Therefore, the results are most reliable when the testing data consists of vehicles of a similar size and position relative the center as those present in the training data.

A strength of the system is the use of GentleBoost as opposed to the more oft-used AdaBoost. This is because GentleBoost is not as easily affected by outliers.

VI. RESULTS AND DISCUSSION

The results of our system were compared against a baseline built using AdaBoost, in particular the SAMME.R version [27]. AdaBoost was chosen as a suitable comparison with our method because of its popularity in the object and vehicle detection fields [20]. The SAMME.R version is a real number-based multiclass classifier that, like our choice of GentleBoost.C, is based on cascaded boosting and does not break the classification problem into binary decisions. SAMME.R also generates sound confidence scores for each class during classification, which was useful when comparing the main system against Approach 1.

A. Individual Vehicle Models: Cars, Big Vehicles

Table XI compares the view classification and vehicle recognition accuracy of the individual car and big-vehicle models, which were trained on images of cars and big vehicles respectively. Except in the vehicle recognition of big vehicles, the GentleBoost version has a higher accuracy than the Adaboost baseline.

Tables XII to XIV compare the vehicle recognition and view classification accuracies of Approach 1 (combined classifiers) and Approach 2 (single classifier) respectively. Once again, the Gentleboost system outperforms the baseline. While the performance of the two approaches is somewhat similar with respect to cars, the single classifier was accurate 72.2% of the time and outperformed the first approach significantly in the case of big vehicles. This difference may be attributed to the difference between testing data for big vehicles and that for cars. Recall that part of the big vehicles' testing data was taken from photographs on the internet, unlike the cars' data, which was generated entirely using a 3D simulator. The results suggest that the single classifier is well-suited to situations where testing data includes samples that are significantly different from those in the training data, in turn suggesting a wider application than what the combined-classifiers model is capable of Tables XIII and XIV show that all classifiers performed less accurately at View Classification.

The GentleBoost single classifier of Approach 2 performs best on cars, and the baseline system, albeit with lower numbers, had a similar trend.

The Gentleboost combined classifiers of Approach 1 performed best on cars again, and lowest on big vehicles. However, the combined baseline models do not maintain the same trend as the Gentleboost classifiers.

Table XV presents the precision of recall of both two approaches. Approach 2 (the single classifier approach) yields 0.92, an improvement over Approach 1.

Overall, Approach 2 using Gentleboost outperformed other classifiers in all experiments.

TABLE XI. PERFORMANCE OF INDIVIDUAL VEHICLE MODELS (%)

	Vehicle classification		Vehicle Recognition	
	GentleBoost	**Baseline**	**GentleBoost**	**Baseline**
Cars	**91.1**	46.7	**98.9**	96.7
Big V	**72.2**	58.3	89.4	**94.4**

TABLE XII. VEHICLE RECOGNITION ACCURACY: SINGLE VERSUS COMBINED MODELS (%)

	Cars		**Big Vehicles**	
	GentleBoost	***Baseline***	***GentleBoost***	***Baseline***
29-class model	**95.6**	73.3	**72.2**	48.9
Combined models	**91.1**	41.1	**55.0**	46.1

TABLE XIII. VIEW CLASSIFICATION ACCURACY (%): SINGLE 29-CLASS MODEL

	GentleBoost	**Baseline**
Cars	**91.1**	44.4
Big Vehicles	**61.1**	13.3
N-V	**75.0**	29.2
Overall	**71.7**	24.5

TABLE XIV. VIEW CLASSIFICATION ACCURACY (%): COMBINED MODELS

	GentleBoost	**Baseline**
Cars	**85.6**	21.1
Big Vehicles	**48.3**	26.7
N-V	**70.8**	83.3
Overall	**62.3**	22.3

TABLE XV. PRECISION/RECALL

	Precision		Recall	
	GentleBoost	***Baseline***	***GentleBoost***	***Baseline***
29-class model	**0.96**	0.89	**0.91**	0.85
Combined models	**0.96**	0.86	0.93	**0.99**

However, the big vehicles did not fare as well. Big vehicles were likely misclassified when seen in the initial views listed in Table I, starting with an inclination angle of 0 and azimuth of 000. Over 10% of the errors, it was found, were confusions between the side views of cars versus of big vehicles, and likewise with front views. This suggests a trade-off between accurate classification of views and of vehicles.

Although many systems explore vehicle detection in spite of occlusions, or from aerial views, and so on, at the time of writing, we do not know of other classification systems which recognize vehicles irrespective of view as well as vehicle class. No immediate comparisons could be made with the current state-of-the-art, since current systems often use datasets such as KITTI, Caltech, Pascal, or Toyota, which lack a comprehensive range of views.

VII. Conclusion

This paper explored the development of a vehicle classifier that can distinguish vehicles regardless of class or view. The classifier was built using a multiclass GentleBoost boosting algorithm trained on 648-length arrays of image HoG features.

While 25 different views of vehicles were initially suggested, so many views were found unnecessary for accurate classification, and in fact likely to reduce accuracy. Therefore an optimal choice of 14-15 views was selected for training.

Another system was built with the same data and choice of views, but with independent classifiers that focused on each type of vehicle. The classifiers' votes were combined to choose the most likely class and view of a test vehicle.

The results showed a single classifier trained over many classes performing significantly better than the results from a combination of individual classifiers trained over subsets of all the training data. The single classifier's performance also showed that this is a better choice in the event that testing data consists of environments and views that are very different from that of the training data. This is because the testing data for big vehicles was different from its training data, and the improvement of performance of the single classifier over the combined classifiers was most pronounced over big vehicles.

The results also showed that large vehicles are more likely to be confused amongst each other than are cars, probably due to the dilution of dissimilar components by similar components.

The experiments in this paper conclude that, without using complex 3D models, a simple multiclass classifier can detect with high precision, various types of vehicles across different environments, and different views of the vehicle, including the top, aerial view.

Acknowledgment

Thanks to Aurelian Tutuianu for his help in the GentleBoost.C implementation.

References

[1] Savarese, S., Li Fei-Fei, "3d generic object categorization, localization and pose estimation", *IEEE International Conference on Computer Vision*, 2007.

[2] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., Ramanan, Deva, "Object Detection with Discriminatively Trained Part-Based Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[3] Clady, X, Negri, P., Milgram, M. and Poulenard, R., "Multi-class Vehicle Type Recognition System", *Proceedings of 3rd IAPR workshop on Artificial Neural Networks in Pattern Recognition*, 2008.

[4] Razavi, N., Gall, J., Van Gool, L., "Scalable Multi-class Object Detection", *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[5] Viola, P., Jones, M. J., "Robust Real-time Object Detection", *International Journal of Computer Vision*, 2001.

[6] Schapire, R. E., "A Brief Introduction to Boosting", Proceedings of the 16th International Joint Conference on Artificial Intelligence, Vol. 2, 1999.

[7] Hastie T., Tibshirani R., Friedman J, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, New York, 2009.

[8] Jazayeri, A, Cai, H., Zheng, J. Y., Tuceryan, M., "Vehicle detection and tracking in car video based on motion model", *IEEE Transactions On Intelligent Transportation Systems*, Vol. 12, No. 2., 2011.

[9] Alonso, J. D., Vidal, E. R., Rotter, A., Muhlenberg, M., "Lane change decision aid system based on motion-driven vehicle tracking", *IEEE Transactions On Vehicular Technology*, Vol. 57, No. 5, 2008.

[10] Thuy Thi Nguyen, Grabner, H., Bischof, H., Gruber, B., "On-line Boosting for Car Detection from Aerial Images", *Proceedings of IEEE International Conference on Research, Innovation and Vision for the Future*, 2007.

[11] Behley, J., Steinhage, V., Cremers, A. B.: "Laser-based Segment Classification Using a Mixture of Bag-of-Words", *International Conference on Intelligent Robots and Systems*, 2013.

[12] Ye Li, Bin Tian, Bo Li, Gang Xiong, Fenghua Zhu, Kunfeng Wang, "Vehicle detection with a part-based model for complex traffic conditions", *IEEE International Conference on Vehicular Electronics and Safety*, 2013.

[13] Zehang Sun, George Bebis, Ronald Miller, "Monocular precrash vehicle detection: Features and classifiers", *IEEE Transactions On Image Processing*, Vol. 15, No. 7, 2006.

[14] Bin-Feng Lin et al, "Integrating appearance and edge features for sedan vehicle detection in the blind-spot area", *IEEE Transactions on Intelligent Transportation Systems*, 2012.

[15] Chi-Chen Raxle Wang, Lien, J.-J.J, "Automatic vehicle detection using local features: A statistical approach", *IEEE Transactions on Intelligent Transportation Systems*, 2008.

[16] Chan, Y.-M., Huang, S., Fu, L., Hsiao, P., Lo, M.-F.: "Vehicle detection and tracking under various lighting conditions using a particle filter", *Intelligent Transport Systems, IET*, Vol. 6, Issue. 1, 2012.

[17] Wen-Chung Chang, Chih-Wei Cho: "Online Boosting for Vehicle Detection", *IEEE Transactions ON Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 40, No.* 3, 2010.

[18] Thuy Thi Nguyen, Grabner, H., Bischof, H., Gruber, B, "On-line Boosting for Car Detection from Aerial Images", *IEEE International Conference on Research, Innovation and Vision for the Future*, 2007.

[19] Sivaraman, S. , Trivedi, M.M., "A general active-learning framework for on-road vehicle recognition and tracking", *IEEE Transactions on Intelligent Transportation Systems*, 2010.

[20] Sivaraman, S., Trivedi, M. M., "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, Issue 4, pp 1773 – 1795, 2013.

[21] Torralba, A., Murphy, K. P., Freeman, W. T.: "Sharing visual features for multiclass and multiview object detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, Issue 5, 2007.

[22] Shalev-Shwartz, S, Wexler, Y., Shashua, A.: "Efficient Multiclass Learning with Feature Sharing", *Advances in Neural Information Processing Systems*, 2011.

[23] Friedman, J., Hastie, T., Tibshirani, R., "Additive Logistic Regression: A statistical view of boosting", *The Annals of Statics*, Vol. 28, No. 2, pp 337-407, 2000.

[24] Rifkin, R., "Multiclass Classification", http://www.mit.edu/~9.520/spring09/Classes/multiclass.pdf, 2008.

[25] Dietterich, T. G., Bakiri, G., "Solving multiclass learning problems via Error Correcting Output Codes", *Journal of Artificial Intelligence Research*, 1995.

[26] Schapire, R., Singer, Y., "Improved Boosting algorithms using confidence-rated predictions", *Machine Learning*, Vol. 37, Issue 3, pp 297-336, 1999.

[27] Zhu, J., Zou, H., Rosset, S., Hastie, T., "Multi-class AdaBoost", *Statistics and Its Interface*, Vol. 2, No. 3, 2009.

[28] Huang, J., Ertekin, S., Song, Y., Zha, H., Giles, C. L., "Efficient Multiclass Boosting Classification with Active Learning", *Seventh SIAM International Conference*, 2007.

[29] Zhang, Z., Chen, C., Dai, G., Li, W.-J., Yeung, D.-Y.-, "Multicategory Large Margin Classification Methods: Hinge Losses vs. Coherence Functions", *Artificial Intelligence*, Vol. 215, pp 55-78, 2014.

[30] Dalal, N., Triggs, B., "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[31] Zhang, Z., Liu, D., Dai, G., Jordan, M. I., "Coherence Functions with Applications in Large-Margin Classification Methods", *Journal of Machine Learning Research*, 2012.

[32] Jones, M., Viola, P., "Fast Multi-view Face Detection", *Tech. Rep. TR2003-96*, Mitsubishi Electric Research Laboratories, 2003.

[33] Zhang, Y. et al, "Research on visual vehicle detection and tracking based on deep learning", *IOP Conf. Ser.: Mater. Sci. Eng.*, 2020.

[34] J. Chen and L. Dai, "Research on Vehicle Detection and Tracking Algorithm for Intelligent Driving," *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2019.