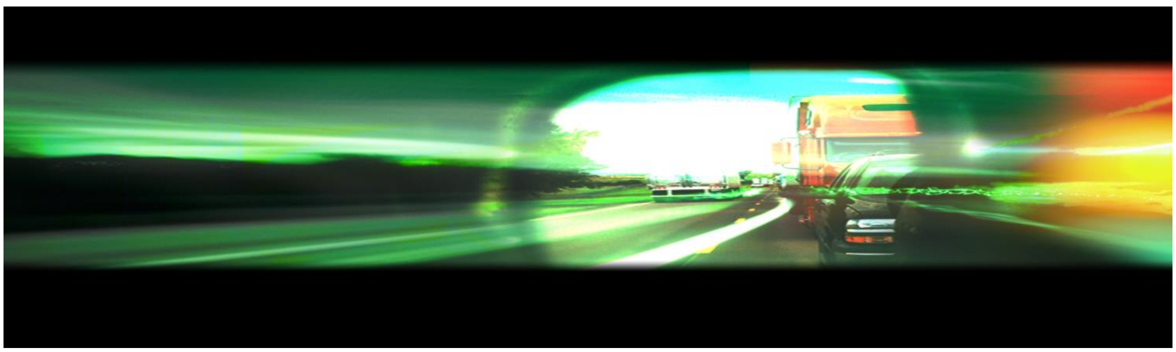
EXPLORING IMAGE-BASED CLASSIFICATION TO DETECT VEHICLE MAKE AND MODEL

FINAL REPORT



TranLIVE

Jeffrey B. Flora, Mahbubul Alam, Amr H. Yousef, and Khan M. Iftekharuddin

December 2013

DISCLAIMER

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Exploring Image-based Classification to Detect Vehicle Make and Model	5. Report Date November 2013 6. Performing Organization Code KLK900-SB-001
7. Author(s) Flora, Jeffrey; Alam, Mahbubul; Yousef, Dr. Amr; Iftekharuddin, Dr. Khan	8. Performing Organization Report No. N13-03
9. Performing Organization Name and Address Old Dominion University 4111 Monarch Way, Suite 204 Norfolk, VA 23506 Source Organization Name and Address National Institute for Advanced Transportation Technology University of Idaho 875 Perimeter Dr. MS0901 Moscow, ID 83844-0901	10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTRT12GUTC17
12. Sponsoring Agency Name and Address US Department of Transportation Research and Special Programs Administration 400 7th Street SW Washington, DC 20509-0001	13. Type of Report and Period Covered Final Report: August 2012 – November 2013 14. Sponsoring Agency Code USDOT/RSPA/DIR-1

15. Supplementary Notes:

16. Abstract

The goal of this work is to improve the understanding of the impact of carbon emissions caused by vehicular traffic on highway systems. In order to achieve this goal, this work obtains a novel pipeline for vehicle segmentation, tracking and classification by leveraging techniques in computer vision and machine learning using the existing Virginia Department of Transportation (VDOT) infrastructure on networked traffic cameras. This vehicle segmentation and classification data can be used to obtain a real-time estimation of carbon emissions. The VDOT traffic video is analyzed for vehicle detection and segmentation using an adaptive Gaussian mixture model algorithm. The segmented vehicles are tracked using speeded up reduced feature (SURF) methods. The morphological properties and histogram of oriented features are derived from the detected and segmented vehicles. Finally, vehicle classification is performed using a multiclass support vector machine classifier. The resulting classification scheme offers an average classification rate of 85% under good quality segmentation.

This work constitutes the first step in estimating carbon emission for highway traffic. In the subsequent steps, we need to extract additional vehicle information (e.g., type, model), speed, and other relevant parameters and use this information for EPA's Motor Vehicle Emissions Simulator (MOVES) or similar tool to estimate the carbon emission. We also need to work on more challenging weather conditions as well as nighttime scenarios to make this tool applicable to real life application.

17. Key Words intelligent transportation system, emissions estimation, computer vision, machine learning	18. Distribution Statement Unrestricted; Document is available to the public through the National Technical Information Service; Springfield, VT.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 22	22. Price …

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

FIGURES

TABLES

EXECUTIVE SUMMARY

In this work, a novel algorithm pipeline has been developed to successfully detect, track, segment and classify vehicles. The VDOT traffic video is analyzed for vehicle detection and segmentation using an adaptive Gaussian mixture model (AGMM) algorithm. The segmented vehicles are tracked using speeded up reduced feature (SURF) methods. The morphological properties and histogram of oriented features are derived from the detected and segmented vehicles. Finally, a multiclass support vector machine classifier is used to classify the vehicles in six different classes (e.g., sedan, passenger truck, motorcycle, bus, long-haul truck, short-haul truck). The resulting classification scheme offers an average classification rate of 85% under good quality segmentation. Our developed methods have been tested on several different video sequences collected from Hampton Roads area VDOT traffic surveillance cameras. Although current testing has only operated on offline video sequences, the performance of our algorithms allows for near real-time implementation.

The vehicle detection, segmentation and classification constitute the first step in estimating carbon emission for highway traffic. In the subsequent steps, we need to extract additional vehicle information (e.g., type, model), speed, and other relevant parameters and use this information for EPA's MOVES or similar tool to estimate the carbon emission. We also need to work on more challenging weather conditions as well as nighttime scenarios to make this tool applicable to real life application.

DESCRIPTION OF PROBLEM

The goal of this work is to develop an image-based solution for detection, tracking, and classification by vehicle type of highway traffic. An image-based technique differs from other roadway sensors, such as radar or inductive loops, which provide data only regarding traffic flow and density, and do not provide information about the type of vehicle in real-time. By leveraging advancements in computer vision algorithms and machine learning techniques, a novel algorithm to perform the detection, tracking, and classification of vehicles may be accomplished by using streaming video input from the infrastructure of networked traffic surveillance cameras that already exist in most metropolitan areas. The proposed system can lead to lower implementation costs as compared to systems requiring additional sensor technologies. The system also provides a higher-resolution, real-time understanding of the types of vehicles on the highway system, in addition to other transportation metrics such as individual vehicle speed, traffic flow and density.

The data generated by the proposed system, in the form of vehicle types, may be used to form an estimate of carbon emissions in real-time for the localized geographic area surrounding the traffic surveillance camera. This carbon emissions estimate is accomplished by the Motor Vehicle Emissions Simulator (MOVES) application as developed by the Environmental Protection Agency (EPA). Further studies of these results may explore the impact of specific vehicle types on carbon emissions and provide motivation for design decisions applied to the development of future transportation systems that may have a more positive environmental impact.

APPROACH AND METHODOLOGY

We developed a novel algorithm pipeline to successfully detect, track, segment, and classify vehicles. The pipeline consists of an AGMM scheme for vehicle segmentation and detection, histogram of oriented gradients (HOG) and morphological properties for feature extraction, and multiclass support vector machine (SVM) for classification. Although these individual processing components are not novel to the application of vision-based, intelligent transportation systems, the proposed algorithm is a novel implementation of these techniques and specifically aims to improve a similar work proposed in [1].

The algorithmic pipeline is shown in Figure 1. Each component is discussed in detail in the proceeding methods section.

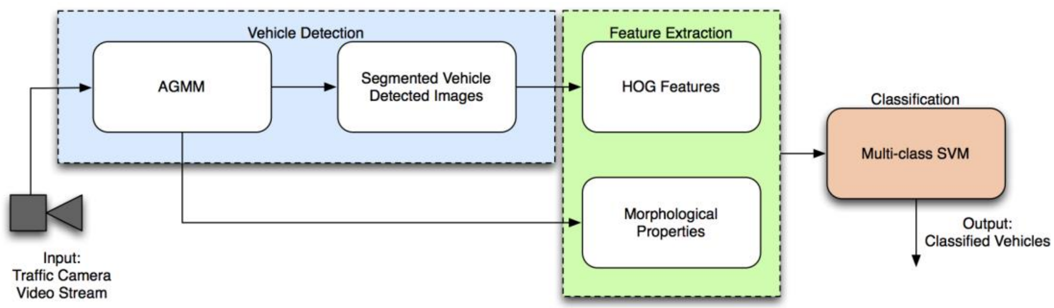


Figure 1. Algorithm pipeline

A model-based approach is used for vehicle segmentation in complex, outdoor scenes. The model chosen is the adaptive Gaussian mixture model (AGMM) [2]. The AGMM considers each pixel as a time domain process, and the pixel is modeled as,

$$\{X_1,\cdots, X_2\}=\{I(x_0,y_0) : 1\leq i\leq t\}. \tag{1}$$

where X represents the image at the i^{th} time step and I represents the pixel at location (x_0, y_0) for the i^{th} time step.

The probable values of each pixel are based on a historical representation of previous pixel values. This representation takes the form of the sum of a weighted mixture of K-Gaussian distributions. This representation is given by,

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta\,(X_t, \mu_{i,t}, \Sigma_{\mathrm{i,t}}) \tag{2}$$

where $\omega_{i,t}$ is the weighting factor for the likelihood of the Gaussian, and $\eta\,(X_t, \mu_{i,t}, \Sigma_{\mathrm{i,t}})$ is the K^{th} Gaussian distribution with parameters X_t being the current pixel value, $\mu_{i,t}$ is the mean value of the pixel, and $\Sigma_{\mathrm{i,t}}$ is the covariance of the pixel. A pixel is determined to be part of the background if the pixel is within the first b distributions such that,

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right) \tag{3}$$

where T is a threshold parameter and the weights, ω_k, are sorted in order of $\frac{\omega}{\sigma}$. This implementation of the Gaussian mixture model is considered adaptive due to the parameters being updated with each iteration of the algorithm. The update equations use two learning rate parameters that influence how fast a corrective behavior occurs. The weights are updated by,

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha\,(M_{k,t}) \tag{4}$$

where α is the first learning parameter and $M_{k,t}$ is 1 if the pixel is considered background and 0 if the pixel is considered foreground. The Gaussian parameters are updated by,

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho\, X_t \tag{5}$$

and,

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho\,(X_t - \mu_t)^T\,(X_t - \mu_t), \tag{6}$$

where ρ is the second learning rate parameter given by,

$$\rho = \alpha\eta\,(X_t|\mu_k\,,\sigma_k\,). \tag{7}$$

The segmentation results of the AGMM algorithm produce a foreground that is noisy and therefore the segmentation of a vehicle contains small holes (gaps) within the image of that vehicle. In order to compensate for this noise, a series of morphological operations are applied to the segmented image. A fill operation is applied to complete any morphological blobs that contain holes in the middle. The following step applies a sequence of morphological open operations, followed by a filter to eliminate any small areas, which are regarded as noise. A small area is regarded as having a connected area of less than 30 pixels. Finally, a second morphological fill operation is performed to correct any holes introduced by the morphological opening operations.

The final step in this algorithm is to detect the vehicles from the segmented background using connected component analysis. A standard, 8-connected labeling approach is used to connect the main segmented blobs. Due to the color intensity of vehicle windows and windshields being similar to that of the roadways, windows often appear as holes in the segmented vehicles and are not easily recovered through morphological operations. In order to cluster these segmented blobs as a single vehicle, each blob is examined to find any overlap of other adjacent blobs, within a certain padding distance that is determined by a parameter provided by the user. The result of this process improves the detection of the entire vehicle.

The vehicle classification task is achieved by using a multiclass SVM classifier as the next step to the framework. The SVM classifier is more suitable to developing a hyperplane of separation for large dimensional data [3]. This improved classification technique first performs a transform on the input data feature dimensions to a higher dimensional plane using a kernel, before performing a linear separation of the transformed data.

The SVM has been introduced as one of the most efficient learning algorithm in computer vision. While many challenging classification problems are inherently multiclass, the original SVM is only able to solve binary classification problems. Due to significant appearance variation across different vehicles, a direct solution of vehicle classification using single SVM module should be avoided. The better method is to use a combination of several binary SVM classifiers to classify vehicles.

The "one against one" and the "one against all" are the two most popular methods for multiclass SVM [4]. In this work, a "one against all" method for vehicle classification is used. In the one-vs.-all (OVA, or one-vs.-rest) method, there are k classifiers, one for each class. In the case of each classifier, a hyperplane is selected to provide the best separation between data points of that class and those data points representing the remaining classes. For example, in the classification experiments for this work, up to six classifiers are created to represent each class presented in the training dataset. The first classifier is trained to find the best classification between cars and all other classes. The second classifier is trained to find the best classification between passenger truck/SUV against all other classes. This process continues for the remainder of the represented classes. This method has been used widely in the support vector literature to solve multiclass pattern recognition problems [5].

A k-fold cross-validation process is used to verify and report the results from each classification experiment. In k-fold cross-validation, the data is randomly segmented into *k* groups. One group is used for the testing data set, and the remaining *k*-1 groups form the training data set. This process is repeated *k* times so that every permutation of the subgroups is tested.

Morphological property features have been extracted from each of the segmented vehicles to form our feature vector as input into the classifier. It is much more computationally and memory efficient to represent the images as a collection of these features and to maintain a database of features rather than images. Twelve different measurements make up the measurement feature vector as shown in Table **1**. These features are derived to define an object from an image blob [1].

Table 1. Morphological Properties

Feature Name	Description
Area	The number of pixels in the region
Bounding Box	The smallest rectangle that encloses the region
Centroid	The center-of-mass for the region
Convex Area	The number of pixels in a fitted convex hull to the region
Eccentricity	The ratio of the minor and major axes of a fitted ellipse to the region
Equivalent Diameter	The diameter of a circle that has the equivalent area of the region
Euler Number	The number of objects in the region minus the number of holes
Extent	The ratio of the region area to the region bounding box
Major Axis Length	The length of the major axis of the fitted ellipse to the region
Minor Axis Length	The length of the minor axis of the fitted ellipse to the region
Orientation	The angle formed by the x-axis and the major axis of the fitted ellipse to the region
Perimeter	The distance around the boundary of the region

The HOG feature descriptor is commonly used in object detection and classification because of its capability to characterize an object's shape and appearance. This characterization is accomplished by encoding the distribution of image intensity gradients, or edge directions [6]. It is relatively invariant to local geometric and photometric transformations. Within cell rotations and translations, it does not affect the HOG values.

The HOG features are derived using a five-step process [6]. The first step in the process is to perform color normalization. Normalization is achieved using a three-channel, power law (gamma) correction. Next, for each pixel and each of the three color channels, the gradient is applied by applying the derivative kernels, $[-1 \quad 0 \quad 1]$ and $[-1 \quad 0 \quad 1]^T$. The channel that contains the largest norm is used as the gradient for the pixel. In the third step, the gradients over a cell are binned to determine the gradient angle for each pixel. In the fourth step, rectangular HOG (R-HOG) descriptor blocks are created to extract the HOG features. R-HOG descriptor blocks use overlapping square or rectangular grids of cells. The descriptor blocks are computed over dense uniformly sampled grids and are usually overlapped. Square R-HOG blocks are used to compute *g x g* grids of *n x n* pixel cells each containing *B* orientation bins. The final step is to perform block normalization. This step is performed using the L2-norm such that

$$v \rightarrow \frac{v}{\sqrt{\| v \|_2^2 + \epsilon}} \tag{8}$$

where v is the unnormalized descriptor vector and ϵ is a small constant. First order image gradients are used to compute oriented histogram voting [6].

FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

The results of the adaptive Gaussian mixture model algorithm are presented. The algorithm output is examined using an example video under ideal conditions: lighting is sufficient without any severe shadows and vehicles are presented directly below the camera field of view. The algorithm is then evaluated by examining the detection of multiple types of vehicles using traffic surveillance cameras from the Virginia Department of Transportation (VDOT) under realistic conditions.

The video frame shown in Figure **2** presents an example frame of a video under ideal conditions. The output of the adaptive Gaussian mixture model algorithm determines the pixels in the frame that are considered to be background and subtracts these pixels. The resulting binary image presents pixels as true for foreground or false for background. The morphological operations and connected components analysis are applied to the binary image to reduce noise and join together the blobs of foreground pixels to form the detected vehicles. Figure **3** shows the output of these operations. The final step is to represent the detected vehicles by the foreground segmentation. This step is represented by forming a boundary rectangle encompassing the area of each detected foreground blob. The image and features found within this boundary rectangle form the input to the classifier for identification of the vehicle type. Figure **4** shows the result of the detected vehicles for this video frame. The results show the three vehicles, having varying color, that are closest to the camera field of view are completely detected. The vehicle at the top of the video frame that is entering the scene is partially detected.

Input video that is sourced from networked traffic surveillance cameras provides challenging conditions for the adaptive Gaussian mixture model algorithm. These challenges include a lower resolution and higher compression artifacts due to internet video streaming restrictions, as well as cameras being placed higher above the roadway and at various angles, resulting in a non-uniform size and shape of vehicle as it progresses through the frame. As a result, a region-of-interest is imposed when applying the algorithm so that detection only occurs when the vehicles are closest to the camera field of view. Frames from two networked traffic surveillance cameras, sourced from VDOT, are analyzed for the ability to detect vehicles.

The first video shows a highway scene under good conditions: traffic is light, the scene is well lighted, and the scene is without strong shadows. The first example frame, shown in Figure **5**, shows five sedans within the region-of-interest. Three of the sedans are fully detected and two of the sedans are partially detected. In the second example frame, Figure **6**, there are five vehicles within the region-of-interest: two sedans, and three passenger trucks. Two of the passenger trucks, to the right of the frame, are fully detected. However, the cluster of the sedans and the passenger truck to the left of the frame, show that in cases of occlusion, the vehicles are grouped together and presented as a single detected vehicle.

Figure 2. Input video frame — ideal conditions

Figure 3. Foreground segmented binary image — ideal conditions

The second video shows a highway scene with heavy compression artifacts, resulting in a blurry image. The first example frame, in Figure **7**, shows six vehicles within the region-of-interest: four sedans, one SUV, and one large-haul utility truck. All vehicles are fully detected. Figure **8** shows four vehicles within the region-of-interest: two SUVs, one utility van, and one motorcycle. The resulting vehicle detection shows that the SUVs and motorcycle are fully detected, and the utility van is contained within two partial detections. The final example frame, Figure **9**, shows three sedans and two short-haul utility trucks. The leftmost sedans and utility truck are fully detected. However, the sedan and utility truck furthest from the camera are grouped together as single vehicle detection. Improvements to the segmentation and detection algorithms are in development. These improvements attempt to classify the detected vehicle blobs as containing only one vehicle or more than one vehicle by analyzing the morphological properties of the detected blobs.

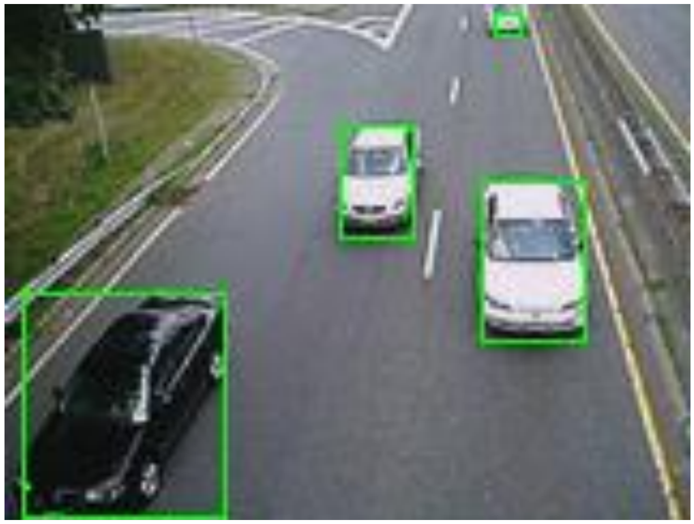


Figure 4. Detected vehicles — ideal conditions

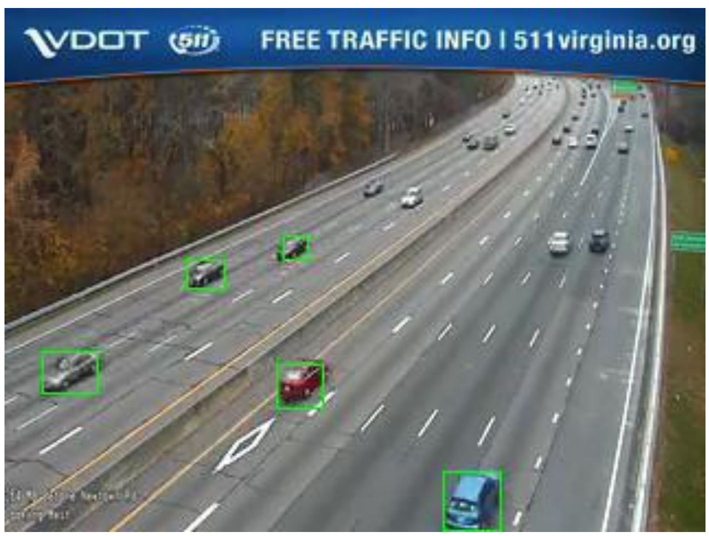


Figure 5. Detected vehicles under good conditions — realistic video 1, example frame 1

The segments that are considered to contain more than one vehicle are further segmented by tracking the SURF features of the detected blob in successive frames, in order to divide and group each vehicle in the detected blob. This enhancement will improve the performance of the vehicle segmentation under conditions of heavier traffic flow when vehicles are in closer proximity.

Figure 6. Detected vehicles under good conditions — realistic video 1, example frame 2

Figure 7. Detected vehicles under heavy compression — realistic video 2, example frame 1

Figure 8. Detected vehicles under heavy compression — realistic video 2, example frame 2

Figure 9. Detected vehicles under heavy compression — realistic video 2, example frame 3

Six different vehicle types are considered for multiclass classification, which were derived from the vehicle types available for emissions estimation from the MOVES application [7]—

i. Car
ii. Passenger truck/SUV
iii. Short-haul truck
iv. Long-haul truck
v. Bus
vi. Motorcycle.

Datasets are generated by manually evaluating the outcome of the segmentation algorithm and assigning a class to each segmented vehicle. A quality label is also applied to each segmented vehicle such that segmentation is considered —

i. Good – If 90% of the vehicle is visible
ii. Partial – If 70-90% of the vehicle is visible
iii. Multiple – If there are multiple vehicles in a single segmented image
iv. Bad – If the segmented image is not identifiable at all.

For the first trial, only good quality images are considered.

In the first dataset there are 773 segmented vehicles, of which 228 results are of good quality images, representing four vehicle classes – car, passenger truck/SUV, short-haul truck, and long-haul truck. The selection of training and testing data is performed manually, using 114 segmentation samples for the training dataset and the remaining 114 segmentation samples for the testing dataset. Table **2** shows the classification result after five-fold cross-validation.

Table 2. Five-fold Cross-validation Result of the First Dataset

	Car	SUV	Short-Haul Truck	Long haul Truck
Car	17	2	0	0
SUV	15	35	0	0
Short-Haul Truck	0	2	10	0
Long-Haul Truck	0	0	4	29

The average accuracy for each class of the first dataset is shown in Table **3**.

Table 3. Average Classification Accuracy for First Dataset

Class	Average Classification Accuracy
Car	91.05%
SUV	69.62%
Short-Haul Truck	84.36%
Long-Haul Truck	87.37%

For the second dataset, there are 3241 segmentation results representing five vehicle classes – car, passenger truck/SUV, short-haul truck, long-haul truck, and bus. There are 490 good quality images. The dataset is manually divided with 245 segmentation samples forming the training data and the remaining 245 samples forming the testing data. Table **4** shows the classification result after ten-fold cross-validation.

Table 4. Ten-fold Cross-validation Result of the Second Dataset

	Car	SUV	Short-Haul Truck	Long-Haul Truck	Bus
Car	31	2	0	0	0
SUV	38	66	2	0	2
Short-Haul Truck	1	1	48	0	4
Long-Haul Truck	1	0	21	18	0
Bus	0	1	1	0	7

The average accuracy for each class of the second dataset is shown in Table **5**.

Table 5. Average Classification Accuracy of Second Dataset

Class	Average Classification Accuracy
Car	93.83%
SUV	60.66%
Short-Haul Truck	87.28%
Long-Haul Truck	45.90%
Bus	78.66%

A second training and testing set is formed from these segmentation results where 300 segmentation samples are selected for the training dataset and the remaining 190 samples form the testing dataset. The ten-fold cross-validation result is shown in Table **6**.

Table 6. Ten-fold Cross-validation Result of the Second Dataset (with larger training set)

	Car	SUV	Short-Haul Truck	Long-Haul Truck	Bus
Car	24	2	0	0	0
SUV	28	51	2	0	3
Short-Haul Truck	1	1	36	1	2
Long-Haul Truck	1	0	17	16	0
Bus	0	1	0	0	5

The average classification accuracy for the second dataset using the larger training set is given in Table **7**.

Table 7. Average Classification Accuracy of Second Dataset (with larger training set)

Class	Average Classification Accuracy
Car	92.36%
SUV	61.07%
Short-Haul Truck	87.96%
Long-Haul Truck	47.40%
Bus	80.60%

From the cross-validation results, it is noted that the classifier confuses the car and passenger truck/SUV classes, as well as the short-haul and long-haul truck classes. The original feature set of morphological properties is extended by features derived from a histogram of oriented gradients (HOG) algorithm in order to improve these results and reduce the confusion of these classes. The second classification dataset, containing 245 results as training data and 245 results as testing data, is considered for the ten-fold cross-validation. The classification result is shown in Table **8**.

Table 8. Ten-fold Cross-validation Result of the Second Dataset with HOG Feature

	Car	SUV	Short-Haul Truck	Long-Haul Truck	Bus
Car	30	3	0	0	1
SUV	24	81	1	0	5
Short-Haul Truck	0	2	44	0	4
Long-Haul Truck	1	0	5	37	0
Bus	0	0	0	0	8

The average accuracy for each individual class is given below in Table **9**.

Table 9. Average Classification Accuracy of Second Dataset (with HOG features)

Class	Average Classification Accuracy
Car	87.01%
SUV	73.44%
Short-Haul Truck	88.43%
Long-Haul Truck	87.11%
Bus	93.78%

Better results are achieved after adding HOG feature, particularly for the passenger truck/SUV and long-haul truck classes. A comparison of classification results with and without HOG features is shown in Table 10.

Table 10. Classification Result Comparison

	Without HOG	With HOG
Car	92.3624%	87.0098%
SUV	61.0739%	73.4353%
Short-Haul Truck	87.9569%	88.4263%
Long-Haul Truck	47.3983%	87.1069%
Bus	80.6032%	93.7778%

Note that the video samples under consideration for these classification experiments did not contain motorcycles, as these vehicles are less common in highway traffic. Future works will attempt to incorporate motorcycles into the classification. Additionally, further classification testing will be performed on segmentation samples of lesser quality.

In conclusion, our novel algorithm pipeline consisting of AGMM scheme for vehicle detection, SURF processing for tracking, HOG and morphological feature for segmentation, and multiclass SVM for classification has demonstrated an average classification rate of 85%. This result is suitable for the work of carbon emissions estimation. The computational complexity of the proposed algorithm allows for near real-time implementation. The data derived from this experiment may be suitable in aiding the development of future intelligent transportation systems.

There are some areas where future works may improve the results of the experiment. Notably, the resolution limitations of the network traffic camera stream, as well as the distance of the cameras from the vehicles, prevent identification of a vehicle by make and model. Further experiments may employ higher resolution cameras that are located closer to the roadway. Additional work may further study the results of the proposed algorithm in cases of inclement weather and under low light or nighttime.

REFERENCES

[1] B. Morris and M. Trivedi, “Robust Classification and Tracking of Vehicles in Traffic Video Streams,” Proceedings of the IEEE ITSC 2006, IEEE Intelligent Transportation Systems Conference Toronto, Canada, September 17-20, 2006, pp. 1078-1083.

[2] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” presented at the Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on, 1999, vol. 2.

[3] K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, “Predicting Time Series with Support Vector Machines,” *ICANN'97*, pp. 999–1004, Nov. 1997.

[4] Z. Qian, H. Shi , J. Yang and L. Duan, “Video-based multiclass vehicle detection and tracking,” IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No. 3, January 2013.

[5] Z. Chen, N. Pears, M. Freeman and J. Austin, “Road Vehicle Classification using Support Vector Machines,” Intelligent Computing and Intelligent Systems, 2009, ICIS 2009, IEEE International Conference, Volume: 4.

[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” presented at the Computer Vision and Pattern Recognition, 2005, CVPR 2005, IEEE Computer Society Conference, 2005, vol. 1, pp. 886–893.

[7] United States Environmental Protection Agency. “MOVES (Motor Vehicle Emissions Simulator).” [online] Updated: December 13, 2013. Accessed: December 16, 2013. Available at: http://www.epa.gov/otaq/models/moves/index.htm

APPENDIX

Publications resulting from this grant are as follows:

[1] A. H. Yousef, J. B. Flora, K. M. Iftekharuddin, "Highway traffic segmentation using super-resolution and Gaussian mixture model," Proceedings of SPIE Vol. 8855, 88550G (2013).

[2] J. B. Flora, “Image-based vehicle classification for real-time emissions estimation,” M.S. thesis, Under Preparation, Dept. Elect. & Comp. Eng., Old Dominion Univ., Norfolk, VA, Spring 2014.

[3] A. H. Yousef, J. B. Flora, and K. M. Iftekharuddin, “Improved highway traffic segmentation in inclement weather conditions using super-resolution,” Journal paper, Under Preparation.

[4] J. B. Flora, and K. M. Iftekharuddin, “Improved emissions estimation of highway traffic using image-based vehicle classification,” Conference/Journal paper, Under Preparation.