THE FLORIDA DEPARMENT OF TRANSPORATION

TRANSPORTATION DATA AND ANALYTICS OFFICE

Truck Taxonomy and Classification Using Video and Weigh-In Motion (WIM) Technology

Final Report

Pan He, Aotian Wu, Anand Rangarajan, Sanjay Ranka

<pan.he@ufl.edu, aotian.wu@ufl.edu,

anand@cise.ufl.edu, sranka@ufl.edu>

Contract: BDV31-977-81

Date: May 2019

DISCLAIMER

The work was supported in part by the Florida Department of Transportation. The opinions, findings and conclusions expressed in this publication are those of the author(s) and not necessarily those of the Florida Department of Transportation or the U.S. Department of Transportation.

METRIC CONVERSION TABLE

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
in	inches	25.4	millimeters	mm
ft.	feet	0.305	meters	m
yd.	yards	0.914	meters	m
mi	miles	1.61	kilometers	km

Table 1: *U.S. UNITS TO METRIC (SI) UNITS*

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft.
m	meters	1.09	yards	yd.
km	kilometers	0.621	miles	mi

Table 2: *METRIC (SI) UNITS TO U.S. UNITS*

DOCUMENTATION PAGE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.		3. Recipient's Catalog No.
4. Title and Subtitle Truck Taxonomy and Classification Using Video and Weigh-In Motion (WIM) Technology Final Report			5. Report Date May 2019
			6. Performing Organization Code
7. Author(s) Pan He, Aotian Wu, Anand Rangarajan, Sanjay Ranka			8. Performing Organization Report No.
9. Performing Organization Name and Address, Computer and Information Science and Engineering, University of Florida 432 Newell Dr, Gainesville, FL 32611			10. Work Unit No. (TRAIS)
			11. Contract or Grant No. BDV31-977-81
12. Sponsoring Agency Name and Address Florida Department of Transportation 605 Suwannee Street, MS 30 Tallahassee, FL 32399			13. Type of Report and Period Covered Final Report 8/21/2017 - 7/22/2019
			14. Sponsoring Agency Code
15. Supplementary Notes			

16. Abstract

The primary objective of the present work was to develop video processing and machine learning methods to automatically detect and classify trucks traveling on Florida highways. The assembly of an automated system to detect, classify, and recognize various truck types from video and to derive their attributes is presented in this report. These extracted attributes were then used to determine commodity types, which can be used downstream for tracking commodity movements. To accomplish this, a set of high resolution videos (made available by the Florida Department of Transportation, FDOT) using freeway roadside passive cameras was utilized to create benchmark datasets. These videos were captured at different times of day, mainly at two freeway locations, and on various days during the past two years. The set of images derived from the videos was processed by our developed system to train and evaluate different approaches. The approaches drew upon recent work in deep convolutional neural networks for object detection and classification, semantic segmentation, and feature extraction, as well as drawing from traditional methods such as decision trees and geometric features (like edges and corners). We developed deep learning algorithms that leveraged transfer learning to determine whether an image frame has a truck and, if the answer is affirmative, localize the area from the image frame where the truck is most likely to be present. In particular, (1) we developed deep learning algorithms for detecting the location of a truck in a video frame followed by determining whether the image corresponds to a truck or a non-truck, (2) we developed a hybrid truck classification approach that integrates deep learning models and geometric truck features for classifying trucks into one of the nine FHWA classes (FHWA classes 5 through 13), (3) we developed algorithms for recognizing and classifying various truck attributes such as tractor type, trailer type, and refrigeration units that are useful in commodity prediction, and (4) we developed techniques for extracting vendor information corresponding to a truck, using logo and text detection.

17. Key Word Machine learning; Deep neural networks; Truck classification; Trailer classification; Logo detection; Commodity identification			18. Distribution Statement No restrictions.
19. Security Classif. (of this report) Unclassified.	20. Security Classif. (of this page) Unclassified.	21. No. of Pages 76	22. Price.

Form DOT F 1700.7 Reproduction of completed page authorized

ACKNOWLEDGMENTS

The authors would like to thank the Florida Department of Transportation (FDOT) for providing real truck video data and for insightful discussions during the course of the project.

EXECUTIVE SUMMARY

Freight analysis is an established approach in many states and major metropolitan areas. It is aimed at integrating various data sources, thus creating a comprehensive picture of freight movement. The main goal of the present work was to develop video processing and machine learning methods to automatically detect and classify trucks traveling on Florida highways. We used the Federal Highway Administration (FHWA) truck taxonomy for truck classification. Next, we automatically performed logo detection and recognition because this allowed for precise determination of commodity types in the detected truck.

The assembly of an automated system for detection, classification, and recognition of various truck types from video and deriving their attributes is addressed in this report. These extracted attributes were used to determine commodity types, which can be used downstream for tracking commodity movements. To accomplish this, a set of high resolution videos (made available by FDOT) using freeway roadside passive cameras was utilized to create benchmark datasets. These videos were captured at different times of day, mainly at two freeway locations, and on various days during the past two years.

The set of images derived from the videos was processed by our developed system to train and evaluate different approaches. The approaches drew upon recent work in deep neural networks for object detection and classification, semantic segmentation, feature extraction, as well as drawing from traditional methods such as decision trees and geometric features (like edges and corners). We developed deep learning algorithms that used transfer learning to determine whether an image frame has a truck and, if the answer is affirmative, localize the area from the image frame where the truck is most likely to be present. In particular,

1. We developed deep learning algorithms for detecting the location of a truck in a video frame followed by whether the image corresponds to a truck or a non-truck.
2. We developed a hybrid truck classification approach that integrates deep learning models and geometric truck features for classifying trucks into one of the nine FHWA classes (FHWA classes 5 through 13).
3. We developed algorithms for recognizing and classifying various truck attributes, such as tractor types, trailer types and refrigeration units, that are useful in commodity pre-

diction.

4. We developed techniques for extracting vendor information corresponding to a truck using logo and text detection. This consisted of two steps: determining the location of the text or logo and detecting company information using text- or logo-based identification. For enclosed trailer types (which correspond to a large fraction of trailer types), this is an effective way of determining the commodity.

Results obtained from our datasets show that our scheme for truck classification has > 90% accuracy for classifying trucks into one of the nine classes and is relatively independent of the actual camera angle. Additionally, our algorithms for trailer detection achieved > 85% accuracy for classifying a tractor and trailer and a 95% accuracy for detecting a refrigerator unit. Additionally, we were able to demonstrate determination of vendor information using text- or logo-based detection and classification, which in turn is associated with commodities using the NAICS database.

Contents

List of Figures

List of Tables

1 Introduction

The Florida Department of Transportation is interested in using a variety of data sources (road sensors, roadside video, etc.) to comprehensively analyze the flow of vehicles and commodities across the state of Florida. Understanding and predicting freight flow patterns is extremely useful in planning and policy decisions at the federal, state, and local levels.

Trucks are largely in charge of transporting freight, both in terms of tonnage and revenue. Federal Highway Administration (FHWA) has proposed a methodology for classifying these trucks into nine categories. Determining the class of the truck is extremely useful in understanding the type of commodity that truck is carrying. According to a study of the American Trucking Association (ATA) [5], the truck industry continues to dominate freight transportation, in terms of both tonnage and revenue. The study estimates that by 2020, total freight tonnage is expected to grow more than 26 percent, along with total freight transportation revenue growing 68 percent. In the first-mile and last-mile (FMLM) challenge, one critical barrier to public transit accessibility is the multimodal freight transportation network, trucks play a key role as well [6]. Hence, understanding and monitoring truck activities becomes an essential component to effectively bolster the development of freight movement.

One important need of transportation agencies is truck classification, as it lays the foundation for freight analysis and transportation planning. Truck classification aims at detecting individual trucks and recognizing their specific types based on certain features in images or video frames. Many techniques for acquiring truck types have been discussed in the transportation community [6, 4, 7, 8]. Among them, prominent and frequently used approaches are image processing techniques for traffic applications, e.g, the automated vehicle systems (AVS).

The goal of this research was to conduct a feasibility and proof-of-concept study on the use of computer vision and machine learning techniques to automatically classify the vehicles in images and to recognize the vendor and the type of commodity that they may be carrying. A set of high resolution videos available from FDOT freeway roadside passive cameras were utilized to create benchmark datasets. These videos were captured over several times of day and periods of the year. They were processed by our developed system to train and evaluate the developed approaches. Our developed approaches took advantage of recent advances in deep neural networks for object detection, semantic segmentation, and edge detection. We developed deep learning algorithms that used transfer learning to determine whether an image frame had a truck and, if the answer is affirmative, localize the area from the image frame where the truck is most likely to be present. We developed a hybrid truck classification approach that integrated deep learning models and geometric truck features with high accuracy. We developed algorithms for recognizing and classifying various truck attributes, such as tractor type, trailer type, and refrigeration units, that are useful in commodity prediction. Using logo and text detection, we developed state-of-the-art techniques for extracting vendor information corresponding to a truck. The overall system has the capability of recognizing and classifying various truck attributes such as truck type, tractor type, trailer type, refrigerator units, and logo and text information.

2 Literature Review

Based on axles, length, or vehicle configuration, groupings of vehicles can be customized into various forms in response to the needs of traffic agencies. In the mid-1980s, the most widely used vehicle classification system was established by the Federal HighwayAdministration (FHWA). It was originally introduced for use in pavement and bridge design [9]. This standardized scheme distinguishes 13 vehicle types by the number of axles and the number of units comprising the vehicle (unit number). The body configuration can be utilized to further distinguish vehicles within each axle-based category, connecting vehicle classification to freight planning, in which the operating characteristics such as the present commodity, drive, and duty cycle are presented.

In the U.S., national shipper/carrier surveys, such as the U.S. Vehicle Inventory and Use Survey (VIUS), serve as the main source providers on correlation of truck activity with body configuration. The VIUS collected samples at both national and state level, with a focus on the statistics of freight truck movement and truck characteristics (e.g., weight, number of axles, length, and body type). Due to the limitation of sampling strategy (conducted every five years), such surveys cannot provide operational data for a particular individual link or route level within a certain day or period.

Most recent research work has focused on developing truck classification models that use various traffic sensor data, including both intrusive sensors and non-intrusive sensors. Intrusive sensors are installed on pavement surfaces, thus requiring interruption of traffic during installation. Intrusive sensors have shown their insensitivity to inclement weather, due to a high signal-to-noise ratio. Pneumatic road tube, inductive loop detectors (ILD), weigh- in-motion (WIM), and piezoelectric sensors belong to this sensor type. Non-intrusive sensors installed at locations have the capability of detecting vehicle parameters (e.g., speed and lane coverage). Popular non-intrusive sensors include vision-based sensors, infrared, radar, acoustic, and GPS, etc. In this section, we present a brief summary of classification techniques, focusing on developments for sensors (especially vision sensors). We refer interested readers to [10] for a more detailed discussion.

Truck bodies are classified via non-vision sensors such as WIM and inductive signature data at weigh stations. Various classifiers (Support Vector Machine (SVM), decision trees, and neural networks) were developed for the classification, based on acquired sensor data [11]. A heuristic method has been proposed to develop a vehicle classification model by combining decision trees and K-means clustering approaches using single-loop inductive signature data [12]. In [6], the Truck Activity Monitoring System (TAMS) was developed for detailed truck classification. Existing traffic detection infrastructure, such as WIM and ILD are utilized in TAMS, along with developed state-of-the-art machine learning algorithms. The major component of TAMS involves the creation of inductive signatures and the integration of them with WIM.

On the other hand, automated vehicle classification (AVC) equipped with vision-based sensors has received increasing attention in the transportation community. Successful approaches can greatly help traffic agencies identify vehicles of certain types, colors, makes (manufacturers), and models [6]. An automated classification system, potentially leveraging the Federal Highway Administration (FHWA) truck classification taxonomy along with other information that can be read from the truck, allows precise determination of commodity types. It can be used to track commodity flows within the state.

Traditional approaches are usually based on estimation of vehicle dimensions. Lai *et al*. [13] presented a method that accurately estimates vehicle length within a reasonable error threshold. However, their method is limited by the requirement of camera calibration, which may not be easy to obtain. Commercial video image processors could perform well under a specific configuration with careful calibration, but they are usually very expensive.

Deep learning techniques have advanced the performance of many research problems, such as object detection [14, 15, 16] and object classification [17, 18]. Many advanced techniques from deep learning have been applied to vehicle type classification [13, 19, 7]. In [19], a deep neural network is used for classifying vehicles (cars, sedans, and vans) in a small test dataset. Its performance on truck classes is unknown. Adu-Gyamfi *et al*. adapted the pretrained deep learning model and fine tuned model parameters for vehicle recognition [7].

Yu *et al* . used CNN and a joint Bayesian network for classification [20]. Probabilistic neural networks were used on frontal view image measurements [21]. Vehicle model verification (i.e., verifying whether two vehicle images belonged to the same vehicle model) and vehicle re-identification (precise vehicle search) were achieved by using deep convolutional networks [22]. Classifying a vehicle based on the type (sedan, bus, truck, *etc*.), in particular, gained most of the interest, playing a key role in intelligent traffic systems. Dong *et al*. classified the vehicle type from front-view images using semi-supervised convolutional neural networks (CNN) [23]. The vehicle type was classified with an SVM after extracting local and structural features [24]. Fu *et al*. classified multiple vehicles in crowded traffic scenes from video surveillance by using multiple SVM classifiers [25]. The 3-D deformable vehicle model with evolutionary computing has been developed to classify types [26]. Dimension estimation was developed to identify vehicle types by finding a simple deformable vehicle model from calibrated cameras [13]. Discrimination between trucks and other vehicles was also achieved by comparing the length in pixels of vehicles from uncalibrated video camera images [27]. A deep transfer learning approach was developed for truck body type classification in [8]. The ensemble approach of deep learning model was discussed and developed in [28].

Based on all of the above, we see that an integrated approach that leverages many of these previous techniques while paying close attention to the FHWA taxonomy. The present work was a foray into achieving this objective.

3 Methodology

3.1 Overview

In this section, we describe the computer vision and machine learning approaches that were developed for the problems at hand. The fixed position of roadside camera sensors allowed us to obtain truck visual information and grab a set of frames in which the truck persists. As lighting conditions were not excessively variable, the difficult problem of text and image recognition from truck bodies was simplified.

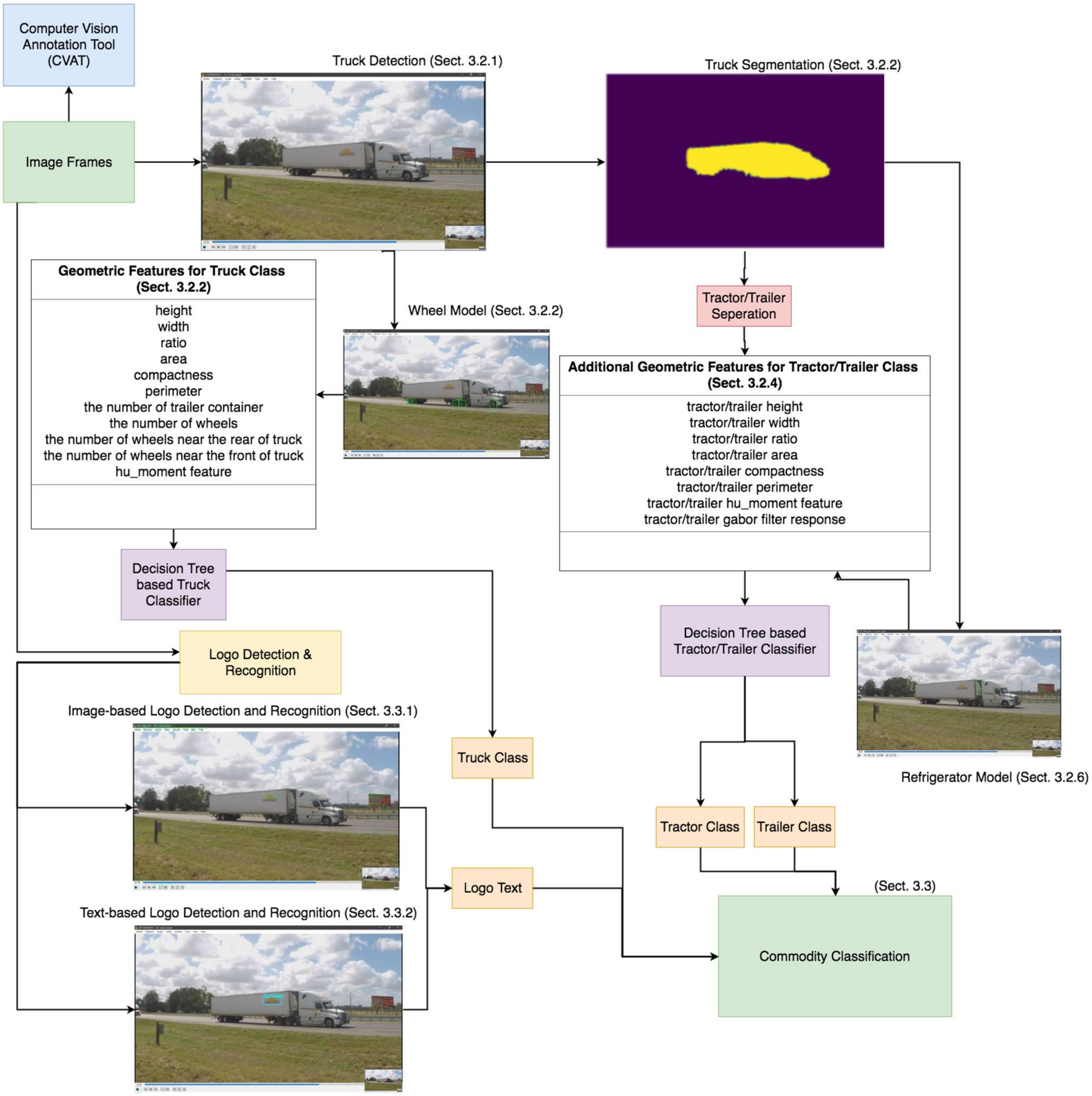

Computer Vision Annotation Tool (CVAT)
Image Frames
Truck Detection (Sect. 3.2.1)
Truck Segmentation (Sect. 3.2.2)
Geometric Features for Truck Class (Sect. 3.2.2)
height
width
ratio
area
compactness
perimeter
the number of trailer container
the number of wheels
the number of wheels near the rear of truck
the number of wheels near the front of truck
hu_moment feature
Wheel Model (Sect. 3.2.2)
Tractor/Trailer Seperation
Additional Geometric Features for Tractor/Trailer Class (Sect. 3.2.4)
tractor/trailer height
tractor/trailer width
tractor/trailer ratio
tractor/trailer area
tractor/trailer compactness
tractor/trailer perimeter
tractor/trailer hu_moment feature
tractor/trailer gabor filter response
Decision Tree based Truck Classifier
Decision Tree based Tractor/Trailer Classifier
Logo Detection & Recognition
Image-based Logo Detection and Recognition (Sect. 3.3.1)
Truck Class
Refrigerator Model (Sect. 3.2.6)
Tractor Class
Trailer Class
Logo Text
(Sect. 3.3)
Commodity Classification
Text-based Logo Detection and Recognition (Sect. 3.3.2)

Figure 1: *A pipeline of all the developed techniques.*

We used an array of techniques for obtaining a set of features that are suitable to truck classification and commodity classification. A pipeline of all the developed techniques is summarized in Figure 1. Specifically, in the task of detecting potential trucks shown in image frames, transfer learning techniques were adopted to accurately find candidate vehicle regions by estimating the bounding box of each vehicle object. A 2-class (truck vs. non-truck) deep learning classifier was developed to decide whether the vehicle candidate was a truck or not, as we were interested in trucks.

Once the truck regions were determined, we moved to the task of determining the truck classes. After a careful examination of the truck classes (Figure 30), we identified key features for discrimination such as the number of wheels (a proxy for the number of axles), number of trailers, size, and aspect ratio (ratio of length to height from the side view). Leveraging these phenomenological observations, we developed an effective truck classification approach that is human-understandable. Along the way, a novel model was developed for locating wheels. Semantic segmentation models were developed to extract truck contours, based on which geometric features were computed. Image processing techniques were developed for identifying trailer and tractor units. Many advanced computer vision algorithms were utilized for extracting truck shape features that were helpful in discrimination.

Similarly, we developed models for tractor and trailer classification. Our observation was that truck classes are closely related to the tractor and trailer types. For example, most of the class 6 trucks, in reality, are bobtail (one of the trailer types). RV trucks (one of the tractor types) can only come from the class 5 truck category. Based on this domain-specific knowledge, we designed customized tractor and trailer features and combined them with truck features to develop tractor and trailer models.

A refrigerator truck is designed to carry perishable freight. The refrigeration unit, or refrigerator unit, is usually attached to the trailer unit. We develop a deep learning model for locating the refrigerator unit. The image annotation tool was utilized to annotate enough refrigerator unit training data to allow the construction of an accurate refrigerator model.

In addition to these truck attributes, logo and text information attached to the truck images were extracted for commodity classification. In order to extract pure text information on the truck, state-of-the-art models were developed for text detection and recognition. The truck images were sent to a fully convolutional network (FCN) model, followed by the post-processing step non-maximum suppression (NMS) that filtered out overlapped detection results. In this way, we obtained text line and word locations represented by oriented bounding boxes. Pure text was made available by a post-processing recognition model. Word correction and string matching techniques were applied to match the result to predefined vendor lists (collected in a database). We could have obtained the final company names from the text information extracted from the trucks if a comprehensive database had been available.

After extracting logo information, deep learning models were developed to localize logo regions. This involved a process of collecting logo training samples and training the detector for logo classes. Several model variants are presented and discussed in Section 3.3.2. A very preliminary effort is described at the end of this report showing anecdotal results for a selected set of vendor/logo pairs.

3.2 Truck Model

We describe the logic and reasoning undergirding the development of our approaches for truck detection and classification. The whole method was mainly separated into two distinct components: truck detection and truck classification. The truck detection component aimed at determining the presence of truck objects within images, followed by estimating the bounding box of each truck object. The truck classification component determined the category of the truck object.

3.2.1 Truck Detection Component

Truck videos contain many complex and challenging backgrounds. If the whole image was used as the input for the truck classification model, we would have obtained unsatisfactory results. A better approach is to first crop out individual truck regions from the images. We adopted the advanced YOLO (You Only Look Once) object detector for the truck detection problem.

The pipeline of YOLO [29] is simple and straightforward, as shown in Figure 2. The method begins with dividing the image into an $S \times S$ grid. Subsequently, it passes the entire image

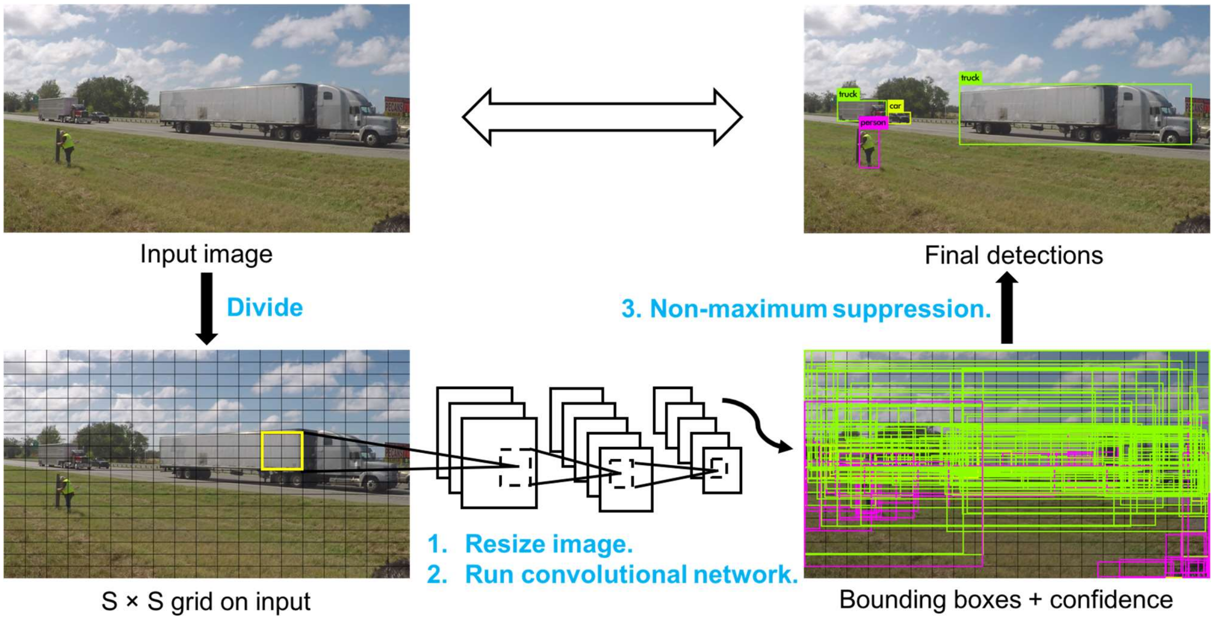


__Figure 2:__ The YOLO detector divides the image into an $S \times S$ grid and, using shared computation, passes the whole image through the model to obtain image features for each cell. Predictions are encoded as a 3D tensor of size $S \times S \times (B \times 5 + C)$. The final results are obtained with the non-maximum suppression.

through a neural network model to obtain image features for each cell (in a shared computational framework). Each cell containing the learned generalizable representation is utilized to predict B bounding boxes, B confidence scores for those boxes, and C class probabilities, resulting in predictions for the whole image encoded as a 3D tensor of size $S \times S \times (B \times 5 + C)$.

Specifically, each boundary box consists of 5 elements: (x, y, w, h) and a box score representing the degree of detection confidence, s. The locations (x, y) correspond to the center of the box. The (w, h) are the predicted vertical and horizontal size relative to the whole image. These elements are normalized to values between 0 and 1. The confidence prediction, s, reflects the model confidence that the box contains an object. The accuracy of box predictions is based on finding the overlap between the predicted box and the ground truth.

We followed the original YOLO work [29] to get multiple candidate detections. Because we were only interested in trucks, we removed predictions not belonging to vehicle classes (e.g., truck, car, bus), followed by NMS to pick up top predictions. We further utilized advanced tracking algorithms, such as the correlation trackers in dlib [30], to speed up the model.

3.2.2 Truck Classification Component

The detected trucks were processed by the truck classifier to determine the class of the truck based on the FHWA classification scheme. We observed that the truck classification problem was more challenging than general classification problems, as we were attempting to differentiate subordinate truck classes (FHWA class 5 to FHWA class 13) of the common superior class (truck class). These subordinate truck classes are defined by transportation experts with complicated rules, focusing on subtle differences in particular regions (e.g., number and spacing of axles and trailer numbers). We therefore sought a solution to integrate deep learning models with geometric truck features, resulting in a hybrid approach for truck classification. Below, we present the deep learning approach, followed by the integrated approach.

Pure Deep Learning Approach. The first approach we pursued during the project was a pure deep learning approach for truck classification. The basic idea was to utilize the transfer learning technique in deep learning. We were able to identify one of the key challenges for the truck classification problem, namely a limited dataset and imbalanced data distributions. As shown in Figure 6, most of the training images were class 9 trucks. This would result in an inaccurate training/evaluation method for measuring model performance because the classifiers trained with this distribution tend to predict the majority class. The features ofthe minority classes are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority classes, compared to the majority classes.

To overcome this limitation, transfer learning-based techniques were explored in this approach. Transfer learning approaches focus on storing knowledge garnered from one domain and then applying it to a different, but related domain. Training a deep neural net using random model weight initialization is impractical because the dataset size requirements are very high. We tried to utilize knowledge gained from other large dataset problems, such as a ConvNet pretrained on a huge dataset (e.g., ImageNet, 1.2 million labeled images with 1,000 categories). The use of pretrained ConvNet was then leveraged as an initialization. We also applied it to the task of interest as a fixed feature extractor. We performed transfer learning as follows.

We modified the original model trained on the large dataset and retrained the classifier on the truck dataset, through back-propagation. We chose to freeze the weights of the earlier layers. The motivation for this strategy was the fact that the lower layers of the ConvNet contain more generic features (e.g., edge detectors or color blob detectors), and they should be useful features for all classification tasks. The higher layers of the ConvNet are generally more specialized to the original classes; thus, we retrained the original model for our own task.

The preliminary results from pure deep learning approaches indicated that models tended to over-fit on the training dataset because our training dataset was relatively small compared to the million training images used in the popular ImageNet Classification problem [31]. We therefore moved to a hybrid approach that leverages recent advances in deep convolutional neural networks. For the hybrid approach, we designed specific algorithms for extracting geometric features that are suitable for distinguishing truck classes. It naturally incorporated domain knowledge from transportation engineering to the machine learning approaches. It presents a human-understandable model that can enable a successful collaboration between traffic agencies and machine learning models, allowing for an effective interaction with the model to make better decisions.

Estimating Truck Size. One of the basic approaches that has been applied in vehicleclassification is the use of the bounding box around the detected vehicle, which covers the initial approximation of vehicle shape. In order to obtain precise shape information, a refinement step must be applied to the detected vehicles obtained from truck detectioncomponents.

Fully convolutional neural networks (FCNNs) have shown to be very useful for pixel-wise segmentation and are very suitable for large-scale traffic video processing. FCNNs can deal with variable size images and take into account context information when identifying the vehicle objects. The flexibility of FCNNs is due to their adaptive network design: a deep convolutional neural network (DCNN) encoder; a decoder that uses bilinear interpolation; predictions that work on similar size images; and popular post-processing techniques such as the conditional random field (CRF) model [32].

We implemented and adapted the popular DeepLabV2 [1] model for estimating the vehicle shape. It introduced the multiple parallel atrous convolutional layers. The atrous convolutional layers do not increase the number of parameters and computational overhead, yet

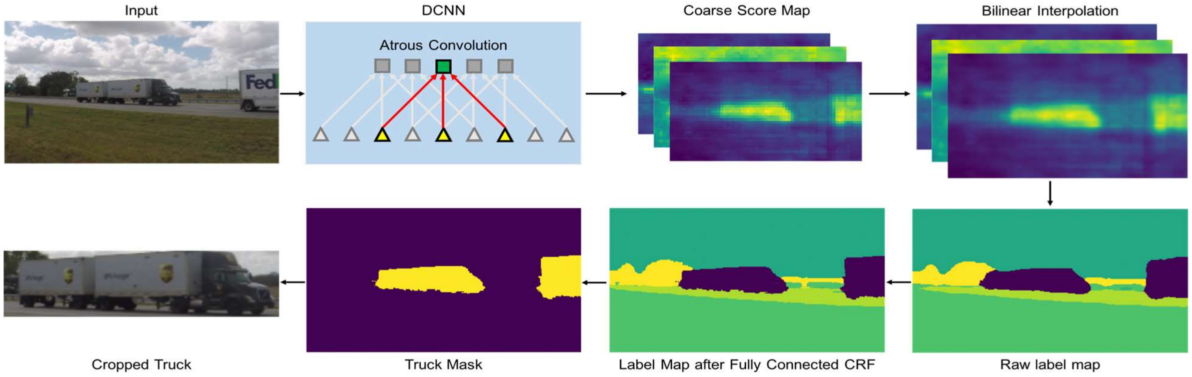


Figure 3: *The pipeline of estimating truck size. Following [1], a fully convolutional ResNet is deployed by adding atrous convolution with different sampling strides to obtain the coarse score map. The feature maps are then upsampled by the bilinear interpolation to the original image resolution. A raw label map is obtained via the Softmax operation. Refining the coarse prediction and utilizing structure information in images, a fully connected conditional random field is then applied to better capture the object boundaries. The truck mask is obtained by removing other non-truck labels. The truck region is cropped out by selecting the largest connected components. In our problem, we could also apply this pipeline to the initial detected truck regions to refine the detection results because the method supports processing images of different sizes.*

$$y(i) = \sum_{k=1}^{K} x[i + r * k]w[k]. \tag{1}$$

they effectively enlarged the field of view of filters. The atrous convolutional layer (also known as dilated convolution) is a variant of convolutional layers. r denotes the sampling stride on the input signal. Given a 1-D input signal $x[i]$, its corresponding output $y[i]$ with atrous convolutional filter $w[k]$ of length K can be described as

By setting the rate r equals to 1, the standard convolution is formulated. The DCNN score map is extracted from the atrous layers with different sampling rates. It is processed by the bilinearly interpolation layer to produce a score feature map of the original image resolution. The final score map are obtained by taking the maximum response at each pixel location.

The score map from DCNN was able to determine the presence and rough position of truck objects, but failed to delineate the borders. To overcome this limitation, a further post-processing step was integrated. Following [1], the fully connected CRF model was employed with the energy function:

$$E(\mathrm{Y}) = \sum_{i} \theta_i(\mathrm{Y}_i) + \sum_{ij} \theta_{ij}(\mathrm{Y}_i, \mathrm{Y}_j) \tag{2}$$

where Y represents the label assignment for all the pixels. Let P (Y$_i$) denote the label assignment probability at pixel i. θ_i(Y$_i$) = $-$ log P (Y$_i$) is then the unary potential. The pairwise potential θ_{ij} is defined as

$$\begin{aligned} \theta_{ij}(P(\mathrm{Y}_i), P(\mathrm{Y}_j)) = 1_{i,j} \Bigg[& \omega_1 \exp(-\frac{||p_i - p_j||}{2\sigma_\alpha^2} - \frac{||I_i - I_j||}{2\sigma_\beta^2}) \\ & + \omega_2 \exp(-\frac{||p_i - p_j||}{2\sigma_\gamma^2} \Bigg] \end{aligned} \tag{3}$$

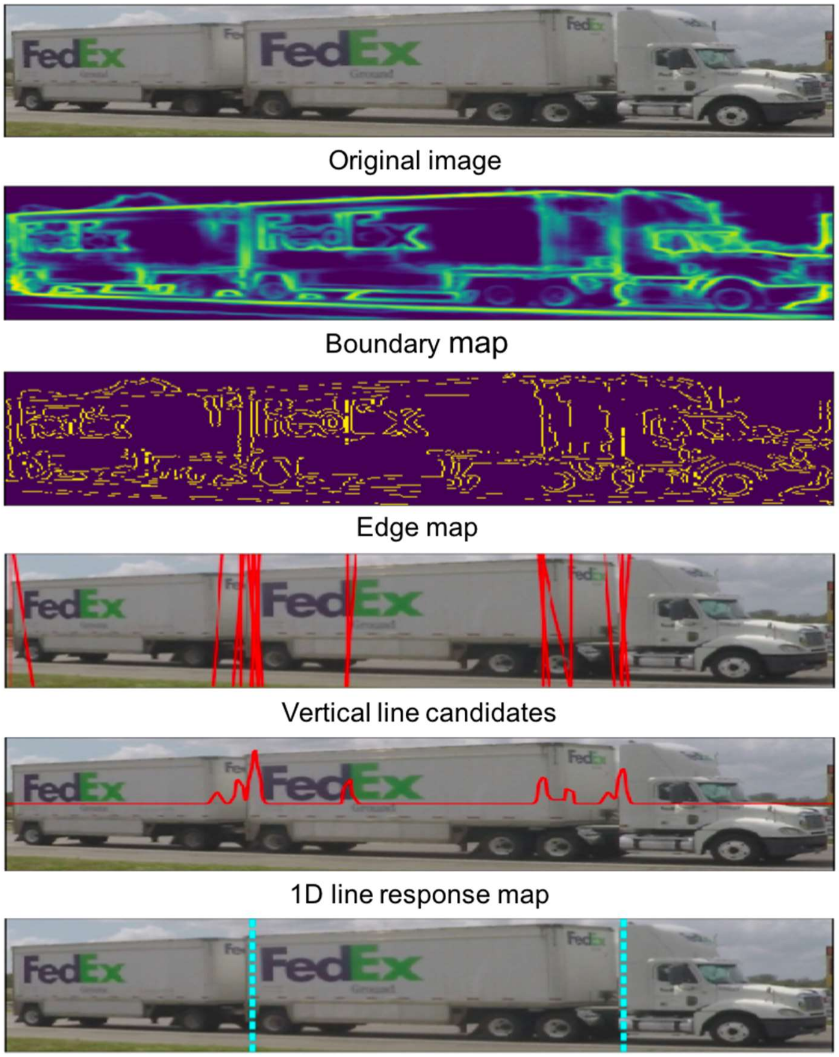
where 1$_{i,j}$ = 1 if Y_i ƒ= Y_j, and zero otherwise. The pairwise potential was modeled with two Gaussian kernels. The first kernel ('bilateral' kernel) is based on both pixel positions (denoted as p) and RGB color (denoted as I). It forces similar label assignments between nearby pixels with similar colors. The second kernel purely depended on pixel position, thus encouraging spatial smoothness.

Estimating Vehicle Trailer Units. The number of trailers is a useful feature for distinguishing the FHWA truck class. To obtain the number of trailers, we developed the TRailer Unit Estimation (TRUE) model. It used the number of truck containers as a proxy for the number of trailers. Our pipeline for the TRUE model involved three main steps: vehicle boundary and edge detection, vertical line detection, and peak finding.

To capture the vehicle boundary, we built our architecture on top of the HED (holistically-nested edge detector) system [33], which is based on the idea of FCNNs([34] and deep-supervised nets [18]. Given the training data set S = {(X_n, Y_n)}, n = 1, ..., N, where sample X_n denotes the raw images and Y_n denotes the corresponding ground truth binary edge map. The goal was to learn a feature mapping function f with a network (parameterized by θ) to produce edge maps. $x_{i,j}$ denotes the data vector at location (i,j) in a particular layer and by $y_{i,j}$ the corresponding output for that layer. Formally, the function of output $y_{i,j}$ is defined as

$$y_{ij} = f_{ks}(\{x_{s*i+\delta_i, s*j+\delta_j}\}_{0<\delta_i,\delta_j<k}) \tag{4}$$

where k and s are the kernel size and the stride factor, respectively. f_{ks} is the layer manipu- lation.



Original image

Boundary map

Edge map

Vertical line candidates

1D line response map

Peak finding

Figure 4: *The novel pipeline for estimating vehicle trailer units. Given an initial cropped-out vehicle image, the boundary map is obtained by the HED detector. Edgelets are extracted from the boundary map with the popular Canny edge detector. Vertical line candidates (red lines) are detected via the Hough transformation algorithm. The 1-D line response map is obtained by merging lines that are close to each other, using morphological image operations, followed by projecting vertical lines via summation along the columns. Finally, a peak-finding algorithm picks up the best separation spacing for trailer units.*

The whole network added such layers multiple times to learn the nonlinear filters. The final vehicle boundary was obtained by further aggregating the generated edge maps of layers (with different down-sampling rates) with in-network bilinear interpolation. One advantage of this design was that the model could take inputs of arbitrary size and efficiently producethe output.

The obtained boundary map was further processed by the Canny edge detector to obtain the edgelets, followed by the Hough transformation for finding straight lines. Because most of the straight lines are borders of truck heads, trailers units, or road lines in vehicle images, we were able to obtain vertical candidate lines for estimating the number of trailer units. However, directly taking the total sum of numbers of vertical lines as the number of trailer units was not satisfactory because the method introduced noisy vertical (or nearly vertical) line detections. We therefore developed a pipeline of peak finding to overcome the limitation.

The process started by computing the line response maps to merge lines that are close to each other by using morphological image operations. We refer to these merged line detections as a $W \times H$ line response map, as shown in Figure 4. These $W \times H$ maps were reduced to $W \times 1$ responses by summing along the columns. The optimal breakpoint placing for separating trailer units was obtained by the peak finding algorithm.

Peaks can be considered as a location where the value is greater than a threshold or a relative threshold. We define a response location as the peak if it satisfies two conditions:

- The value is higher than a minimum peak height α. We set $\alpha = 0.25H$.
- The value is higher than the values of its nearby peaks within a peak distance β. We set $\beta = 0.25W$.

Estimating Vehicle Wheels. Detecting and recognizing wheels in vehicle images can serve as a foundation for the FHWA classification method. For example, based on the center of each wheel and distances between subsequent wheels belonging to the same vehicle, agencies could compare the generated distances with a known table and assign the corresponding vehicle class. Therefore, it is very important to know the location of each wheel in order to classify a vehicle.

We began with a baseline wheel detection model based on traditional hand-crafted features and support vector machine classifier. This baseline model turned out not to be a robust solution for wheel detection for several reasons: (1) the viewpoint variations (or equivalently, pose variations) present huge challenges to wheel detection because they introduce unwanted perspective effects, where local descriptors are not robust enough to handle these appearance variations; (2) the illumination and background clutter make it very difficult to extract discriminative features; (3) the deformation of wheels and scale variations further degrade system performance.

We observed that wheel detection problems are similar to face detection problems (especially tiny face detection problems) as both of them must detect round shapes (usually with small sizes). Within faces or wheels regions, pixels tend to have almost the same intensity values or the same color code. Because face detection has been intensely studied for a while, we sought a wheel solution from that field.

Inspired by recent advances in tiny face detection [16], we developed a lightweight deep model for wheel detection. It was built upon a generic object detector called the region proposal network (RPN). Our problem is a single-category detection task (wheel vs. non-wheel), and RPN is a detector concerning just one category.

Because wheel boxes are usually square, we only used a 1 : 1 aspect ratio for the default anchors. We scaled the anchors in layers in different depths with different rates to handle wheel size variations. Similarly, we followed [16] and used the anchor densification strategy to eliminate the tiling density imbalance problem. Several data augmentation strategies were followed. These include the following: random cropping, color distortion, scale transformation, and horizontal flipping. The loss function was the same as RPN [15] with sigmoidal loss for binary classification and smooth L1 loss for regression.

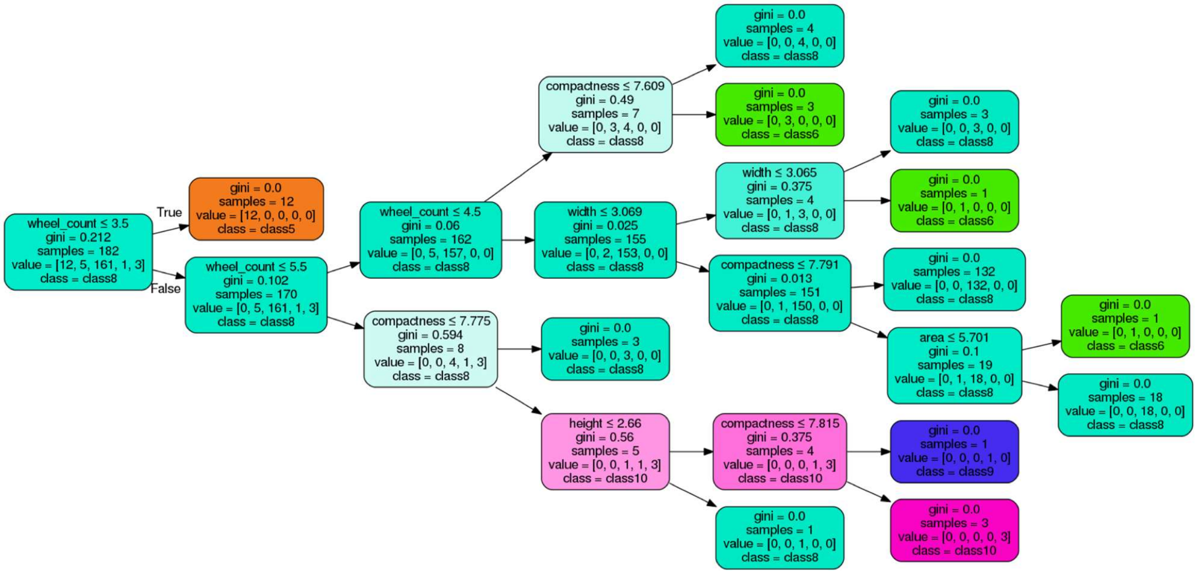


Figure 5: *Visualization of the learned decision tree classifier. The learned rules are human-understandable, which can enable a successful collaboration between traffic agencies and machine learning models and allow an effective interaction with the model to make better decisions.*

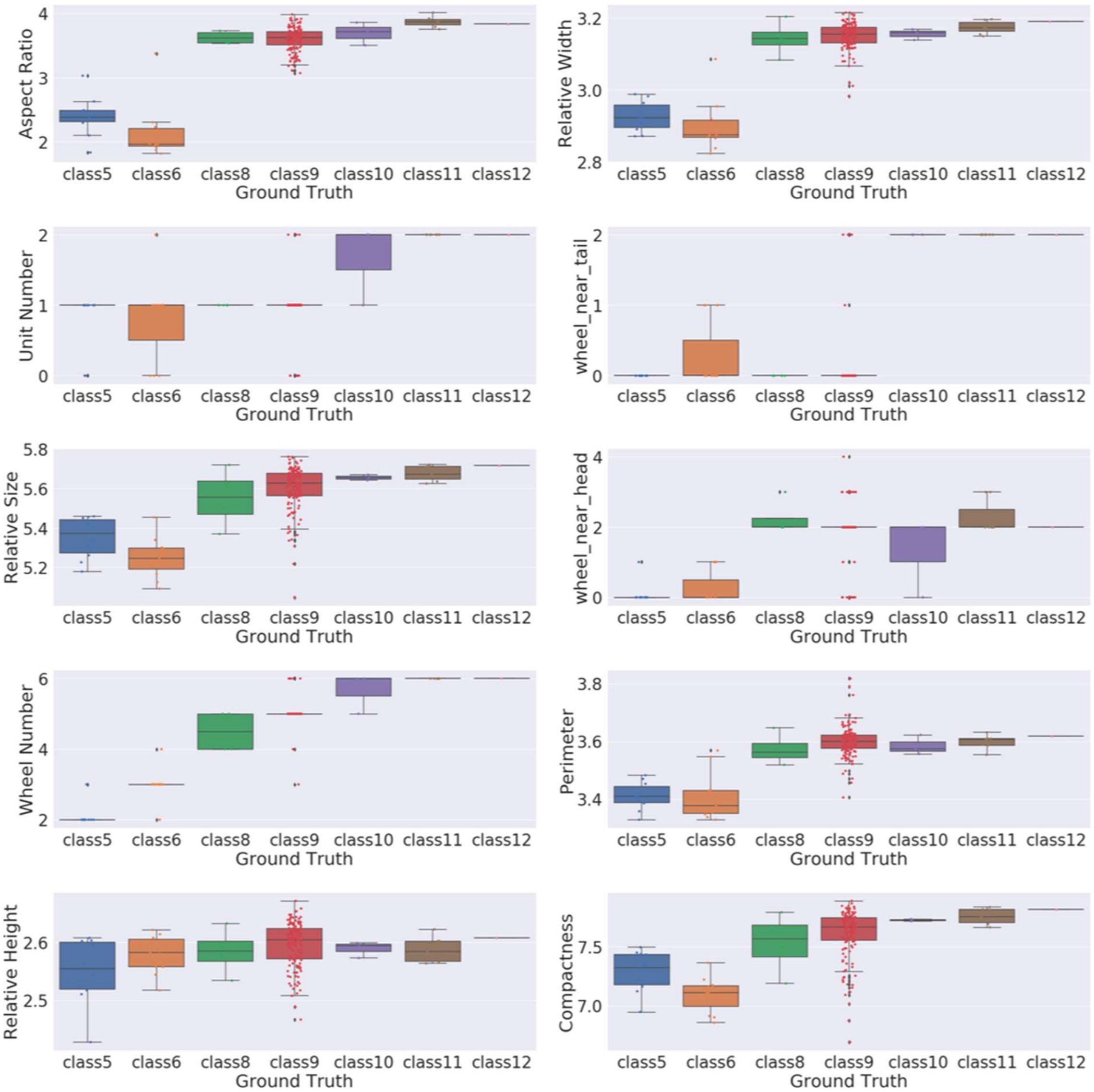


Figure 6: *The sample distribution of acquired geometric features from our annotated truck dataset.*

Decision Tree with Geometric Features. We acquired all the geometric features from the developed algorithms (e.g., the number of wheels, number of trailers, size, and aspect ratio). Based on the relative relation between the wheels and trailer unit, we also considered two features called $wheel\_near\_tail$ and $wheel\_near\_head$. The $wheel\_near\_tail$ calculates the number of wheels around the tail trailer unit while $wheel\_near\_head$ calculates the number of wheels around the truck head. As shown in both Figure 5 and Figure 6, these two features, used to describe the location distribution of wheels, are discriminative features for truck classification. For features with large values, we standardized them by applying the log scale transformation. The sample distribution of acquired geometric features from our annotated truck datasets can be found in Figure 6. We leveraged the popular CART (classification and regression trees) decision tree for the truck classifier. One main motivation for us to choose this algorithm was that the learned model is human-understandable (as shown in Figure 5), which can enable a successful collaboration between traffic agencies and machine learning models, allowing an effective interaction with the model to make better decisions.

3.2.3 Results for Truck Classification

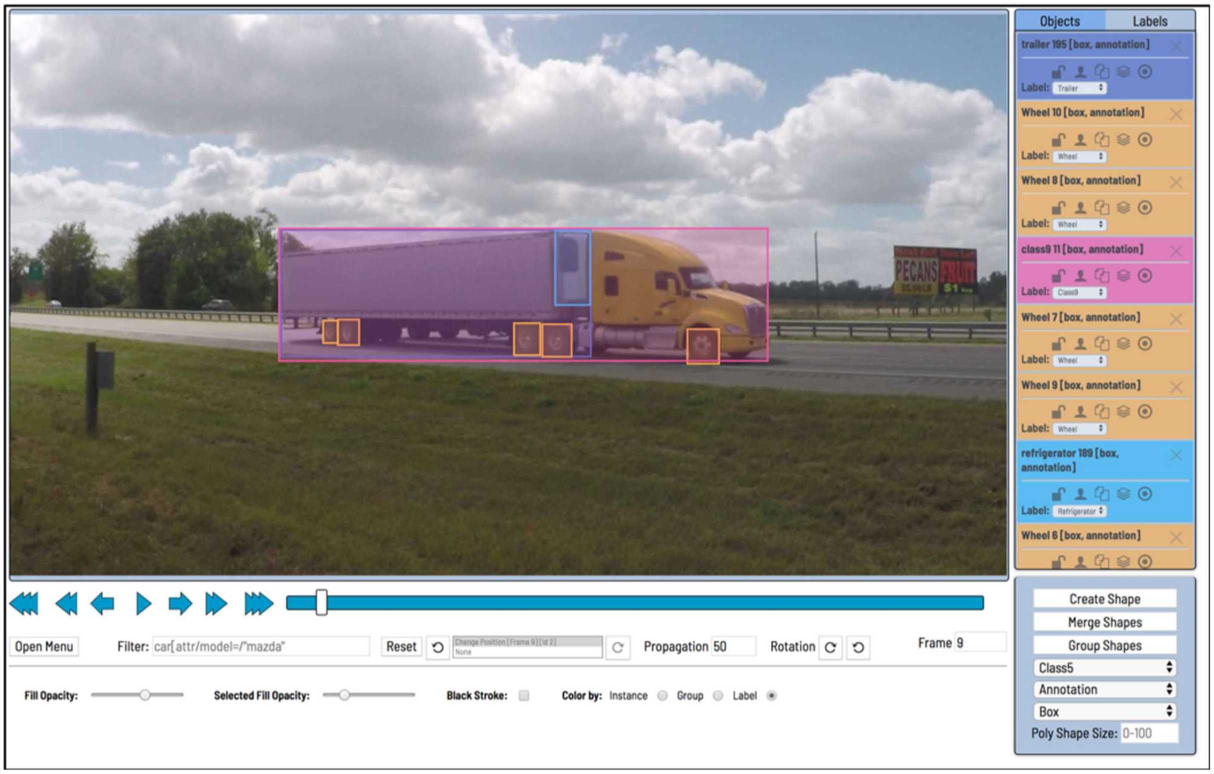
We evaluated our developed truck classification approaches on the proposed benchmark datasets, including the Annotated Truck Dataset and the Annotated Wheel Dataset.

Image Collection Procedure. The primary source of data for evaluating our approach were video frames captured by roadside video cameras deployed at a WIM station in Florida. As shown in Figure 7, we used an available annotation tool called Computer Vision Annotation Tool (CVAT) [35], to carefully annotate the acquired images, thus enabling a quantitative comparison across various algorithms. Our collected datasets were used mainly for two purposes: measuring the performance of truck recognition and evaluating wheel detection models.

Annotated Wheel Dataset. Wheel images were collected for training and evaluating the wheel models. We manually annotated 6, 648 wheels from 1, 634 traffic images. Among them, 1, 234 images were used for training. The remaining ones were used for evaluation.

hen evaluating the performance of wheel detection algorithms, we followed [15] and used the most commonly used metric, Average Precision (AP), which is derived from precision and recall. Detected results were assigned to ground truth wheel annotations. The classification accuracy was determined by calculating the bounding box overlap using the IoU (Intersection over Union) criterion. We considered a detection to be correct if the area of overlap between the IoU exceeded a certain threshold. A threshold of 0.5 was used in our experiments.

Results for Annotated Wheel Dataset. To demonstrate the advantages of our wheel detection method, we developed a baseline approach based on popular HOG (Histogram of Oriented Gradients) + SVM detection pipeline. In addition to this, for the baseline model, we



trailer 195 [box, annotation]
Label: Trailer

Wheel 10 [box, annotation]
Label: Wheel

Wheel 8 [box, annotation]
Label: Wheel

class9 11 [box, annotation]
Label: Class9

Wheel 7 [box, annotation]
Label: Wheel

Wheel 9 [box, annotation]
Label: Wheel

refrigerator 189 [box, annotation]
Label: Refrigerator

Wheel 6 [box, annotation]

Filter: car[attr/model=/"mazda"

Change Position [Frame 9] [id 2]
None

Propagation 50

Frame 9

Fill Opacity:

Selected Fill Opacity:

Black Stroke: □

Color by: Instance ○ Group ○ Label ●

Class5

Annotation

Box

Poly Shape Size: 0-100

Figure 7: *Visualization of data annotation tool. For a given truck image, attributes such as truck class, tractor class, trailer class, refrigerator units, and wheel units are annotated manually and saved to XML format.*

precomputed the perspective transformation matrix for camera calibration. As illustrated in Fig 33, our wheel detection model performed well on the annotated wheel dataset, achieving an average precision of 96.63%. The qualitative results are shown in Figure 32. As can be seen, our wheel model was even able to handle the situation where wheels are extremely tiny and blurred, which indicated its robustness on wheel detection.

Annotated Truck Dataset. Two datasets were acquired to develop and evaluate truck classification systems. The first one (Dataset A) was collected and annotated directly by the traffic agencies—the Florida Department of Transportation (FDOT) in our case. It contains 372 truck images with a fixed camera view angle. The second one (Dataset B) was our self-annotated dataset, which contains 1, 251 truck images from different camera angles.

We present two types of evaluations based on our modified scheme. The first one directly follows the FHWA vehicle classification scheme. Considering that some of the classes in the FHWA scheme only have subtle differences, we present the second evaluation that cast

	9-class Experiment			
	Dataset A		Dataset B	
	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
1	98.65	94.35	94.89	92.25
2	99.19	94.08	93.77	90.97
3	98.39	93.83	93.68	91.05
4	98.92	92.74	93.52	91.53
5	98.65	94.34	93.69	90.81
Average	98.76	93.87	93.91	91.32

Table 3: *Nine-class performance evaluations on the two annotated truck datasets.*

the original FHWA truck classes into a 3-class problem, consisting of group 1 (class 5, 6, 7), group 2 (class 8, 9, 10), and group 3 (class 11, 12, 13). The K-fold cross validation (KCV) procedure was exploited in all the truck classification experiments. It consisted of splitting the dataset into k subsets, where k was fixed in advance: $k -$ 1 folds were used for training the classifier, and the remaining fold was used for the evaluation. We set K = 2, repeated the K-fold experiment five times, and reported the average results.

Results for Annotated Truck Dataset. All our results are reported in Tab. 3. In the 9-class experiment, we achieved an average test accuracy of 93.87% on dataset A. It achieved slightly lower accuracy on dataset B (91.32%). Dataset B was generally more challenging than dataset A as it contained truck images from different camera view angles. In the 3-class experiment, we achieved average test accuracies of 97.36% and 96.34% on Dataset A and Dataset B, respectively.

3.2.4 Tractor and Trailer Classification Component

The semantic segmentation scheme, as described for truck classification, can be used to determine the tractor (or trailer) contour, as shown in Figure 8 and Figure 9. Based on the extracted contour, additional shape features such as compactness, perimeter, area and hu_moment features can be computed.

Shape features can characterize the appearance of an object. We identified features that

	3-class Experiment			
	Dataset A		Dataset B	
	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
1	100.00	97.04	98.40	96.09
2	99.73	96.50	98.08	96.33
3	100.00	97.31	98.32	96.49
4	100.00	97.85	98.24	96.48
5	99.73	98.11	98.32	96.33
Average	99.89	97.36	98.27	96.34

Table 4: *Three-class performance evaluations on the two annotated truck datasets.*

would be helpful for tractor and trailer classification. Because a bobtail truck has a short length, the ’area’ and ’perimeter’ shape feature could be used to separate them from other trailer types.

Objects which have an elliptical shape, or a boundary that is irregular rather than smooth, tend to have a smaller compactness value defined as

$$\text{Compactness} = \frac{4\pi \times \text{area}}{(\text{perimeter})^2} \tag{5}$$

Because many tractor and trailer types can be characterized as blob shapes, we utilized advanced shape-matching techniques in computer vision for generating discriminant features. An image moment is defined as a weighted average of image pixel intensities. Given an image I, the simplest image moment is given below:

$$M = \sum_x \sum_y I(x, y) \tag{6}$$

which calculates the sum of all pixel intensities. This feature is relatively robust with respect to rotation. Generalizing the above idea, a more complex moment is given by:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{7}$$

where the moment now depends on both the intensity of pixels and their locations in the image. Given the fact that the centroid of a binary image is simply its center of mass, we

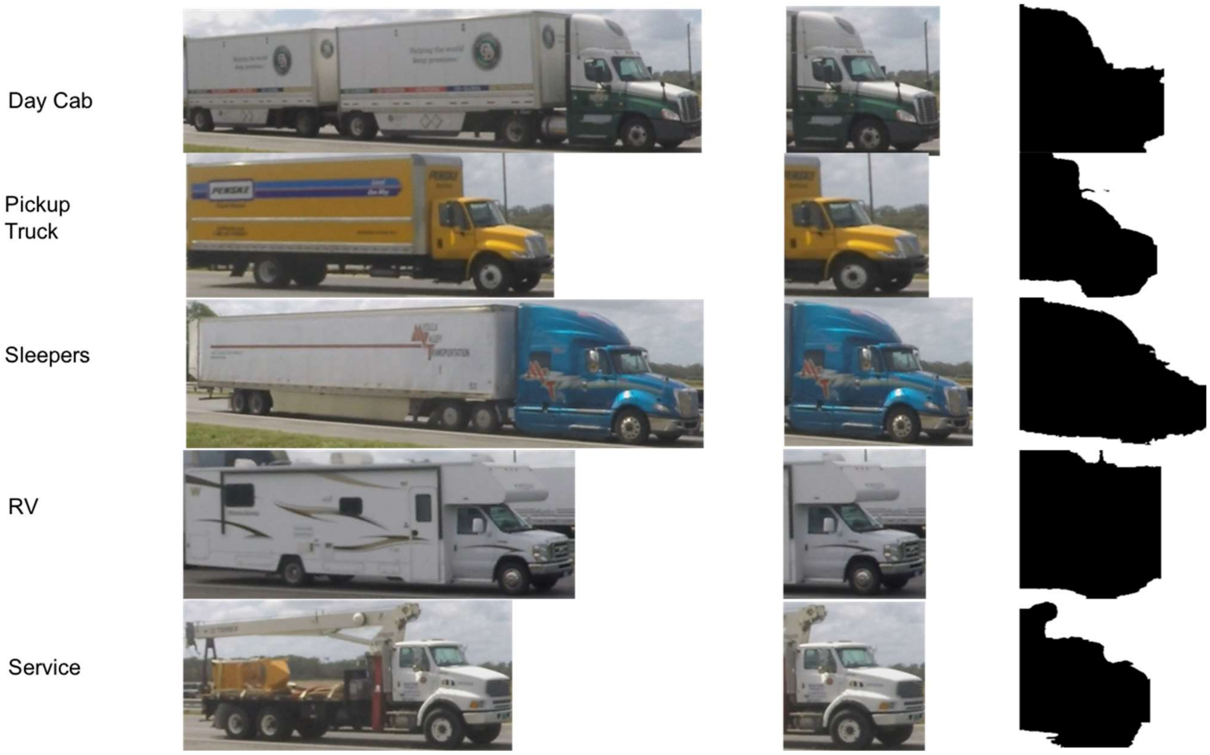


Figure 8: *Sample-derived tractors and their corresponding contours. The results were derived from videos taken at I-75 site 9956.*

were able to transform the original image moment to central moments by subtracting the centroid of the image as follows:

$$\mu_{ij} = \sum_x \sum_y (x - \hat{x})^j (y - \hat{y})^i I(x, y) \tag{8}$$

It can be shown that this computation is **translation invariant**. The central moments were further normalized as shown below:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j)/2+1}} \tag{9}$$

The Hu moments are advanced image moments. The Hu moments are a set of seven values calculated using central moments. These are invariant to several image transformations. The first six moments have been shown to be invariant to translation, scale, rotation, and reflection; the seventh moment changes sign for image reflection. This feature is very helpful in distinguishing tractor and trailer classes.

These features were concatenated with original truck classification features and then used for generating a decision tree for tractor (or trailer) classification. The main motivation for

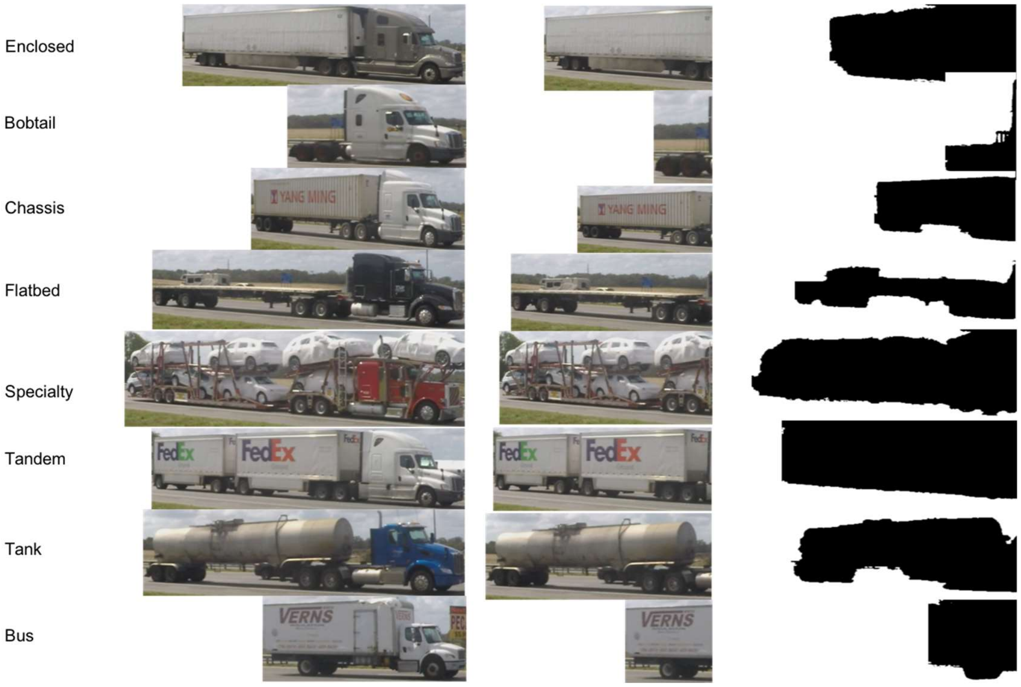


Figure 9: *Sample-derived trailers and their corresponding contours. The results were derived from videos taken at I-75 site 9956.*

using the original truck classification feature was that several of the truck classes were highly correlated to the tractor and trailer types. For example, most of the class 6 trucks were bobtail truck (one of the trailer types). RV trucks (one of the tractor types) generally belonged to class 5 truck category. Thus, we expected that utilizing the truck type would implicitly help to determine the correct tractor and trailer types.

3.2.5 Results for Tractor and Trailer Classification

Results from the tractor classification are shown in Table 5. Most of the tractor samples came from sleeper and day cab categories. Overall, we were able to obtain an average accuracy of 85.14%.

Results for the trailer classification derived from the FDOT dataset are presented in Table 6. The average accuracy was around 76%. The main challenge was that the "Enclosed" class is a significantly large fraction of the data. So, we decided to annotate other data to make the class more balanced. As shown in Table 7, we combined the enclosed class and the chassis class because the subtle difference is not trivial to distinguish. We then obtained an

***Table 5:** Performance evaluations on the tractor classification datasets. The benchmark dataset is built by FDOT on videos from I-75 site 9956.*

Train Accuracy	0.986	0.981	0.992	0.978	0.983
Test Accuracy	0.859	0.842	0.864	0.828	0.864

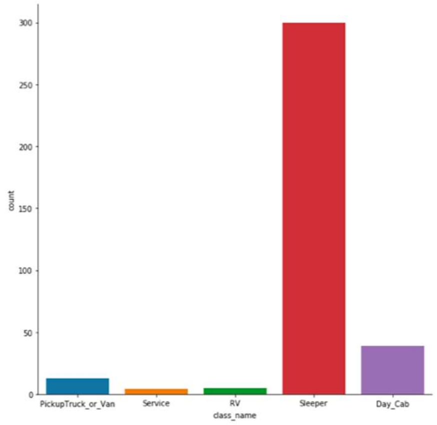


***Figure 10:** Tractor class data distribution.*

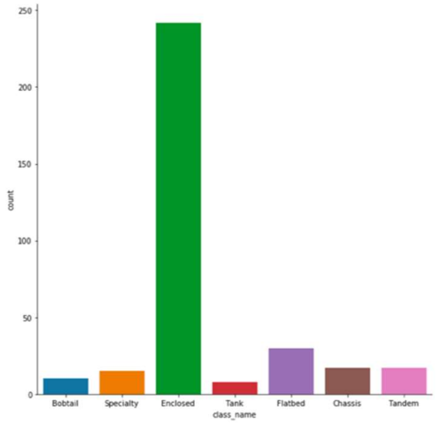


***Figure 11:** Trailer class data distribution.*

***Table 6:** Performance evaluations on the trailer classification datasets. The benchmark dataset is built by FDOT on videos taken at I-75 site 9956.*

Train Accuracy	0.972	0.933	0.964	0.972	0.958
Test Accuracy	0.765	0.781	0.762	0.795	0.745

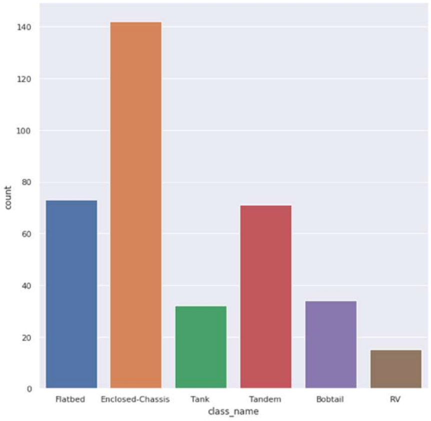


Figure 12: *Trailer class data distribution for 'DT' evaluation.*

Table 7: *Performance evaluations on the trailer classification datasets. The dataset was annotated by UF on videos from I-75 site 9956. 'DT' and 'RF' denote the decision tree classifier and the random forest classifier, respectively. 'RF' results take the 'car hauler' into consideration and treat it as an individual trailer class.*

DT Train Accuracy	0.989	0.989	0.978	0.981	0.992
DT Test Accuracy	0.867	0.852	0.853	0.845	0.850
RF Train Accuracy	0.997	0.992	0.997	1.000	0.997
RF Test Accuracy	0.904	0.915	0.889	0.902	0.915

average accuracy of 85.34% using a decision tree classifier.

Considering that trailer attributes are crucial for commodity classification, we further improved classification by adding more specialized features and a more powerful classifier, and we obtained an average accuracy of 90.05%. To determine the ’car hauler’ trailer class, we ran an object detector over the truck image to obtain object candidates. Ideally, if the truck was not a ’car hauler’, the detector would have reported one truck object for thewhole truck image. If the truck was carrying vehicles such as cars or small tractors, we expected to receive multiple object predictions from the object detector. Based on the number of reported vehicle objects within this single truck image, we made an estimation that it was likely to be a ’car hauler’ if the number was greater than 1. Random forest was introduced to further improve the model performance. We simply considered a random forest classifier as a collection of decision trees whose results are aggregated into one final result. The Random forest was considered as a stronger modeling technique and much more robust than a single decision tree. As multiple decision trees were aggregated, overfitting the dataset was less likely, and therefore, better results were yielded.

3.2.6 Refrigerator Unit Detection Component

A refrigerator truck is designed to carry perishable freight. The refrigerator unit is usually attached to the trailer unit. Based on visual exploration of underlying datasets, we observed that the unit appears with a relatively consistent pattern. Such patterns have the potential of being learned by deep learning. We developed a deep learning approach for refrigerator unit detection similar to the approach that we used for estimating the vehicle wheels. The image annotation tool was utilized to annotate enough refrigerator unit training data, thereby obtaining an accurate refrigerator model. Figure 31 indicates that the model was able to detect refrigerator units from truck images. We obtained > 95% accuracy on the FDOT dataset comprising videos from I-75 site 9956.

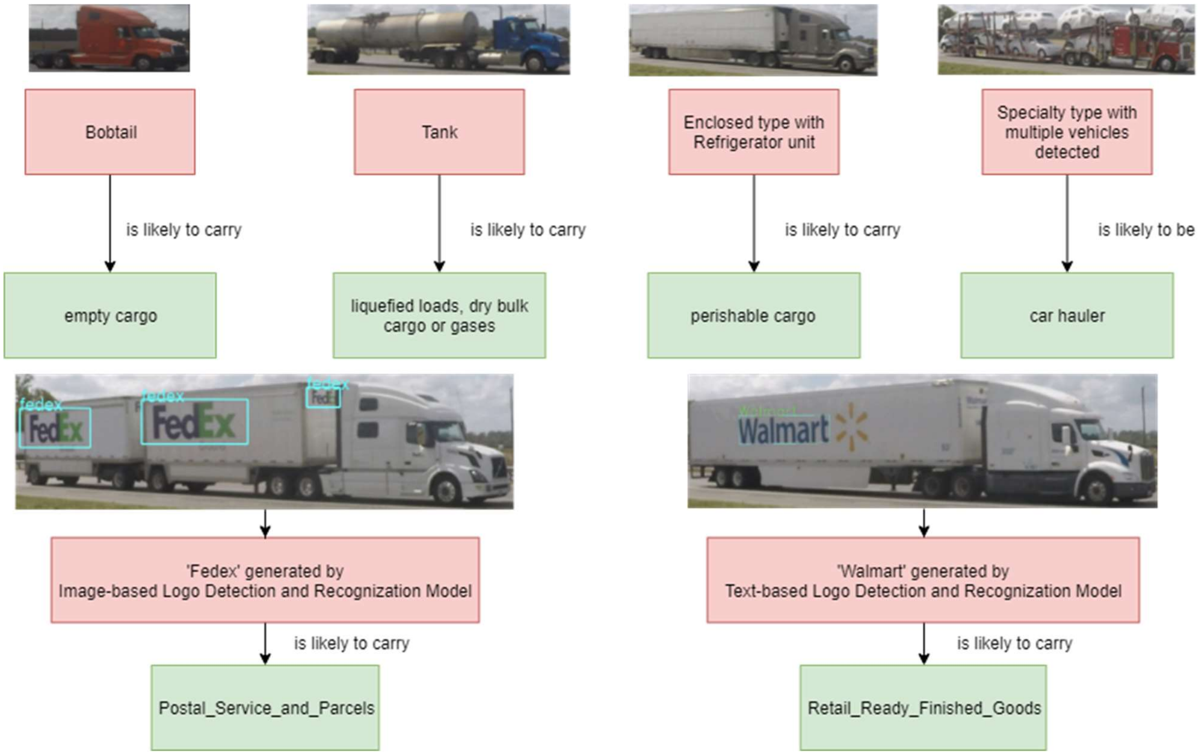

Bobtail
is likely to carry
empty cargo
Tank
is likely to carry
liquefied loads, dry bulk cargo or gases
Enclosed type with Refrigerator unit
is likely to carry
perishable cargo
Specialty type with multiple vehicles detected
is likely to be
car hauler
'Fedex' generated by Image-based Logo Detection and Recognization Model
is likely to carry
Postal_Service_and_Parcels
'Walmart' generated by Text-based Logo Detection and Recognization Model
is likely to carry
Retail_Ready_Finished_Goods

Figure 13: *Typical relations between trailer types or logos and commodities. Considering that a large fraction of trucks are enclosed, one approach to figure out the cargo in enclosed trucks is based on logo and text information shown on the trucks (if available). Having obtained the company name, an North American Industry Classification System (NAICS) code lookup table can be utilized to find the commodity.*

3.3 Commodity Classification

Trailer type is an important piece of information in determining the type of commodity in the trailer (Figure 13). Consequently, for detected trucks, only after a trailer is also detected could we continue the process of commodity detection. For many trailers, the corresponding commodity could be directly determined by their type. In case of enclosed trailers, logo detection, recognition, and database lookup was the primary way of determining commodity type and is the focus of the rest of this section. The overall pipeline can therefore be summarized as follows: (i) truck detection from video, (ii) truck identification, (iii) trailer identification, (iv) potential logo detection, (v) potential logo recognition, and (vi) NAICS database lookup for commodity identification.

The overall goal of the entire project was commodity identification using information obtained

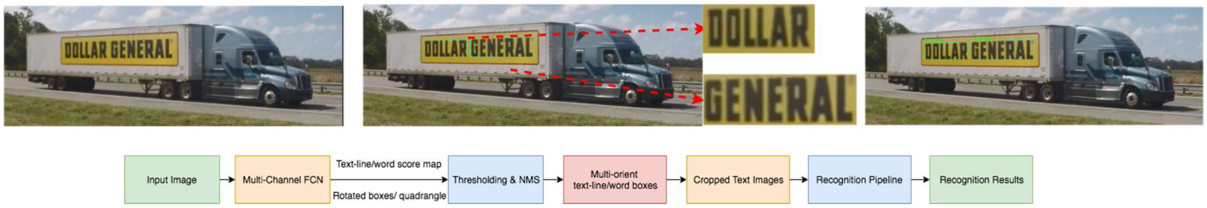


Figure 14: *The recognition pipeline of truck images, which corresponds to Task 3, subtask 1, text information retrieval.*

from freeway video.

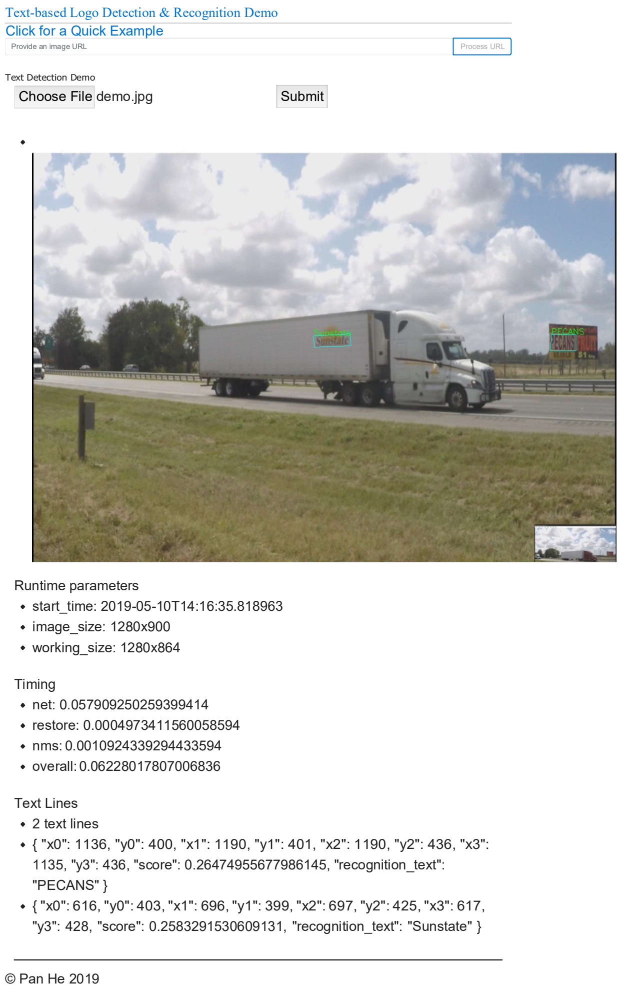
Because the camera was positioned to mainly obtain information from the side of the trucks passing on the freeway (as opposed to information from the rear), the process of identifying commodities was fundamentally restricted by the types of vendor image, logo, or text information that could be gleaned from the trucks themselves.

Below, we describe the process of logo detection and recognition. Different solutions are presented because multiple scenarios can be gleaned from the data. Sometimes, the logo appeared as text on the side of the truck (cf. Figure 15). This was the most straightforward case. In other cases, the logo appeared as a brand image (cf. Figure 18). This was less straightforward, but identification proceeded via brand logo image recognition. Finally, some logos appeared as stylized text within an image. This idiosyncratic case was the most difficult and appeared at a reasonable frequency.

3.3.1 Text-based Logo Detection and Recognition

We begin with a description of commodity identification via text detection and recognition—the most straightforward case.

Our first component for commodity classification was a solution to text detection and recognition. We developed state-of-the-art solutions, extending our previous research work CTPN [36] for text detection and DTRN [37] for text recognition. The entire pipeline is shown in Figure 14. Given a cropped truck image (generated by our previous truck models), we forwarded it into a multi-channel FCN model to get a text line/word score map. A post-processing step followed to filter out overlapped detection results by applying the standard NMS technique. After this step, we obtained results of text line/word locations represented by oriented bound-



demo.jpg

Runtime parameters

- start_time: 2019-05-10T14:16:35.818963
- image_size: 1280x900
- working_size: 1280x864

Timing

- net: 0.057909250259399414
- restore: 0.0004973411560058594
- nms: 0.0010924339294433594
- overall: 0.06228017807006836

Text Lines

- 2 text lines
- { "x0": 1136, "y0": 400, "x1": 1190, "y1": 401, "x2": 1190, "y2": 436, "x3": 1135, "y3": 436, "score": 0.26474955677986145, "recognition_text": "PECANS" }
- { "x0": 616, "y0": 403, "x1": 696, "y1": 399, "x2": 697, "y2": 425, "x3": 617, "y3": 428, "score": 0.2583291530609131, "recognition_text": "Sunstate" }

© Pan He 2019

Figure 15: *The Web demo of text-based logo detection and recognition developed by us. After uploading the truck image to our deployed server, we can return the recognition result packaged with JavaScript Object Notation (JSON) format.*

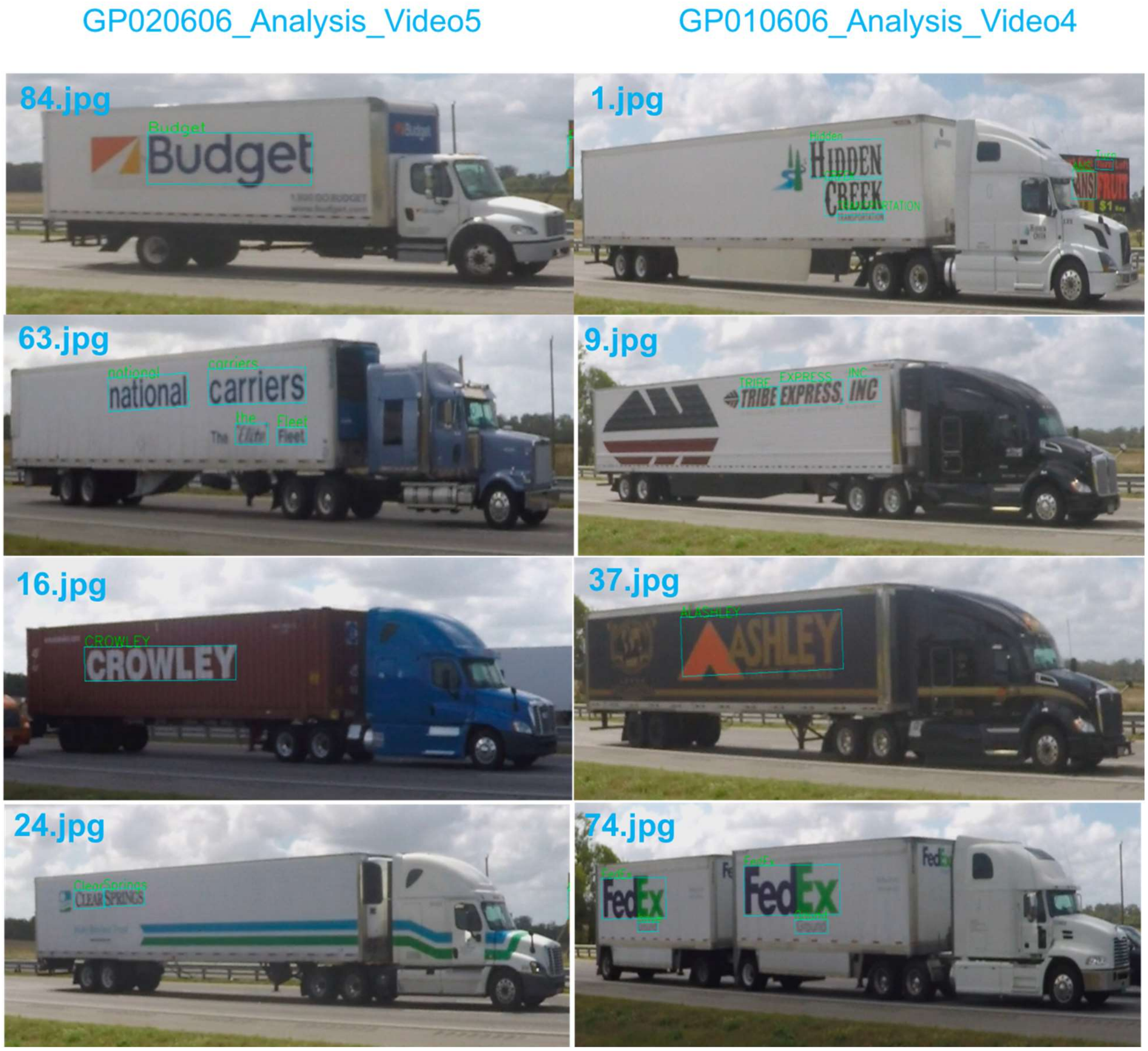


Figure 16: *Results for our developed algorithms for text detection and recognition of videos from I-75 site 9956. The developed algorithms achieved a high recall with a competitive recognition accuracy. Notice that some of the recognition results missed or wrongly predicted one or a few characters, which in reality should not cause many problems because the recognition results are further processed by matching the most similar results.*

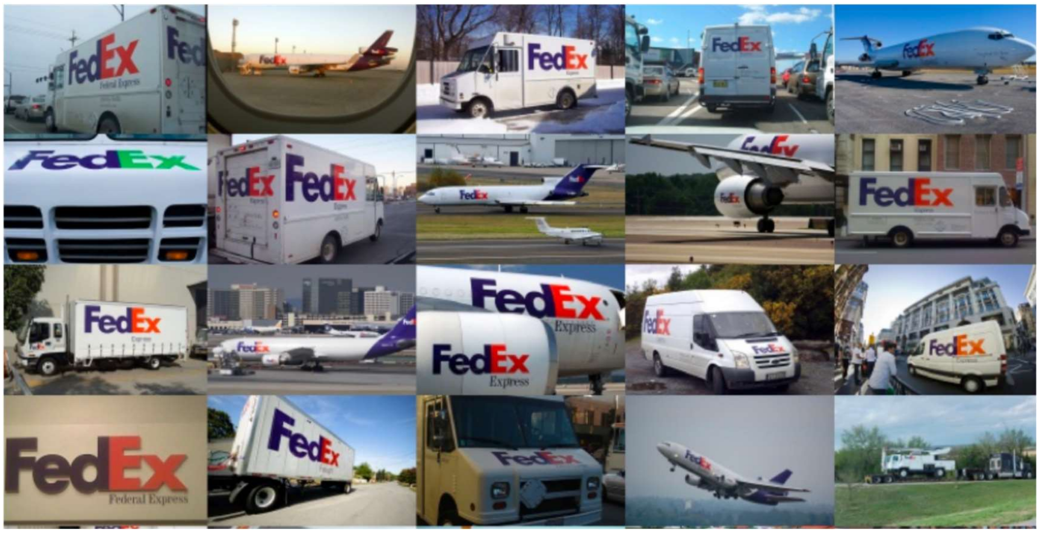


Figure 17: *Training samples from the dataset in Romberg et al. [2]. This public dataset is utilized to train our universal logo detector.*

ing boxes. Cropped images containing pure text were then available and further processed by a recognition model to get recognition results. In the final stage, word correction and string matching techniques were applied to match the result to predefined vendor lists (stored in a database). We would have been able to obtain the final company names from the text or logo information extracted from the truck images if a comprehensive database had been available. For the purposes of this project, we demonstrated the use of this technique on a selected set of vendor-logo pairs.

We show results of the approach in Figure 16. The developed algorithms achieved a high recall with a competitive recognition accuracy. Notice that some of the recognition results missed or incorrectly predicted one or a few characters, which in reality should not cause problems because the recognition results are further processed using a number of publicly available spelling correction methods.

3.3.2 Image-based Logo Detection and Recognition

Some of the logos do not contain text (or the text is complex with stylized fonts) and have to be recognized as entire images. Deriving the company name from such logos is a challenging object recognition and classification problem. A logo can be conceptualized of as a brand image expression, comprising a (stylized) letter or text; a graphical figure for figures;

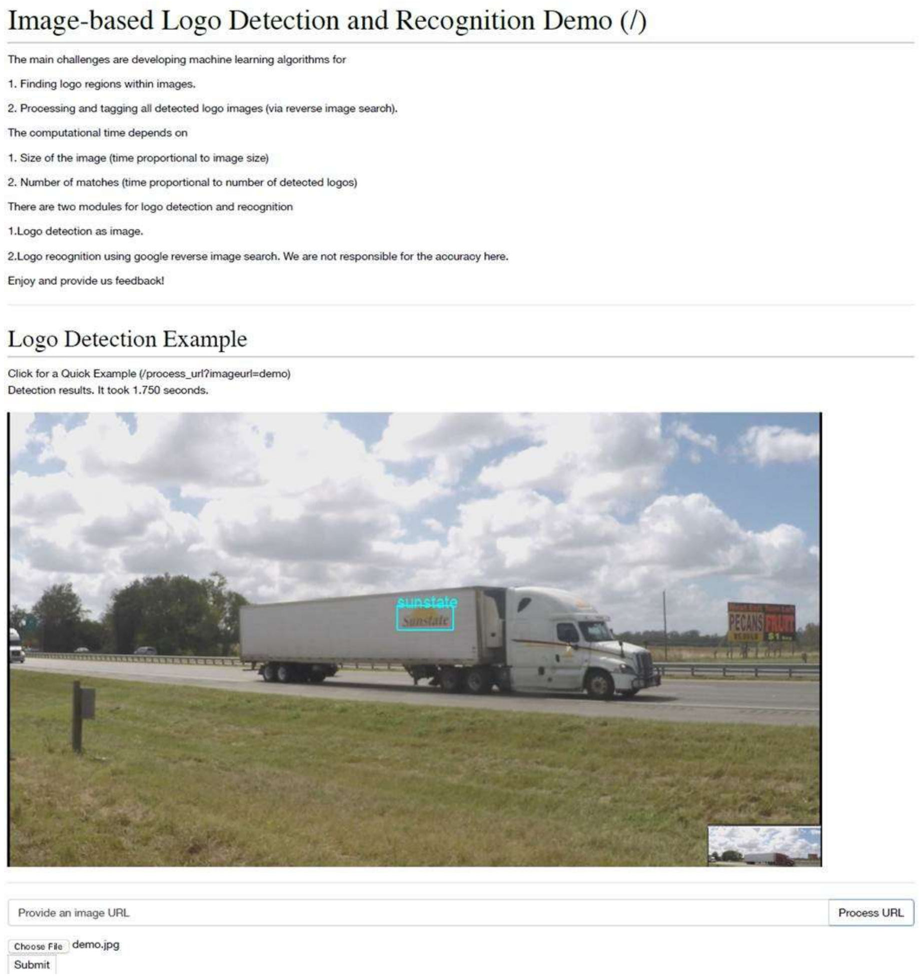


Image-based Logo Detection and Recognition Demo (/)

The main challenges are developing machine learning algorithms for

1. Finding logo regions within images.

2. Processing and tagging all detected logo images (via reverse image search).

The computational time depends on

1. Size of the image (time proportional to image size)

2. Number of matches (time proportional to number of detected logos)

There are two modules for logo detection and recognition

1.Logo detection as image.

2.Logo recognition using google reverse image search. We are not responsible for the accuracy here.

Enjoy and provide us feedback!

Logo Detection Example

Click for a Quick Example (/process_url?imageurl=demo)
Detection results. It took 1.750 seconds.

Figure 18: *The Web demo of image-based logo detection and recognition developed by us.*

or a combination thereof [3]. Additionally, many logo images vary significantly in color and contain specialized, unknown fonts. Furthermore, it is difficult to guarantee their context or placement because logos can be placed anywhere in the images. This problem is worsened by low image resolution, poor light/weather condition, and variable view angle.

Previous work on logo detection assumed that large training datasets for each logo class were available with fine-grained bounding box annotations. Such assumptions are often invalid in realistic scenarios where it is impractical to exhaustively label fine-grained training data for every new class. In the following, we present three potential solutions to address this, along with their corresponding advantages and disadvantages.

Solution 1: Logo Detector with Fixed Classes. Our first implemented solution was a logo detector with predefined logo classes. The process was to collect as many as possible training samples specialized for a certain logo (such as FedEx shown in Figure 17) and train a detector for this logo class.

- **Advantage**. The main advantage of our first solution was its simplicity in model design. Users already know in advance their interested logos and brands and ignore others. Once we established a fixed set of logo classes and trained the model based on them, the model successfully recognized logo images belonging to these classes. The design also simplified the whole algorithm, thus showing a faster inference speed compared to other solutions mentioned in the following sections.
- **Disadvantage**. The drawback for solution 1 was that the whole model cannot localize and recognize logo images not previously seen and not belonging to predefined logo classes. The model must be retrained when new logo classes are added.

Solution 2: Universal Logo Detector with Few-shot Recognizer. Our second solution was to scale up the problem and develop a generic logo detector for dynamic real-world applications. This method does not require fine-grained labelled training data for new incoming logo classes. As shown in Figure 19, we designed a universal logo detector (ULD) to localize any potential logos. The training was based on data images with bounding box annotations of logos (some samples are shown in Figure 17). Once we obtained the location of a logo, we cropped around the logo and forwarded the resulting image to a few-shot logo recognition model to extract logo/brand features. The recognition model first applied a

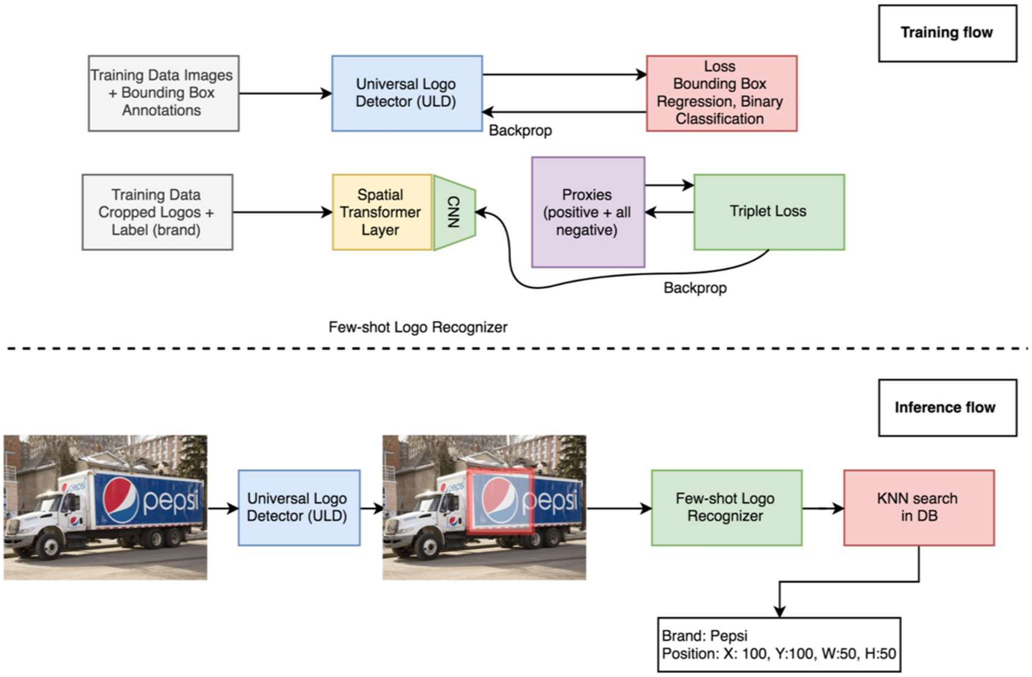

Training flow
Training Data Images + Bounding Box Annotations
Universal Logo Detector (ULD)
Loss Bounding Box Regression, Binary Classification
Backprop
Training Data Cropped Logos + Label (brand)
Spatial Transformer Layer
CNN
Proxies (positive + all negative)
Triplet Loss
Backprop
Few-shot Logo Recognizer
Inference flow
Universal Logo Detector (ULD)
Few-shot Logo Recognizer
KNN search in DB
Brand: Pepsi
Position: X: 100, Y:100, W:50, H:50

Figure 19: *The flow diagram of solution 2 on both training and inference. Inspired by Fehervari and Appalaraju [3], we train our few-shot model, which was used to compute the triplet loss. In the inference stage, we utilized the trained universal logo detector to get positions of interested logos. Then, the few-shot logo recognizer was applied to extract brand features. Finally, this was compared with database entries using a KNN search to get the final results. The flow chart has been taken from [3].*

spatial transformation layer to rectify logos with affine transformations. Then a triplet loss with proxies strategy was followed to backpropagate errors. This strategy helped learn a more discriminative classifier and feature representation once we had trained both the logo detector and logo recognizer. In the test stage, given new unseen logo images, we were able to extract brand/logo features. They were treated as the query feature and were compared to precomputed features of each company stored in the database. A k-nearest neighbors (KNN) search output the best matching result.

- **Advantage.** The main advantage of our second solution was that it is a generic logo detector that can detect any potential logo image, i.e., a universal logo detector. Some of the results for the FDOT dataset are shown in Figure 20. The model was able to quickly adapt to new unseen classes due to our few-shot learning schema.
- **Disadvantage.** The drawback for solution 2 was that few samples for new classes

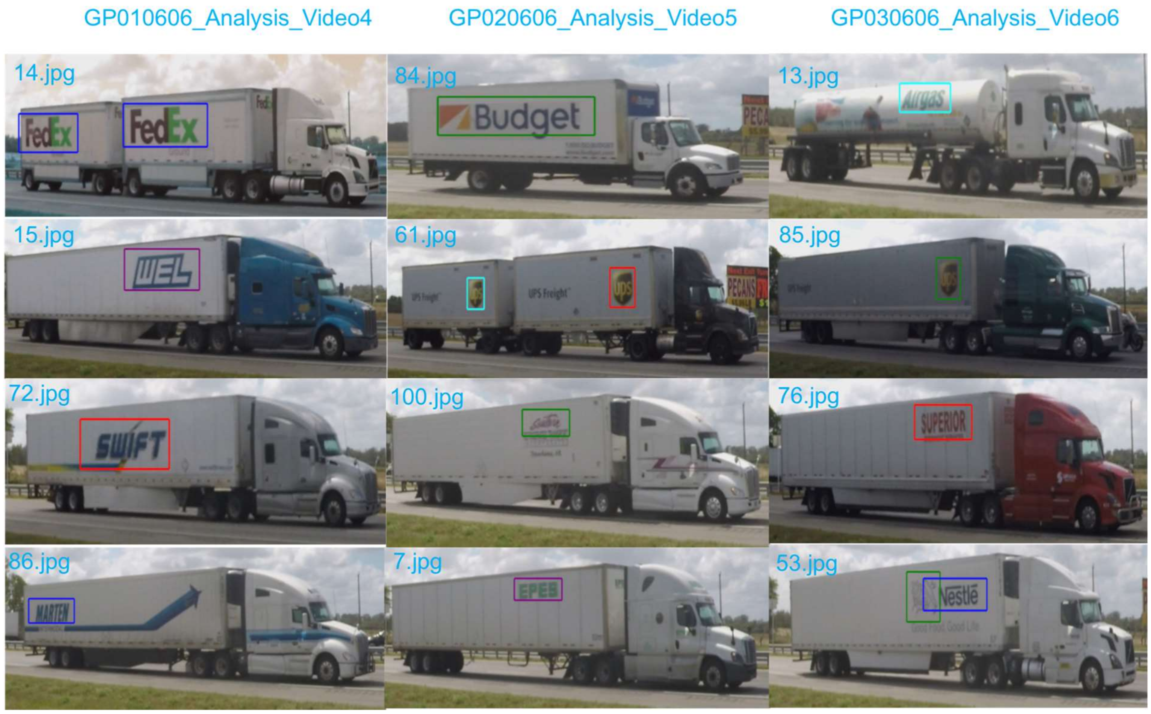


Figure 20: *Samples of logo detection from videos from I-75 site 9956.*

must still be manually annotated. Also, the performance is usually not satisfactory due to limited training samples.

Solution 3: Universal Logo Detector with Reverse Image Search. This approach is a content-based image retrieval (CBIR) query approach in which we provide the system with a sample image (search query) to search related concepts about this image. For example, Google's Search by Image allows users to search for related images from a large database. It provides an interface for loading an image or image URL. Google analyzes the submitted picture and comparing it to a large number of images in Google databases, and returning similar images and their annotations. Images on the Internet usually contain metadata information, such as caption, title location, label, or URL. Formally, there are three main categories of image metadata:

- **Technical metadata**. This is camera generated and contains information. Examples of such information include aperture, shutter speed, focal depth, and resolution. Additional information includes date, time, and GPS location of image creation.
- **Descriptive metadata**. This information corresponds to the name of the image creator, keywords related to the image, user comments, etc. Such information can be useful

for image searches.

- **Administrative metadata**. This includes licensing rights, restrictions on reuse, owner contact information, etc.

The pipeline for solution 3 is outlined in Figure 34 and contains two parts: universal logo detector and reverse image search. The universal logo detector outputs the same results as solution 2, providing essential visual logo information to the later stage. For the reverse image search, we automated the process by developing an HTMLParser that can parse results sent back from the Google Image Search (or similar image search engines).

- **Advantage**. The universal logo detector with reverse image search is different from solution 1 and solution 2 in that it does not require many annotations for new logo classes. The reverse image search directly utilizes the existing commercial service. It can return reasonable results with richer meta information related to the logo image. For example, we were able to process recommendation URLs of the company or brand related to the logo.
- **Disadvantage**. An additional cost is the maintenance of our developed HTMLParser because the service provider (Google) constantly changes their application program interfaces (APIs). Because service provider issues are beyond our control, reverse image search would sometimes not return results oriented to brand and logo.

We developed a logo detector with fixed classes; a universal logo detector; and a universal logo detector with reverse image search. These are shown in Figure 34. Once we forwarded truck images into our universal logo detector, we were able to estimate the rough location for each potential logo image. We then cropped around logo regions from the original truck image, which can be considered as a ’zoom-in’ operation for the truck image. This process enabled the model to focus on pure logo content information and ignore non-relevant background distraction information.

We now describe preliminary proof of concept results for the third approach. (This approach needs to be validated much more thoroughly before we can make reasonable claims regarding its effectiveness.)

Using the generic logo detector, we obtained patches containing target logos. The remaining

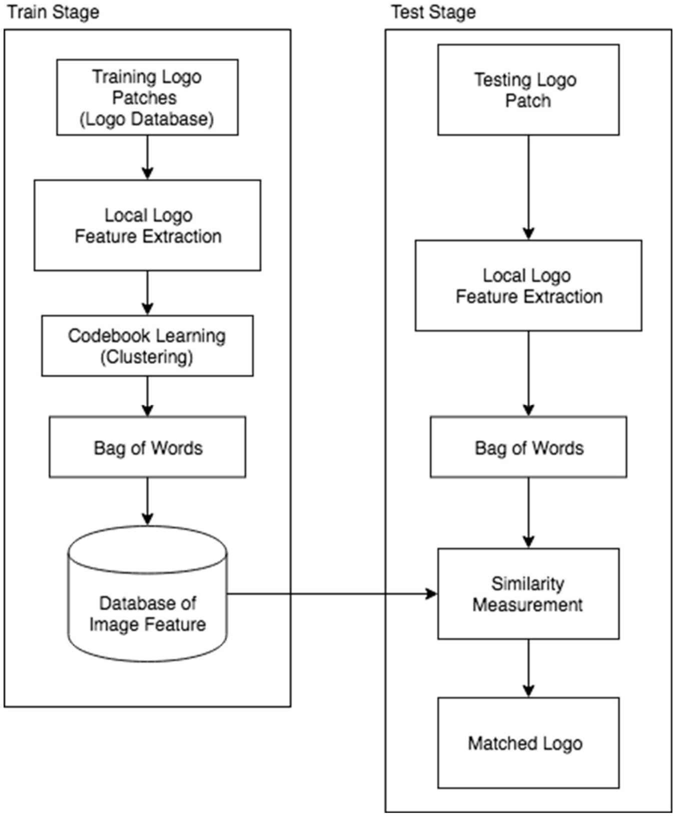

Train Stage
Test Stage
Training Logo Patches (Logo Database)
Local Logo Feature Extraction
Codebook Learning (Clustering)
Bag of Words
Database of Image Feature
Testing Logo Patch
Local Logo Feature Extraction
Bag of Words
Similarity Measurement
Matched Logo

Figure 21: *BoW model for logo recognition and matching.*

problem was to classify each patch into pre-defined classes. To this end, we developed an approach based on a Bag of Words (BoW) model. The overall pipeline is shown in Figure 21.

In the training stage, we first extracted the local features using the SIFT (scale-invariant feature transform) descriptor. This resulted in a large number of features (several hundred) for each image patch. These were then quantized by using a BoW approach. This was accomplished by training a visual codebook, followed by quantization of local features to visual words in the codebook. After finding the representation of all training patches, we stored the mean representation of each class in a database as the final representation of that class.

In the testing stage, for each testing patch, we computed local features using the same SIFT descriptors and then used the trained codebook to get the compact representation. After the representation of a testing patch was obtained, we assigned the class label by comparing all representations in the database and returned the class of the most similar one.

The evaluation for the BoW model was conducted on 30 min of videos GOPR0604 and

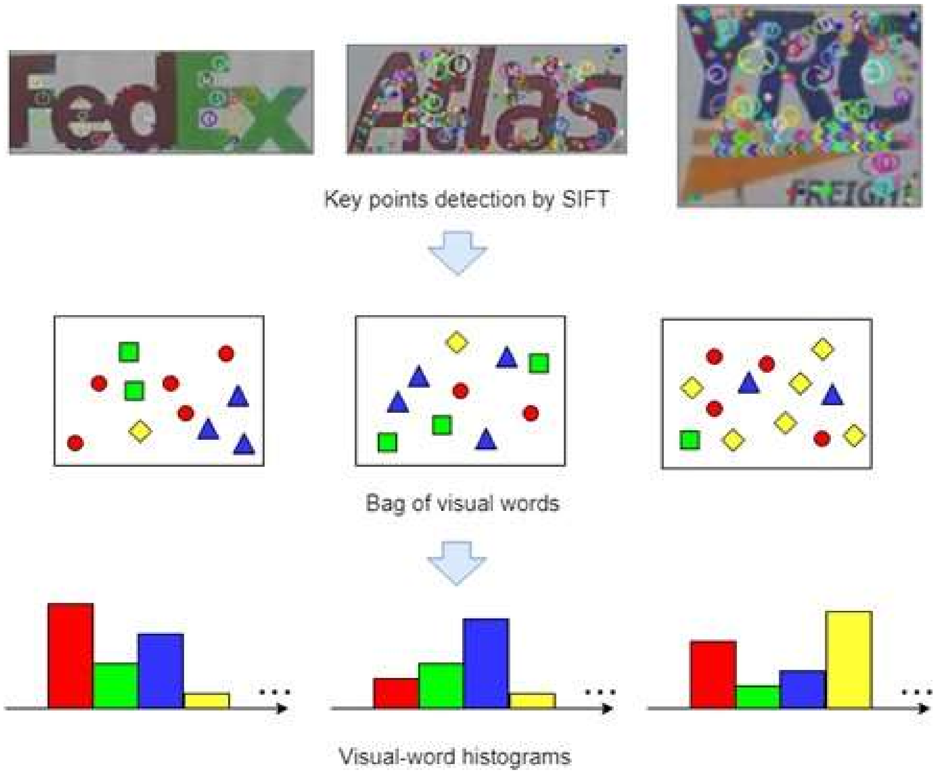


Figure 22: *Sample BoW features.*

GP010257. Logo images were first cropped and collected. The data were shuffled and split into 70% and 30% for training and testing purposes, respectively. To begin with, the ten categories, namely FedEx, RBI, XTRA, Landstar, UPS, Heartland Express, Premier, Southern AG, Dollar General, and US Foods were collected for the evaluation. The statistics of samples for each class are shown in Table 8.

We extended the algorithm by ensemble learning with soft voting that arrives at the best result by weighting or averaging out the probabilities calculated by individual algorithms. In addition to the bag of visual words features, we utilized three extra types of features: deep learning features, shape context features, and color histogram features.

The deep learning features utilized transfer learning. We extracted features for each input logo image by forwarding it into a pretrained ResNet model, resulting in a vector length of 256. This vector forms a compact representation of the logo image. For the shape context features, we mainly computed these features based on the contours of logo images because each logo class has a certain contour pattern. Color histogram features are based on the color pattern of each logo class. We began by converting the original RGB image into

	Train Dataset	Test Dataset
FedEx	996	425
XTRA	385	164
Landstar	302	134
UPS	282	121
RBI	276	120
Dollar General	272	111
Heartland Express	208	90
Southern AG	167	71
Premier	145	67
US Foods	131	53

Table 8: *Statistics of samples for each class.*

HSV (hue, saturation, value) image color space. We then focused on the H (Hue) image channel as it is less sensitive (if not invariant) to lighting variations. Based on the H channel, we computed the image histogram. Some results are shown in Figure 23. Logos within the same class (FedEx) illustrate a similar histogram pattern, while a different logo class (Dollar General) is drastically different in the histogram pattern. Thus, color features were discriminative features for logo classification.

With these four features, we trained four individual support vector machine (SVM) classifiers, each with a prediction probability of the query image belonging to a certain class. Finally, we

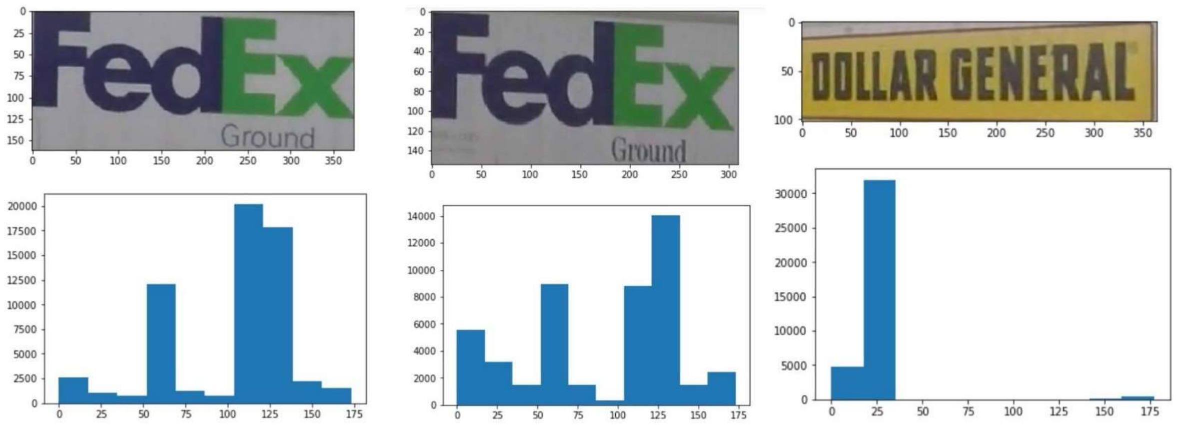


Figure 23: *Sample color features using hue image histogram.*

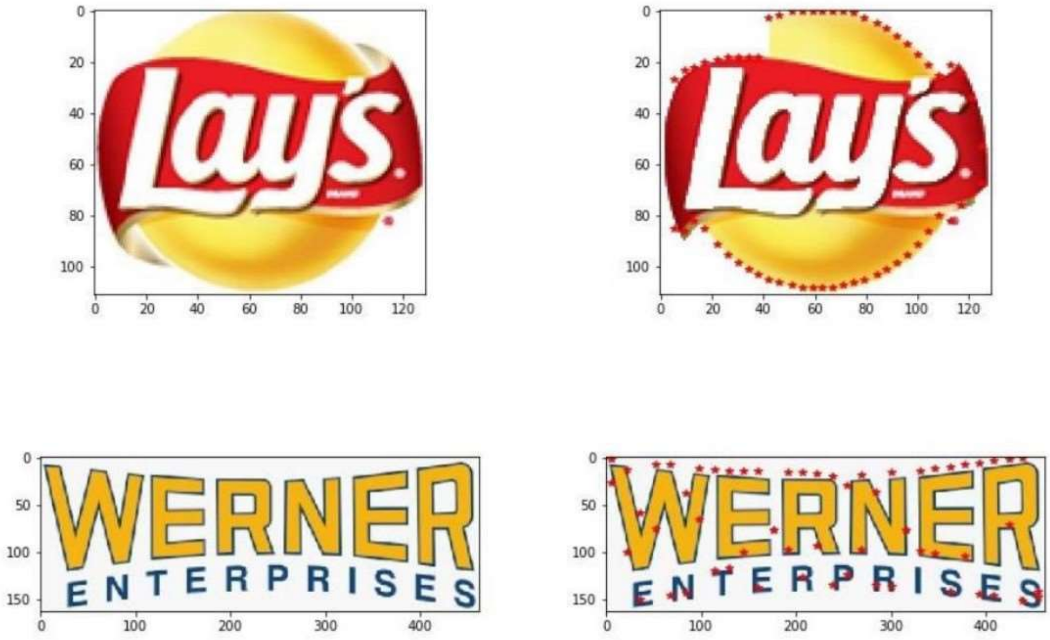


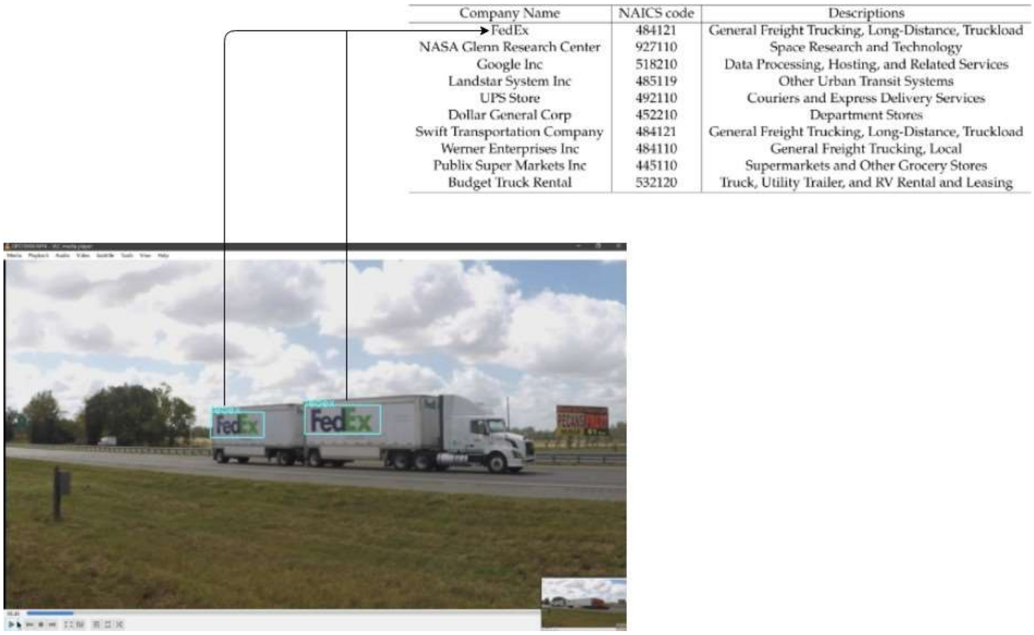
Figure 24: *Sample features using shape context method. The red dots indicate the shape context features which are computed based on the contours.*

averaged the results from these classifiers with different weights, where the weights were determined by how well each classifier performed on the test set. Final results are shown in Table 9. We achieved the best result by combining all four features with a top-1 accuracy of 84% and a top-3 accuracy of 98%.

The above description and results of the third approach to logo identification clearly serve to demonstrate its potential. However, we stress that these results should be viewed as somewhat anecdotal. A more systematic effort is required, followed by careful validation

Feature Types	Top 1 Accuracy	Top 3 Accuracy
Bag of Words features	0.83	0.96
CNN features	0.63	0.71
Shape features	0.31	0.52
Color features	0.14	0.36
Ensemble	0.84	0.98

Table 9: *Logo matching results: We achieved the best result by combining all four features with a top-1 accuracy of 84% and a top-3 accuracy of 98%.*



Company Name	NAICS code	Descriptions
FedEx	484121	General Freight Trucking, Long-Distance, Truckload
NASA Glenn Research Center	927110	Space Research and Technology
Google Inc	518210	Data Processing, Hosting, and Related Services
Landstar System Inc	485119	Other Urban Transit Systems
UPS Store	492110	Couriers and Express Delivery Services
Dollar General Corp	452210	Department Stores
Swift Transportation Company	484121	General Freight Trucking, Long-Distance, Truckload
Werner Enterprises Inc	484110	General Freight Trucking, Local
Publix Super Markets Inc	445110	Supermarkets and Other Grocery Stores
Budget Truck Rental	532120	Truck, Utility Trailer, and RV Rental and Leasing

Figure 25: *Example of our solution for commodity classification. Given the FedEx truck image, our developed algorithm predicted and placed the label 'FedEx' over the logo. It was then used as the query string for the U.S. company list to obtain results such as industry information (General Freight Trucking, Long-Distance, Truckload) and NAICS code 484121.*

before this approach can take its place alongside the other two approaches.

In our project, after developing all the aforementioned algorithms, we were able to extract text and company information. Each detected truck potentially carries text that provides company information, which can help in figuring out the cargo. We can search for companies information via NAICS code lookup from vendors recognized from their corresponding text (identified from the side of the truck). It remains to build a database that can comprehensively cover vendors to link the companies and trucks to commodities and to then obtain the NAICS codes from which the commodity information can be approximately identified. We provide anecdotal results of a complete end-to-end solution at the end of this report.

Company Lookup Tool. NAICS is an industry classification system for grouping establishments into industries based on production processes used. It is a comprehensive system covering all economic activities.

Inspired by this, we developed our solution for commodity identification. It was based on results obtained from our previous two components mentioned in Section 3.3.1 and Sec-

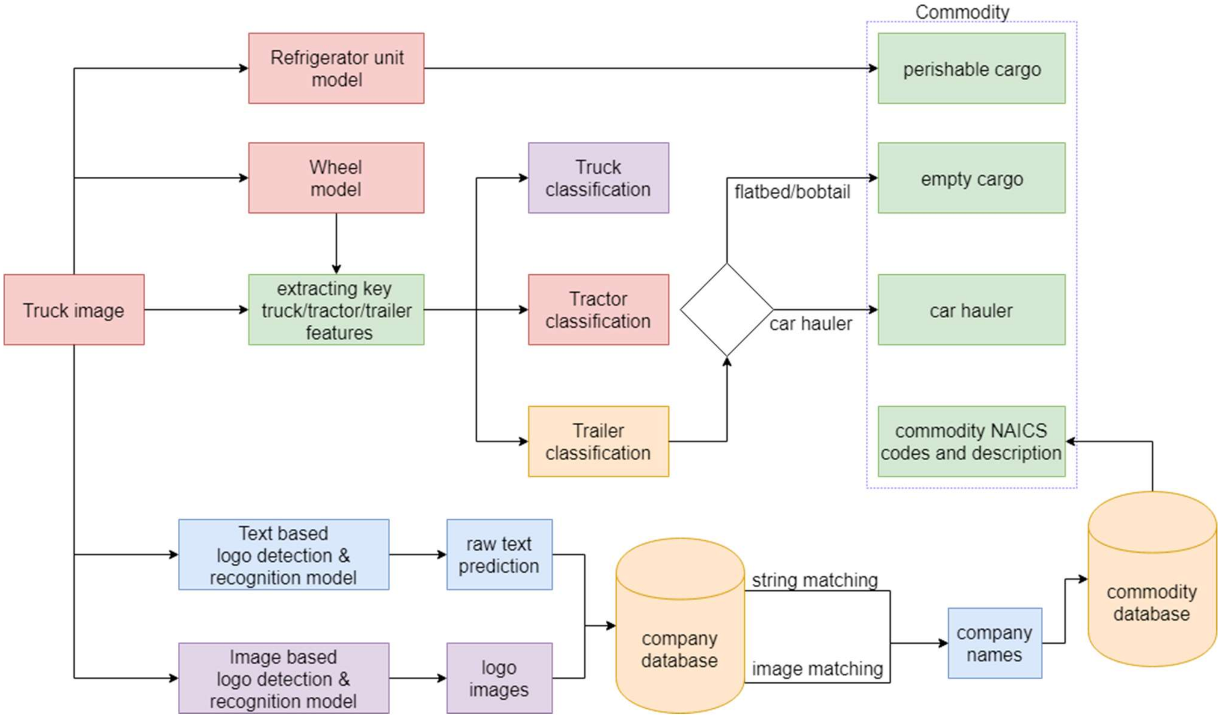


Figure 26: *End-to-end pipeline for determining the carried commodities.*

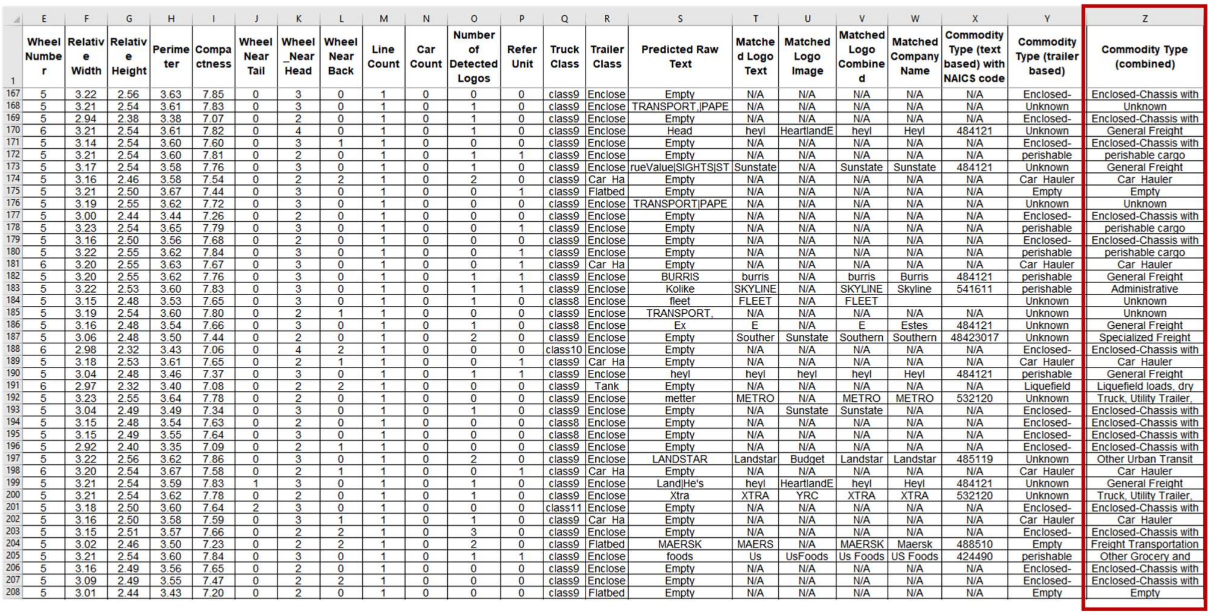
tion 3.3.2. Here, we show a proof of concept of our method. Once we extracted the text and company name, we forwarded it to our collected company list to search for the NAICS code and commodity information, as shown in Figure 25. Because the extracted text may not be precise and match the company name exactly, we also developed a string-matching method for finding the most similar name in the database.

In summary, we have made progress towards the following:

1. Developed new algorithms for image based logo classification. For 26 company logos, our accuracy is 83% (top 1) and 95% (top 3).
2. Improved classification accuracy of the trailer classifier using alternate decision tree approaches.
3. Performed end-to-end work flow that starts from a truck image and detects the potential commodity.

To determine the commodities carried on the trucks, we developed the following end-to-end pipeline (as shown in Figure 26):

1. Determine the key features using semantic segmentation. Detect wheels and refriger-



	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Wheel Number	Relative Width	Relative Height	Perimeter	Compactness	Wheel Near Tail	Wheel _Near Head	Wheel Near Back	Line Count	Car Count	Number of Detected Logos	Refer Unit	Truck Class	Trailer Class	Predicted Raw Text	Matched Logo Text	Matched Logo Image	Matched Logo Combined	Matched Company Name	Commodity Type (text based) with NAICS code	Commodity Type (trailer based)	Commodity Type (combined)
167	5	3.22	2.56	3.63	7.85	0	3	0	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
168	5	3.21	2.54	3.61	7.83	0	3	0	1	0	1	0	class9	Enclose	TRANSPORT,|PAPE	N/A	N/A	N/A	N/A	N/A	Unknown	Unknown
169	5	2.94	2.38	3.38	7.07	0	2	0	1	0	1	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
170	6	3.21	2.54	3.61	7.82	0	4	0	1	0	1	0	class9	Enclose	Head	heyl	HeartlandE	heyl	Heyl	484121	Unknown	General Freight
171	5	3.14	2.54	3.60	7.60	0	3	1	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
172	5	3.21	2.54	3.60	7.81	0	2	0	1	0	1	1	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	perishable	perishable cargo
173	5	3.17	2.54	3.58	7.76	0	3	0	1	0	1	0	class9	Enclose	rueValue|SIGHTS|ST	Sunstate	N/A	Sunstate	Sunstate	484121	Unknown	General Freight
174	5	3.16	2.46	3.58	7.54	0	2	2	1	0	2	0	class9	Car Ha	Empty	N/A	N/A	N/A	N/A	N/A	Car Hauler	Car Hauler
175	5	3.21	2.50	3.67	7.44	0	3	0	1	0	0	1	class9	Flatbed	Empty	N/A	N/A	N/A	N/A	N/A	Empty	Empty
176	5	3.19	2.55	3.62	7.72	0	3	0	1	0	0	0	class9	Enclose	TRANSPORT|PAPE	N/A	N/A	N/A	N/A	N/A	Unknown	Unknown
177	5	3.00	2.44	3.44	7.26	0	2	0	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
178	5	3.23	2.54	3.65	7.79	0	3	0	1	0	0	1	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	perishable	perishable cargo
179	5	3.16	2.50	3.56	7.68	0	2	0	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
180	5	3.22	2.55	3.62	7.84	0	3	0	1	0	0	1	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	perishable	perishable cargo
181	6	3.20	2.55	3.63	7.67	0	3	0	1	0	0	1	class9	Car Ha	Empty	N/A	N/A	N/A	N/A	N/A	Car Hauler	Car Hauler
182	5	3.20	2.55	3.62	7.76	0	3	0	1	0	1	1	class9	Enclose	BURRIS	burris	N/A	burris	Burris	484121	perishable	General Freight
183	5	3.22	2.53	3.60	7.83	0	3	0	1	0	1	1	class9	Enclose	Kolike	SKYLINE	N/A	SKYLINE	Skyline	541611	perishable	Administrative
184	5	3.15	2.48	3.53	7.65	0	3	0	1	0	1	0	class8	Enclose	fleet	FLEET	N/A	FLEET			Unknown	Unknown
185	5	3.19	2.54	3.60	7.80	0	2	1	1	0	0	0	class9	Enclose	TRANSPORT,	N/A	N/A	N/A	N/A	N/A	Unknown	Unknown
186	5	3.16	2.48	3.54	7.66	0	3	0	1	0	1	0	class8	Enclose	Ex	E	N/A	E	Estes	484121	Unknown	General Freight
187	5	3.06	2.48	3.50	7.44	0	2	0	1	0	2	0	class9	Enclose	Empty	Souther	Sunstate	Southern	Southern	48423017	Unknown	Specialized Freight
188	6	2.98	2.32	3.43	7.06	0	4	2	1	0	0	0	class10	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
189	5	3.18	2.53	3.61	7.65	0	2	1	1	0	0	1	class9	Car Ha	Empty	N/A	N/A	N/A	N/A	N/A	Car Hauler	Car Hauler
190	5	3.04	2.48	3.46	7.37	0	3	0	1	0	1	1	class9	Enclose	heyl	heyl	heyl	heyl	Heyl	484121	perishable	General Freight
191	6	2.97	2.32	3.40	7.08	0	2	2	1	0	0	0	class9	Tank	Empty	N/A	N/A	N/A	N/A	N/A	Liquefield	Liquefield loads, dry
192	5	3.23	2.55	3.64	7.78	0	2	0	1	0	0	0	class9	Enclose	metter	METRO	N/A	METRO	METRO	532120	Unknown	Truck, Utility Trailer,
193	5	3.04	2.49	3.49	7.34	0	3	0	1	0	1	0	class9	Enclose	Empty	N/A	Sunstate	Sunstate	N/A	N/A	Enclosed-	Enclosed-Chassis with
194	5	3.15	2.48	3.54	7.63	0	2	0	1	0	0	0	class8	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
195	5	3.15	2.49	3.55	7.64	0	3	0	1	0	0	0	class8	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
196	5	2.92	2.40	3.35	7.09	0	2	1	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
197	5	3.22	2.56	3.62	7.86	0	3	0	1	0	2	0	class9	Enclose	LANDSTAR	Landstar	Budget	Landstar	Landstar	485119	Unknown	Other Urban Transit
198	6	3.20	2.54	3.67	7.58	0	2	1	1	0	0	1	class9	Car Ha	Empty	N/A	N/A	N/A	N/A	N/A	Car Hauler	Car Hauler
199	5	3.21	2.54	3.59	7.83	1	3	0	1	0	1	0	class9	Enclose	Land|He's	heyl	HeartlandE	heyl	Heyl	484121	Unknown	General Freight
200	5	3.21	2.54	3.62	7.78	0	2	0	1	0	1	0	class9	Enclose	Xtra	XTRA	YRC	XTRA	XTRA	532120	Unknown	Truck, Utility Trailer,
201	5	3.18	2.50	3.60	7.64	2	3	0	1	0	0	0	class11	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
202	5	3.16	2.50	3.58	7.59	0	3	1	1	0	1	0	class9	Car Ha	Empty	N/A	N/A	N/A	N/A	N/A	Car Hauler	Car Hauler
203	5	3.15	2.51	3.57	7.66	0	2	2	1	0	3	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
204	5	3.02	2.46	3.50	7.23	0	2	2	1	0	2	0	class9	Flatbed	MAERSK	MAERS	N/A	MAERSK	Maersk	488510	Empty	Freight Transportation
205	5	3.21	2.54	3.60	7.84	0	3	0	1	0	1	1	class9	Enclose	foods	Us	UsFoods	Us Foods	US Foods	424490	perishable	Other Grocery and
206	5	3.16	2.49	3.56	7.65	0	2	0	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
207	5	3.09	2.49	3.55	7.47	0	2	2	1	0	0	0	class9	Enclose	Empty	N/A	N/A	N/A	N/A	N/A	Enclosed-	Enclosed-Chassis with
208	5	3.01	2.44	3.43	7.20	0	2	0	1	0	0	0	class9	Flatbed	Empty	N/A	N/A	N/A	N/A	N/A	Empty	Empty

Figure 27: *Sample spreadsheet for commodity recognition.*

ator unit.

2. Use these to determine the trailer type.
3. If a logo is present, use text-based and image-based logo detection to determine the most likely company.
4. If the trailer type suggests “empty”, then report empty; If a trailer type suggests car hauler, report car hauling, etc.
5. If a trailer type is enclosed and logo is present, use logo detection to derive company name, and then use the NAICS lookup table to determine potential commodity.

A sample spreadsheet generated by our developed approaches is shown in Figure 27. Using the extracted features, we predict the commodity type as shown in column Z, using the built commodity database.

4 Conclusions

This report presents an automated system for detection, recognition, and classification of trucks that can be used to determine commodity types—indispensable downstream for tracking commodity movements. To accomplish this, a set of high resolution videos (made available by FDOT) using freeway roadside passive cameras were utilized.

The approaches rely on recent advances in deep learning for object detection and classification as well as traditional methods such as decision trees and geometric features. We developed deep learning algorithms that leveraged transfer learning to determine whether an image frame shows a truck and, if the answer is affirmative, localize the area from the image frame where the truck is most likely to be present. We developed a hybrid truck classification approach that integrates deep learning models and geometric truck features resulting in a method that achieves high accuracy. We also developed algorithms for recognizing and classifying various truck attributes such as tractor type, trailer type, and refrigeration units. Subsequently, logo and text information were extracted from the detected trucks through a two-step process: first, the existence of a logo was detected, followed by logo/text detection via text models and logo image matching algorithms. Three different logo classification algorithms were presented: one based on text recognition and two based on image matching. The first image-matching approach used Google reverse image search, whereas the second deployed a bag of words matching model.

All the results obtained by the developed models are summarized in Figure 28. These results show that our scheme for truck classification has > 90% accuracy for classifying trucks into one of the nine classes (FHWA classes 5 through 13) and is relatively independent of the actual camera angle and has potential for wide deployment. Additionally, our algorithms for trailer recognition have > 85% accuracy for classifying tractor and trailer types. Furthermore, we demonstrated a proof of concept system for detecting refrigeration units on trucks. Finally, we were able to demonstrate three different logo/text detection and classification approaches. These result in vendor identification via logo recognition. Once the vendor information is available, this can be associated with commodities via the NAICS database.

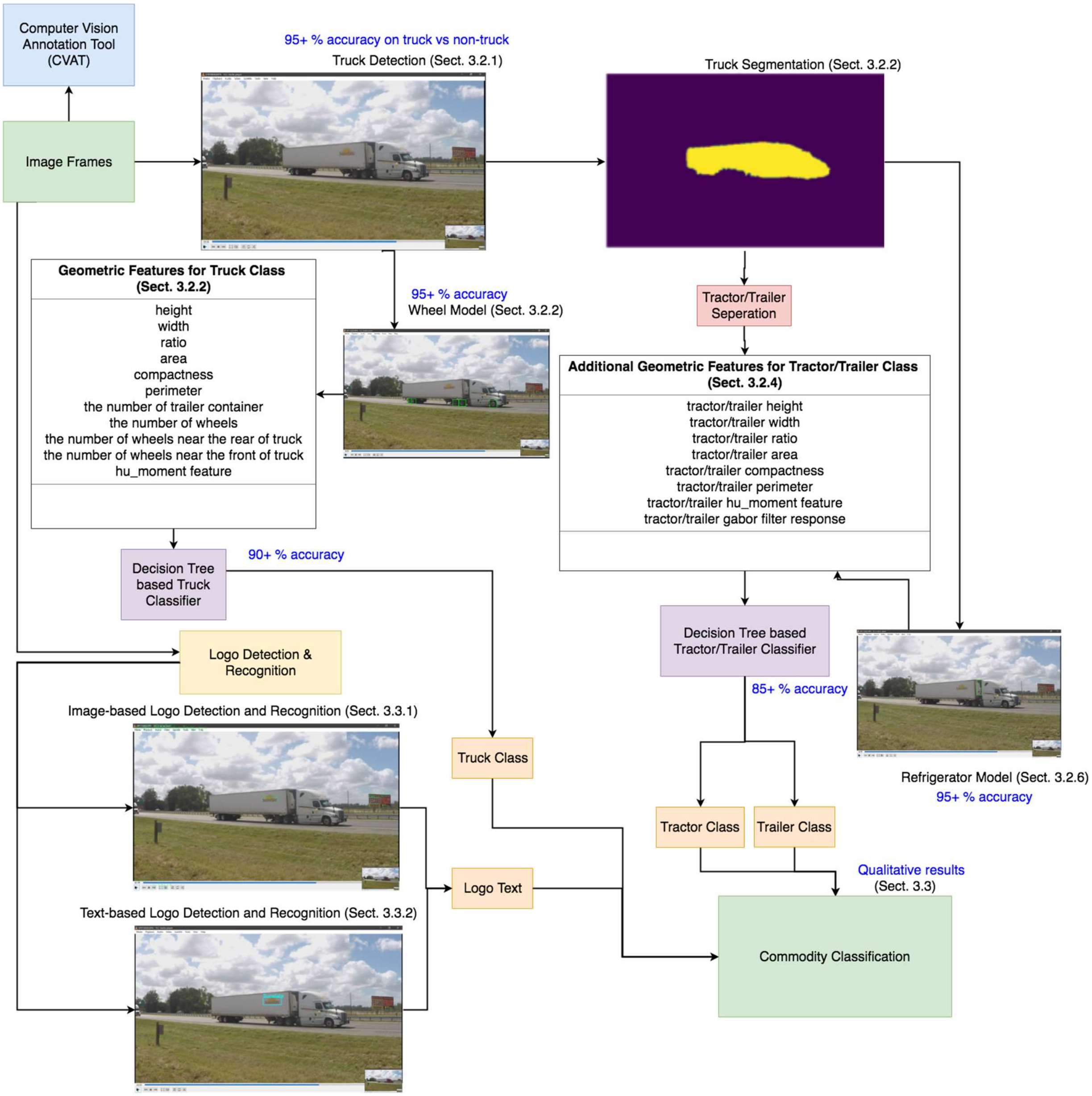

Computer Vision Annotation Tool (CVAT)
95+ % accuracy on truck vs non-truck
Truck Detection (Sect. 3.2.1)
Truck Segmentation (Sect. 3.2.2)
Image Frames
Geometric Features for Truck Class (Sect. 3.2.2)
height
width
ratio
area
compactness
perimeter
the number of trailer container
the number of wheels
the number of wheels near the rear of truck
the number of wheels near the front of truck
hu_moment feature
95+ % accuracy
Wheel Model (Sect. 3.2.2)
Tractor/Trailer Seperation
Additional Geometric Features for Tractor/Trailer Class (Sect. 3.2.4)
tractor/trailer height
tractor/trailer width
tractor/trailer ratio
tractor/trailer area
tractor/trailer compactness
tractor/trailer perimeter
tractor/trailer hu_moment feature
tractor/trailer gabor filter response
Decision Tree based Truck Classifier
90+ % accuracy
Decision Tree based Tractor/Trailer Classifier
85+ % accuracy
Logo Detection & Recognition
Image-based Logo Detection and Recognition (Sect. 3.3.1)
Truck Class
Refrigerator Model (Sect. 3.2.6)
95+ % accuracy
Tractor Class
Trailer Class
Logo Text
Qualitative results (Sect. 3.3)
Text-based Logo Detection and Recognition (Sect. 3.3.2)
Commodity Classification

Figure 28: *All the results obtained by the developed models.*

5 References

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[2] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol, “Scalable logo recognition in real-world images,” in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, p. 25, ACM, 2011.

[3] I. Fehérvári and S. Appalaraju, “Scalable logo recognition using proxies,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 715–725, IEEE, 2019.

[4] H. Refai, B. Naim, J. Schettler, and O. A. Kalaa, “The study of vehicle classification equipment with solutions to improve accuracy in Oklahoma,” tech. rep., Oklahoma City, OK: Oklahoma Department of Transportation, 2014.

[5] “Freight Transportation Forecast,” tech. rep., American Trucking Associations, 2017. `http://www.atabusinesssolutions.com/`.

[6] A. Tok, K. Hyun, S. Hernandez, K. Jeong, Y. Sun, C. Rindt, and S. G. Ritchie, “Truck Activity Monitoring System (TAMS) for Freight Transportation Analysis,” in *Transportation Research Board 96th Annual Meeting*, (Washington DC, USA), 2017.

[7] Y. O. Adu-Gyamfi, S. K. Asare, A. Sharma, and T. Titus, “Automated vehicle recognition with deep convolutional neural networks,” *Transportation Research Record*, vol. 2645, no. 1, pp. 113–122, 2017.

[8] R. V. Nezafat, B. Salahshour, and M. Cetin, “Classification of truck body types using a deep transfer learning approach,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3144–3149, IEEE, 2018.

[9] M. E. Hallenbeck, O. I. Selezneva, and R. Quinley, “Verification, refinement, and applicability of long-term pavement performance vehicle classification rules.,” tech. rep., FHWA-HRT-13-091. Federal Highway Administration, Washington, D C, 2014.

[10] L. E. Y. Mimbela and L. A. Klein, “Summary of vehicle detection and surveillance technologies used in intelligent transportation systems,” in *Joint Program Office for Intelligent Transportation Systems, Washington, D.C.*, 2000.

[11] S. V. Hernandez, A. Tok, and S. G. Ritchie, “Integration of Weigh-in-Motion (WIM) and inductive signature data for truck body classification,” *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 1–21, 2016.

[12] S.-T. Jeng and S. Ritchie, “Real-time vehicle classification using inductive loop signature data,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2086, pp. 8–22, 2008.

[13] A. H. Lai, G. S. Fung, and N. H. Yung, “Vehicle type classification from visual-based dimension estimation,” in *Intelligent Transportation Systems, 2001. Proceedings*, pp. 201–206, IEEE, 2001.

[14] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” *arXiv preprint arXiv:1612.08242*, 2016.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, pp. 91–99, 2015.

[16] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, “Faceboxes: A CPU real-time face detector with high accuracy,” in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–9, IEEE, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[18] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” in *Artificial Intelligence and Statistics*, pp. 562–570, 2015.

[19] Y. Zhou, H. Nejati, T.-T. Do, N.-M. Cheung, and L. Cheah, “Image-based vehicle analysis using deep neural network: A systematic study,” in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 276–280, IEEE, 2016.

[20] S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, “A model for fine-grained vehicle classification based on deep learning,” *Neurocomputing*, 2017.

[21] A. Psyllos, C.-N. Anagnostopoulos, and E. Kayafas, “Vehicle model recognition from frontal view image measurements,” *Computer Standards & Interfaces*, vol. 33, no. 2, pp. 142–151, 2011.

[22] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, “Deep relative distance learning: Tell the difference between similar vehicles,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2167–2175, 2016.

[23] Z. Dong, Y. Wu, M. Pei, and Y. Jia, “Vehicle type classification using a semisupervised convolutional neural network,” *IEEE transactions on intelligent transportation systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[24] Z. Xiang, X. Huang, and Y. Zou, “An effective and robust multi-view vehicle classification method based on local and structural features,” in *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pp. 68–73, IEEE, 2016.

[25] H. Fu, H. Ma, Y. Liu, and D. Lu, “A vehicle classification system based on hierarchical multi-svms in crowded traffic scenes,” *Neurocomputing*, vol. 211, pp. 182–190, 2016.

[26] Z. Zhang, T. Tan, K. Huang, and Y. Wang, “Three-dimensional deformable-model-based localization and recognition of road vehicles,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 1–13, 2012.

[27] G. Zhang, R. Avery, and Y. Wang, “Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1993, pp. 138–147, 2007.

[28] P.-K. Kim and K.-T. Lim, “Vehicle type classification using bagging and convolutional neural network on multi view surveillance image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 41–46, 2017.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[30] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, vol. 60, pp. 84–90, May 2017.

[32] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected CRFs withgaussian edge potentials,” in *Advances in neural information processing systems*, pp. 109–117, 2011.

[33] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[34] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[35] C. Vondrick, D. Patterson, and D. Ramanan, “Efficiently scaling up crowdsourced video annotation,” *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, 2013.

[36] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, “Detecting text in natural image with connectionist text proposal network,” in *European conference on computer vision*, pp. 56–72, Springer, 2016.

[37] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, “Reading scene text in deep convolutional sequences,” in *Proceedings of AAAI Conference on Artificial Intelligence, (AAAI)*, 2016.

[38] J. Canny, “A computational approach to edge detection,” in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.

Appendices

A Machine Learning Terminology and Acronyms

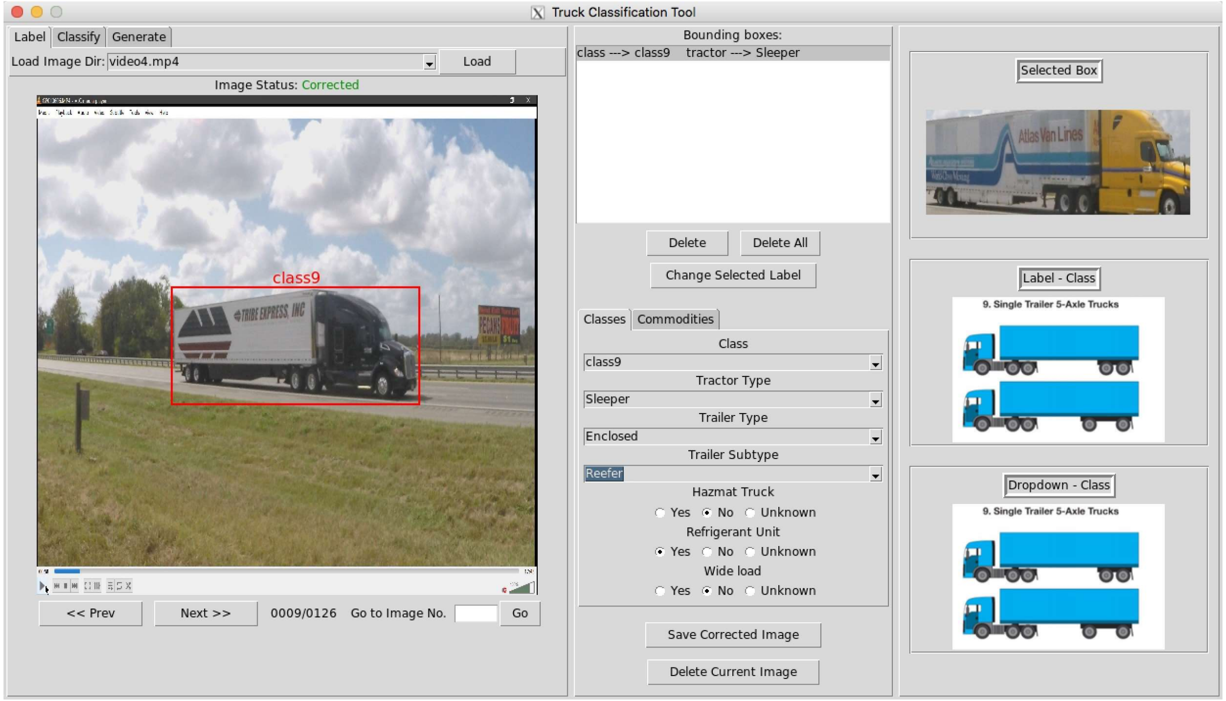
- 3D Deformable Vehicle Model, a vahicle classification model developed by [26].
- 3D tensor. A tensor is a generalization of vectors and matrices to potentially higher dimensions. A 3D tensor is a tensor with 3 dimensions.
- An application programming interfaces (APIs), is a set of subroutine definitions, communication protocols, and tools for building software.
- Atrous Convolution. The atrous convolutional layer (also known as dilated convolution) is a variant of convolutional layers.
- AVC, automated vehicle classification.
- The Bag of Words (BoW) Model, is originally developed for extracting features from text. It is adapted to image classification for counting occurrence of a vocabulary of local image features.
- Bilinear Interpolation Operation, an image operation to downsample or upsample images.
- Blob objects are objects having no distinct shape or definition, such as soap bubbles.
- Bobtail trucks, are trucks without a trailer attached.
- Canny Edge Detector, a classic method proposed by John F. Canny to detect image edges [38].
- Decision Tree (DT) and Random Forest (RF), two popular machine learning methods used for both classification and regression.
- CNN or ConvNet, abbreviation for Convolutional Neural Network.
- Coarse Score Maps are score maps that cannot clearly delineate the borders.
- Content-based Image Retrieval (CBIR) Query, one computer vision application of searching for digital images in databases.

- CONVs, abbreviation for convolutional neural network layers.
- Conditional random field (CRF) model, are statistical techniques applied in pattern recognition and machine learning and used for structured prediction [32].
- CVAT, abbreviation for Computer Vision Annotation Tool.
- FCs, abbreviation for Fully Connected Layers.
- DeepLabV2, an extension work of DeepLab [1], a classic semantic segmentation method.
- Dlib, is a c++ based general purpose cross-platform software library [30].
- Edgelets, a very short, locally straight image segment of what may be a longer, possibly curved, line.
- Euclidean Distance Matching, matching based on Euclidean distances.
- Fully convolutional neural networks (FCNNs) are used for pixel-wise segmentation [1]. One of the state-of-the-art FCNN methods is DeepLab [1].
- Few Shot Logo Recognition Model, one logo recognition model using few shot learning algorithms.
- The Fractionally Strided Convolution, a.k.a. deconvolution or transposed convolution, is one type of convolution operator.
- Gaussian Kernel is a popular kernel function.
- Google Reverse Engine Search, is developed to help users find the original source of photographs, memes and profile pictures, by uploading images or image URLs.
- HED (holistically-nested edge detector) is an edge detection algorithm that uses a deep learning model [33].
- HOG (Histogram of Oriented Gradients), a image feature descriptor used for object detection or object classification.
- Hough Transformation is a feature extraction technique used image processing. It is proposed to find objects of certain shapes by a voting procedure.
- HTMLParser parses a web page’s HTML (Hypertext Markup Language) content.

- ILDs, abbreviation for Inductive Loop Detectors.
- ImageNet is a huge image database organized according to the WordNet hierarchy.
- IOU, abbreviation for Intersection over Union.
- K-fold cross validation (KCV) is a procedure used to estimate the skill of the model on new data.
- k-means clustering is used for clustering analysis.
- K-nearest neighbors (KNN) Search aims at finding the point in a given set that is closest (or most similar) to a given point.
- Log Scale Transformation, a data transformation method to reduce variability of data.
- Technical metadata. This is camera generated and contains information. Examples of such information include aperture, shutter speed, focal depth, and resolution. Additional information includes date, time, and GPS location of image creation.
- Descriptive metadata. This information corresponds to the name of the image creator, keywords related to the image, user comments, etc. Such information can be useful for image searches.
- Administrative metadata. This includes licensing rights, restrictions on reuse, owner contact information, etc.
- An image moment is defined as a weighted average of image pixel intensities.
- NMS, abbreviation for Non-Maximum Suppression, a post processing method of object detection to remove duplicate detection.
- NAICS, abbreviation for North American Industry Classification System.
- A precision-recall curve is a plot that shows tradeoffs between precision (y-axis) and the recall (x-axis).
- The refrigeration unit, or refrigerator unit, is usually attached to the trailer unit.
- Region proposal network (RPN) [15] is one of the most popular real-time object detectors.
- Scale-Invariant Feature Transform (SIFT), a popular key-point detector.

- The Softmax is a function that converts a vector of K real numbers into a probability distribution.
- SVM, abbreviation for Support Vector Machine.
- TAMS, abbreviation for Truck Activity Monitoring System, developed by [6].
- TRUE Model, abbreviation for TRailer Unit Estimation (TRUE) model.
- Universal Logo Detector (ULD), a detector that reports all potential logos within images.
- A trailer is an unpowered vehicle that is towed by a powered vehicle. Trailers are frequently used for transport of goods and materials. There are various trailer types such as Day Cab, Enclosed, Chassis, Flatbed, Sleeper, Specialty, Tandem, Tank.
- Car Hauler, a trailer hauling cars.
- WIM, abbreviation for weighing-in-motion.
- YOLO (You Only Look Once), a state-of-the-art object detector [29].

B Visualization and Annotation Tool Development



Load Image Dir: video4.mp4

Image Status: Corrected

0009/0126

Bounding boxes:
class ---> class9 tractor ---> Sleeper

Class: class9

Tractor Type: Sleeper

Trailer Type: Enclosed

Trailer Subtype: Reefer

Hazmat Truck ○ Yes ● No ○ Unknown

Refrigerant Unit ● Yes ○ No ○ Unknown

Wide load ○ Yes ● No ○ Unknown

Figure 29: *The annotation tool developed by us. At the early stage of the project, we developed an convenient visualization and annotation tool that helps speed up the annotation process. The tool is divided into 3 main panels. The image panel (left) lets the users select a gallery of labeled images at the top and then move through the images using the bottom pane of the panel. The label panel (middle) allows the user to select and modify the current images labels. The example panel (right) shows the user the bounding box image separately to better see what was selected. The example panel also helps to define classes and provide visual guidance on labeling.*

C FHWA Vehicle Classification

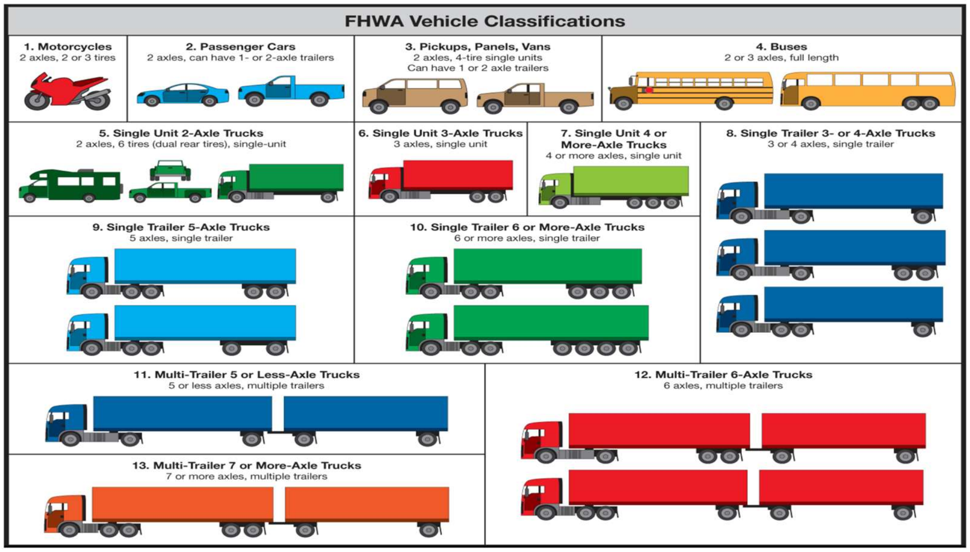
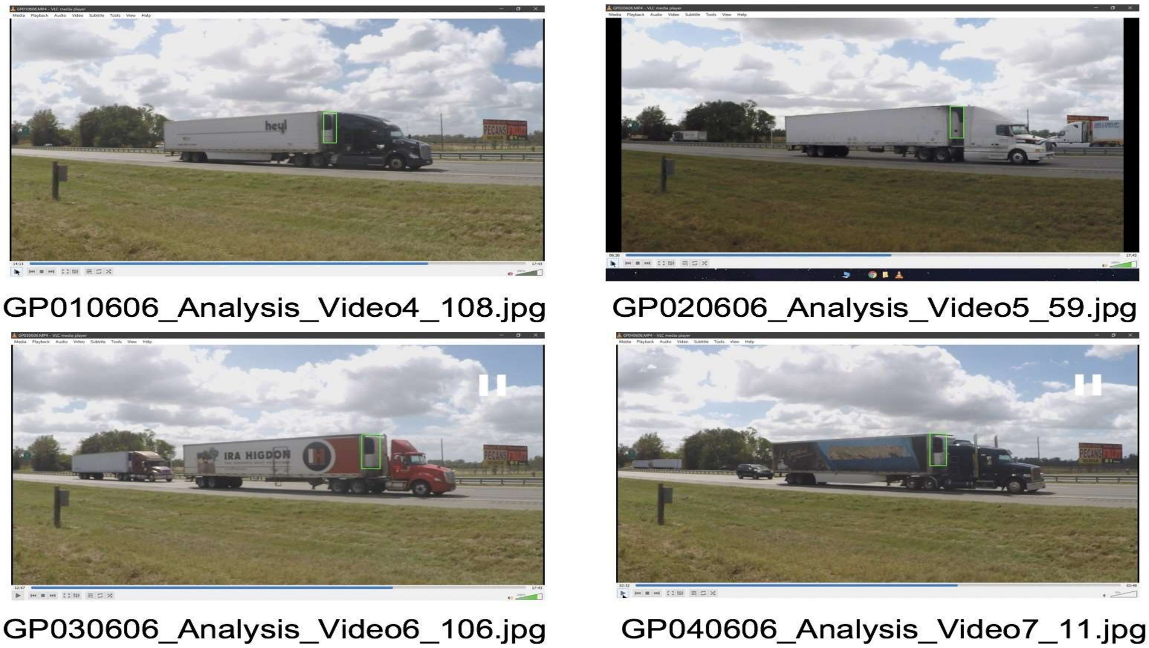

FHWA Vehicle Classifications
1. Motorcycles
2 axles, 2 or 3 tires
2. Passenger Cars
2 axles, can have 1- or 2-axle trailers
3. Pickups, Panels, Vans
2 axles, 4-tire single units
Can have 1 or 2 axle trailers
4. Buses
2 or 3 axles, full length
5. Single Unit 2-Axle Trucks
2 axles, 6 tires (dual rear tires), single-unit
6. Single Unit 3-Axle Trucks
3 axles, single unit
7. Single Unit 4 or More-Axle Trucks
4 or more axles, single unit
8. Single Trailer 3- or 4-Axle Trucks
3 or 4 axles, single trailer
9. Single Trailer 5-Axle Trucks
5 axles, single trailer
10. Single Trailer 6 or More-Axle Trucks
6 or more axles, single trailer
11. Multi-Trailer 5 or Less-Axle Trucks
5 or less axles, multiple trailers
12. Multi-Trailer 6-Axle Trucks
6 axles, multiple trailers
13. Multi-Trailer 7 or More-Axle Trucks
7 or more axles, multiple trailers

***Figure 30:** FHWA Vehicle Classification [4]. This standardized scheme distinguishes 13 vehicle types by the number of axles, unit numbers, and body configuration.*

D Refrigerator Unit Model



GP010606_Analysis_Video4_108.jpg

GP020606_Analysis_Video5_59.jpg

GP030606_Analysis_Video6_106.jpg

GP040606_Analysis_Video7_11.jpg

Figure 31: *Qualitative detection examples of our refrigerator unit model. The results were derived from videos taken at I-75 site 9956.*

E Wheel Model

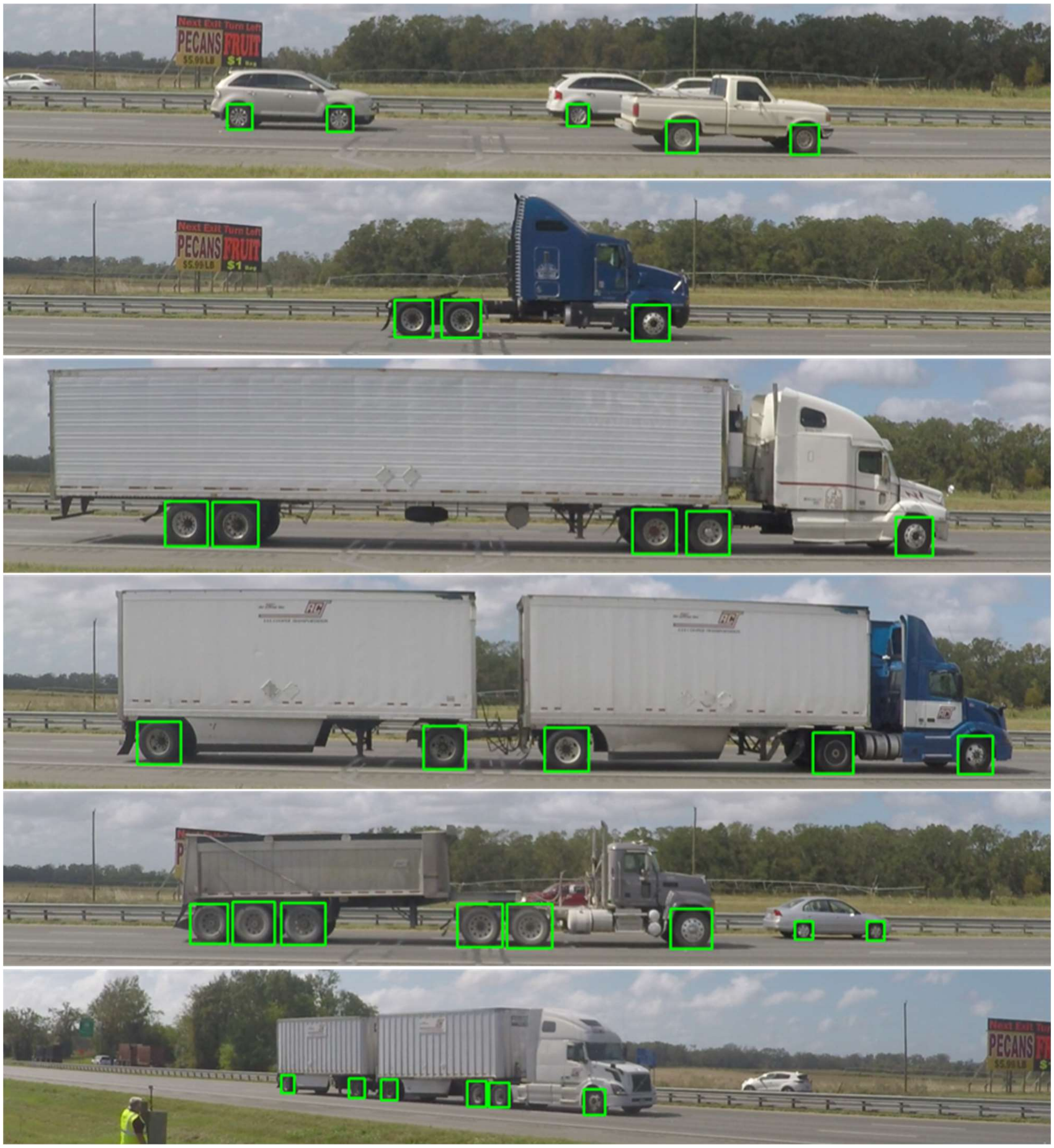


Figure 32: *Qualitative results for annotated wheel dataset.*

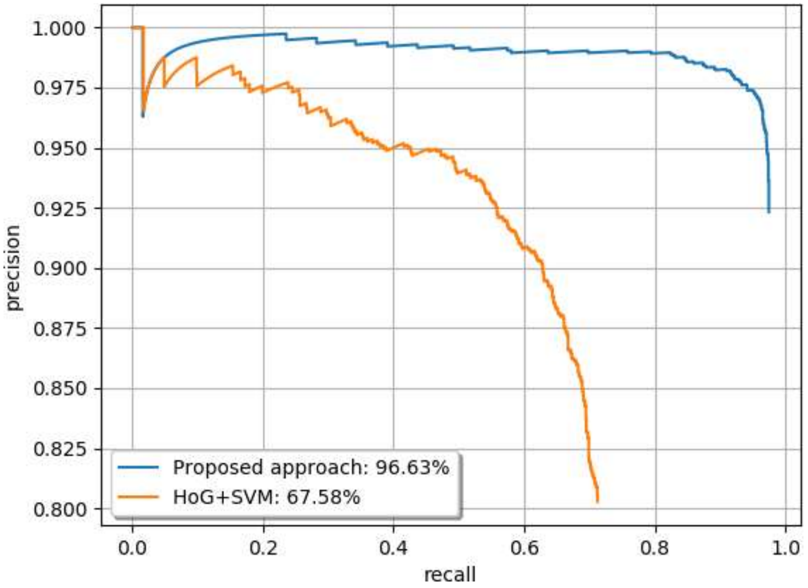


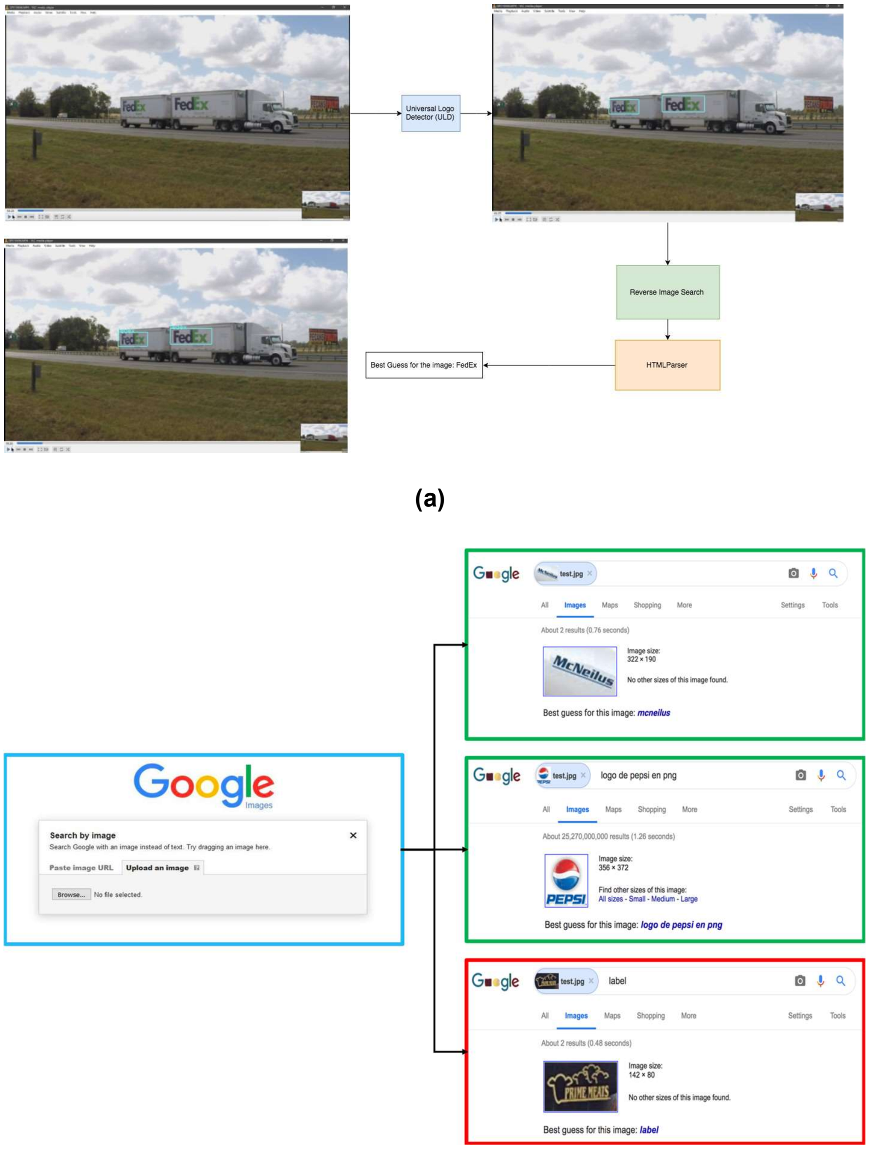
***Figure 33:** Precision-recall curves on annotated wheel dataset.*

F FDOT Image Library

The primary source of data for evaluating our approach were video frames captured by roadside video cameras deployed at a WIM station in Florida. Tracking spreadsheets are provided by the Florida Department of Transportation with rich annotations for truck attributes. Figure 35 shows a typical tracking spreadsheet provided by FDOT.

Two datasets were acquired to develop and evaluate truck classification systems. The first one (Dataset A) was collected and annotated directly by the traffic agencies—the Florida Department of Transportation (FDOT) in our case. It contains 372 truck images with a fixed camera view angle. The second one (Dataset B) was our self-annotated dataset, which contains 1,251 truck images from different camera angles.

G Universal Logo Detector


Universal Logo Detector (ULD)
Reverse Image Search
HTMLParser
Best Guess for the image: FedEx
Google Images
Search by image
Search Google with an image instead of text. Try dragging an image here.
Paste image URL
Upload an image
Browse... No file selected.
test.jpg
All Images Maps Shopping More Settings Tools
About 2 results (0.76 seconds)
Image size: 322 × 190
No other sizes of this image found.
Best guess for this image: mcnellus
test.jpg logo de pepsi en png
About 25,270,000,000 results (1.26 seconds)
Image size: 356 × 372
Find other sizes of this image: All sizes - Small - Medium - Large
Best guess for this image: logo de pepsi en png
test.jpg label
About 2 results (0.48 seconds)
Image size: 142 × 80
No other sizes of this image found.
Best guess for this image: label

(a)

(b)

Figure 34: *Our solution for Universal Logo Detector and Reverse Image Search. (a) We obtain logo images within truck images. (b) We feed them into Google Reverse Image Search. A green box means a successful query, and a red box means a failed query. The conclusion is that the logo recognition module needs more work to achieve a higher accuracy. Even though Google Reverse Image Search is already a state-of-the-art commercial product, it still cannot achieve satisfactory results when compared to the success achieved with our text-based scheme.*

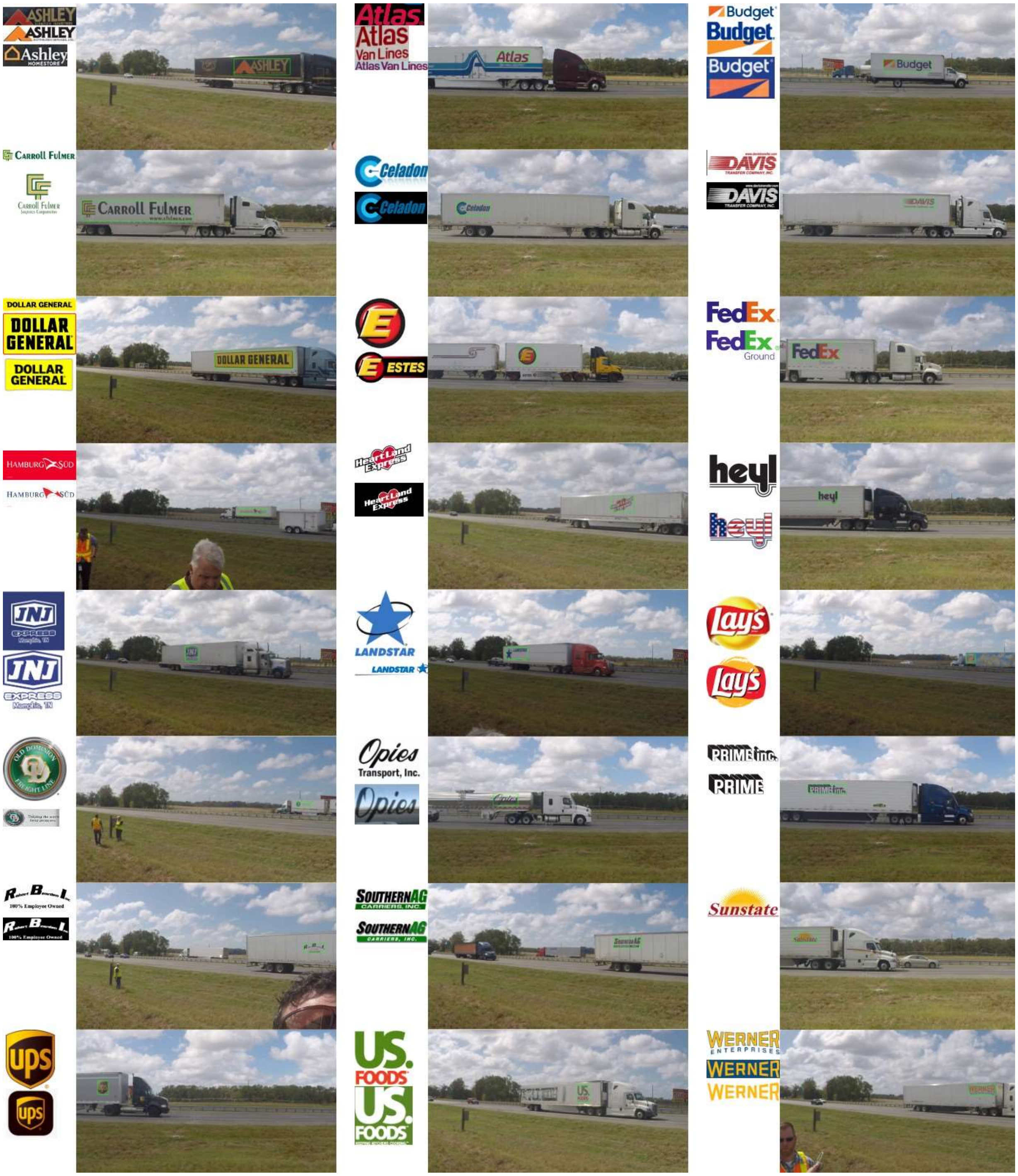


Figure 35: *Sample logo preprocessing results (These logos are copyrights or trademarks of their respective companies).*