Truck and Trailer Classification With Deep Learning Based Geometric Features

Pan He, *Graduate Student Member, IEEE*, Aotian Wu, Xiaohui Huang, Jerry Scott, Anand Rangarajan, *Member, IEEE*, and Sanjay Ranka, *Fellow, IEEE*

***Abstract*— In this paper, we present a novel and effective approach to truck and trailer classification, which integrates deep learning models and conventional image processing and computer vision techniques. The developed method groups trucks into subcategories by carefully examining the truck classes and identifying key geometric features for discriminating truck and trailer types. We also present three discriminating features that involve shape, texture, and semantic information to identify trailer types. Experimental results demonstrate that the developed hybrid approach can achieve high accuracy with limited training data, where the vanilla deep learning approaches show moderate performance due to over-fitting and poor generalization. Additionally, the models generated are human-understandable.**

***Index Terms*— Truck and trailer classification, deep learning, intelligent transportation system.**

I. Introduction

TRUCKS are largely in charge of transporting freight, both in terms of tonnage and revenue. The 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 the truck is carrying. According to a study of the American Trucking Association (ATA) [1], 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, in which trucks play a key role as well. Hence, understanding and monitoring truck activities become an essential component to effectively bolster the development of freight movement.

One important need of transportation agencies is truck classification, which 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 [2]–[4]. Among them, prominent and frequently used approaches are image processing techniques for traffic applications, e.g, the automated vehicle systems (AVS). Traditional approaches are usually based on the estimation of vehicle dimensions. Lai *et al.* [5] 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 [6]–[8] and object classification [9], [10]. Many advanced techniques from deep learning have been applied to vehicle type classification [3], [5], [11]. In [11], a deep neural network classifies vehicles (cars, sedans, and vans) in a small test dataset. Its performance in truck classes is unknown. Adu-Gyamfi *et al.* adapted the pre-trained deep learning model and fine-tuned model parameters for vehicle recognition [3]. However, several limitations are presented in these vanilla deep learning approaches, which we summarized: i) Large datasets are necessary for preventing overfitting and poor generalization; ii) It is difficult for humans to understand these models because of the large number of weights and layers; and iii) Most models do not fully exploit domain expert knowledge, e.g., specific truck configurations. Our preliminary study shows that the vanilla deep learning models fine-tuned with limited training data can achieve high accuracy at the training stage but show much lower accuracy at the inference stage.

Inspired by these, we carefully examined the truck classes and identified key features for discriminating truck types (e.g., the number of wheels (a proxy for the number of axles), number of trailers, size and aspect ratio, i.e., the ratio of length to height from a side view).[1] Leveraging these phenomenological observations, we developed an effective human-understandable truck classification approach [12], [13], enabling a successful collaboration between traffic agencies

Manuscript received September 21, 2019; revised April 16, 2020; accepted June 15, 2020. This work was supported in part by the Florida Department of Transportation. The Associate Editor for this article was S. Sacone. *(Corresponding author: Sanjay Ranka.)*

Pan He, Aotian Wu, Xiaohui Huang, Anand Rangarajan, and Sanjay Ranka are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: pan.he@ufl.edu; aotian.wu@ufl.edu; xiaohuihuang@ufl.edu; anand@cise.ufl.edu; ranka@cise.ufl.edu).

Jerry Scott is with the Florida Department of Transportation, Tallahassee, FL 32399 USA (e-mail: jerry.scott@dot.state.fl.us).

Digital Object Identifier 10.1109/TITS.2020.3009254

[1] A preliminary version of this manuscript was published in ITSC 2019, where we first developed an effective human-understandable truck classification approach.

and machine learning models. Since then, we generalized the developed approach by adding new features. In particular, based on the observation that trailer types are closely related to freight analysis, we customized the developed approach into trailer classification, which bridges the gap between truck taxonomy and freight analysis or, more precisely, commodity recognition.

Our approach leverages these phenomenological observations to develop effective classification approaches supporting truck and trailer classification. It drew upon recent work in deep convolutional 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). In particular,

1) We developed deep learning algorithms that leverage 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;
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 several algorithms for recognizing and classifying various truck attributes, such as trailer types and refrigeration units, that are useful in commodity recognition.

Results obtained from our datasets show that our scheme for truck classification has >90% accuracy for classifying trucks into one of the nine classes. Additionally, our algorithms achieved >85% accuracy for classifying a trailer and >95% accuracy for detecting a refrigerator unit.

II. Background

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 Highway Administration (FHWA). It was originally introduced for use in pavement and bridge design [14]. 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., surveys of national shippers and carriers, such as the U.S. Vehicle Inventory and Use Survey (VIUS), serve as the main source providers for correlation of truck activity with body configuration. The VIUS collected samples at both national and state levels, 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 tubes, 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 summary of classification techniques, focusing on developments for sensors (especially vision sensors). We refer interested readers to [15] 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 [16]. 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 [17]. In [18], 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 [18]. 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.

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 is required at present. The present work is a foray into achieving this objective.

III. Methodology

In this section, we describe the logic and reasoning undergirding the development of our proposed approach. To meet requirements for finer scale traffic engineering, we present the hybrid classification approach that integrates deep learning models (e.g., object detection, semantic segmentation, and edge detection) for geometric feature extraction for trucks and trailers. The pipeline of our developed method can be mainly separated into three distinct parts:

- The truck detection part aims at determining the presence of truck objects within images, followed by estimating the bounding box of each truck object.
- The truck classification part further determines the category of the truck object, by extracting geometric features for truck objects.
- The trailer classification groups trucks into categories of the trailer objects. A detailed description of the various geometric features of interest is presented, where we propose trailer shape, textual, and semantic features.

A. 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 crop out individual truck regions from the images. We follow the popular YOLO (You Only Look Once) object detector [6] for the truck detection problem. Precisely, we use the improved version YOLOv3 [20] across the entire experiments. In general, we prefer one-stage (unified) frameworks such as YOLO [6] and SSD [21], and their variants rather than two-stage (region-based) frameworks such as Fast R-CNN [22], Faster R-CNN [7]. The main advantage is that they show state-of-the-art detection performance while achieving a much faster speed. We refer the reader to [23] for more detailed discussions.

The pipeline of YOLOv3 [20] is simple and straightforward. The method begins with dividing the image into a $S \times S$ grid. Subsequently, it passes the entire image 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.

At inference time, only one network evaluation is required for obtaining the prediction of a test image. We followed YOLOv3 [20] to get multiple candidate detections. Because we were only interested in trucks, we removed predictions not belonging to vehicle classes (e.g., car, bus), followed by NMS to pick up top predictions. We further utilized state-of-the-art tracking algorithms, such as the correlation trackers in dlib, to speed up the model.

The experimental result in Table III in Section IV demonstrates that the YOLOv3 can achieve a high truck detection performance with an average precision of $>$90%.

B. 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.

1) Estimating Truck Size: One basic approach that has been applied in vehicle classification is the use of the bounding box around the detected vehicle, which covers the initial approximation of vehicle shape. To obtain precise shape information, a refinement step must be applied to the detected vehicles obtained from truck detection components.

Fully convolutional neural networks (FCNNs) are currently the most successful methods for pixel-wise segmentation and are very suitable for large-scale traffic video processing. FCNNs can deal with images of any size, taking into account wider 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 or fractionally strided convolutions; predictions that share the same size of the original image; and popular post-processing techniques such as the conditional random field (CRF) model [24].

We implemented and adapted the popular DeepLabV2 [19] 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 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

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

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 bilinear interpolation layer to produce a score feature map of the original image resolution. The final score map is obtained by taking the maximum response at each pixel location.

The final score map 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 [19], the fully connected CRF

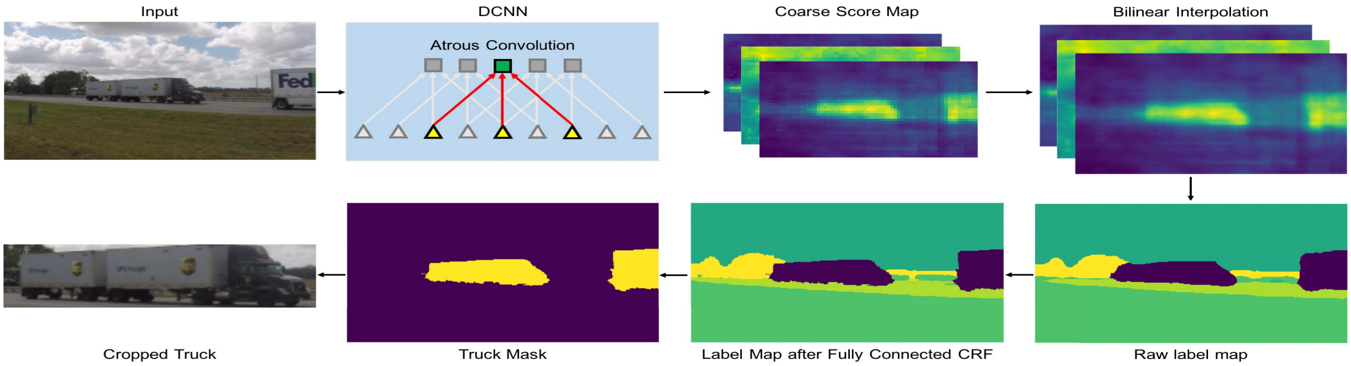


Fig. 1. The pipeline of estimating truck size. Following [19], a fully convolutional ResNet is deployed by adding atrous convolution with different sampling strides to obtain the coarse score map. The feature maps are then upsampled by the bilinear interpolation to the original image resolution. A raw label map is obtained via the Softmax operation. Refining the coarse prediction and utilizing structure information in images, a fully connected conditional random field model 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 initially detected truck regions to refine the detection results because the method supports processing images of different sizes.

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 $P(\mathrm{Y}_i)$ denote the label assignment probability at pixel i. $\theta_i(\mathrm{Y}_i) = -\log P(\mathrm{Y}_i)$ is then the unary potential. The pairwise potential θ_{ij} is defined as

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

where $\mathbb{1}_{i,j} = 1$ if $Y_i \neq Y_j$, 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 forced similar label assignments between pixels with similar positions and colors. The second kernel purely depended on pixel position, thus encouraging spatial smoothness.

2) 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 [25], which is based on the idea of FCNNs [26] and deep-supervised nets [10]. Given the training data set $S = \{(X_n, Y_n)\}, n = 1, \ldots, N$, 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 the 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 manipulation (e.g., convolution, pooling, or activation function).

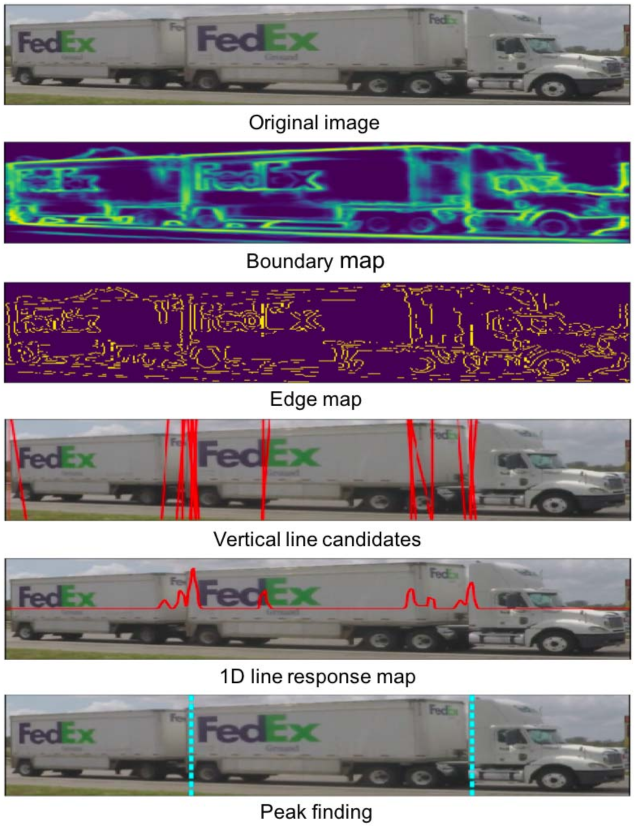
The whole network added such layers multiple times to learn the nonlinear filters. The final vehicle boundary was obtained by further aggregating the generated edge maps of layers (with different down-sampling rates) with in-network bilinear interpolation. One advantage of this design was that the model could take inputs of arbitrary size and efficiently produce the 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 2. These $W \times H$ maps were reduced to $W \times 1$ responses by summing over 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 distance β. We set $\beta = 0.25W$.


Original image

Boundary map

Edge map

Vertical line candidates

1D line response map

Peak finding

Fig. 2. 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 over the columns. Finally, a peak-finding algorithm picks up the best separation spacing for trailer units.

3) Estimating Vehicle Wheels: Detecting and recognizing wheels in vehicle images can serve as a foundation for the FHWA classification method. We began with a baseline wheel detection model based on traditional hand-crafted features and support vector machine classifiers. 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; and (3) the deformation of wheels and scale variations further degrades system performance.

Inspired by recent advances in tiny face detection [8], 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. Adding more aspect ratios can further improve the performance, mostly on improving the recall. However, it incurs an extra computational cost due to predictions for more anchor boxes. We find out that using 1 : 1 aspect ratio already achieves a fairly good performance. It makes a good accuracy-speed trade-off. Several data augmentation strategies were followed, including color distortion, random cropping, scale transformation, and horizontal flipping. The loss function was the same as RPN [7] with the sigmoidal loss for binary classification and smooth L1 loss for regression.

4) 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. For features with large values, we standardized them by applying the log scale transformation. 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, which can enable a successful collaboration between traffic agencies and machine learning models, allowing an effective interaction with the model to make better decisions.

C. Trailer Classification Component

While truck classification approaches have been extensively discussed and developed in the transportation community, fewer efforts have been made in trailer classification. The pioneer work [16] developed a novel approach that integrates sensor data collected from WIM systems and advanced ILDs. They trained eight separate truck body classification models and explored classification tasks for semi-tractor trailers with high accuracy. The limitation of their approach is that WIMs and ILDs are intrusive sensors that require interruption of traffic during installation. It is difficult to further minimize facilities installation, maintenance, and repair costs because these sensors are installed on pavement surfaces. A trailer classification model based on cameras seems to be more promising because cameras are easy to install. They can provide rich visual information at low costs.

Both tractor and trailer types are crucial information for deciding the carried cargo in trucks. The tractor types can assist in determining if the trip is a long-haul or short-haul trip. If the tractor type contains a sleeper cab, then that trip is more likely to be utilized for a long haul trip. If the tractor type contains a day cab, then that trip will probably not be making any long haul trips, and the trip would be for a short-haul trip. The identification of the trailer type (and trailer sub type) will narrow the different types of commodities which may be hauled during the trip. As shown in Figure 3. If the truck is bobtail or flatbed, traffic agencies can tell immediately that it contains empty cargo. For the tank trailer, it is likely to carry liquefied loads, dry bulk cargo, or gases. For enclosed trucks, if they contain any refrigerator units, the possible commodities are perishable cargo. Due to the variety of specialty trailer

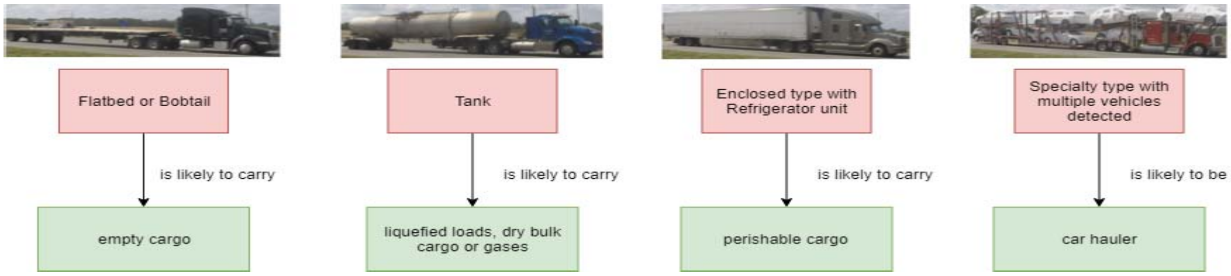


Fig. 3. Typical relations between trailer types and commodity types. The trailer type is an important piece of information in determining the type of commodity carried in trucks. Consequently, for detected trucks, only after a trailer is detected could we continue the process of commodity identification. For many trailers, the corresponding commodities could be directly determined by their types.

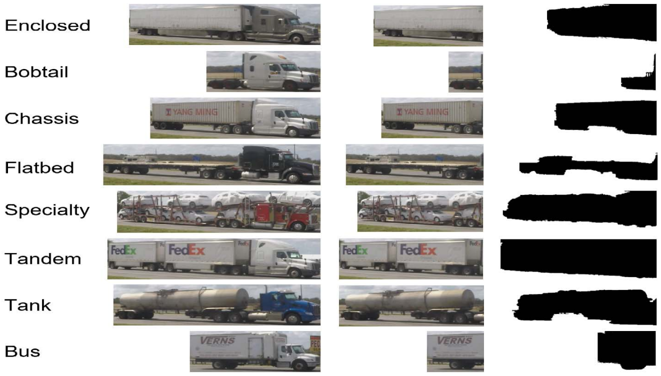


Fig. 4. Sample-derived trailers and their corresponding contours.

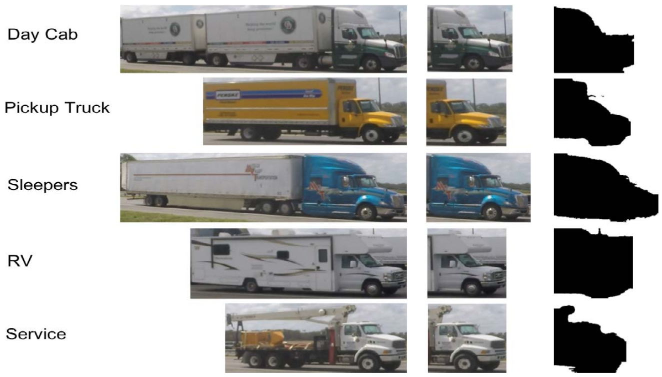


Fig. 5. Sample-derived tractor and their corresponding contours.

types, we are only able to determine the car hauler if multiple vehicles are detected over trucks. As we are more interested in identifying carried commodities, we focus on designing more powerful trailer models. Nevertheless, the proposed features apply to tractor classification as well.

Inspired by these observations, we developed models for trailer classification. Our observation was that truck classes are closely related to the trailer types. For example, most of the class 9 trucks, in reality, are enclosed trucks. Based on this domain-specific knowledge, we designed customized trailer features and combined them with truck features to develop trailer models.

The semantic segmentation scheme, as described in truck classification, can be used to determine the trailer contour. Using the peak-finding algorithm of the TRUE model described in Section III-B.2, we can separate the tractor and trailer part of the truck, as shown in Figure 4 and Figure 5.

Based on the extracted contour, additional features can be computed. For determining the trailer types, we carefully examined trailer images and identified several discriminative features suitable for trailer classification. The designed model involves various features based on shape, texture, and semantic information. The shape features can directly characterize the appearance of an object. The textural features can capture the surface characteristics of trucks, providing useful information to distinguish trailers made with different materials, e.g., aluminum, steel, or composite materials. The semantic features provide us additional information based on the functionality and characteristics of trucks. For example, by deciding whether the truck contains a refrigerator unit, we can separate chassis or enclosed trailers from other trailer types because the refrigerator units are mostly contained in these two trailer types.

1) Trailer Shape Features: Shape features can characterize the appearance of an object. The basic shape features we have identified are size information such as length, height, area, and perimeter. As a bobtail truck[2] has a short length without any trailer attached, the area and perimeter shape feature can be utilized to separate them from other trailer types. The compactness feature is also added to describe the trucks, which is defined as

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

Trailers that have an elliptical shape or a boundary that is irregular rather than smooth tend to have a smaller compactness value. Many trailer types can be characterized as blob shapes. We further utilized advanced shape-matching techniques (such as image moment) 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:

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

[2] Though a bobtail truck does not have a trailer attached, we add it as a special class of trailer types. The main reason is that for truck freight analysis, we are interested in determining whether the trucks carry any cargo and further classifying them. Bobtail trucks are common trucks on the highway roads.

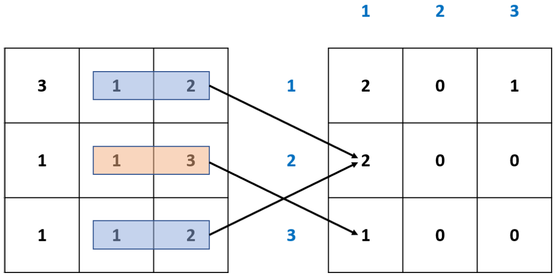


Fig. 6. Gray Level Co-occurrence Matrix (GLCM). The adjacency calculation considers only left-to-right. Generally, we consider four types of adjacency: left-to-right, top-to-bottom, top left-to-bottom right, and top right-to-bottom left.

which calculates the sum of all pixel intensities. This feature is relatively robust to the 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 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: a set of seven values calculated using central moments. They are invariant to several image transformations. The first six moments are invariant to translation, scale, rotation, and reflection; the seventh moment changes sign for image reflection. This feature is very helpful in distinguishing trailer classes.

2) Trailer Textural Features: The textural features capture the surface characteristics of trucks. They describe the spatial arrangement of intensities or color within trucks. We already extracted the trailer contours, which effectively removed the background interference of the truck images. In [27], Haralick *et al.* suggested the use of gray level co-occurrence matrices (GLCM), where the joint probability distributions of pairs of pixels are used to compute the GLCM. The first order texture measures are computed from the original image such as histogram and variance. Such computations do not consider pixel relationships. The GLCM instead belongs to second-order texture information, as it captures the relative positional relationships between groups of two pixels. It describes how frequently two pixels with the same gray-level occur in a window within a certain distance in direction of θ.

As shown in Figure 6, from left to right ($\theta = 0$), we repeatedly compute the occurrence between two adjacent pixels with a one-pixel offset. We can compute GLCM with different orientations (such as left-to-right, top-to-bottom, top left-to-bottom right, top right-to-bottom left). Haralick *et al.* [27] then proposed 14 statistics (such as correlation, contrast, entropy, and sum variance) based on the co-occurrence matrix to describe the texture of the image. We followed the same approach and extracted the first 13 Haralick features for four types of adjacency. The feature is reduced by taking the mean along the four adjacency directions, which results in a feature vector of 13 dimensions.

3) Trailer Semantic Features: Unlike shape and textural features that describe trucks based on pixel information, semantic features are image context as perceived by humans. Traffic experts can identify potential trailer types based on the unique materials, components, and commodities appearing on the trailers. For example, if the trailer contains any HAZMAT (hazardous materials) symbols, the trailer is likely to be a tank or specialty trailer, because hazardous materials are usually carried by emergency vehicles (e.g., fire trucks, ambulances) or tank trucks (carrying fuel or gases).

The first semantic feature is related to the refrigerator unit. If the truck contains a refrigerator unit, the feature is 1, otherwise 0. As discussed, it is a good feature to separate chassis and enclosed trailers from other trailer types. A refrigerator truck is designed to carry perishable freight. The refrigerator unit is usually attached to the trailer unit. Based on the 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. The model was able to detect refrigerator units from truck images. We obtained >95% accuracy on one internal dataset collected from highway videos.

Similarly, 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 the whole 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 estimated that it was likely to be a ‘car hauler’ if the number was greater than 1.

On identifying tank trailer type, we observed that the rear wheels of tank trailers are very close to the back of the trucks (the distance is referred to as the rear overhang as shown in Figure 7). Tank trailers tend to have smaller rear overhang values, compared to enclosed trailers. As the wheel model can report the location distributions of detected wheels, we use this information and count the rear overhang feature as 1 if the distance is less than 2 times the width of the rear wheel, otherwise 0.

4) Ensemble Features: Once we computed all the features described, we concatenated these features with original truck classification features. The main motivation for using the original truck classification feature was that several of the truck

TABLE I

PERFORMANCE EVALUATIONS ON THE TWO ANNOTATED TRUCK DATASETS, EACH WITH THE 2-FOLD CROSS VALIDATION REPEATED FIVE TIMES

	9-class experiment				3-class experiment			
	Dataset A		Dataset B		Dataset A		Dataset B	
	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
1	98.65	94.35	94.89	92.25	100.00	97.04	98.40	96.09
2	99.19	94.08	93.77	90.97	99.73	96.50	98.08	96.33
3	98.39	93.83	93.68	91.05	100.00	97.31	98.32	96.49
4	98.92	92.74	93.52	91.53	100.00	97.85	98.24	96.48
5	98.65	94.34	93.69	90.81	99.73	98.11	98.32	96.33
Average	98.76	93.87	93.91	91.32	99.89	97.36	98.27	96.34

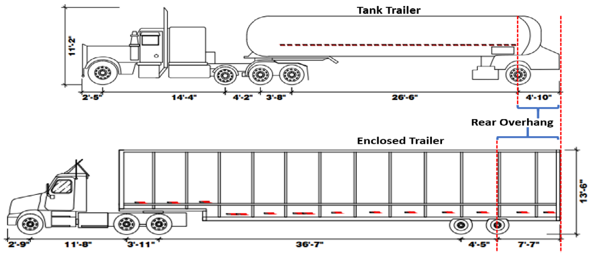


Fig. 7. Samples of rear overhang [28], which measures the length of the truck which extends beyond the wheelbase at the rear. Tank trailers tend to have smaller rear overhang values, compared to enclosed trailers.

classes were highly correlated to the trailer types. For example, most of the class 9 trucks are enclosed truck (one of the trailer types). Thus, we expected that utilizing the truck type would implicitly help to determine the correct trailer types. Decision trees are utilized for trailer classification on top of these features following the truck classification. 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 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.

IV. EXPERIMENTS

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

A. Image Collection Procedure

Our approach to vehicle detection and recognition is based on video frames captured by roadside video cameras deployed at a WIM station in Florida, in the United States. These image frames were captured at different times of the day and at different periods of the year. We used the available annotation tool named Computer Vision Annotation Tool (CVAT) 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.

1) 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 agency—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 a dataset that we annotated, which contains 1,251 truck images from different camera angles. Most of the classes are class 9 ($>$60%), which coincides with the observed traffic at this location.

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 casts 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, as shown in Table I.

Besides, we manually labeled 405 images with the background for evaluating the truck detector. Only the largest truck in the image is considered, which is the main interest of the FDOT traffic agency. We adopted the standard evaluation protocol for object detection [29]. We follow the new evaluation protocol of the Pascal VOC challenge where they use all data points, rather than interpolating only 11 equally spaced points [29]. The results are shown in Table III.

2) Annotated Trailer Dataset: Similarly, we annotated 377 truck images with trailer types. We considered the following trailer types: flatbed, enclosed, chassis, tank, tandem, RV, and car-hauler. Though bobtail trucks do not have any trailer attached, we added it to the trailer types as discussed. We combined the enclosed class and the chassis class because the subtle difference is not trivial to distinguish.

3) 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, and the rest were used for evaluation. The evaluation is similar to truck detection. We set the IoU threshold to 0.5 in the experiment.

B. Experimental Results

1) Results on Annotated Truck Dataset: As shown in Table I, each row i indicates the i_{th} K-fold experimental results. We repeated the experiments 5 times. In the 9-class experiment, we achieved an average test accuracy of 93.87% on dataset A. It achieved slightly lower accuracy on dataset

TABLE II

PERFORMANCE EVALUATIONS ON THE ANNOTATED TRAILER DATASET. 'DT' AND 'RF' DENOTE THE DECISION TREE CLASSIFIER AND THE RANDOM FOREST CLASSIFIER, RESPECTIVELY

	Train Accuracies with Repeated Experiments (%)					Average Train Accuracy (%)
DT with Truck Features	95.49	93.65	95.75	94.70	94.97	94.91
DT with Trailer Features	91.25	91.77	91.51	92.58	92.56	91.93
DT with Both Features	96.28	96.29	96.55	96.81	95.23	96.23
RF with Truck Features	99.47	98.94	98.94	99.47	99.21	99.21
RF with Trailer Features	99.47	99.74	99.47	99.21	99.20	99.42
RF with Both Features	99.74	99.21	99.47	99.74	99.74	99.58

	Test Accuracies with Repeated Experiments (%)					Average Test Accuracy (%)
DT with Truck Features	81.43	79.58	83.57	79.57	84.10	81.65
DT with Trailer Features	86.47	79.59	83.55	84.08	81.95	83.13
DT with Both Features	86.20	83.56	83.83	86.19	86.71	85.30
RF with Truck Features	86.75	82.22	84.36	83.82	84.88	84.41
RF with Trailer Features	83.55	83.83	86.17	83.80	85.14	84.50
RF with Both Features	89.12	89.40	88.10	89.39	89.41	89.08

TABLE III

EVALUATION OF TRUCK DETECTION USING YOLOV3 WITH DIFFERENT IOU THRESHOLD VALUES

	IoU=0.3	IoU=0.5	IoU=0.7
Average Precision	99.68	94.71	90.89

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.

2) Results on Annotated Trailer Dataset: Results from the trailer classification are shown in Table II. If only using truck features (originally used for truck classification models), we can obtain an average test accuracy of 81.65% with a decision tree classifier. Because these features are originally designed for truck classification and are not customized for trailer classification. This approach cannot achieve competitive performance compared to other model variants, While we replaced the features with trailer features, we unsurprisingly improved the results to 83.13%, indicating that the customized trailer features are indeed more suitable for trailer classification. As we discussed, truck features and trailer features are highly correlated and complementary to each other. We, therefore, concatenated them together to obtain hybrid features. This significantly improved results on both 'DT' (the decision tree classifier) and 'RF' (the random forest classifier) experiments, which validates our observations. Considering that trailer attributes are crucial for commodity classification, we further improved the model by using the more powerful random forest classifier, and we were able to obtain an average test accuracy of 89.08%. The Random Forest classifier automates the feature selection by indicating the max number of features (usually less than the total number of features) to consider when looking for the best split, which can partly ease the problem of the potential existence of redundant features.

3) Results on 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. This pipeline achieved an average precision of 67.58%. In addition to this, for the baseline model, we precomputed the perspective transformation matrix for camera calibration. Our wheel detection model performed well on the annotated wheel dataset, achieving an average precision of 96.63%.

V. SUMMARY AND CONCLUSION

The proposed hybrid approach took advantage of recent advances in deep convolutional neural networks. The developed approach achieved high average precision rates on the annotated truck dataset. On the wheel detection problem, our developed method significantly surpassed the baseline HOG+SVM framework. Our approach leveraged geometric features to develop an effective classification approach that is human-understandable, enabling a successful collaboration between traffic agencies and machine learning models and permits an effective interaction with the model to make better decisions. Future studies can focus on model architectural design to increase the number of classes being accurately classified by the proposed vision system. Inter-frame features or characteristics, such as motion vectors or optical flows can be utilized to further improve the detection and classification.

ACKNOWLEDGMENT

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.

REFERENCES

[1] (2017). *Freight Transportation Forecast*. [Online]. Available: http://www.atabusinesssolutions.com/

[2] H. Refai, B. Naim, J. Schettler, and O. A. Kalaa, "The study of vehicle classification equipment with solutions to improve accuracy in Oklahoma," Oklahoma Dept. Transp., Oklahoma City, OK, USA, Tech. Rep. FHWA-OK-14-17, 2014.

[3] Y. O. Adu-Gyamfi, S. K. Asare, A. Sharma, and T. Titus, "Automated vehicle recognition with deep convolutional neural networks," *Transp. Res. Rec.*, vol. 2645, no. 1, pp. 113–122, 2017.

[4] R. V. Nezafat, B. Salahshour, and M. Cetin, "Classification of truck body types using a deep transfer learning approach," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3144–3149.

[5] A. H. S. Lai, G. S. K. Fung, and N. H. C. Yung, "Vehicle type classification from visual-based dimension estimation," in *Proc. IEEE Intell. Transp. Systems. (ITSC)*, Aug. 2001, pp. 201–206.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[8] 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 *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 1–9.

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

[10] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 38. PMLR, 2015, pp. 562–570.

[11] 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 *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Oct. 2016, pp. 276–280.

[12] P. He, A. Wu, X. Huang, J. Scott, A. Rangarajan, and S. Ranka, "Deep learning based geometric features for effective truck selection and classification from highway videos," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019.

[13] P. He, A. Wu, A. Rangarajan, and S. Ranka, "Truck taxonomy and classification using video and Weigh-In Motion (WIM) technology final report," Florida Dept. Transp., Univ. Florida, Gainesville, FL, USA, Tech. Rep. BDV31-977-81, 2019.

[14] M. E. Hallenbeck, O. I. Selezneva, and R. Quinley, "Verification, refinement, and applicability of long-term pavement performance vehicle classification rules," Federal Highway Admin., Washington, DC, USA, Tech. Rep FHWA-HRT-13-091, 2014.

[15] L. E. Y. Mimbela and L. A. Klein, "A summary of vehicle detection and surveillance technologies used in intelligent transportation systems," in *Joint Program Office for Intelligent Transportation Systems*. Washington, DC, USA: Federal Highway Administration, 2000.

[16] S. V. Hernandez, A. Tok, and S. G. Ritchie, "Integration of Weigh-in-Motion (WIM) and inductive signature data for truck body classification," *Transp. Res. C, Emerg. Technol.*, vol. 68, pp. 1–21, Jul. 2016.

[17] S.-T. Jeng and S. G. Ritchie, "Real-time vehicle classification using inductive loop signature data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2086, no. 1, pp. 8–22, Jan. 2008.

[18] A. Tok, K. Hyun, S. Hernandez, K. Jeong, Y. Sun, C. Rindt, and S. G. Ritchie, "Truck activity monitoring system for freight transportation analysis," *Transp. Res. Rec., J. Transp. Res. Board*, no. 2610, pp. 97–107, 2017.

[19] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[21] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2167–2175.

[22] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[23] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[24] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

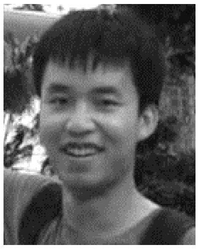
[25] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[27] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

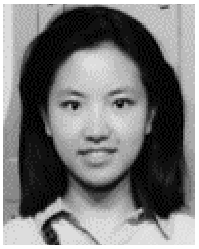
[28] (2017). *US Vehicle Inventory*. [Online]. Available: https://www.fhwa.dot.gov/policyinformation/travel_monitoring/

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.



Pan He (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. His research interests include deep learning and computer vision.

Aotian Wu is currently pursuing the Ph.D. degree with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA.



Xiaohui Huang is currently pursuing the Ph.D. degree with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA.

Jerry Scott is currently a Multimodal Data System Coordinator with the Florida Department of Transportation, Tallahassee, FL, USA.

Anand Rangarajan (Member, IEEE) is currently a Professor with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. His research interests include computer vision, machine learning, medical and hyperspectral imaging, and the science of consciousness.

Sanjay Ranka (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, USA, in 1988. He was the Chief Technology Officer at Paramark, where he developed real-time optimization software for optimizing marketing campaigns. He was one of the main architects of the Syracuse Fortran 90D/HPF compiler. He is currently a Professor with the University of Florida, where he directs the University of Florida Transportation Institute. He has coauthored two books: *Elements of Neural Networks* and *Hypercube Algorithms*. His research interests include data mining, informatics and grid computing for data-intensive applications in high energy physics, bioterrorism, and biomedical computing. He was a past member of the Parallel Compiler Runtime Consortium and the Message Passing Initiative Standards Committee. He is a fellow of AAAS, an Advisory Board Member of the IEEE Technical Committee on Parallel Processing, and a member of IFIP Committee on System Modeling and Optimization. He serves on the Editorial Board of the *Journal of Parallel and Distributed Computing*.