FINAL REPORT

Feasibility of Estimating Commodity Flows on Highways with Existing and Emerging Technologies

Prepared by:

Mecit Cetin, Olcay Sahin,
Reza Nezafat Vatani

Transportation Research Institute
Old Dominion University
135 Kaufman Hall
Norfolk, VA, 23529

Wael Zatar, and Andrew P. Nichols

Marshall University
One John Marshall Drive
Huntington, WV 25537

Date: Mar 2019

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Feasibility of Estimating Commodity Flows on Highways with Existing and Emerging Technologies

5. Report Date
March, 2019

6. Performing Organization Code

7. Author(s)
Mecit Cetin, Olcay Sahin, Reza Nezafat Vatani, Wael Zatar, and Andrew P. Nichols

8. Performing Organization Report No.

9. Performing Organization Name and Address

Old Dominion University 135 Kaufman Hall Norfolk, VA, 23529	Marshall University One John Marshall Drive Huntington, WV 25537

10. Work Unit No. (TRAIS

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
US Department of Transportation
Office of the Secretary-Research
UTC Program, RDT-30
1200 New Jersey Ave., SE
Washington, DC 20590

13. Type of Report and Period Covered

Final Report 7/1/16 – 3/18/19

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
Understanding the types and volumes of different commodities being hauled over the roadway network is important for freight demand modeling and planning. Categorizing trucks or trailers into subclasses (e.g., container, dry van, refrigerated vans, tank, and car transporter) will help narrow down the type of commodity that is being carried. For example, refrigerated trailers are commonly used to transport perishable produce and meat products, tank trailers are for fuel and other liquid products, and livestock is carried in specialized trailers. The main goal of this project is to investigate the feasibility of using data from non-intrusive sensors (e.g., camera and LIDAR) to identify the type of trailer. Algorithms are developed to demonstrate the feasibility of accurately detecting and classifying the trailer types. Data collected by a 3D LIDAR sensor are analyzed to extract pertinent information for classifying truck trailers into subtypes. The collected data include enough samples of four trailer types: Intermodal container, refrigerated container, dry van, and refrigerated dry van. After data processing, various machine learning algorithms are developed to automatically classify the observed trucks into these four subcategories. The tested machine learning algorithms include an SVM (support vector machines) model which requires a feature vector as the input, and deep CNN (convolutional neural networks) which do not require extracting features from raw data. Instead, LIDAR data are converted to a 2D image as the input to the CNN. The results from these machine learning algorithms show that these four truck body types could be classified with over 90% accuracy. The report present further details on how raw data are processed and the classification algorithms are developed.

17. Key Words
Vehicle classification, freight modeling, machine learning, LIDAR, truck body types.

18. Distribution Statement

No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	26	

ACKNOWLEDGMENTS

This project was funded by Mid-Atlantic Transportation Sustainability Center – Region 3 University Transportation Center (MATS UTC). The authors would like to thank Virginia Department of Transportation (VDOT) for assisting with the installation of the sensors for field data collection. Dr. Andrew Nichols contributed to the development the research plan and assisted with the initial phase of this project before leaving Marshall University. The author and other coauthors are thankful to Dr. Nichols for his contributions to the initial phase of this project.

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROBLEM

Understanding the types and volumes of different commodities being hauled over the roadway network is an important aspect of the freight planning process. To adequately allocate resources for accommodating commercial freight delivery and distribution across the highway network public agencies need to predict the quantity and type of the commodities. However, collecting commodity-specific freight data is a challenge. The Census Bureau conducts a Commodity Flow Survey (CFS) every five years at the national level. The CFS is a primary source for the freight analysis framework. However, the CFS may not be representative at the state or metropolitan levels, therefore, there is a need to collect this information for planning and monitoring purposes using the currently available technologies. There are various types of sensors for vehicle detections and classification (e.g., inductive loops, weigh-in-motion (WIM) sensors, radar), and they provide the raw data needed to classify vehicles into a predefined set of classes such as FHWA's 13-class scheme. These schemes or algorithms work mostly based on the number of axles and number of trailers and do not provide much details about the type of trailer attached to the truck. Categorizing trailers into subclasses (e.g., container, dry van, refrigerated vans, tank, and car transporter) will help narrow down the type of commodity that is being carried. For example, refrigerated trailers are commonly used to transport perishable produce and meat products, tank trailers are for fuel and other liquid products, and livestock is carried in specialized trailers. The main goal of this project is to investigate the feasibility of using data from non-intrusive sensors (e.g., camera and LIDAR) to identify the type of trailer. Algorithms are developed to demonstrate the feasibility of accurately detecting and classifying the trailer types.

APPROACH

The main focus of this project is to investigate how emerging sensors could be utilized in categorizing trucks into more distinct classes beyond the common classes (e.g., FHWA's 13 classes). In particular, data collected by a 3D LIDAR sensor are analyzed to extract pertinent information for classifying truck trailers into subtypes. Raw point-cloud data from these sensors include distance to the target and intensity of the reflected light. From these data, it is possible to construct a 3D representation of the objects within the range of the sensor. As explained later in the report, a LIDAR unit along with a set of surveillance cameras are used to collect data at a specific location on I-64 in Hampton Roads, VA. The collected data include samples of four trailer types: Intermodal container, refrigerated container, dry van, and refrigerated dry van. After data processing, various machine learning algorithms are developed to automatically classify observed trucks into these four subcategories.

It should be mentioned that data from other types of sensors could also be utilized to classify trucks into subcategories. For example, recently, researchers show how WIM and inductive loop signature data [1] could be used to detect truck body types. Since such options have been explored previously, this study is focused on the emerging technology of LIDAR for traffic data collection. The classification results obtained based on the LIDAR data will be compared to those in the literature [1].

METHODOLOGY

The data needed to conduct this research come from a LIDAR unit and a set of surveillance cameras. The video collected by the cameras is used for ground truth and model validation. The LIDAR and surveillance cameras continuously collected data for 36 days. After data collection is completed, the data are preprocessed for further analyses and feature extraction. For each individual vehicle observed, the point cloud data from multiple LIDAR frames are combined to generate a 2-dimensional profile. These profiles form the basis or input data to the machine learning algorithms. Two machine learning methods are developed to classify truck body types:

1. The first method involves extracting useful features from the 2D-profiles which are then taken as inputs to a Support Vector Machines (SVM) model. Support Vector Machines (SVM) solves an optimization problem to find the optimal hyperplane to separate classes [2] based on the features and labels of the training data. The SVM model classifies the trucks into three groups: Dry van, intermodal container, and other. After this grouping, a heuristic is developed to detect whether a truck classified as either dry van or intermodal container has a refrigerated unit or not. In the end, these two steps (SVM and the heuristic) will classify the trucks into the four types described earlier.
2. In the second method, deep Convolutional Neural Networks (CNNs) are used which do not require extracting specific features as inputs. In particular, rather than developing and training new deep CNNs, some of the well-known pre-trained models (VGGNet, AlexNet, and ResNet) are customized to solve the classification problem. This method is called "transfer learning" since many of the parameters of these pre-trained models are kept the same. These models (VGGNet, AlexNet, and ResNet) are extensively trained to solve challenging object recognition problems.

FINDINGS

The two methods described above are first developed based on some training data. They are then applied to test data to evaluate their performance in terms of how accurately they classify the truck body types. Their performance is summarized in terms of a confusion matrix that shows both accurate classification and misclassified trucks.

Table 1 shows the confusion matrix for the first method where the rows represent the actual class and columns the predicted class. The whole numbers correspond to the number of samples and the values in parenthesis are percentage of them with respect to the total number of samples in that category (i.e., total number of samples in the row). Most of the samples are correctly classified. While the overall accuracies are higher than 90%, more refinements can be made to the models to improve these accuracies.

Table 1 Confusion Matrix for the SVM and Heuristic Method

		Predicted			
		Container	**Ref Container**	**Ref Enclosed Van**	**Enclosed Van**
True	**Container**	**279 (93.62%)**	**12 (4.03%)**	**1 (0.34%)**	**6 (2.01%)**
	Ref Container	**0 (0.00%)**	**29 (93.55%)**	**2 (6.45%)**	**0 (0.00%)**
	Ref Enclosed Van	**0 (0.00%)**	**2 (0.99%)**	**185 (91.13%)**	**16 (7.88%)**
	Enclosed Van	**4 (1.08%)**	**0 (0.00%)**	**18 (4.88%)**	**347 (94.04%)**

For the second method, the confusion matrix of each model summarizing the testing results is shown in Table 2. It is evident that all three models produce comparable accuracies and misclassification is a slightly skewed towards containers. The accuracies are a bit lower for refrigerated trailers (i.e., Ref Containers and Ref Enclosed VAs). For example, in the AlexNet model, the containers are classified with 98% accuracy whereas refrigerated containers with 90%. Perhaps, the relatively lower accuracy in the refrigerated containers category could be attributed to the lower number of samples in this category (201 samples) as compared to the other three. It should be noted that the misclassifications are almost always between a trailer type and its refrigerated counterpart. If these are ignored, i.e., if the trailer type and its refrigerated counterpart are considered as one class, the AlexNet model produces about 98% accuracy in distinguishing between these two more aggregate classes.

Table 2 The confusion matrix of all three CNN models

AlexNet					
		Predicted			
		Container	Ref Container	Ref Enclosed van	Enclosed van
True	Container	1832 (98.38 %)	7 (0.38%)	3 (0.16 %)	20 (1.07 %)
	Ref Container	20 (9.95 %)	181 (90.05 %)	0 (0 %)	0 (0 %)
	Ref Enclosed van	1 (0.1 %)	0 (0 %)	1003 (95.70 %)	44 (4.20 %)
	Enclosed van	14 (0.87 %)	1 (0.06 %)	24 (1.5 %)	1564 (97.57 %)
VGGNet					
		Predicted			
		Container	Ref Container	Ref Enclosed van	Enclosed van
True	Container	1802 (96.78 %)	32 (1.72 %)	5 (0.27 %)	23 (1.24 %)
	Ref Container	65 (32.34 %)	136 (67.66 %)	0 (0.00 %)	0 (0.00 %)
	Ref Enclosed van	6 (0.57 %)	0 (0.00 %)	976 (93.13 %)	66 (6.30 %)
	Enclosed van	19 (1.19 %)	1 (0.06 %)	106 (6.61 %)	1477 (92.14 %)
ResNet					
		Predicted			
		Container	Ref Container	Ref Enclosed van	Enclosed van
True	Container	1821 (97.80 %)	5 (0.27 %)	2 (0.11 %)	34 (1.83 %)
	Ref Container	31 (15.42 %)	169 (84.08 %)	0 (0.00 %)	1 (0.5 %)
	Ref Enclosed van	2 (0.19 %)	0 (0.00 %)	1006 (95.99 %)	40 (3.82 %)
	Enclosed van	19 (1.19 %)	1 (0.06 %)	30 (1.87 %)	1553 (96.88 %)

CONCLUSIONS

This study demonstrates that LiDAR data could be effectively utilized to accurately predict truck trailer types. The presented methodology for processing the data involves a series of statistical models, heuristics, and machine learning algorithms to extract pertinent features to distinguish between different truck body types. After extracting key features, a Support Vector Machine (SVM) model is trained to determine whether the subject truck is hauling an enclosed van or an intermodal container. The results of the SVM model on test data show very high level of accuracies at around 98%. After determining the body type, the presence of a refrigerator unit is detected by a heuristic method that searches the density of points at the frontend of the container or dry van. This method also yields relatively high accuracy rates between 92% and 96% on the test data.

In addition to SVM method a transfer learning model is developed. Since the simple or basic features for any type of image datasets are the same, it was possible to transfer features learned by a pre-trained model (ResNet_152, VGGNet, AlexNet) to another classifier and build highly accurate models with relatively small datasets. Four of the most challenging categories of trucks are chosen to train the model. Four different experiments are conducted to find the optimal level of complexity for transferring learned features. It is shown that the first 33 layers of the ResNet_152, the first 7 layers of VGGNet, and the first 3 layers of AlexNet have the best performance on this dataset. With any one of the three CNNs analyzed here, as long as the optimal point for transferring the learning (or features) is selected, it is possible to get around 97% classification accuracy. The computation time for feature extraction shows that it is better to use a shallower CNN models with less parameters for this specific problem since the accuracy of more complex models are almost the same. Results from the confusion matrices show that the models are very accurate (~98%) in distinguishing between enclosed vans and containers, and trucks with refrigerator units (the refrigerator containers or refrigerator enclosed vans) are more prone to be misclassified. The AlexNet model was found to be computationally more efficient to implement and yielded classification accuracies higher than 90% for each one of the four truck body types.

Clearly, the scope of this study is limited in terms of the variety of truck body types considered. However, the authors believe that similar models could be developed to capture salient features for other truck body types, some of which have easily distinguishable characteristics such as tanks and empty flatbeds. This research will ultimately enhance freight and commodity modeling research by providing a detailed breakdown of truck body types at observation stations where a LiDAR sensor is installed.

COMPLETE DOCUMENTATION

This section provides a detailed overview of all the work undertaken in this project including literature synthesis, data collection process, data description, post-estimation analysis, and experimental work.

Past Literature:

Vehicle classification plays a significant role in almost all aspects of transportation engineering and planning applications. In mid 1980s, the FHWA developed a standardized 13-category vehicle classification rule set which meet the needs of many traffic data user applications, such as highway and pavement design, performance monitoring, tolling, transportation planning, and freight planning and modeling, etc. The rule set is designed to classify visual descriptions of vehicles using axle-related metrics, such as the number of axles, axle spacing, number of trailers, and vehicle length with the available intrusive and non-intrusive equipment. The intrusive equipment includes inductive loops, road tubes, and piezo sensors, etc. [3]. The non-intrusive equipment includes video detection system, passive infrared, radar, etc. [3]. The complete list of common and current data collection technologies can be seen in Table 3 [4]. In general, rule-set separates vehicles into categories depending on the passenger vehicles or commodities. For some engineering and planning analyses, people use a generalized 4-bin categorization which include cars, small trucks, large trucks, and multi-trailer trucks.

Table 3 Common technologies for vehicle classification.

Axle-Based	Length-Based
Infrared (passive) (NI)	Dual inductive loops (I)
Laser radar (NI)	Inductive loops (loop signature) (I)
Piezo-electric (I)	Magnetic (magnetometer) (I)
Quartz sensor (I)	Video detection system (NI)
Fiber optic (I)	Microwave radar (NI)
Inductive Loop Signatures (I)	CW Doppler sensors (NI)
Capacitance mats (I)	
Bending plates (I)	
Load cells (I)	
Contact switch closures (e.g., road tubes)	
Specialized inductive loop systems	

Key: Non-Intrusive (NI), Intrusive (I), Source: (FHWA 2016)

However, 4 bins category is not sufficient for some applications, such as estimation of pavement loads or freight planning and modeling. In order to have suitable data for analysis, in 2003, the Transportation Research Board (TRB) Expert Task Group (ETG) on Long-Term Pavement Performance (LTPP) Traffic Data Collection and Analysis developed a new set of rules for classifying vehicles based on sensor outputs available from WIM systems [5]. This rule-set is sufficient for pavement and bridge designs but for freight planning and modeling there is still need for more detailed information of the truck characteristics, such as refrigerated, tank, specialized trailers, etc. FHWA and LTPP rule-sets cannot distinguish detailed characteristics of the truck trailers. It is important to know the type of commodity to support freight demand modeling [6]. For classifying truck body types, researchers applied machine learning algorithms to loop signature and WIM data [1]. Researchers also used WIM data to unanimously re-identify trucks between two observation sites to support freight modeling [7, 8].

Having access to more detailed truck characteristics may help reveal information about the commodity being carried. For example, refrigerated trailers are commonly used to transport perishable produce and meat products, tank trailers are for fuel and other liquid products, and livestock is carried in specialized trailers. It is obvious that not all commodity types can be easily inferred from the externally observable characteristics of the truck or trailer, however, it is possible to narrow the possible types of commodities if the trailers are classified into distinct categories (e.g., car-transporter, tank, enclosed van, intermodal container, empty platform or trailer). Non-intrusive sensors are also widely used for vehicle classification but again they do not provide truck trailer or commodity type. However, researchers have attempted to classify vehicles from data from surveillance cameras [9, 10]. Another non-intrusive sensor is Light Detection Area Ranging (LIDAR) which is becoming widely available in the transportation field due to the recent research and developments in automated driving technologies. In this technology, the raw point-cloud data from these sensors include distance to the target and intensity of the reflected light. From these data, it is possible to construct a 3D representation of the objects within the range of the sensor. Researchers have recently investigated how these sensors could be used for monitoring traffic flow and estimating vehicle trajectories [11-13] and for evaluating highway safety issues [14, 15].

The Artificial Neural Network (ANN) idea has been around since 1960 [16]. There have been many studies using ANN as a machine learning approach in all engineering fields including the transportation community [17]. The advantage of ANN over classical machine learning approaches such as SVM is the ability of this method to perform feature extraction and selection automatically. To get the best performance out of classical methods, researchers need to select the best features to represent the data, which is time-consuming and involves some heuristic procedures. In a sense, for classical methods, some part of the learning has to be done by the researcher. Up to a few years ago, the performance of ANNs and classical methods were almost the same. With the recent advancements in computational power and an increase in the accumulation of data, researchers have noticed an interesting pattern in the performance of machine learning algorithms. The performance of ANNs increases rapidly with more data while the performance of classical methods would not get better after a certain point. This observation has led to increase in new studies about ANNs in the field of computer science. For some tasks, such as image retrieval [18], object detection, and tracking [19, 20] ANN has reached a state of the art performance. Convolutional Neural Network (CNN) image classification is one of the most successful methods in neural network research. It can find the properties of different categories much more accurate than other methods. The drawback of CNN is the need for the tremendous amount of training data and computation power because the model has to optimize numerous parameters in its network structure [21]. Recently, this method got popular among transportation researchers. One study was able to detect cracks on hot-mix asphalt and Portland cement concrete using pavement images with the help of deep CNN [22]. Other researchers have used CNNs for vehicle detection based on satellite images [23] and vehicle classification based on video from surveillance camera [24].

The training process of deep CNNs is time-consuming and needs a huge amount of computational power. Moreover, it can easily lead to overfitting. Some researchers have tried pre-training and fine-tuning strategy to overcome this limitation [25]. They have pre-trained a GoogleNet model [26] on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset to find the initial model. Then using 13,700 images extracted from surveillance cameras, they have fine-tuned the initial model. This approach has reached around 98% accuracy in classification of vehicles. This strategy would solve the overfitting problem but the pre-training procedure is still computationally intensive. By visualizing learned features of CNN [27], researchers have noticed that the network always learn low-level features at the beginning layers and consecutively features would become more complex as you go deeper into the model. Independent from the dataset, low-level features are always the same for almost any type of images. It would be intuitive to keep low-level features learned from one dataset and transfer that knowledge to perform classification on different dataset. This method uses the feature descriptor parts of an already existing trained model such as AlexNet [28] and replaces the classifier part with a new task-specific model. This type of modeling practice is called “transfer learning.” Many researchers have used transfer learning to improve the accuracy and proficiency of new models with limited training data [29-32].

There are many CNN models trained on ImageNet dataset [21] that can be used as pre-trained CNN including AlexNet [28], VGGNet [33], and ResNet [34]. In this study, we have investigated the implementation of transfer learning on these three pre-trained models to find the best performance for the classification of truck images generated from data collected by a 3D LiDAR sensor.

Data Collection

A Velodyne VLP-16 LIDAR [35] along with a camera system was installed on I-64 WB in Hampton Roads, VA. More specifically, it is installed on the post carrying the last VMS gantry before the Willoughby Bridge on the Willoughby Spit side, as seen in Figure 1. Surveillance camera and LIDAR timestamps were synchronized so that the same truck can be found in LIDAR files.

Since trucks are prohibited from traveling in the left lane, LIDAR data collection is limited to the rightmost lane. The LIDAR sensor is approximately 22 ft. above ground and 20 ft. away from the edge of the travel lane. The LIDAR sensor and the surveillance cameras are oriented to get good coverage of the vehicles traveling in the right lane.

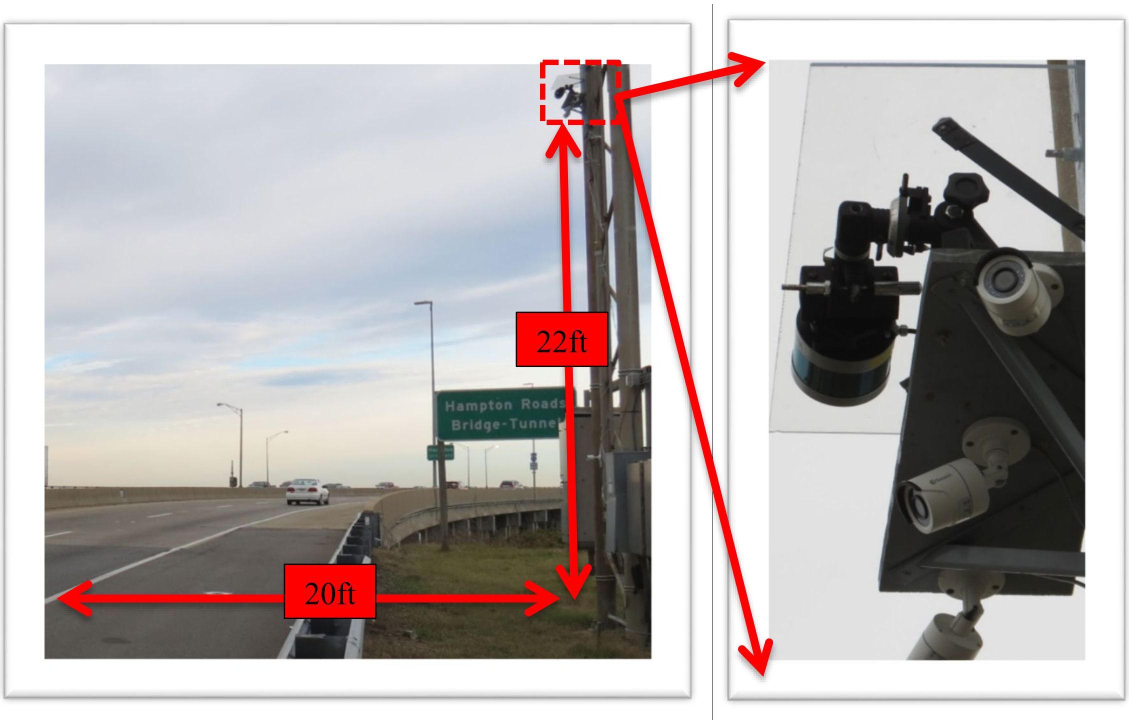


Figure 1 LIDAR and surveillance cameras mounted on the gantry pole before the Willoughby Bridge

The VLP-16 LIDAR sensor comes with 16 beams, which covers a 30° view angle with 360° rotation around its internal z-axis. The LIDAR frequency is 10 Hz, which provides a very rich cloud point dataset. The data points measured by its 16 beams within one complete rotation are called a scan or frame. The unit can be installed in a vertical or horizontal (or any other angle) scanning mode depending on the application. If it is mounted horizontally (Figure 2a) on the roadside, it covers a maximum of 50 meters from the sensor in the longitudinal direction of the roadway, but it will not result in a dense sets of points for each individual vehicle observed. In a vertical configuration (Figure 2b), it covers about a 3.5 meters longitudinal section of the roadway and provides a greater density of points per vehicle. Most passenger vehicles can fit within this range, but vehicles longer than 3.5 meters will not. Therefore, multiple scans or frames need to be combined to create full 3D or 2D profiles of trucks. For this research, the LIDAR is configured in the vertical orientation as shown in Figure 2.

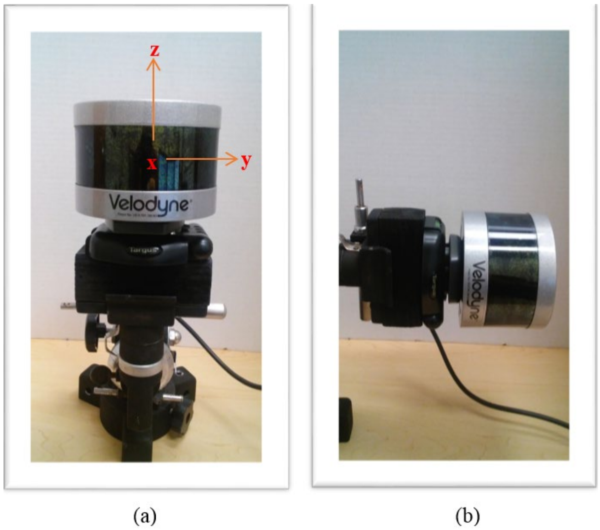


Figure 2 LIDAR orientation (a) horizontal and (b) vertical.

Since trucks are travelling only in the right lane, LIDAR points reflected from objects elsewhere can be excluded from the dataset. Thresholds were established to eliminate these redundant data points. Figure 3 shows a complete LIDAR scan, whereas Figure 4 has the remaining data points after removing the redundant data. All the analyses are performed with the subset of points belonging to vehicles traveling on the right lane, as in Figure 4.

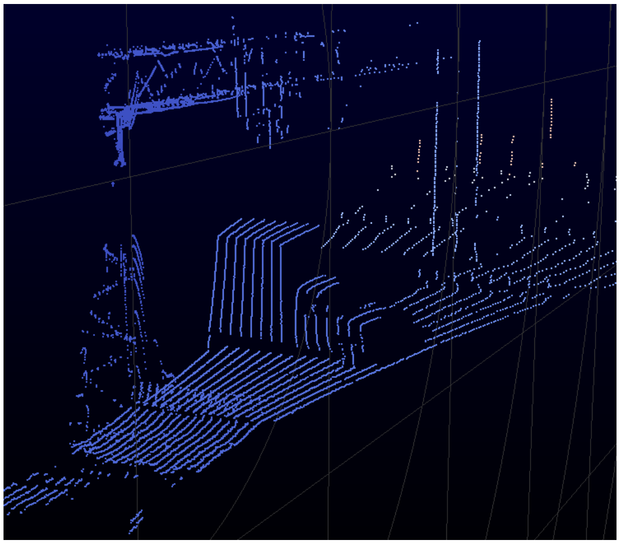


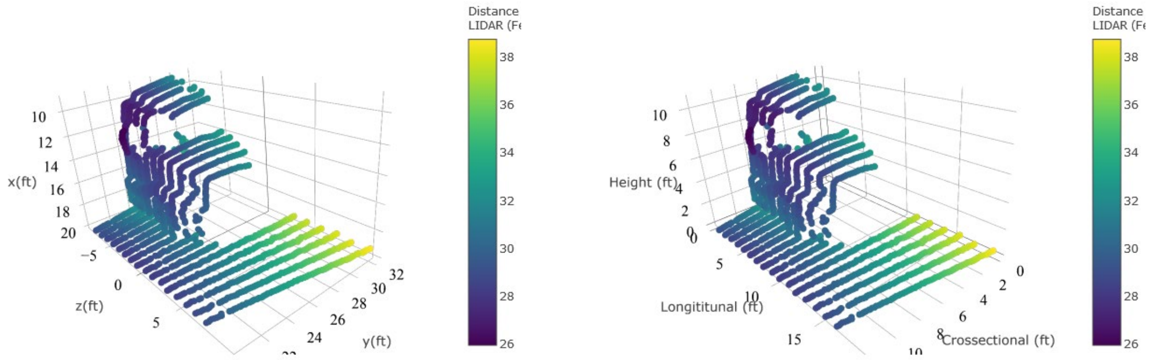
Figure 3 A full scan or frame from the LIDAR sensor

Data Preprocessing

Coordinate Transformation

The LIDAR sensor provides the position of each point in its own a 3D Cartesian coordinate system, as defined in Figure 2 (a). This coordinate system may not be fully aligned with the travel lane. Therefore, the raw data are transformed to a new coordinate system where the x-axis is along the travel direction, the y-axis is perpendicular to the roadway surface, and the z-axis the lateral direction. This transformation is simply accomplished by identifying unit vectors along these three directions. Then, the points are transformed to the new coordinate system by employing a rotation matrix, a commonly used coordinate transformation method.

As seen in Figure 4 (a), a truck enters the field of view of the LIDAR. After coordinate transformation is applied to the raw data, it is recreated as shown in Figure 4 (b) with all the original data preserved. Working in the new coordinate system facilitates the remaining steps where speed is estimated for merging multiple frames as explained next.



(a) (b)

Figure 4 Before (a) and after (b) coordinate transformation

Speed Estimation

As explained above, the entire truck does not fit within the detection zone or the field of view of the LIDAR. Therefore, to generate the full truck profile, multiple frames need to be merged. This can be done if the speed at which the truck is traveling is known. Using the first two consecutive frames and the time instances when the truck enters the scan zone of each beam, the speed of the vehicle can be estimated since the distance between individual beams is known. Likewise, as the truck is departing the detection zone, the last two frames can be utilized in the same manner to estimate another speed. In fact, as long as the vehicle is not occupying the entire set of 16-beams in two consecutive frames, data from such frames can be used similarly to estimate speed. These speeds are then averaged to find a constant average speed for the vehicle. It should be noted that this precision of this method is limited since the distance can only be measured in increments of the distance between two consecutive beams. For this installation, this increment is about one foot. Given the fact that the time between two frames is 0.1 seconds, this discretized measurement of travel distance translates to approximately ±7 mph maximum error (worst case) for a truck traveling around 50 mph. However, since multiple estimates are utilized the actual error is expected to be lower than this. The research team did not have a speed-measuring sensor at the site to quantify the error in the estimated speeds.

Merging Multiple Frames

Based on the estimated average speed, the frames belonging to the same trucks are then merged by shifting the consecutive frames accordingly. A reconstructed sample 2D profile of a truck is shown in Figure 5. To facilitate more efficient computation, the 3D LIDAR points are projected onto a 2-dimentional x-y grid where each cell is 2 inch by 2 inch. A typical FHWA Class 9 truck spends about 1-2 seconds within the LIDAR detection zone at free flow speed. Within this time, LIDAR generates around 30,000 points. Projecting the data to the 2D grid reduces the computation load needed to process this large number of points per truck. This projection reduces the data points drastically to around 1,000 points. The bottom plot in Figure 5 represents the 2D profile of the truck that is generated after merging the cloud points of the truck in the top picture. Each cell of the truck profile is 2in by 2in and color scale represents the average of z-coordinates (or depth) of raw 3D points corresponding to that grid point.

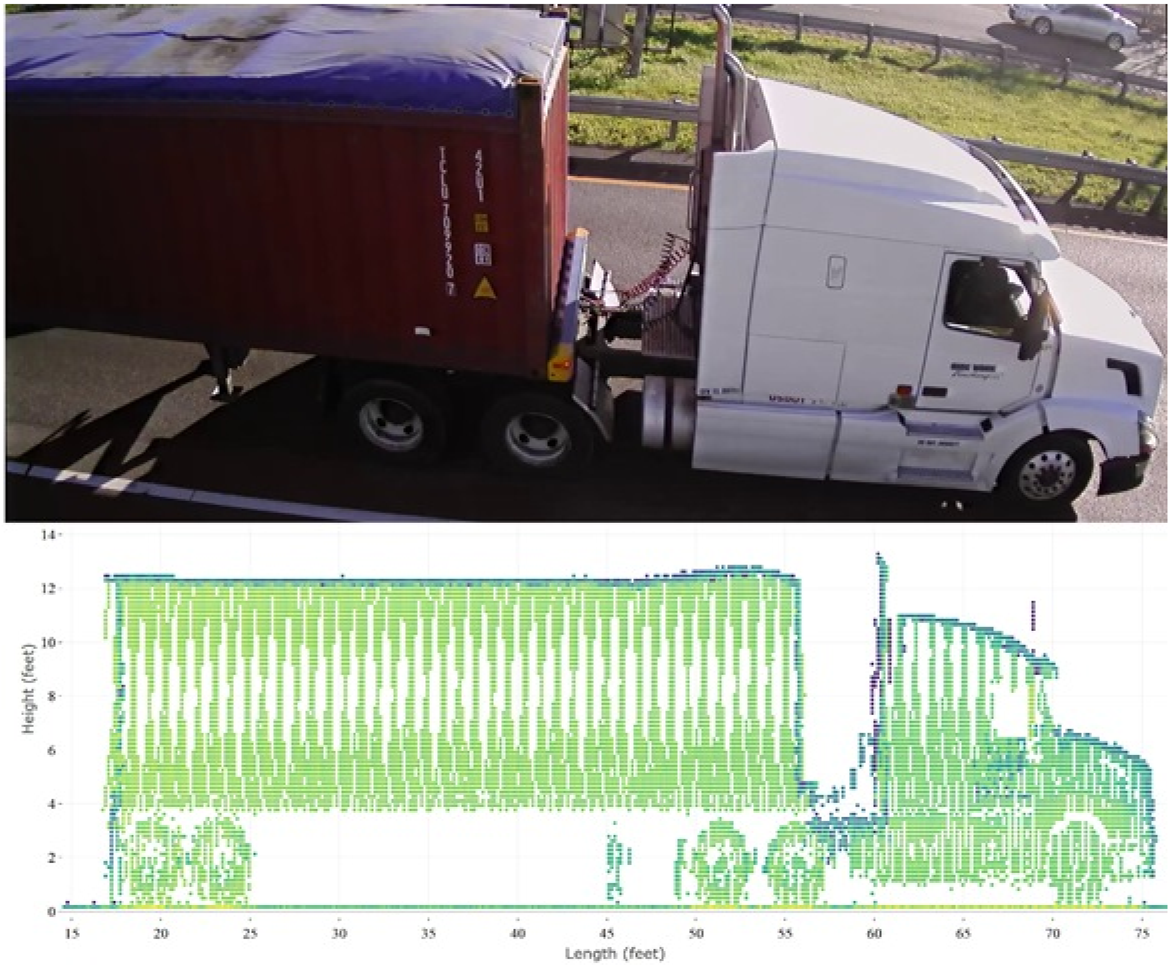
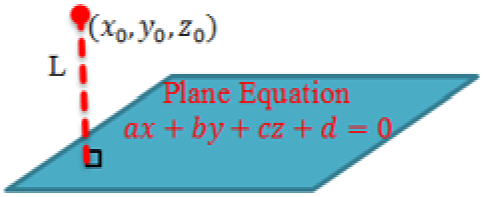


Figure 5 Merged cloud point data projected onto a 2D grid (bottom) and the corresponding truck image (top).

Height Estimation

The heights of individual points of LIDAR data from the roadway surface are found by the following process. First, LIDAR frames belonging to a specific truck are labeled. This process involves consecutively numbering all vehicles passing under the detection zone and assigning each LIDAR beam and frame to the corresponding vehicle. Appendix A shows a sample table demonstrating how point cloud data belonging to individual vehicles (or vehicle IDs) are labeled and extracted from the raw LIDAR data tables. Once points are grouped by vehicle IDs, the height of each individual point is computed with respect to the roadway surface. This is accomplished by using basic geometry and the equation of a point to a plane (i.e., roadway surface) – see Figure 6. The plane equation is found by fitting a surface plane to the data collected when there is no vehicle in the detection zone (roadway surface of the right lane).

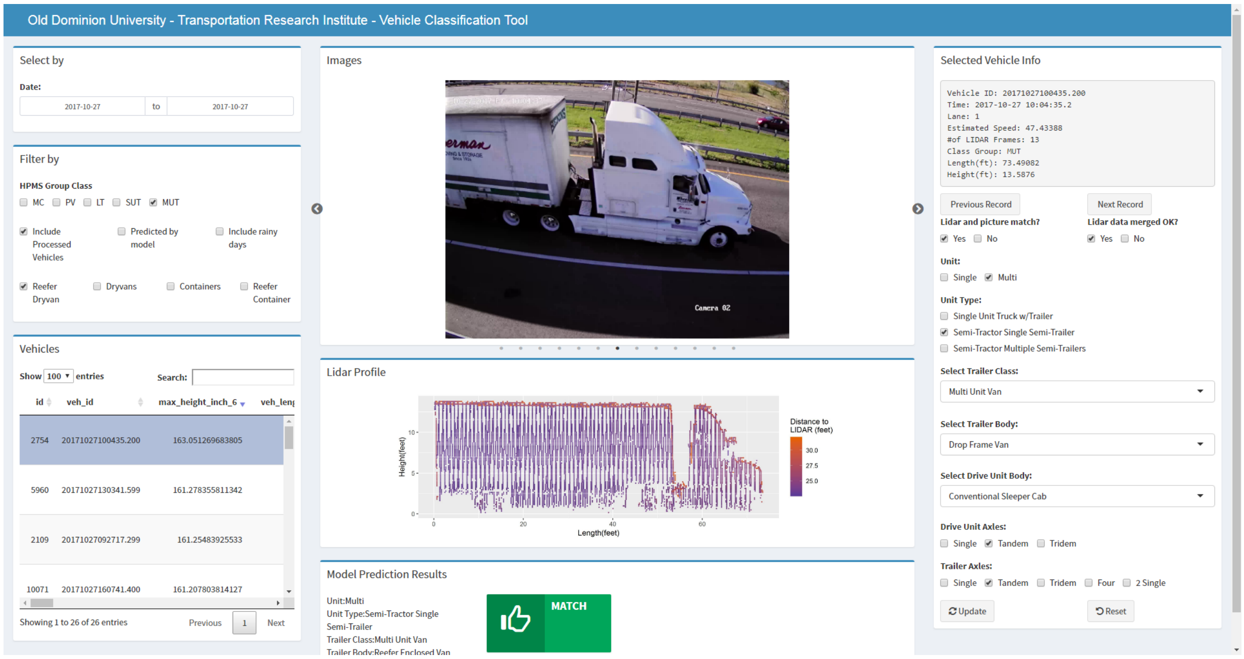


$$L = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}}$$

Figure 6 Equations used to calculate the height of a given point on a vehicle to the roadway surface represented by a plane

Ground Truth Labeling

The process explained above is applied to every vehicle passing under the LIDAR detection zone. Then each vehicle's LIDAR cloud points and actual images which is extracted from the video are stored in a relational database. A custom tool shown in Figure 7 is developed for visually matching LIDAR data with vehicle images along with fields for inputting the vehicle configuration such as truck body type, number of axles, etc.



Old Dominion University - Transportation Research Institute - Vehicle Classification Tool

Select by

Date: 2017-10-27 to 2017-10-27

Filter by

HPMS Group Class

□ MC □ PV □ LT □ SUT ☑ MUT

☑ Include Processed Vehicles □ Predicted by model □ Include rainy days

☑ Reefer Dryvan □ Dryvans □ Containers □ Reefer Container

Vehicles

Show 100 entries Search:

id	veh_id	max_height_inch_6	veh_leng
2754	20171027100435.200	163.051269683805	
5960	20171027130341.599	161.278355811342	
2109	20171027092717.299	161.25483925533	
10071	20171027160741.400	161.207803814127	

Showing 1 to 26 of 26 entries Previous 1 Next

Images

Lidar Profile

Model Prediction Results

Unit:Multi
Unit Type:Semi-Tractor Single Semi-Trailer
Trailer Class:Multi Unit Van
Trailer Body:Reefer Enclosed Van

Selected Vehicle Info

```
Vehicle ID: 20171027100435.200
Time: 2017-10-27 10:04:35.2
Lane: 1
Estimated Speed: 47.43388
#of LIDAR Frames: 13
Class Group: MUT
Length(ft): 73.49082
Height(ft): 13.5876
```

Lidar and picture match? ☑ Yes □ No

Lidar data merged OK? ☑ Yes □ No

Unit: □ Single ☑ Multi

Unit Type:
□ Single Unit Truck w/Trailer
☑ Semi-Tractor Single Semi-Trailer
□ Semi-Tractor Multiple Semi-Trailers

Select Trailer Class: Multi Unit Van

Select Trailer Body: Drop Frame Van

Select Drive Unit Body: Conventional Sleeper Cab

Drive Unit Axles: □ Single ☑ Tandem □ Tridem

Trailer Axles: □ Single ☑ Tandem □ Tridem □ Four □ 2 Single

Figure 7 Interface developed to manually label vehicle classes

Testing and Training Dataset

For this study, only a small subset of the data are utilized since manually labeling vehicles into different categories is time consuming. We selected several days without precipitation, and time periods during daylight times to extract images and LIDAR data. Overall, 3,336 trucks are manually labeled and verified from the video files. These trucks and their classes are listed in Table 4. Sample truck images belonging to each class are shown in Figure 13. The sample is split into two groups: training and testing. The data for the set of trucks in the training group are used for developing and training the classification algorithms (presented in the next section). The testing data are used for quantifying the accuracy of the models.

Table 4 Training and testing data

Trailer Type	Training	Testing	Total
Intermodal container	691	298	990
Refrigerated container	81	31	112
Dry van	854	369	1,225
Refrigerated dry van	493	203	698
Other (Platform, auto transport, tank, dump, etc.)	218	98	319
Total Samples	2,337	999	3,336

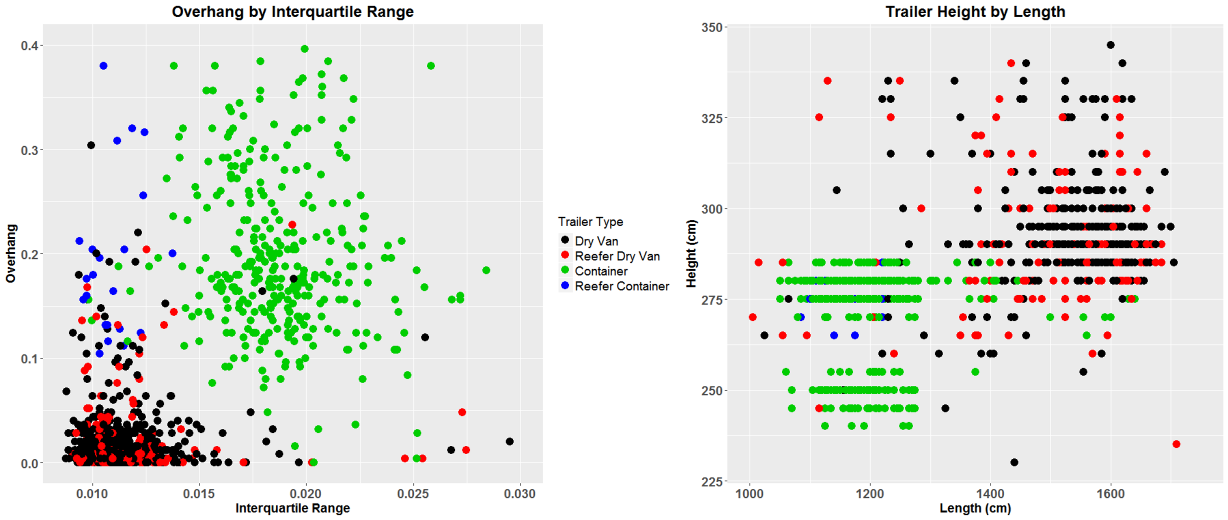
SVM Method

To be able to categorize trucks into distinct groups, pertinent features that help distinguish trucks in each group are needed. After experimenting with different combination of variables, the six variables listed in Table 5 are found to be effective for the purpose of this study and are used as input features.

Table 5 Features extracted for each truck

Feature	Abbreviation	
Trailer length	TrL	
Trailer height	TrH	
Trailer surface interquartile range	TrIQR	
Top density	TD	
Bottom density	BD	
Overhang	OH	

These variables show enough variation across the two types of the truck trailers as shown in plots in Figure 8(b). In the sample data, refrigerated containers have similar surface smoothness as dry vans. Therefore, their TrIQR or TrSSD would be similar to that of dry vans but their overhang (OH) measure is not as can be observed in Figure 8(a).



(a) **(b)**

Figure 8 (a) Overhang by Interquartile Range (b) Trailer height by trailer length

Trailer Length (TrL) and Trailer Height (TrH)

To find the trailer dimensions, a rectangular box is drawn as shown in Figure 9. While not all enclosed vans have a perfect rectangular side profile, this assumption is accurate enough for the analyses conducted here. To find the needed dimensions, four sides of the rectangle (top, bottom, front, and back) needs to be determined. Since all trailers analyzed here have a relatively straight side profile, this is exploited to determine the four sides needed.

For example, to determine the x-coordinate for the frontend of the trailer, a rectangular region (e.g., 2ft high and 6.5ft long) much smaller than the trailer is selected where the ends of this region are safely away from the four boundaries of the trailer. As indicated earlier each grid cell contains the average z-coordinate (depth) of all LIDAR points corresponding (i.e., projected) to that cell. We then collapse the vertical dimension of this sample (2ft by 6.5ft) rectangle by taking the average of all z-coordinates for all y-levels at a given x coordinate. This results in one-dimensional vector (along x-axis) with z-coordinates as the variables. We then fit a simple linear regression model of the form $z = mx + b$ to this vector. We then use this model to predict z-coordinate (depth) as function of x, where x is now extended further towards the tractor unit where the frontend of the trailer is expected to be. By measuring the difference between predicted z (depth) and actual z (depth) we can identify a sudden drop or change in the surface depth. The x-coordinate where the first sudden drop occurs will be identified as the frontend of the trailer. This technique is repeated to determine the backend, top, and bottom of the trailer as well by scanning in the relevant direction. Once these four coordinates are found the trailer dimensions are straightforward to compute.

Trailer Surface Standard Deviation (TrSSD) and Trailer Surface Interquartile Range (TrIQR)

The surface smoothness of intermodal containers is quite different from that of enclosed vans. Typical intermodal containers are made of corrugated sheet metal. The VLP-16 LIDAR sensor is accurate enough to detect the depth variation on the side surface of these containers. Therefore, two variables (*TrSSD* and *TrIQR*) are defined to capture that information. To do so, similar to the analysis above, a rectangular region away from the estimated sides of the trailer is selected. A 2D-plane is fit to the depth data (z-coordinate) by using an ordinary linear regression model. This plane in essence represents the average z-coordinates of the side surface of the trailer. The deviations or residuals from this plane are then calculated. The two variables, *TrSSD* and *TrIQR,* are defined as the standard deviation and interquartile range of these residuals, respectively.

Top Density (TD) and Bottom Density (BD)

Dry vans and refrigerated dry vans look the same except the later has a refrigerator unit attached to the dry van's frontend. The refrigerator unit doesn't cover the whole spaces on the frontend. There is a gap between tractor chassis and underneath of the refrigerator unit, as seen in Figure 9. Therefore, in the 2D profile of a refrigerated unit, there are more LIDAR points observed at the top section protruding forward from the trailer. This information can be captured by defining two regions (rectangles) at the frontend of the trailer. Since the x-coordinate of the frontend is estimated, we can go a certain distance towards the tractor (about 2.5ft) from the trailer edge (see the highlighted section in Figure 9) and calculate the density of points in a rectangle close to the top of the trailer and another rectangle to the bottom of the trailer. The heights of the bottom and top rectangles are taken as 2ft and 4ft respectively. We then simply count the number of grid cells that are not empty and divide the count by the area of the rectangle to find the density of points.

Overhang

Another useful variable is the overhang distance or the distance from the trailer end to where the rear tandem axle is. This distance is quite small for intermodal trailer containers whereas it is typically larger for enclosed or dry vans. Rather than attempting to measure the overhang distance from the 2D profiles, we use a 2ft high by 4ft long rectangle starting at the origin of the coordinate system and measure the density of points within this rectangle. This turns out to be a good surrogate variable to capture the variation in rear overhang among the trailers of different types.

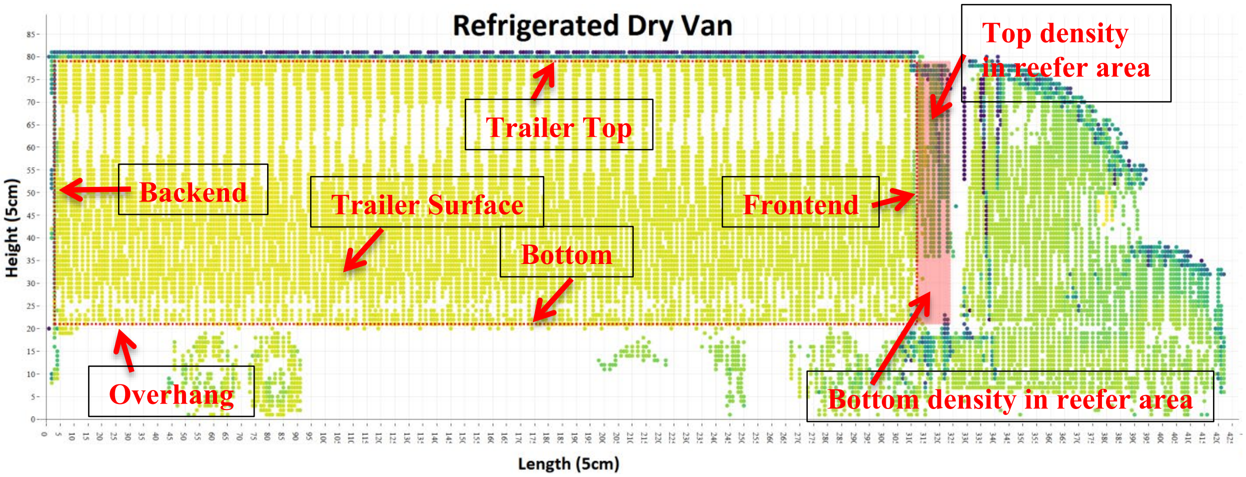


Figure 9 Extracted features annotated on a 2D profile of a refrigerated dry van

Heuristics Method

After applying the SVM model, the unknown truck data are now labeled as either dry van or intermodal container. In the second step, a heuristic is implemented to detect whether there is a refrigerator on the trailer unit. This is accomplished by using the following two variables:

- Top density (TD)
- Bottom density (BD)

The ratio of BD to TD is calculated. If this ratio is less than a threshold (here taken to be 0.7 per performance on the training samples) then, this will indicate that there are many more LIDAR points observed at the top region, and hence a refrigerator unit is assumed to be present on the trailer.

Table 6 shows the results of the two-step approach explained above. The SVM model correctly identifies the trailer types with 98%, 98%, and 93% accuracy for intermodal containers, dry vans, and other trailers, respectively. In step 2, in order to separate refrigerated and non-refrigerated trailers from the dry vans and intermodal trailers, the heuristic method explained above is applied. This method also shows high accuracy ranging between 90% and 100%. Table 6 also includes results from Hernandez et al. study [1] which proposes a classification method for truck body configuration using weigh-in-motion and inductive loop signature data. For all cases, the proposed method in this study gives more accurate results most likely due to the richer LIDAR data and more pertinent information and features extracted from the raw data. It is interesting to note that both methods have the best performance in the refrigerated intermodal category and lowest in the refrigerated dry (or enclosed) vans.

Table 6 Results from the SVM and Heuristic

	Proposed Method		**Hernandez et al (2016) Study**	
Trailer Type	**Number of test samples**	**CCR** (Correct Classification Rate**)**	**Number of test samples**	**Best CCR**
Step 1 (Classification Method):		**SVM**		
Dry van	572	98%		
Intermodal container	329	98%		
Other	98	93%		
Step 2 (Heuristic):				
Dry van	369	94%	2,329	83.8%
Refrigerated dry van	203	90%	1,565	75.3%
Intermodal container	298	94%	131	87.8%
Refrigerated intermodal container	31	100%	16	93.8%

Transfer Learning Method

In the transfer learning method, a pre-trained model would be chosen as a feature descriptor or extractor. Over the years, researchers developed various architectures of deep learning to solve challenging object recognition problems. Of these, VGGNet, AlexNet, and ResNet are well-known and popular as each of these achieved high levels of accuracy in classifying objects in large image databases. Therefore, in this study, to find out which pre-trained model would be more suitable for transferring the learnings to the truck trailer classification, three deep CNN models, i.e., VGGNet, AlexNet, and ResNet, have been investigated. Features extracted from these pre-trained models become the selected features and can be used as input to any classification algorithm. In our previous study [9], we demonstrated how the features from a pre-trained ResNet model could be utilized for truck classification based on camera images. The features from ResNet are used as input to a Multi-Layer Perceptron (MLP) neural network which is shown to outperform other machine learning algorithms such as SVM or Kth Nearest Neighbors [9]. In this study, we extend the transfer learning idea for solving the truck classification problem by considering additional well-known deep NNs and applying them to a new type of dataset, i.e., LIDAR point cloud data expressed as an image.

All three CNN models listed above are already trained on the ILSVRC dataset. Since the model is pre-trained, extractions of the already learned features and using them directly will save a great amount of computation time. However, features in a CNN grow in complexity as we step deeper into the network. Therefore, a key task is determining the optimal point at which the pre-trained model structure should be cut or stopped in order to get the right level of feature complexity for our task. Four different positions for the feature extraction has been investigated on all three models as shown in Figure 10 & Figure 12. The features extracted from these models are used as the feature descriptors for the respective MLP classifier which has two fully connected layers. Hyper parameter optimization of this classifier was done through a simple grid-search to find the optimum number of hidden units in each fully connected layer. The implemented MLP classifier has 128 hidden units for the first layer and 64 hidden units for the second one, the learning rate is 0.001, and 200 epochs of training was done. Some specific details of the three deep networks are provided below.

In 2012, Krizhevsky *et al.* have developed AlexNet which outperformed all the models at the ILSVRC 2012 competition. AlexNet is one of the most popular CNNs and usually would be considered as a baseline model. It has five convolutional layers followed by three fully connected layers. The size of the input image for this model is $227 \times 227 \times 3$ and consecutively, number of units in each layers are 96, 256, 384, 384, 256, 4096, 4096, and 1000. The first-, second-, and fifth-convolutional layers are followed by maxpooling layers. The first and second maxpooling layer is followed by local contrast normalization. The nonlinear activation function of each unit is the rectified linear units (ReLUs). The architecture of the model with the details and proposed position for feature extraction is shown in Figure 10.

The VGGNet had an outstanding performance in the ILSVRC 2014 competition by winning the second place. It consists of 16 convolutional layers and the structure is uniform. The model characteristics are similar to AlexNet but number of filters are more. One drawback of this model would be the big structure, which has around 138 million parameters. It takes lots of computation power to train this model on ImageNet dataset. The high number of filters in this model makes it one of the most preferred choices as a feature extractor. The architecture of the model with the details and proposed position for feature extraction is shown in Figure 10.

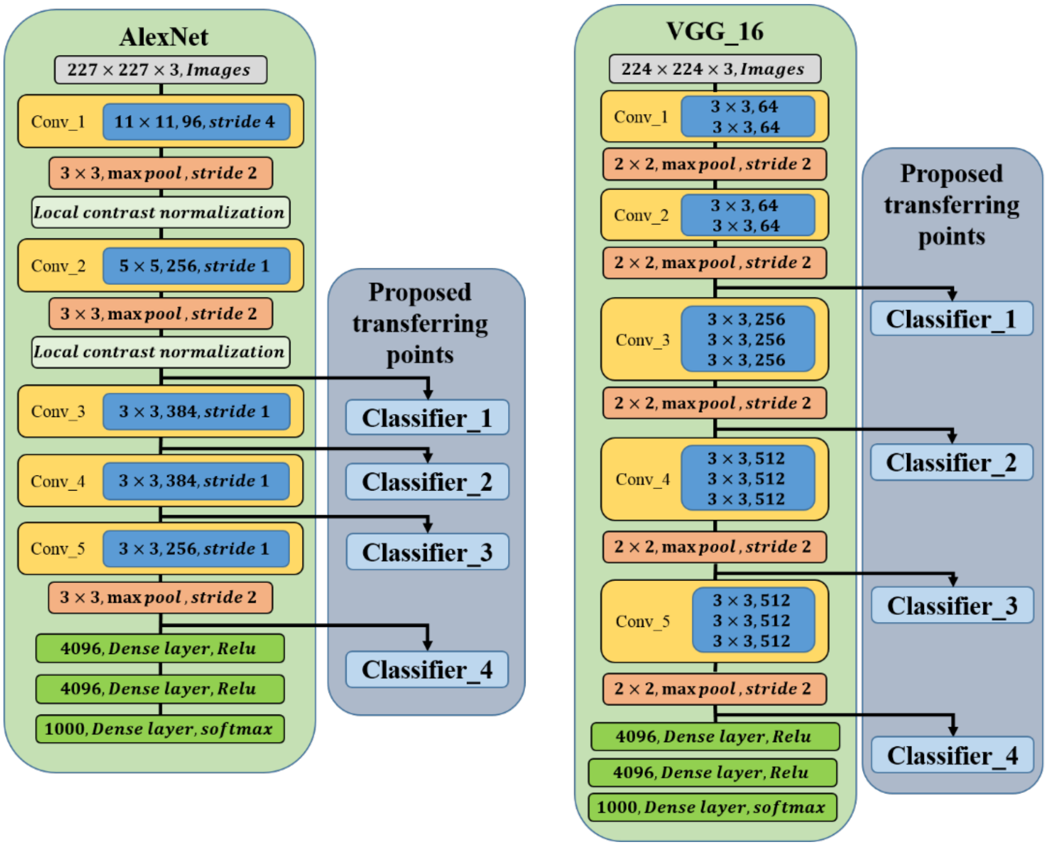


Figure 10 The architecture of AlexNet (left) and VGGNet (right)

ResNet model proposes residual learning blocks to solve the degradation problem caused by multiple nonlinear layers. By using residual learning, if the optimal solution for a specific case is closer to an identity mapping (i.e., the output is a slightly altered version of the input), the solvers can reach it by simply driving the weights of the multiple nonlinear layers toward zero. This way, the solver should converge easier by retaining the input rather than learning the function like a new one. The mathematical formulation of the added residual learning units can be expressed as:

$$y = F(x, \{W_i\}) + x$$

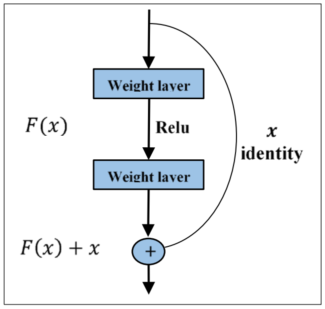


Figure 11 Identity block

Where x and y are, respectively, the input and output vectors of the layers considered; and the function $F(x, \{W_i\})$ represents the residual mapping to be learned. The architecture of this building block is represented in Figure 11. The added shortcut solves the degradation problem without introducing extra parameters or computation complexity. The ResNet architecture used in this article has 151 convolutional layers and a final dense layer with a Softmax activation function. The structure of the model, along with its respective hidden units, is presented in Figure 12. As it can be seen, a residual learning block is defined for every few stacked layers (yellow boxes). Building blocks are shown in white boxes with the numbers of residual blocks stacked written on the right (i.e.,× 3). Down-sampling is done by blocks conv3_1, conv4_1, and conv5_1 with a stride of 2.

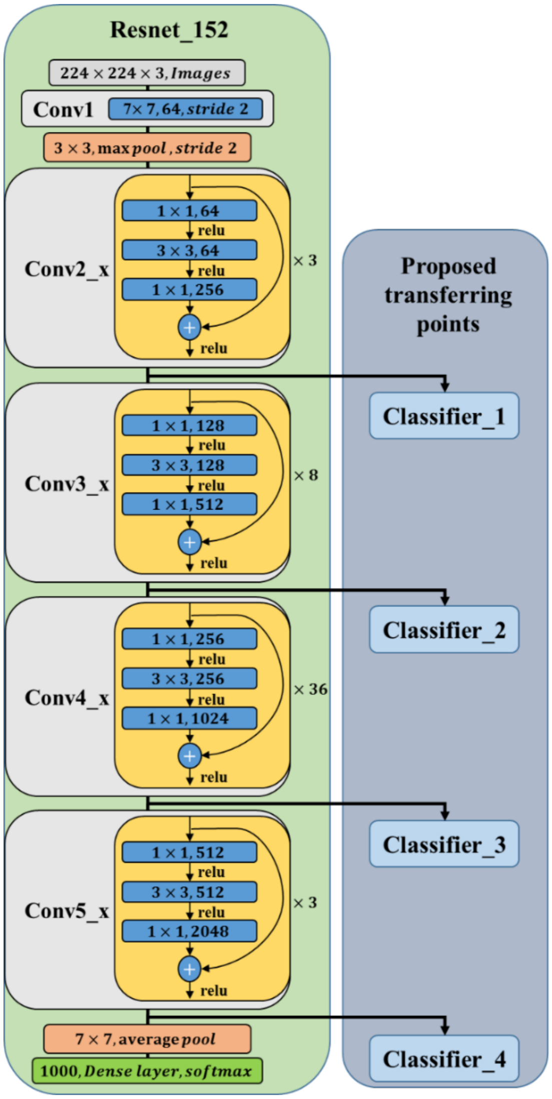


Figure 12 The architecture of ResNet

In summary, Table 7 presents some of the characteristics of these three models.

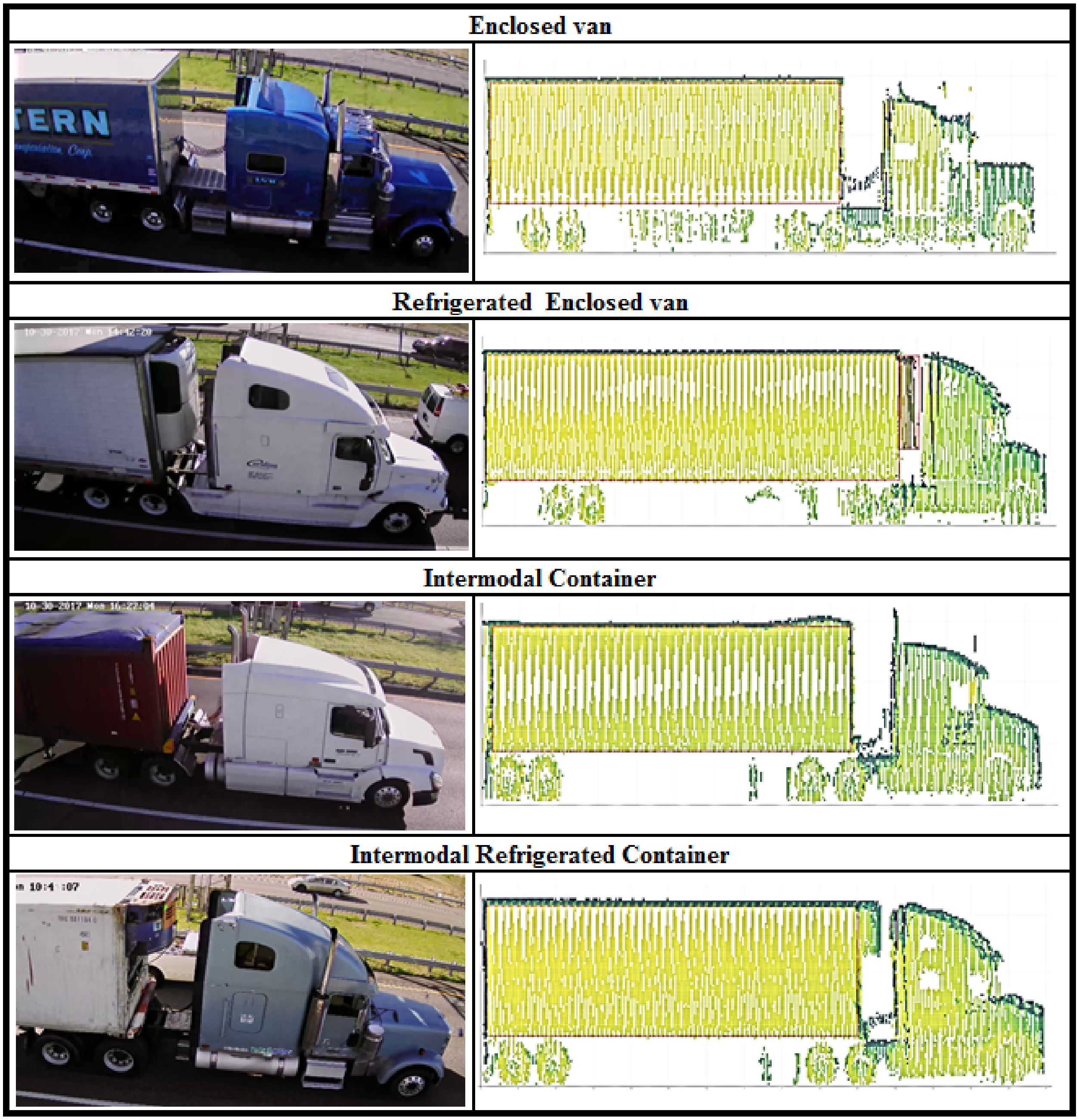
Table 7 Some of the characteristics of the pre-trained models

	ResNet	**VGGNet**	**AlexNet**
Number of layers	152	16	8
Number of convolutions	151	13	5
Number of parameters (M)	60.2	138	62.3

There are many categories of truck body configurations that one can consider [1]. In this study, four of the most challenging categories have been selected for classification, i.e., trailers with containers and enclosed vans with or without refrigerator. As it is shown in Figure 13, the four examined body types are very similar in structure and shape. The total number of truck profiles used in this study is 4,714 out of which 1,628 are enclosed vans, 1,032 are refrigerated enclosed vans, 1,869 are containers and the rest are refrigerated container. The profiles have been saved as images resized to be consistent with the pre-trained input of CNNs. 80% of the data are used for training, and the rest is set aside to be used as the test data. Ground truth labeling of these images was done manually. All computations in this article were conducted with Tensorflow platform on Windows 7 OS with Intel Xeon E5-2630 2.40 GHz and an NVIDIA Quadro K4200 GPU with 4 GB memory. Since parameters of the pre-trained models are fixed and would not change during the training, it is intuitive to do the feature extraction only once and save them to perform the training faster. Table 2 presents computation time needed for feature extraction of each algorithm on four level of complexity on all datasets. While the number of parameters for ResNet and AlexNet are almost the same (Table 7) but their computation time is significantly different which is due to the deep network of ResNet (152 layers). The VGGNet is not as deep as ResNet but still has a significant difference with the AlexNet in terms of computation time which is because it has a much wider network (number of parameters).

Table 8 Computation time for feature extraction for all images in the dataset

	Computation time (min)		
Level of complexity	**ResNet**	**VGGNet**	**AlexNet**
Features_for_Classifier_1	**10.8**	**11.3**	**1.4**
Features_for_Classifier_2	**15.8**	**18.8**	**1.5**
Features_for_Classifier_3	**40.3**	**26.5**	**1.6**
Features_for_Classifier_4	**44.4**	**29.7**	**1.7**
Total	**111.3**	**86.2**	**6.2**


Enclosed van

Refrigerated Enclosed van

Intermodal Container

Intermodal Refrigerated Container

Figure 13 Sample truck types and their projected merged cloud point data onto a 2D surface.

Four different placements of the MLP classifier, as shown in Figure 10 and Figure 12, have been tested to identify the optimal point at which the pre-trained structure should be cut. A K-fold cross-validation procedure has been used to measure accuracy with K equals to 5. Accuracy is the fraction of correct predictions over total number of predictions. This method of accuracy calculation ensures that model is not over fitted. The average accuracy results for 5-fold cross-validation of each model on the test data are presented in Table 9.

Table 9 Average accuracy (%) of 5-fold cross-validation for proposed positions for the classifier

Models	**ResNet**	**VGGNet**	**AlexNet**
Classifier_1	**89.3**	**88.6**	**85.7**
Classifier_2	**96.5**	**93.4**	**97.5**
Classifier_3	**92.2**	**92.3**	**92.4**
Classifier_4	**90.4**	**90.7**	**89.9**

In all models, the Classifier_1 is the worst performing model in the classification task. Features are primary and basic at this level of the network and the Classifier_1 struggles to correctly identify the correct truck types based on these features. Examples of possible features at this level will be a color change, the shape of lines, edges, etc. It is evident that it is hard to identify between a container and an enclosed van using these simplistic features. Features grow in complexity as we go deeper in the network and Classifier_2 will get more complex features from the pre-trained CNN compared to Classifier_1. By the same logic, Classifier_3 and Classifier_4 should be more accurate than their proceeding peers. However, the performance of the last two proposed placements are lower than Classifier_2. This happens because the features beyond Classifier_2 are becoming unnecessarily complicated for the classifier to distinguish between these two vehicle classes. The features at these deeper layers might be appropriate for the wide range of images they are trained on but not for the relatively simpler problem being solved in this study. This trend is consistent between all three models. In other words, the optimal point for selecting the best-suited features for classification of vehicle classes is about $^{2}/_{4}$ down the given network. In this case, the first 33 layers of ResNet_152, the first 7 layers of VGGNet, and the first 3 layers of AlexNet have the best performance for the feature extraction task.

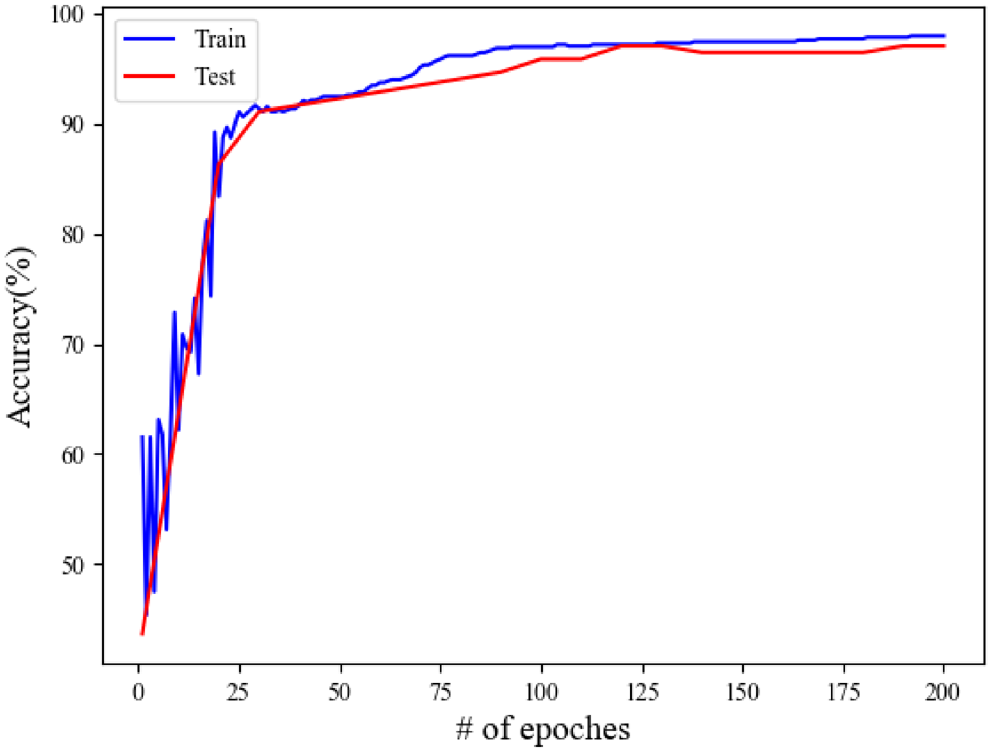


Figure 14 Convergence of the MLP model accuracy for Classifier_2 of ResNet

The convergence of model accuracy for the Classifier_2 of ResNet as an example is presented in Figure 14. An epoch is when all the training samples are used once to update the weights of the MLP by the optimization algorithm that iteratively improves the model variables (e.g., weights). The accuracy of test data follows approximately the same trend as the accuracy of training data, and after around 100 epochs the model becomes steady.

The confusion matrix of each model summarizing the testing results is shown in Table 2. The whole numbers correspond to the number of samples and the values in parenthesis are percentage of them with respect to the total number of samples in that category (i.e., total number of samples in the row). It is evident that all three models produce comparable accuracies and misclassification is a slightly skewed towards containers. The accuracies are a bit lower for refrigerated trailers (i.e., Ref Containers and Ref Enclosed VAs). For example, in the AlexNet model, the containers are classified with 98% accuracy whereas refrigerated containers with 90%. Perhaps, the relatively lower accuracy in the refrigerated containers category could be attributed to the lower number of samples in this category (201 samples) as compared to the other three. It should be noted that the misclassifications are almost always between a trailer type and its refrigerated counterpart. If these are ignored, i.e., if the trailer type and its refrigerated counterpart are considered as one class, the AlexNet model produces about 98% accuracy in distinguishing between these two more aggregate classes.

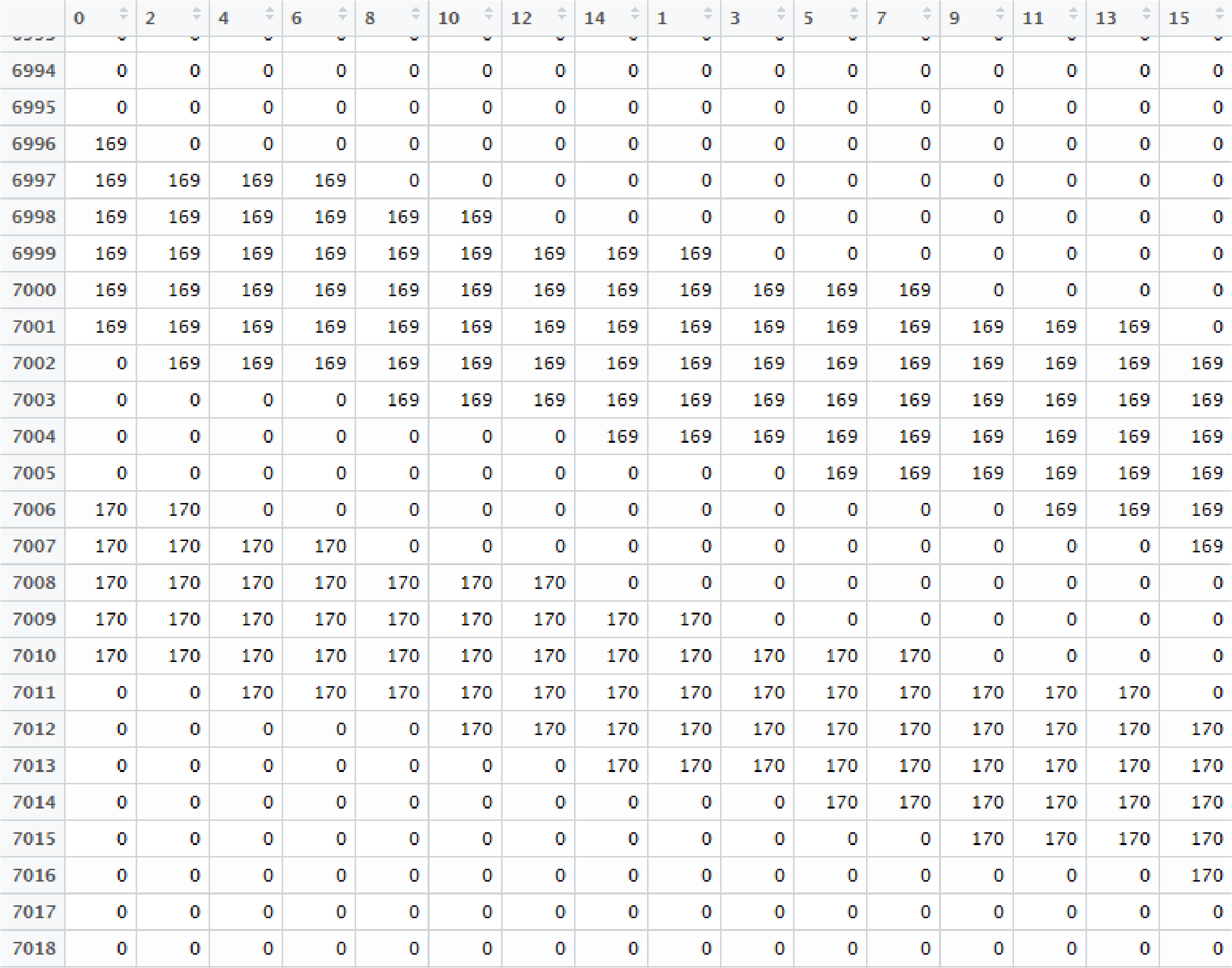
REFERENCES

1. Hernandez, S.V., 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, 2016. **68**: p. 1-21.
2. Burges, C.J., *A tutorial on support vector machines for pattern recognition.* Data mining and knowledge discovery, 1998. **2**(2): p. 121-167.
3. Martin, P.T., Y. Feng, and X. Wang, *Detector technology evaluation*. 2003, Citeseer.
4. FHWA, *Traffic Monitoring Guide*. 2016.
5. Nichols, A.P. and M. Cetin, *Numerical Characterization of Gross Vehicle Weight Distributions from Weigh-in-Motion Data.* Transportation Research Record, 2007. **1993**(1): p. 148-154.
6. de Dios Ortúzar, J. and L.G. Willumsen, *Modelling Transport*. 2011: Wiley.
7. Cetin, M., C.M. Monsere, and A.P. Nichols, *Bayesian Models for Reidentification of Trucks Over Long Distances on the Basis of Axle Measurement Data.* Journal of Intelligent Transportation Systems, 2011. **15**(1): p. 1-12.
8. Cetin, M., et al., *Key Factors Affecting the Accuracy of Reidentification of Trucks over Long Distances Based on Axle Measurement Data.* Transportation Research Record, 2011. **2243**(1): p. 1-8.
9. Vatani Nezafat, R., S. Behrouz, and M. Cetin, *Classification of truck body types using a deep transfer learning approach*, in *The 21st IEEE International Conference on Intelligent Transportation Systems*. 2018.
10. Zhang, G., 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, 2007. **1993**: p. 138-147.
11. Sun, Y., et al., *3-D Data Processing to Extract Vehicle Trajectories from Roadside LiDAR Data.* Transportation Research Record, 2018. **0**(0): p. 0361198118775839.
12. Sazara, C., R.V. Nezafat, and M. Cetin. *Offline reconstruction of missing vehicle trajectory data from 3D LIDAR*. in *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017.
13. Aijazi, A.K., et al. *Automatic detection of vehicles at road intersections using a compact 3D Velodyne sensor mounted on traffic signals*. in *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016.
14. Aycard, O., et al. *Intersection safety using lidar and stereo vision sensors*. in *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011.
15. Khattak, A., S. Hallmark, and R. Souleyrette, *Application of Light Detection and Ranging Technology to Highway Safety.* Transportation Research Record: Journal of the Transportation Research Board, 2003. **1836**: p. 7-15.
16. Lippmann, R., *An introduction to computing with neural nets.* IEEE Assp magazine, 1987. **4**(2): p. 4-22.
17. Faghri, A. and J. Hua, *Evaluation of artificial neural network applications in transportation engineering.* Transportation Research Record, 1992. **1358**: p. 71.
18. Ku, W.-L., H.-C. Chou, and W.-H. Peng. *Discriminatively-learned global image representation using CNN as a local feature extractor for image retrieval*. in *Visual Communications and Image Processing (VCIP), 2015*. 2015. IEEE.
19. Girshick, R., *Fast r-cnn.* arXiv preprint arXiv:1504.08083, 2015.

20. Ren, S., et al. *Faster r-cnn: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*. 2015.
21. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International Journal of Computer Vision, 2015. **115**(3): p. 211-252.
22. Gopalakrishnan, K., et al., *Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection.* Construction and Building Materials, 2017. **157**: p. 322-330.
23. Chen, X., et al., *Vehicle detection in satellite images by hybrid deep convolutional neural networks.* IEEE Geoscience and remote sensing letters, 2014. **11**(10): p. 1797-1801.
24. Adu-Gyamfi, Y.O., et al., *Automated vehicle recognition with deep convolutional neural networks.* Transportation Research Record, 2017. **2645**(1): p. 113-122.
25. Zhuo, L., et al., *Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks.* Machine Vision and Applications, 2017. **28**(7): p. 793-802.
26. Szegedy, C., et al. *Going deeper with convolutions*. 2015. Cvpr.
27. Zeiler, M.D. and R. Fergus. *Visualizing and understanding convolutional networks*. in *European conference on computer vision*. 2014. Springer.
28. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.
29. Hu, F., et al., *Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery.* Remote Sensing, 2015. **7**(11): p. 14680.
30. Pan, S.J. and Q. Yang, *A survey on transfer learning.* IEEE Transactions on knowledge and data engineering, 2010. **22**(10): p. 1345-1359.
31. Razavian, A.S., et al. *CNN features off-the-shelf: an astounding baseline for recognition*. in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. 2014. IEEE.
32. Donahue, J., et al. *Decaf: A deep convolutional activation feature for generic visual recognition*. in *International conference on machine learning*. 2014.
33. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.
34. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
35. Velodyne. *PUCK™ VLP-16 LIDAR*. 2018 [cited 2018 September 15]; Available from: https://velodynelidar.com/vlp-16.html.

APPENDIX A

Labeling of each LIDAR frame (rows) and beam (columns) belonging to individual vehicles numbered consecutively (numbers in the table)



	0	2	4	6	8	10	12	14	1	3	5	7	9	11	13	15
6994	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6995	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6996	169	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6997	169	169	169	169	0	0	0	0	0	0	0	0	0	0	0	0
6998	169	169	169	169	169	169	0	0	0	0	0	0	0	0	0	0
6999	169	169	169	169	169	169	169	169	169	0	0	0	0	0	0	0
7000	169	169	169	169	169	169	169	169	169	169	169	169	0	0	0	0
7001	169	169	169	169	169	169	169	169	169	169	169	169	169	169	169	0
7002	0	169	169	169	169	169	169	169	169	169	169	169	169	169	169	169
7003	0	0	0	0	169	169	169	169	169	169	169	169	169	169	169	169
7004	0	0	0	0	0	0	0	169	169	169	169	169	169	169	169	169
7005	0	0	0	0	0	0	0	0	0	0	169	169	169	169	169	169
7006	170	170	0	0	0	0	0	0	0	0	0	0	0	169	169	169
7007	170	170	170	170	0	0	0	0	0	0	0	0	0	0	0	169
7008	170	170	170	170	170	170	170	0	0	0	0	0	0	0	0	0
7009	170	170	170	170	170	170	170	170	170	0	0	0	0	0	0	0
7010	170	170	170	170	170	170	170	170	170	170	170	170	0	0	0	0
7011	0	0	170	170	170	170	170	170	170	170	170	170	170	170	170	0
7012	0	0	0	0	0	170	170	170	170	170	170	170	170	170	170	170
7013	0	0	0	0	0	0	0	170	170	170	170	170	170	170	170	170
7014	0	0	0	0	0	0	0	0	0	0	170	170	170	170	170	170
7015	0	0	0	0	0	0	0	0	0	0	0	0	170	170	170	170
7016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	170
7017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0