U.S. Department
of Transportation
Federal Railroad
Administration

Office of Research,
Development and Technology
Washington, DC 20590

Intelligent Crossing Assessment and Traffic Sharing System (i-CATSS)

DOT/FRA/ORD-22/15

Final Report | April 2022

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
May 2022	Technical Report	August 2019 – February 2021

4. TITLE AND SUBTITLE
Intelligent Crossing Assessment and Traffic Sharing System (i-CATSS)

5a. CONTRACT NUMBER
693JJ619C000009

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Yu Qian: 0000-0001-8543-2774
Yi Wang: 0000-0002-5750-3181
Dimitris Rizos: 0000-0001-5764-7911
Csilla Farkas: 0000-0002-1469-1522
Chin-tser Huang: 0000-0003-3983-972X

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of South Carolina
300 Main St.
Columbia, SC 29208

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Federal Railroad Administration
Office of Railroad Policy and Development
Office of Research, Development and Technology
Washington, DC 20590

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
DOT/FRA/ORD-22/15

12. DISTRIBUTION/AVAILABILITY STATEMENT
This document is available to the public through the FRA website.

13. SUPPLEMENTARY NOTES
COR: Francesco Bedini Jacobini

14. ABSTRACT
Railroad crossings may cause unexpected blockages and considerable traffic delays, especially in urban areas. However, there is no practical system available to share potential crossing blockage information to the public. An unexpected blockage threatens first responders while they respond to emergencies. This research project has developed an Intelligent Crossing Assessment and Traffic Sharing System (i-CATSS) that can detect and predict highway-rail blockages at grade crossings and provide first responders with real-time information of traffic conditions at crossings.

15. SUBJECT TERMS
Crossing safety, crossing monitoring, first responders, delay time, train signal, artificial intelligence, train control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		69	**19b. TELEPHONE NUMBER** *(Include area code)*

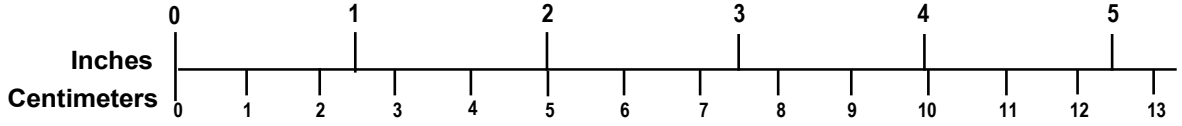
Standard Form 298 (Rev. 8/98)

Prescribed by ANSI Std. Z39.18

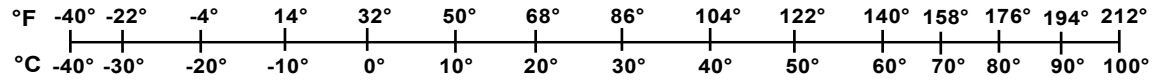
METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC	METRIC TO ENGLISH
LENGTH (APPROXIMATE)	**LENGTH (APPROXIMATE)**
1 inch (in) = 2.5 centimeters (cm)	1 millimeter (mm) = 0.04 inch (in)
1 foot (ft) = 30 centimeters (cm)	1 centimeter (cm) = 0.4 inch (in)
1 yard (yd) = 0.9 meter (m)	1 meter (m) = 3.3 feet (ft)
1 mile (mi) = 1.6 kilometers (km)	1 meter (m) = 1.1 yards (yd)
	1 kilometer (km) = 0.6 mile (mi)
AREA (APPROXIMATE)	**AREA (APPROXIMATE)**
1 square inch (sq in, in^2) = 6.5 square centimeters (cm^2)	1 square centimeter (cm^2) = 0.16 square inch (sq in, in^2)
1 square foot (sq ft, ft^2) = 0.09 square meter (m^2)	1 square meter (m^2) = 1.2 square yards (sq yd, yd^2)
1 square yard (sq yd, yd^2) = 0.8 square meter (m^2)	1 square kilometer (km^2) = 0.4 square mile (sq mi, mi^2)
1 square mile (sq mi, mi^2) = 2.6 square kilometers (km^2)	10,000 square meters (m^2) = 1 hectare (ha) = 2.5 acres
1 acre = 0.4 hectare (he) = 4,000 square meters (m^2)	
MASS - WEIGHT (APPROXIMATE)	**MASS - WEIGHT (APPROXIMATE)**
1 ounce (oz) = 28 grams (gm)	1 gram (gm) = 0.036 ounce (oz)
1 pound (lb) = 0.45 kilogram (kg)	1 kilogram (kg) = 2.2 pounds (lb)
1 short ton = 2,000 pounds (lb) = 0.9 tonne (t)	1 tonne (t) = 1,000 kilograms (kg) = 1.1 short tons
VOLUME (APPROXIMATE)	**VOLUME (APPROXIMATE)**
1 teaspoon (tsp) = 5 milliliters (ml)	1 milliliter (ml) = 0.03 fluid ounce (fl oz)
1 tablespoon (tbsp) = 15 milliliters (ml)	1 liter (l) = 2.1 pints (pt)
1 fluid ounce (fl oz) = 30 milliliters (ml)	1 liter (l) = 1.06 quarts (qt)
1 cup (c) = 0.24 liter (l)	1 liter (l) = 0.26 gallon (gal)
1 pint (pt) = 0.47 liter (l)	
1 quart (qt) = 0.96 liter (l)	
1 gallon (gal) = 3.8 liters (l)	
1 cubic foot (cu ft, ft^3) = 0.03 cubic meter (m^3)	1 cubic meter (m^3) = 36 cubic feet (cu ft, ft^3)
1 cubic yard (cu yd, yd^3) = 0.76 cubic meter (m^3)	1 cubic meter (m^3) = 1.3 cubic yards (cu yd, yd^3)
TEMPERATURE (EXACT)	**TEMPERATURE (EXACT)**
[(x-32)(5/9)] °F = y °C	[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Acknowledgments

Special thanks go out to Mayor Steve Benjamin, the City Council, and the first responders at the City of Columbia, South Carolina, for their tremendous support. Robert Anderson, David Brewer, John Stroke, and other members at the Public Works Department provided great help with traffic monitoring.

CSX Corporation also provided significant support and useful advice. Todd Echler and Joshua Marschke from Department of Track, Standards & Capital Planning, and Jason Schroeder and other experts from Department of Signaling and Control provided great help and useful information.

The opinions expressed in this report are solely those of the authors.

Contents

Illustrations

Executive Summary

From August 2019 through February 2021, researchers at the University of South Carolina developed an innovative Intelligent Crossing Assessment and Traffic Sharing System (i-CATSS) that can detect and predict highway-rail blockages at grade crossings and provide first responders with real-time information of traffic conditions at crossings. The system evaluates the total expected delay time due to both passing trains and vehicle congestion in front of the railroad crossing.

The research team developed a graphic user interface to display the estimated arrival time of the train and the estimated departure time for the monitored crossing, given the information shared from the partner railroad. Traffic status is consistently updated when new incoming information is received.

The team also developed an artificial intelligence (AI) model to detect the number of vehicles waiting in front of the blocked crossing. The system automatically starts whenever a train is detected within the area of interest by the surveillance camera. The correlations between the number of the waiting vehicles and the delay time are established based on the AI model. The model training and validation are performed using the surveillance videos recorded at the crossing of interest.

First responders from Columbia, SC, offered opinions on the impact of the unexpected railroad blockages through a survey. They also assisted in system development. The South Carolina Department of Transportation for the City of Columbia assisted in location identification and video collection. Industry partner CSX provided the train operation information for system development and improvement.

1. Introduction

1.1 Background

According to the Federal Railroad Administration (FRA), there were about 210,000 public and private grade crossings nationwide, and most of them were public-approachable (Ogden, B. D., & Cooper, C., 2019). Since the 1980s, highway-rail collisions at grade crossings have been decreasing continuously as a result of upgrading many unprotected crossings. However, the cost of upgrading existing systems, including signaling units and gate arms, can easily exceed $200,000 per crossing. Although there are significant improvements for railroad safety over the past decades, crashes at grade crossings are still one of the leading causes of railroad-related fatalities. According to FRA, in the U.S. from 2010 to 2014 there were approximately 2,100 collisions between trains and motor vehicles per year (Baron, W., & da Silva, M., 2019) (Brabb, D. C., Vithani, A., & Martin, K., 2017). More than 250 people were killed per year, which translates to an average of about 5 fatal accidents per week. These collisions disrupt both highway and rail operations and damage local economies and communities.

Today, trespass casualties represent roughly 70 percent of accidents on railroad rights-of-way (ROW) in North America. Ironically, more than 60 percent of collisions occur at crossings with automatic warning systems, and 34.7 percent occur at crossings that have flashing lights and gates (Federal Railroad Administration, 2019). There are several issues with the existing grade crossing warning system, including:

1) Flashing lights and gate arms only indicate an approaching train without quantitative information of estimated arrival time of the train. Out of all highway-rail collisions, 94 percent can be attributed to driver behavior or poor judgment (Federal Railroad Administration, 2019). On many occasions, poor judgment was mostly caused by the lack of quantitative situational awareness, especially when an approaching train was beyond sight of the driver or pedestrian. The availability of such information and corresponding alerts could potentially enhance situational assessment and rational decisions made by the drivers before entering the crossing or abandoning a vehicle promptly to avoid catastrophic consequences.
2) Current systems are limited by "one-way" communication—that is, they only offer warning signals to the vehicles and pedestrians while trains are not notified about any real-time information at the crossings. Onboard engineers can respond only when unusual activities at the crossing occur within their sight, which often is too late for taking effective countermeasures. Indeed, under many circumstances, collisions could be prevented if the early, real-time traffic information were exchanged between the train and vehicles and/or pedestrians through "two-way" communication.
3) The current system does not have self-diagnosis capabilities and relies on fixed inspection and maintenance schedules to ensure proper operation as in the video "Truck Breaks Crossing Gate During Malfunction." For instance, FRA requires railroads to perform monthly tests of automatic warning devices. Therefore, an "intelligent" system for automatic grade crossing surveillance, mutual and quantitative information sharing, and self-diagnosis for condition-based maintenance (CBM), is strongly needed.

Many studies (Eluru, N., Bagheri, M., Miranda-Moreno, L. F., & Fu, L., 2012) (Haleem, K., 2016) (McCollister, G., & Pflaum, C., 2007) (Ogden, B. D., Korve Engineering, a Division of

DMJM + Harris, 2007) (Yan, X., Richards, S., & Su, X., 2010) on highway-railroad grade crossing safety led by FRA, the U.S. Department of Transportation (USDOT), and/or public/private educational institutions have been conducted to understand and reduce collisions at the grade crossings. However, most of those studies focused on accident severity or frequency analysis.

Some researchers have investigated the impact of traffic congestions on highway safety (Marchesini, P., & Weijermars, W. A. M., 2010) (Cambridge Systematics, Inc., & Meyer, M. D., 2008) (Wang, C., 2010). However, at present, there is no systematic study to quantify traffic congestion in front of the grade crossing and assess its impact on railroad safety or public concerns.

To address such an urgent need, in 2018 the research team proposed the first-ever Intelligent Camera Aided Railway Emergency System (i-CARES), relying on image-based monitoring and surveillance, quantitative situational awareness assessment, and direct, two-way communication and information sharing. i-CARES features salient wireless communication, computer vision, Artificial Intelligence (AI) innovation, and in-situ preventive actions on an embedded and autonomous "cyber-physical systems/CPS" platform installed at grade crossings, and represents a holistic solution to grade crossing safety that has never been explored before for railroad engineering. Other innovations and benefits offered by i-CARES include automatic fault detection and notification for CBM and imagery evidence for trespassing violation (similar to the Electronic Police Reports Online [ePRO] system). This project will be focusing on developing the automatic target recognition algorithm and the AI to track vehicles to analyze vehicle behavior at railroad crossings.

1.2 Objective

Researchers set out to develop an affordable and field-deployable Intelligent Crossing Assessment and Traffic Sharing System (i-CATSS) that can detect and predict highway-rail blockages at grade crossings due to a slow-moving or stopped train, and provide real-time information of traffic conditions at crossing to facilitate route planning/reconfiguration for motorists, especially first responders.

1.3 Overall Approach

The system will integrate low-cost surveillance camera, real-time communication unit, and an AI model on a secured computing platform.

1.4 Scope

Research activities in this project consisted of three work packages.

Work Package 1 (WP1) established a communication channel between a railroad signaling and control unit and i-CATSS in order to obtain train operation information. Note that WP1 only established one-way communication – that is, reading the information from the Positive Train Control (PTC) system or other similar train signal systems without interference to normal train operations. The information obtained from WP1 was passed to WP3 to determine the possible delaying situation.

Work Package 2 (WP2) evaluated the traffic flow through the grade crossing by detecting moving vehicles. WP2 used a similar automatic target recognition (ATR) technique previously

used in other engineering disciplines (i.e., detection and recognition of small moving vehicle or aircraft using machine learning techniques).

Work Package 3 (WP3) developed a self-learning AI and situational awareness capability that extracted models for predicting total delay/waiting time based on the estimated time of arrival (ETA), passing time of the train (ETP), traffic intensity Q, PTC schedule, and historical values. The developed software can perform in-situ mobile computing for real-time sharing of traffic information.

1.4.1 Train Information Extraction and Estimated Train Arrival Time Calculation

Train information was extracted by the industry partner from its dispatching system and shared with the research team. To calculate the estimated train arrival time for a specific crossing, the current position and moving speed were needed. Because the PTC system contains multiple layers and more information than just the train real-time location and speed, part of the PTC information was decoded by the industry partner, and the train location and speed information was shared with the team with a refreshing time interval of approximately 1 minute. With the train location and moving speed, the estimated train arrival time to a monitored crossing could be calculated.

1.4.2 Street Traffic Detection and Vehicle Counting

In rural area, street traffic clearance time is not very important. However, for a populated urban area, it would take longer for the street traffic to go back to normal after a train has passed a crossing. For first responders, time is of the essence, and the traffic decongestion time may alter their route selection. Thus, a computer vision-based monitoring system was developed to detect and count vehicles waiting in the crossing.

1.4.3 Artificial Intelligence Model for Decongestion Time Estimation

Based on the recorded vehicle number and the time needed for street traffic to return to normal after a train passing, an AI model can be developed to correlate the detected vehicles and the estimated traffic decongestion time. The traffic decongestion time and the time for a train to pass a specific crossing would be the total expected delay time. Depending on the street traffic condition, train length, and train speed, the time component may be different from case to case.

1.5 Organization of the Report

Section 2 provides a literature review of both train control systems and the applications of AI in traffic monitoring. Section 3 describes the train information shared from the industry partner and how the estimated train arrival time and departure time were calculated. Section 4 gives the details of how the street traffic was monitored and the model that was developed to predict the traffic decongestion time. Section 5 summarizes the findings and offers suggestions for future studies. Appendix A provides the survey questionnaire and Appendix B contains the survey results.

2. Background and Literature Review

Decongestion time is a crucial component for intelligent crossing assessment and traffic sharing systems at a grade crossing. Predicting decongestion time facilitates evaluating the severity of traffic blockage and is the key to taking proactive measures that mitigate potential delays. There are roughly 210,000 public and private grade crossings in the U.S. (Ogden, B. D., & Cooper, C., 2019), and each train can easily take 5–10 minutes or more to pass. Thus, providing real-time information about traffic conditions at grade crossings to motorists, and especially first responders (i.e., firefighters, police, and ambulance) for route planning is indeed important (Gephardt, M., & Poe, M., 2018) (U.S. Department of Transportation, 2020a). In 2015, former FRA Administrator Sarah E. Feinberg acknowledged that the most serious consequence of grade crossing blockage was the delay caused to first responders in the case of emergency (Njus, E., 2019).

Unfortunately, due to a variety of reasons, railroads' operating information is typically not shared with the public. There are no navigation apps in the market to assess delays at a railroad grade crossing or to plan traffic rerouting. Consequently, severe traffic jams occur around grade crossings in highly populated areas (Arnott, R., De Palma, A., & Lindsey, R., 1991). Even worse, trains can move at an extremely slow speed or stop at the grade crossing, leading to an extended period of traffic blockage and adding large uncertainties to the delay time prediction. Train blockages can last for hours, rendering it difficult for motorists to make a rational decision between waiting and taking an alternative route; the scenario is far more critical when first responders are involved. The situation assessment at a blocked crossing essentially depends on the real-time information from two sources: the estimated time of the train operation and the traffic flow through the crossing. A robust and reliable intelligent transportation system should provide quantitatively accurate traffic condition assessment in a complex grade crossing environment that involves the train, vehicles, pedestrians, gates, road signs, and unforeseen circumstances. Furthermore, to avoid the unnecessary delay, it will also be beneficial to share the processed traffic information with the first responders for their decision making (Njus, E., 2019).

2.1 Positive Train Control System

As described in a report by Association of American Railroads (2021), PTC systems aims to automatically stop a train to prevent accidents due to human errors or poor judgements. PTC systems facilitate several key functions, including command, control, communications, and information sharing. With data interchanged among subsystems, PTC can accurately control the operation of a train. With PTC systems, certain types of accidents could be prevented, especially those due to human errors or lack of communication. Because of the great potential advantages to improve operation safety and efficiency, the Rail Safety Improvement Act of 2008 (RSIA) originally set December 31, 2015, as the deadline for PTC implementation, but extensions were granted later due to technical and practical reasons.

The are several key components of the PTC system, as shown in Figure 1 (Zhang, Z., Liu, X., & Holt, K., 2018)—the on-board computer, wayside device, communication channels, and back office. The on-board computer communicates with other components from the train to report information such as speed, train length, origin, and destination, while the wayside devices monitor track information at specific locations, such as the switch and signal information, and pass that information to the train and the back office. The back office is a data processing and

communication center which handles all the information from all the trains and other devices in the network and coordinate all the parties to work together. To facilitate reliable communications, a proper network is required to transmit and receive all the signals. The network could use different ways to transmit PTC information, such as radio (specified at 220 MHz), Wi-Fi, or cellular service.

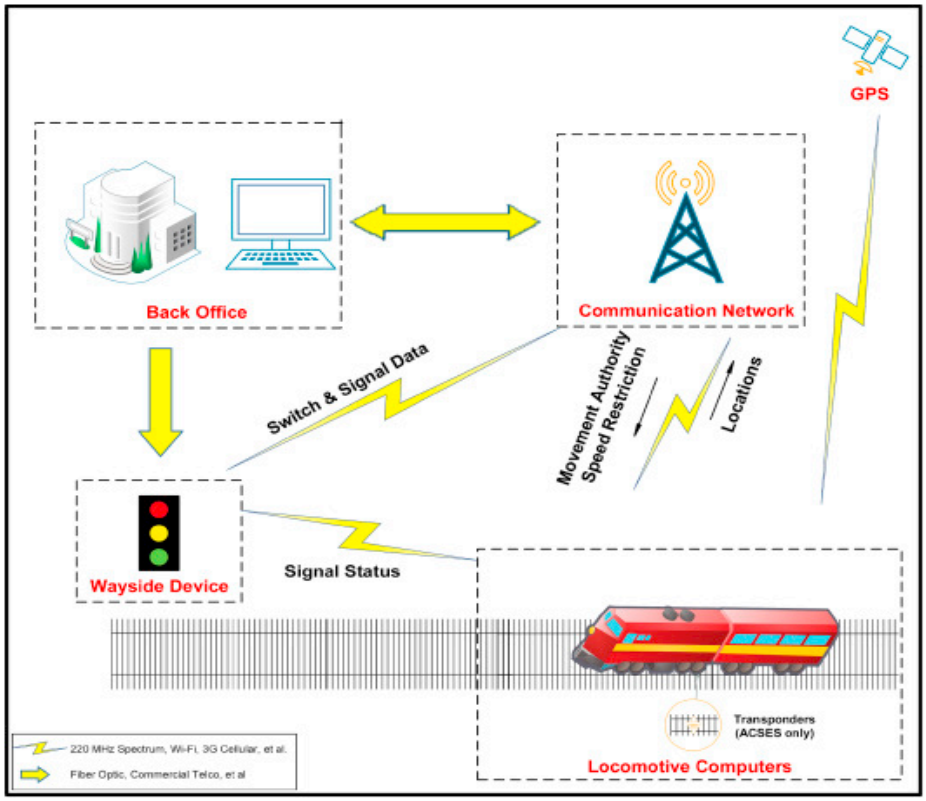


Figure 1. Schematic illustration of a general PTC system (Zhang, Z., Liu, X., & Holt, K., 2018)

2.1.1 Brief History of PTC Development in the U.S.

A system that can automatically handle train operation to avoid man-made mistakes has been planned for decades. Three decades ago, the National Transportation Safety Board described PTC as an important technology that would improve train operation safety, and stimulated its development and implementation in the years that followed. Earlier attempts included the Advanced Civil Speed Enforcement System and the Incremental Train Control System for passenger service by Amtrak and the onboard GPS system for freight service by CSX Transportation, Inc. (CSX).

After a collision accident between a Metrolink commuter train and a Union Pacific Railroad train in 2008, Congress passed the Rail Safety Improvement Act later that year (U.S. Congress, RSIA), which requires approximately 58,000 miles of railroad track equipped with PTC by the end of 2015. Although the specific deadline was extended multiple times due to technical and practical challenges, this act greatly accelerated PTC development and implementation.

2.1.2 Current PTC Implementation Status in the U.S.

Driven by congressional expectations and the railroads' motivations to improve the safety, PTC systems were installed for all required routes by the end of 2018 (Federal Railroad Administration, 2020). According to FRA data, PTC systems in 91 percent of the required routes in Class I railroads are already in service (see Figure 2 for the latest status). It is expected all the installed PTC systems in the required Class I railroad tracks will be in working condition to assist train operation by the end of 2020 (Federal Railroad Adminstration, 2020).

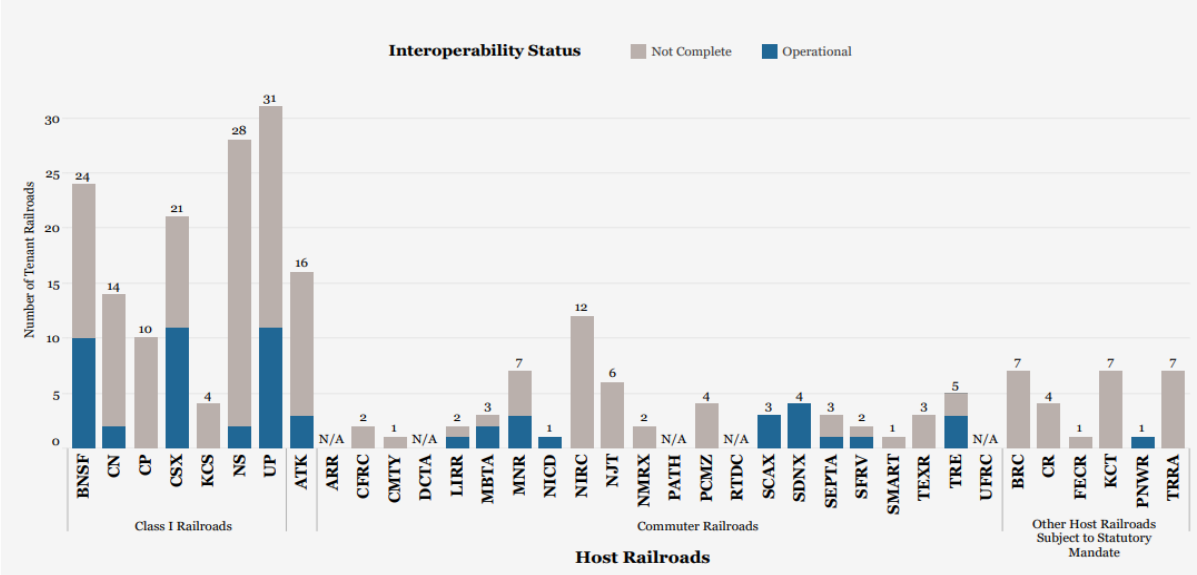


Figure 2. PTC Interoperability status – based on railroads' self-reported progress as of September 30, 2019 (Federal Railroad Administration, 2020)

2.1.3 Possible Improvement on PTC for Railroad Crossing

According to Barkouk et al. (2017), PTC systems were under development to better exchange information and improve efficiency. Currently, communications for all the installed or working PTC systems are limited within the railroad system. In other words, the information between the train, wayside devices, and the back office is kept within the loop and is not shared with other systems. Note the airline operations information, such as the flight status, and street traffic information, such as the congestion condition, is available to the public. Railroads are not integrated into the grand transportation systems even within current PTC systems. As Barkouk et al. (2017) pointed out, currently, there is no information exchange between street vehicles and PTC systems. If train operation information can be exchanged with street traffic information, it may benefit both the street traffic management and the train operations. For example, if the PTC system can obtain and share the information that a certain segment of track or a certain crossing is blocked by a vehicle, an accident may be prevented. Alternatively, if the traffic management system could obtain the crossing blockage time and duration due to an incoming train, the congestion could be reduced in front of the crossing if the traffic can be directed properly. Thus, to develop systems that can securely facilitate the information sharing between the railroads and other transportation modes would extend the function of the PTC data and potentially benefit both the railroads and the public. Possible extension may include:

- Information exchange directly between the trains and street vehicles
- Information exchange between the trains and street vehicles through PTC back office
- Information exchange between the trains and street vehicles through street traffic management office
- Information exchange between the PTC back office and the street traffic management office

Figure 3 illustrates possible ways of communication between PTC system and the street traffic proposed by Barkouk et al. (2017).

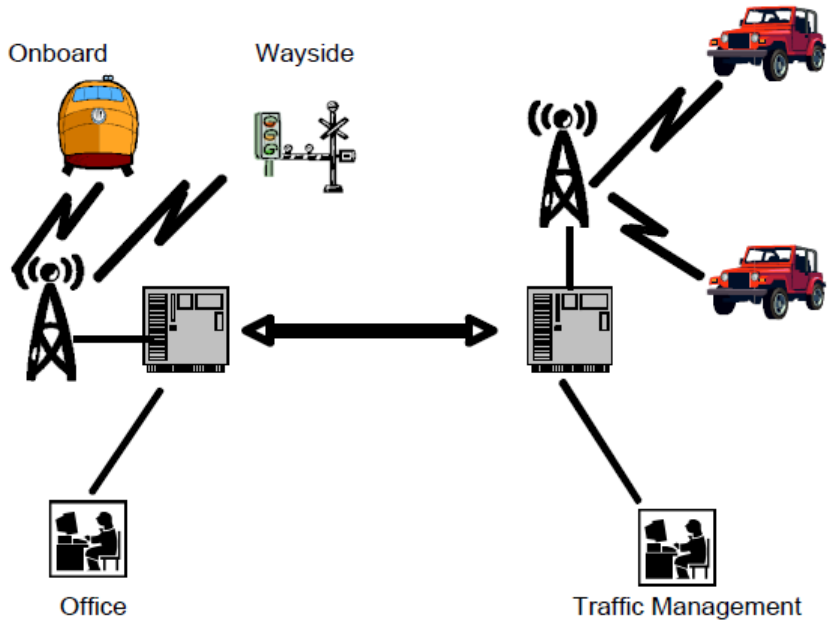


Figure 3. Example of PTC function extension (Barkouk, H., En-Naimi, E.M., & Mahboub, A., 2017)

2.2 Vehicle Detection and Counting

According to a report, in general, there are two categories of conventional practices for traffic counting: intrusive and non-intrusive methods (Leduc, G., 2008). Intrusive methods consist of a data recorder and a counting sensor placed on or embedded in the pavement. Most commonly used intrusive methods include but are not limit to pneumatic road tubes, piezoelectric sensors and magnetic loops, etc. Although these intrusive methods have been successfully employed for many years with their high accuracy nature, note that some limitations on urban applications still exist:

- These intrusive detectors can provide accurate data only when properly installed (e.g., traffic counting using existing video detection cameras). However, in practice, not only the installation process of sensors causes pavement damage but also the lane closure will have an impact on the detectors' position and data accuracy.
- The pavement maintenance will be influenced because of their intrusive nature, especially for quick maintenance during the peak hours.

- In a busy traffic section, retrieving data will be very inconvenient, as each lane is embedded with a detector.
- Extreme weather conditions like rain or snow can cause the pavement surface to become wet which leads to miscounting and misclassification. Meanwhile, the removal of snow and other maintenance process can break the sensors embedded in the pavement.
- All intrusive practices have potential safety concerns. Installing intrusive detectors in a heavy traffic area is very difficult, as there is little time window for staff to safely install sensors.
- The limited budget of transportation agencies allows for few locations to install traffic counting detectors, which means that there is not enough data for researchers and government agencies to fully understand the real-world traffic situation, not to mention future traffic planning and management needs.

Non-intrusive methods are traffic detection sensors that cause minimal disruption to normal traffic operations during installation, operation, and maintenance (Minge, E., Kotzenmacher, J., & Peterson, S., 2010). Most of the non-intrusive methods are based on remote observations and include but not limit to manual counts, passive and active infrared, passive magnetic, microwave radar, and video image detection, etc. As its definition and purpose for the traffic survey, non-intrusive methods can be more safely utilized than intrusive practices. Since non-intrusive methods do not need to cut the pavement to install sensors, it minimizes traffic disruption and minimizes cost. The advantages and disadvantages include:

- Manual counting is the most common practice. It does not need the complex equipment for the observer, and the test station is much more flexible. However, this practice would be time-consuming, and it cannot provide more information like vehicle occupation, vehicle classification even for a trained observer compared with auto counting.
- Passive and active infrared/laser can detect traffic volume, length, speed, and the number of axles by detecting the infrared/laser energy whose frequencies range from 1,011 to 1,014 Hz from the detection region. It can work well both day and night; however, it is limited by occlusions, road dirt on the lens, and extreme weather conditions.
- A passive magnetic infra-red laser can be used to detect the presence, speed, type, and number of vehicles by using the magnetic flux change measured by the corresponding changes in electric fields in the sensors. The installation is relatively complicated as it needs to be fixed on or under the top of roadbed, and it cannot detect stopped cars as there is no change on the magnetic flux.
- Radar can detect the presence, volume, classification, speed of vehicles by using the particular signal to calculate the time delay and the return signal. It can detect stationary vehicles but cannot detect the vehicles in the “dead zone” – like areas with obstructions or barriers.
- Video-based image processing analyzed the video image on a target area and compared the change of the same target area to collect more traffic information like presence, volume, speed, density, occupation, queue length, vehicle number, acceleration, classification by analyzing the video contents. However, it can be easily affected by the poor video quality caused by meteorological and environmental conditions like rain,

snow, poor light etc. Proper installation and calibration are important for video system to collect accurate traffic data.

2.2.1 Object Detection with Deep Learning Networks

Deep learning is an advanced subset of machine learning in AI and is largely based on the artificial neural networks (ANN), inspired by the human biological systems (Dabiri, S., 2019) (O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J., 2019). Deep learning networks have multiple hidden layers which are hidden in between the input layer and outer layer to perform computations on the weighted inputs and produce net input for activation functions. They integrate feature learning and model construction in one model by selecting different kernels or tuning the parameters via end-to-end optimization Wang et al., 2018). In DL's deep networks, the multiple hidden layers are used to map original input data to its new representations and then are transferred to higher layers to extract more efficient features. Finally, these abstracted representations are mapped into the outer layer to finish its classification and regression. In short, DL architecture learns features directly from the training data via the end-to-end process with minimum human inference (Hwang et al., 2019; Wang et al., 2018). Compared with conventional intrusive, non-intrusive detectors and especially the classical video-based image processing system, deep learning networks can operate in real time, work on end-to-end, collect much more traffic information, and achieve high accuracy with superior flexibility (Dabiri, S., 2019) (O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J., 2019) (Wang et al., 2018). Figure 4 shows the workflows of traditional computer vision and deep learning (Wang et al., 2018).

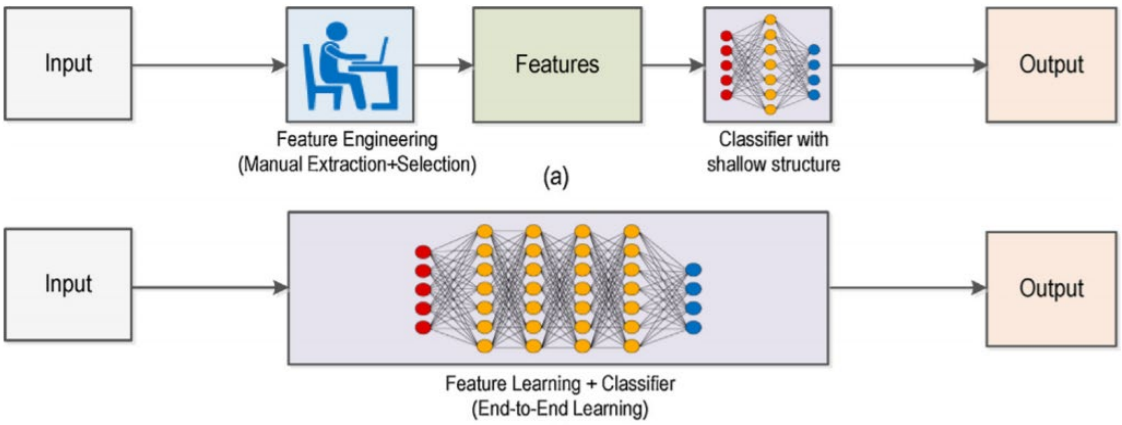


Figure 4. a) Workflow of traditional computer vision and b) workflow of end-to-end deep learning structure (Wang et al., 2018)

2.2.2 Real-Time Traffic Data Collection via Deep Learning Networks

Huang et al. (2014) first applied deep learning methods on transportation research. To solve the previous issues in transportation modeling they proposed a deep learning architecture, including a deep brief network (DBN) and a multitask task regression layer. Specifically, DBN is responsible for unsupervised feature learning, which can help researchers get rid of the hand-engineered feature extraction and selection. The multitask regression layer is responsible for supervised training. The network structure is shown in the Figure 5. In addition, the idea of

multiple-task learning (MTL) was absorbed into their deep architecture and proved MTL was useful in improving overall performance. Experimental tests showed that MTL has a 5 percent improvement compared with competing approaches.

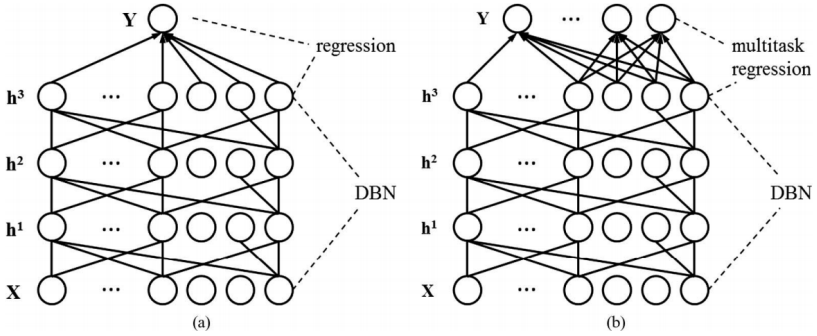


Figure 5. Deep architecture for traffic flow prediction: a) DBN structure and b) multitask regression layer (Huang, W., Song, G., Hong, H., & Xie, K., 2014)

Yi et al. (2017) used a deep learning neural-network based on the newest TensorFlow libraries to predict real-time traffic flow conditions. In their paper, congested and non-congested traffic conditions were distinguished by using the proposed deep learning architecture and logistic regression analysis. The experimental results demonstrated that the model's accuracy rate can achieve 99 percent; however, these results are based on each day's 1 percent traffic data as the limitations of memory storage. Note that even though their data processing has much room to improve, it still shows considerable potential for the application of the state-of-the-art TensorFlow library on transportation.

Lv et al. (2014) proposed a novel deep learning-based method which first applied autoencoders as building blocks on a deep architecture model to predict traffic flow conditions. In their approach, the SAE model, which can discover the nonlinear, spatial, and temporal correlations to represent latent traffic flow features, was used for the prediction work. Greedy layer-wise unsupervised learning was applied to pre-train the network and the fine-tuning process was employed to update the model's parameters for improving prediction performance. The experimental results showed their method was better than competing ones such as SVM, BP NN, etc. The architecture can be seen in the Figure 6 (Lv, Y., Duan, Y., & Kang, W., 2014).

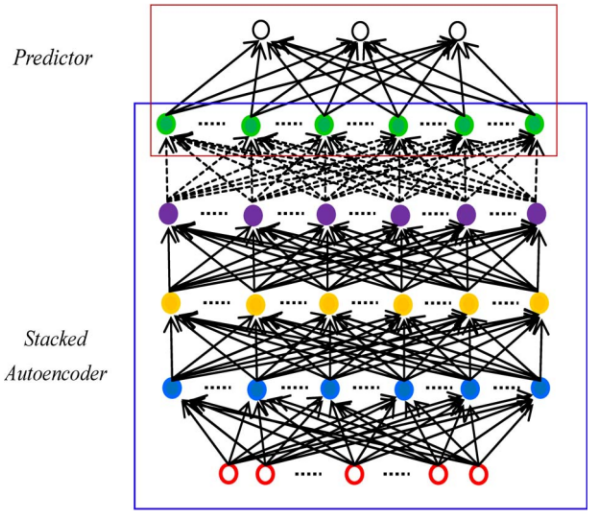


Figure 6. Deep learning architecture for traffic flow prediction (Lv, Y., Duan, Y., & Kang, W., 2014)

Polson et al. (2017) showed that deep learning can contribute precise, short-term traffic flow predictions in two cases with their developed model, which combined a linear model for regularization and a sequence of tanh layers. To capture the nonlinear spatio-temporal features in traffic flow, they carried the first layer to identify spatio-temporal relations, and other layers were used to model nonlinear relations. According to their tests, deep learning showed a great improvement over linear models. The recurrent neural network (RNN) and a long short-term memory (LSTM) network can work better on time-series data. They also mentioned one drawback of their model was its low explanatory power.

2.2.3 Possible Improvement on Vehicle Counting for Railroad Crossing

Presently, the research and development of intelligent transportation assessment systems is mainly focused on highway intersections rather than grade crossings (i.e., highway-railroad crossing), and the applications primarily concentrate on traffic flow prediction (U.S. Department of Transportation, 2020b), autonomous driving (Hoel, C.-J., Driggs-Campbell, K., Wolff, K., Laine, L., & Kochenderfer, M. J., 2019), traffic signal control (Wan, J., Yuan, Y., & Wang, Q., 2017), and others. Previous research efforts on theories, algorithms, and implementations of traffic analysis have been developed using mathematical models or DNN models. In the past few years, a prevalent trend in intelligent transportation system development has seen a shift from conventional methods (Hensher, D. A., & Mannering, F. L., 1994) (Paselk, T. A., & Mannering, F. L., 1994) to deep learning (Gruden, C., Otković, I. I., & Šraml, M., 2020) (Khan, A. M., 2010) approaches. The traditional congestion analysis methods build a dynamical model of traffic congestion based on the equation of motion of each vehicle (Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y., 1995) (Komatsu, T. S., & Sasa, S.-i, 1995). The simulated model can observe the evolution of traffic congestion with the elapse of time. However, with this approach, the model performance might deteriorate with an increase in the number of vehicles. The deep learning-based congestion analysis (Wan, J., Yuan, Y., & Wang, Q., 2017), (Ma, X., Yu, H., Wang, Y., & Wang, Y., 2015) and related studies, such as traffic congestion detection and prediction (Bauza, B., & Gozalvez, J., 2013) (Kurniawan, J., Syahra, S. G., & Dewa, C. K.,

2018) (Ma, X., Yu, H., Wang, Y., & Wang, Y., 2015) (Wang, J., Gu, Q., Wu, J., Liu, G., Xiong, Z., 2016) (Wang, Q., Wan, J., & Yuan, Y., 2018) and traffic flow prediction (Zhang, S., Yao, Y., Hu, J., Zhao, Y., Li, S., & Hu, J., 2019), (Polson, N. G., & Sokolov, V. O., 2017) have also been developed to improve the intelligent transportation system. However, to the authors' knowledge, studies on analyzing and predicting traffic decongestion time is limited. Huang et al., 2014 proposes to predict decongestion time using congestion patterns. However, this method might not be suitable for complex crossing scenes due to limited congestion patterns.

It is often challenging to extract key quantitative information from a complex video scene for traffic assessment. Counting vehicles is one way to estimate traffic flow. To provide accurate and robust traffic information, deep learning-based vehicle counting is utilized in the proposed framework. A rich body of literature in crowd counting has been published recently, and vehicle crowd counting methods can be classified into three categories: detection-based counting, direct count regression, and density map estimation. Individual detection and tracking-based counting algorithms in general are not accurate for dense crowds, and hence, are not well-suited for addressing congested scenes (Lee, K., Hong, B., Jeong, D., & Lee, J., 2014) (Li., Y., Zhang, X., & Chen, D., 2018). Besides, it also requires bounding annotation, which is a laborious and ambiguous process due to the heavily overlapped vehicles. To avoid explicit detection of individuals, regression methods (Chan, A. B., Liang, Z.-S. J., & Vasconcelos, N., 2008) (Ge, W., & Collins, R. T., 2009) are proposed to estimate the number of people directly from low-level features, such as texture, color, and gradient. But their results are less interpretable. Density map-based methods are currently the most popular approach to crowd counting since its performance can be dramatically improved by utilizing spatial information. Density maps are typically generated by convolving the dot maps with Gaussian kernels, in which each dot represents an object (or a person) in an image (Chan, A. B., & Vasconcelos, N., 2009). Different network architectures are designed to handle various challenges, such as the scale changes (Babu Sam, D., Surya, S., & Babu, R. V., 2017) (Lempitsky, V., & Zisserman, A., 2010) (Li., Y., Zhang, X., & Chen, D., 2018) (Babu Sam, D., Sajjan, N. N., Babu, R. V., & Srinivasan, M., 2018), the poor quality of density maps (Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., & Shao, L., 2019) (Walach, E., & Wolf, L., 2016), and limited contextual information (Liu, C., Weng, X., & Mu, Y., 2019) (Ranjan, V., Le, H., & Hoai, M., 2018) (Xiong, F., Shi, X., & Yeung, D.-Y., 2017).

Almost all research in intelligent transportation systems are conducted for highway intersections. There are scarce efforts on analyzing the temporal behavior of the congestion events caused by the passing trains at the grade crossing despite their importance in real world. One of the major challenges of studying this problem is the lack of traffic congestion data at grade crossings. For example, recent advancements in intelligent transportation systems are driven by deep learning (Jin, J., & Ma, X., 2019) (Li, Y., 2017) (Lin, Y., Dai, X., Li, L., & Wang, F.-Y., 2018) (Veres, M., & Moussa, M., 2019) (Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J., 2017). Deep learning-based methods are data-driven and entail a large number of instances (and data) to achieve a relevant performance through end-to-end training (Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., & Bayen, A. M., 2017). This requirement is important because it prohibits it from being applied to congestion behavior analysis, decongestion time prediction, and traffic condition understanding due to the lack of an extensive dataset consisting of different grade crossing scenes and instances for each scene. Another challenge related to this problem is that railroad crossing environments vary. Take the scene in Figure 7 below as an example—the vehicles can choose to wait until the traffic congestion clears or take an alternative route (turn

left), as shown by red arrows in Figure 7. The complex behaviors of each vehicle make the traffic behavior and decongestion time even harder to model. Under these circumstances, there appears to be a need to propose a comprehensive system for traffic behavior understanding and decongestion time prediction at the grade crossing.

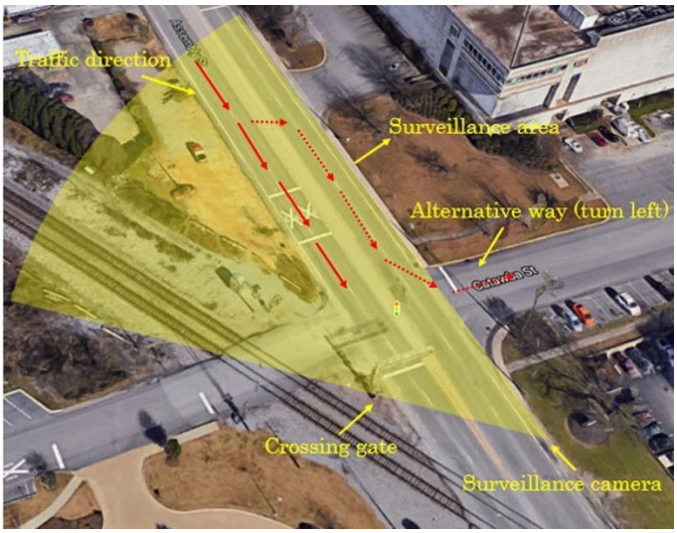


Figure 7. A railroad grade crossing scene at the intersection between Catawba Street and the Assembly Street in Columbia, SC

3. Train Information and Estimated Train Arrival Time Calculation

The ETA of the approaching train and the estimated total time (ETT) of the whole blockage counted from the current time are important for situational awareness assessment and need to be shared with the first responders, i.e.,

$$ETT = ETA + ETP + ETC \quad \textbf{and} \quad ETC = F\left(Q, ETP\right) \tag{1}$$

where ETP is the estimated time of passing of the train, and can be readily computed using the length and the current speed of the train. ETC is the estimated time of clearance (or decongestion). The most critical finding from Eq. (1) is that if the notified drivers cannot cross the track line before ETA, the total time they have to wait would be ETT regardless of their current location. The result, F in Eq. (1) that will be used to predict ETC is nonlinear and will be determined through a training process using the self-learning AI engine in an autonomous manner. First, ETA and ETP will be evaluated using information from the information shared by the railroads, and then the computer-vision-based traffic monitoring system will compute Q and ETC by analyzing collected videos at the grade crossing without human intervention. The mapping relationship F will be established using the machine learning techniques, such as ANN and support vector regression, which will construct a quantitative mapping relationship between the estimated time of clearance (ETC) and the traffic intensity Q and the estimate time of passing (ETP) of the train.

3.1 Identify the Potential Crossing

To develop the proposed system, a railroad crossing needs to be identified so the railroad can share their train information with the research team. Due to many railroad crossings within the City of Columbia, a survey was distributed to the first responders as the "Railroad Grade Crossing Congestion Condition Assessment." This survey helped the research team to better understand the crossing delay issue to the first responders and identify the most troublesome crossing in town. A complete survey form and the results are attached in Appendix A and Appendix B. Based on the number of responses from the survey. three crossings: CSX 634647A, 634632K and 634630W, located on Assembly Street, were identified as the potential crossings to monitor.

3.2 Information Shared by the Railroad

Once the grade crossings had been identified, CSX provided the relevant train information. As mentioned earlier, instead of directly communicating with the PTC system, which is still under development, CSX extracted the information needed for this study—train location, operation speed, and train length. The information was passed directly to the research team, and sample data is shown in Figure 8, while sample train length data is shown in Figure 9. *Please note all the train information shown in this report has been modified to not reflect the actual information per the request from CSX due to security reasons.* The system was developed based on the true data shared by CSX.

	A	B	C	D	E	F	G
1	@timestamp	train_uid.train6	loco_location.loco_id	loco_location.speed	loco_location.point	loco_location.point_altitude	loco_location.heading
2	Nov 10, 2020 @ 17:05:00.000	F77910	csxt3206	9	{ "lon": -81.04417419433594, "lat": 33.97754669189453}	173	236
3	Nov 10, 2020 @ 17:04:35.000	F77910	csxt3206	10	{ "lon": -81.04316711425781, "lat": 33.978275299072266}	178	218
4	Nov 10, 2020 @ 17:04:10.000	F77910	csxt3206	12	{ "lon": -81.0425796508789, "lat": 33.97919464111328}	187	198
5	Nov 10, 2020 @ 17:03:50.000	F77910	csxt3206	13	{ "lon": -81.04219055175781, "lat": 33.98017120361328}	189	202
6	Nov 10, 2020 @ 17:03:25.000	F77910	csxt3206	13	{ "lon": -81.04158020019531, "lat": 33.981197357177734}	186	207
7	Nov 10, 2020 @ 17:03:05.000	F77910	csxt3206	13	{ "lon": -81.04097747802734, "lat": 33.982173919677734}	192	206
8	Nov 10, 2020 @ 17:02:45.000	F77910	csxt3206	12	{ "lon": -81.04037475585938, "lat": 33.98316955566406}	199	206
9	Nov 10, 2020 @ 17:02:25.000	F77910	csxt3206	12	{ "lon": -81.03978729248047, "lat": 33.98413848876953}	201	204
10	Nov 10, 2020 @ 17:02:05.000	F77910	csxt3206	13	{ "lon": -81.03921508789062, "lat": 33.98515319824219}	199	207
11	Nov 10, 2020 @ 17:01:55.000	F77910	csxt3206	13	{ "lon": -81.03910804728952, "lat": 33.98530766493499}	0.055	0
12	Nov 10, 2020 @ 17:01:45.000	F77910	csxt3206	13	{ "lon": -81.03865051269531, "lat": 33.98613739013672}	192	207
13	Nov 10, 2020 @ 17:01:25.000	F77910	csxt3206	13	{ "lon": -81.0378646850586, "lat": 33.98706817626953}	195	219
14	Nov 10, 2020 @ 17:01:05.000	F77910	csxt3206	13	{ "lon": -81.0372085571289, "lat": 33.988033294677734}	200	191
15	Nov 10, 2020 @ 17:00:40.000	F77910	csxt3206	13	{ "lon": -81.03736877441406, "lat": 33.98906326293945}	200	158
16	Nov 10, 2020 @ 17:00:20.000	F77910	csxt3206	12	{ "lon": -81.03788757324219, "lat": 33.99013137817383}	198	157
17	Nov 10, 2020 @ 17:00:14.000	F77910	csxt3206	12	{ "lon": -81.03787984479436, "lat": 33.990129396267875}	0.056	0
18	Nov 10, 2020 @ 16:59:55.000	F77910	csxt3206	12	{ "lon": -81.03838348388672, "lat": 33.99118041921875}	201	159
19	Nov 10, 2020 @ 16:17:45.000	F77210	csxt411	10	{ "lon": -81.04336547851562, "lat": 33.97806930541992}	184	224
20	Nov 10, 2020 @ 16:17:20.000	F77210	csxt411	11	{ "lon": -81.04264068603516, "lat": 33.97898864746094}	186	202
21	Nov 10, 2020 @ 16:17:00.000	F77210	csxt411	12	{ "lon": -81.042236328125, "lat": 33.98000717163086}	184	199
22	Nov 10, 2020 @ 16:16:40.000	F77210	csxt411	14	{ "lon": -81.04168701171875, "lat": 33.980979919433594}	184	207
23	Nov 10, 2020 @ 16:16:20.000	F77210	csxt411	14	{ "lon": -81.04112243652344, "lat": 33.981903076171875}	188	207
24	Nov 10, 2020 @ 16:16:00.000	F77210	csxt411	14	{ "lon": -81.04048919677734, "lat": 33.982933044433594}	195	206
25	Nov 10, 2020 @ 16:15:40.000	F77210	csxt411	13	{ "lon": -81.0398941040039, "lat": 33.983917236328125}	197	205
26	Nov 10, 2020 @ 16:15:21.000	F77210	csxt411	13	{ "lon": -81.0396213396611, "lat": 33.98442771940601}	0.055	0
27	Nov 10, 2020 @ 16:15:20.000	F77210	csxt411	13	{ "lon": -81.03932189941406, "lat": 33.98494338989258}	199	206
28	Nov 10, 2020 @ 16:15:03.000	F77210	csxt411	13	{ "lon": -81.03910952664114, "lat": 33.98530513595111}	0.055	0
29	Nov 10, 2020 @ 16:15:00.000	F77210	csxt411	13	{ "lon": -81.0387191772461, "lat": 33.98591613769531}	196	207

Figure 8. Sample train location, speed, and GPS information

	A	B	C	D	E	F	G	H
1	@timestamp	train_id	loco_id	train_length				
2	Nov 10, 2020 @ 17:21:12.000	M71310	CSXT 5203	148				
3	Nov 10, 2020 @ 17:20:57.000	Q35310	CSXT 3209	2,820				
4	Nov 10, 2020 @ 17:20:13.000	D79110	CSXT 2056	1,461				
5	Nov 10, 2020 @ 17:19:53.000	Q22009	CSXT 3150	3,908				
6	Nov 10, 2020 @ 17:18:09.000	Q51510	CSXT 833	11,784				
7	Nov 10, 2020 @ 17:17:43.000	Q22009	CSXT 3150	3,908				
8	Nov 10, 2020 @ 17:17:18.000	Q54210	CSXT 748	7,990				
9	Nov 10, 2020 @ 17:17:18.000	Q68710	CSXT 3184	2,240				
10	Nov 10, 2020 @ 17:17:18.000	Q54210	CSXT 748	7,990				
11	Nov 10, 2020 @ 17:17:18.000	Q50010	CSXT 5501	7,934				
12	Nov 10, 2020 @ 17:17:18.000	Q50010	CSXT 5501	7,934				
13	Nov 10, 2020 @ 17:17:17.000	D76510	CSXT 3056	3,109				
14	Nov 10, 2020 @ 17:17:17.000	Q56710	CSXT 5442	8,138				
15	Nov 10, 2020 @ 17:17:17.000	Y22210	CSXT 6088	408				
16	Nov 10, 2020 @ 17:17:17.000	D76510	CSXT 3056	3,109				
17	Nov 10, 2020 @ 17:17:17.000	Y22210	CSXT 6088	408				
18	Nov 10, 2020 @ 17:17:16.000	Q69209	CSXT 3026	8,667				
19	Nov 10, 2020 @ 17:17:16.000	Q69209	CSXT 3026	8,667				
20	Nov 10, 2020 @ 17:17:08.000	G63210	CSXT 7205	5,502				
21	Nov 10, 2020 @ 17:14:50.000	E80810	CSXT 5332	7,038				
22	Nov 10, 2020 @ 17:14:37.000	M71509	CSXT 976	2,688				
23	Nov 10, 2020 @ 17:13:59.000	Q64110	CSXT 920	11,485				
24	Nov 10, 2020 @ 17:13:38.000	Q21410	CSXT 859	1,386				
25	Nov 10, 2020 @ 17:11:34.000	Q56709	CSXT 312	2,987				
26	Nov 10, 2020 @ 17:11:34.000	Q56709	CSXT 312	2,987				
27	Nov 10, 2020 @ 17:09:56.000	Q64810	CSXT 3265	6,032				
28	Nov 10, 2020 @ 17:09:32.000	Q13610	CSXT 3213	11,990				
29	Nov 10, 2020 @ 17:09:32.000	U20608	CSXT 7225	5,172				
30	Nov 10, 2020 @ 17:09:31.000	Q26410	CSXT 706	12,651				
31	Nov 10, 2020 @ 17:09:31.000	Q49109	CSXT 5461	2,583				
32	Nov 10, 2020 @ 17:09:31.000	Q15809	CSXT 7201	2,507				
33	Nov 10, 2020 @ 17:09:28.000	Q62110	CSXT 466	4,608				

Figure 9. Sample train length information

It was relatively simple to calculate the estimated arrival time of any train once the GPS location, the train speed, and the train length were known. Note the train information was dynamically updated; thus, the calculated estimated arrival time and departure time were also updated based on the newest information received.

The estimated arrival time can be calculated as:

$$\text{Estimated arrival time} = \frac{Distance\ beteen\ train\ and\ crossing}{train\ speed} \quad (2)$$

Once a train has reached a certain crossing, that crossing is blocked until the entire train passes that crossing. The time the entire train passes a crossing is defined as the estimated departure

time. To estimate departure time by train, one needs both the train length and speed and calculated the time with the equation below:

$$\text{Estimated passing crossing time by train} = \frac{locomotive\ length}{train\ speed} \quad (3)$$

An example of calculated train arrival time of the selected crossings are given in Figure 10.

	loco_id	date	time	speed	Estimated arrival time at Rosewood Dr	Estimated arrival time at S Assembly St	Estimated arrival time at Huger St
1	csxt113	Jul 14, 2020	12:36:07	9	2020-07-14 12:43:25	2020-07-14 12:40:50	2020-07-14 12:36:46
2	csxt113	Jul 14, 2020	12:36:32	9	2020-07-14 12:43:50	2020-07-14 12:41:15	2020-07-14 12:37:11
3	csxt113	Jul 14, 2020	12:36:57	9	2020-07-14 12:44:15	2020-07-14 12:41:40	2020-07-14 12:37:36
4	csxt113	Jul 14, 2020	12:37:27	9	2020-07-14 12:44:45	2020-07-14 12:42:10	2020-07-14 12:38:06
5	csxt113	Jul 14, 2020	12:37:57	10	2020-07-14 12:44:31	2020-07-14 12:42:12	2020-07-14 12:38:32
6	csxt113	Jul 14, 2020	12:38:22	9	2020-07-14 12:45:40	2020-07-14 12:43:05	2020-07-14 12:39:01
7	csxt113	Jul 14, 2020	12:38:52	8	2020-07-14 12:47:04	2020-07-14 12:44:11	2020-07-14 12:39:36
8	csxt113	Jul 14, 2020	12:39:22	9	2020-07-14 12:46:40	2020-07-14 12:44:05	2020-07-14 12:40:01
9	csxt113	Jul 14, 2020	12:39:52	10	2020-07-14 12:46:26	2020-07-14 12:44:07	2020-07-14 12:40:27
10	csxt113	Jul 14, 2020	12:40:22	9	2020-07-14 12:47:40	2020-07-14 12:45:05	2020-07-14 12:41:01
11	csxt113	Jul 14, 2020	12:40:47	9	2020-07-14 12:48:05	2020-07-14 12:45:30	2020-07-14 12:41:26
12	csxt113	Jul 14, 2020	12:41:17	9	2020-07-14 12:48:35	2020-07-14 12:46:00	2020-07-14 12:41:56
13	csxt113	Jul 14, 2020	12:41:42	10	2020-07-14 12:48:16	2020-07-14 12:45:57	2020-07-14 12:42:17
14	csxt113	Jul 14, 2020	12:42:12	9	2020-07-14 12:49:30	2020-07-14 12:46:55	2020-07-14 12:42:51
15	csxt113	Jul 20, 2020	06:44:58	17	2020-07-20 06:48:50	2020-07-20 06:47:28	2020-07-20 06:45:19
16	csxt113	Jul 20, 2020	06:45:18	19	2020-07-20 06:48:45	2020-07-20 06:47:32	2020-07-20 06:45:36
17	csxt113	Jul 20, 2020	06:45:28	18	2020-07-20 06:49:07	2020-07-20 06:47:50	2020-07-20 06:45:47
18	csxt113	Jul 20, 2020	06:45:43	18	2020-07-20 06:49:22	2020-07-20 06:48:05	2020-07-20 06:46:02
19	csxt113	Jul 20, 2020	06:45:59	16	2020-07-20 06:50:05	2020-07-20 06:48:38	2020-07-20 06:46:21
20	csxt113	Jul 20, 2020	06:46:19	13	2020-07-20 06:51:22	2020-07-20 06:49:35	2020-07-20 06:46:46
21	csxt113	Jul 20, 2020	06:46:39	14	2020-07-20 06:51:20	2020-07-20 06:49:41	2020-07-20 06:47:04
22	csxt113	Jul 20, 2020	06:46:59	16	2020-07-20 06:51:05	2020-07-20 06:49:38	2020-07-20 06:47:21
23	csxt113	Jul 20, 2020	06:47:14	15	2020-07-20 06:51:37	2020-07-20 06:50:04	2020-07-20 06:47:37
24	csxt113	Jul 20, 2020	06:47:34	14	2020-07-20 06:52:15	2020-07-20 06:50:36	2020-07-20 06:47:59
25	csxt113	Jul 20, 2020	06:47:54	11	2020-07-20 06:53:52	2020-07-20 06:51:46	2020-07-20 06:48:26
26	csxt113	Jul 20, 2020	06:48:14	10	2020-07-20 06:54:48	2020-07-20 06:52:29	2020-07-20 06:48:49
27	csxt113	Jul 20, 2020	06:48:44	12	2020-07-20 06:54:12	2020-07-20 06:52:16	2020-07-20 06:49:13
28	csxt113	Jul 20, 2020	06:49:09	3	2020-07-20 07:11:02	2020-07-20 07:03:19	2020-07-20 06:51:05

Figure 10. Example of the calculated estimated train arrival time

3.3 Graphic User Interface Development

Although the calculated train arrival time is a valuable reference for a dispatching center, it is not very intuitive. Especially when there is an emergency call, more information may overwhelm the dispatcher to make the best decision. Thus, a graphic user interface (GUI) was developed for this project.

Based on GoogleMapPlotter package, new features have been implemented to display train and the three selected crossing locations on Google Maps. A pop-up information window on Google Maps with train estimated arrival and departure times for each crossing will be shown once a new train enters the area of interest, as shown in Figure 11. The green label denotes train location at a certain timestamp and the red labels denote the three crossings. The fist value in the information box is estimated arrival time and the second value is estimated departure time. For example, (09:26:12, 09:31:25) means train CSXT934 is expected to arrive at the crossing at 09:26:12 and depart at 09:31:25. The GUI may help first responders better understand the estimated delay and make a better decision accordingly.

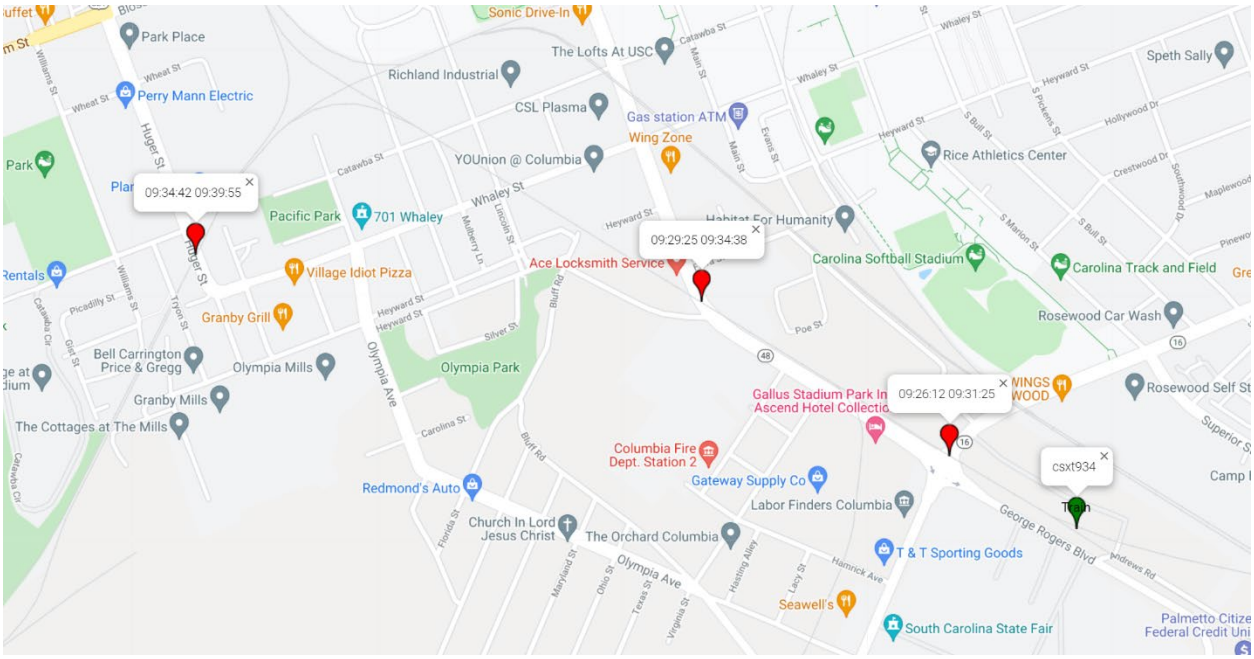


Figure 11. Example GUI pop-up window to show the crossing and train locations

3.4 Train Trajectory and Estimated Arrival Time for Other Crossings

For this project, only three crossings were selected for developing the system. However, it is possible to extend the information to construct the train trajectory and calculate the estimated arrival time for other crossings or any locations for future transportation planning.

To make the best use of the train information shared from the industry partner, the GoogleMapPlotter Python package was further customized by creating another Python drawing script. This script can read the updated train longitude and latitude information from the file shared from CSX and display train location on Google Maps under each timestamp (each small red dot is the train location when last updated). By showing all the locations of a specific train, it is easy to see the train trajectory and predict the direction the train. Thus, future crossing blockages along the route can be estimated. Figure 12 and Figure 13 give two examples of the reconstructed train route. Again, the information could be used for proactive traffic planning ahead of a crossing blockage instead of rerouting the highway during the blockages.

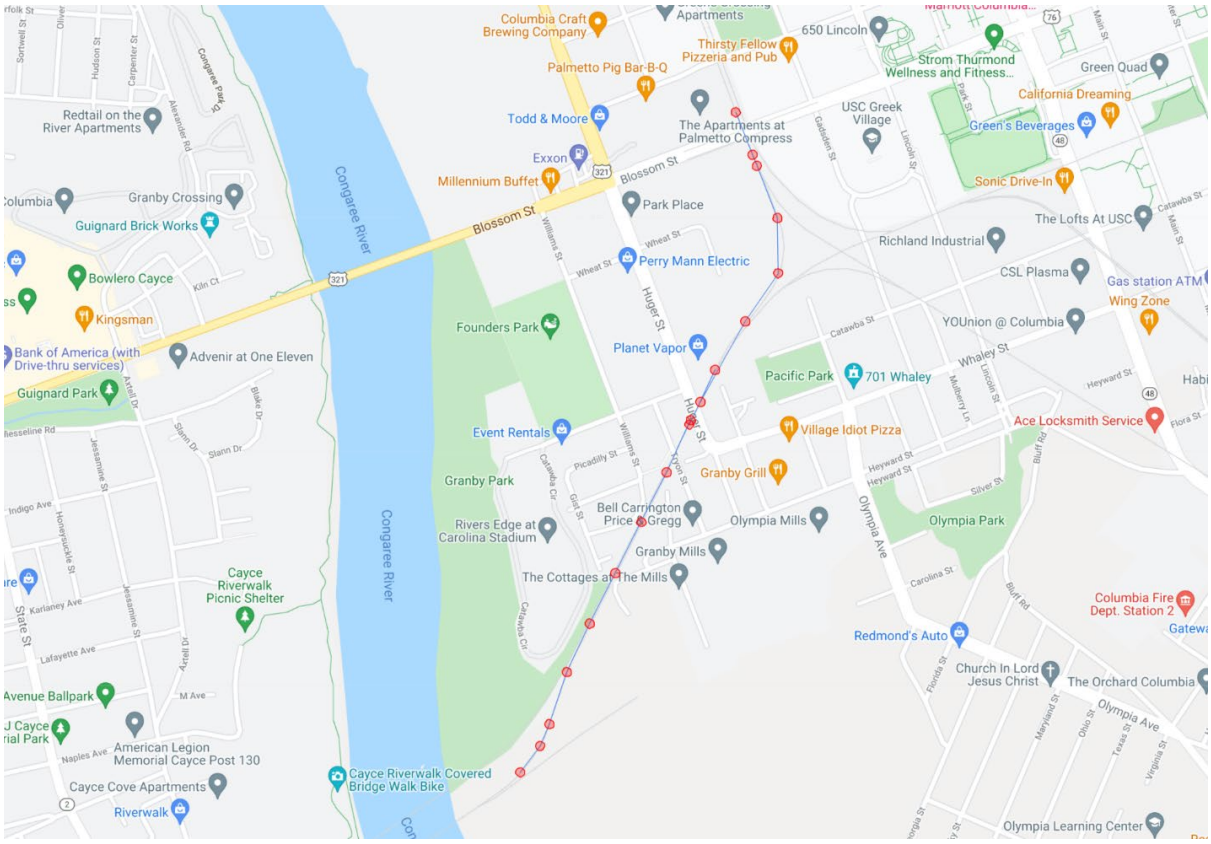


Figure 12. Example train movement route on November 4, 2020

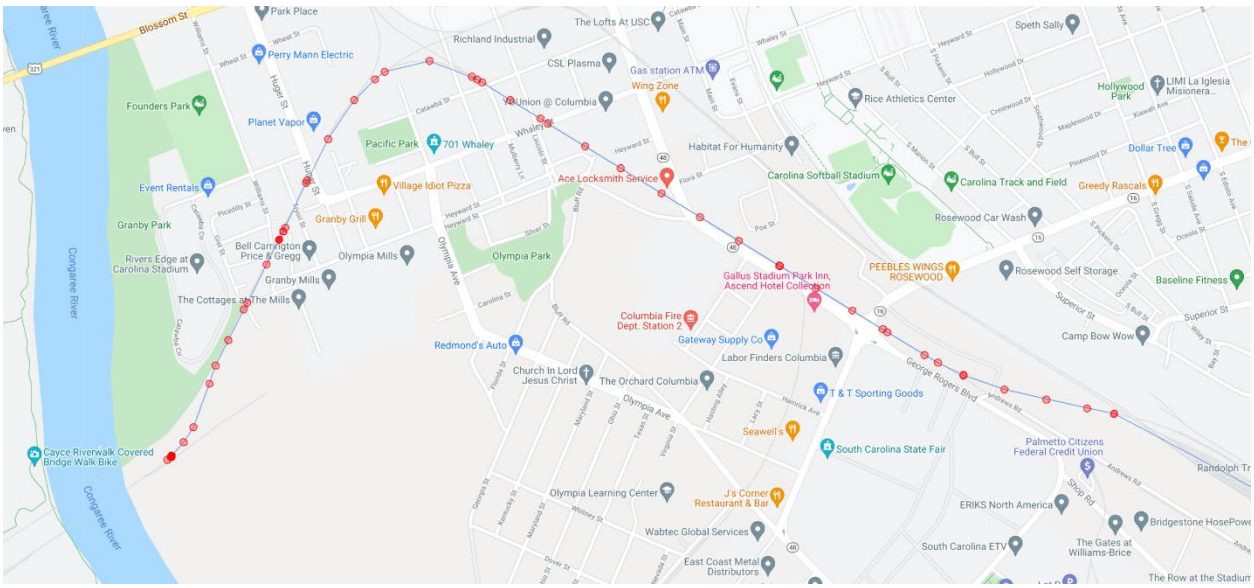


Figure 13. Example train movement route on November 4, 2020

4. Traffic Monitoring and AI-based Decongest Time Calculation

4.1 Traffic Monitoring Site and Video Collection

The test grade crossing in this research is located at the intersection of Catawba Street and Assembly Street in Columbia, SC, near the University of South Carolina campus (Figure 14). Traffic video data was collected using the COUNTcam2 traffic recorder provided by the City of Columbia, and the surveillance camera was installed on a power pole. A total of 96 hours of video records were collected in two sets, from 10:44 a.m., November 19, 2019, to 10:44 a.m. to November 21, 2019 (dataset 1), 12:03 a.m., December 3, 2019, to 12:03 a.m. December 5, 2019 (dataset 2). The camera's battery allows it to continuously monitor the traffic for up to 50 hours. Note that incoming trains do not follow any specific schedule. Therefore, the time of a train approaching the selected crossing in this study was random, which is typical for most of the crossings. In the future, a telecommunication unit facilitating communication with the onboard PTC system will be integrated with the camera to automatically trigger the video recording when a train is approaching the crossing.

Figure 14. Site of grade crossing at Columbia, SC (map data: 2019 Google maps)

Figure 15(a)(b) show two examples of collected data under the normal and congested traffic conditions, respectively. In the former, the vehicles cross the grade crossing continuously with a relatively constant spacing; while in congested traffic, vehicles were stopped in front of the crossing gate due to a passing train. Vehicles could either wait until the train completely passed or take an alternative route by turning left (as shown by the black sedan at the right of the figure). Correspondingly, the number of vehicles would be different under the normal and congested conditions.

(a) (b)

Figure 15. Examples of collected data during normal and congestion traffic condition. (a) Normal traffic condition (NTC); and (b) congested traffic condition when a train was passing

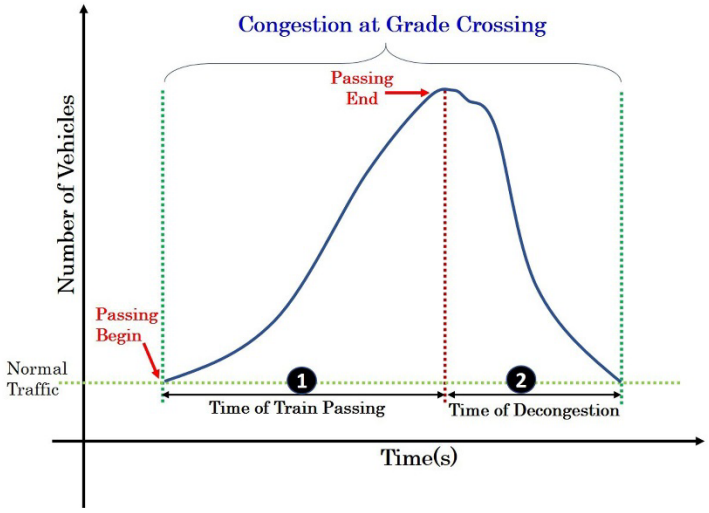


Figure 16. A schematic of congestion based on the variation in the number of vehicles

Based on these observations, a mathematical model is proposed to decipher and assess the temporal behavior of the traffic condition and the number of vehicles at the grade crossing as shown by the bold curve in Figure 16, which then can be used to estimate decongestion time. Figure 16 shows that when the train arrives and the grade crossing gate lowers, all the vehicles yet to pass the gates will slow down and eventually stop, leading to traffic congestion. As a result, the vehicle number in front of the gate will increase, which will continue during the entire period of the train's passing. Once the train completely passes through the grade crossing and the gate rises up, the vehicles at the front of the queue will move first and the congested traffic starts to be cleared. The process of decongestion will continue until the traffic flow returns to the normal condition, viz., the number of the vehicles in the images becomes similar to that prior to the congestion event. Therefore, as revealed by Figure 16, the ETT of a congestion event includes two parts: the ETP and the estimated time of decongestion (ETD), i.e., Eq. (4).

$$ETT = ETP + ETD. \tag{4}$$

Normally, the time of train passing is difficult to predict unless the dispatch center of the railroad company shares train operation information. Such a platform of information sharing, however, is

currently unavailable and will be developed in this project's future work. Therefore, in the present research, the ETP is estimated by detecting the moving train in the recorded videos.

The ETD is defined as the period between the time when the train completely transits the grade crossing and the time when the traffic returns to the normal condition. The present research aims to establish a mathematical model for predicting the ETD that needs to be shared with first responders the moment the train passes through the grade crossing (or the crossing gate raises up) for their situational assessment and decision-making. The research team hypothesizes that the ETD may follow a predictable pattern **F** and depend on two parameters, i.e., the NTC and ETP, where the NTC represents the number of vehicles on the road before the train arrives, i.e., the NTC at the grade crossing. It will take more time to decongest the accumulated traffic if the normal traffic is busier (e.g., rush hours) and/or the ETP is longer, because both contribute to more serious vehicle blockage (but to different extents). The relationship F can be expressed as

$$ETD = F(NTC, ETP) \tag{5}$$

4.2 The Proposed System for Decongestion Time Estimation

The proposed system and the specific methods to model the NTC, ETP, and ETD are discussed in this section. The constructed models can then be used for predicting ETD. As show in Figure 17, the proposed system includes two stages:

(1) **The offline model generation stage.** In this stage, ground truth (GT) values of the NTC, the number of vehicles during non-normal traffic condition (NNTC) (i.e., during congestion), ETP and ETD, are manually extract from recorded video clips. The GT-NNTC is found by counting the number of vehicles during congestion events. The GT-NTC is obtained by averaging the number of vehicles in each frame before the train arrives or the frame without a congestion event at all. The GT-ETP, i.e., the black arrow labeled "1" in Figure 16, is directly observed from the video. To obtain the GT-ETD, the GT-NTC and GT-ETP are employed to measure the period between the moment the train leaves the ROI and the moment the traffic condition returns normal during a congestion event, i.e., the black arrow labeled "2" in Figure 16. With these GT data, three models are built during the offline stage, including (i) the deep convolutional neural network (CNN) for vehicle crowd counting, which is trained using GT-NTC and GT-NNTC. Note that a large amount of vehicle traffic data (including before and during the congestion or without congestion at all) is available to train the deep CNN; (ii) the train detection model built by the running average-based motion detection using GT-ETP; and (iii) the ETD prediction model to establish the relationship **F** between GT-ETD and (GT-NTC, GT-ETP) in Eq. (5). Due to data deficiency (30 congestion events in this study), it would be a formidable task to build a DNN model; instead, a multi-variate polynomial regression model is constructed to fit the extracted values. Note that the present method combines different image analysis and modeling approaches to tackle the challenge associated with the limited data of congestion events; such a hybrid method is different from the solely end-to-end deep learning and represents the most tangible novelty of the present study.

(2) **The online testing stage using the offline generated to predict the ETD**. In this stage, the two quantitative features – the NTC and the ETP – need to be attained in an automated manner without human intervention. Specifically, the deep CNN and the train

motion detection model constructed offline are used to count the number of vehicles at the NTC and the ETP in the images, both were used as the inputs of the regression model, i.e., **F** to predict the ETD.

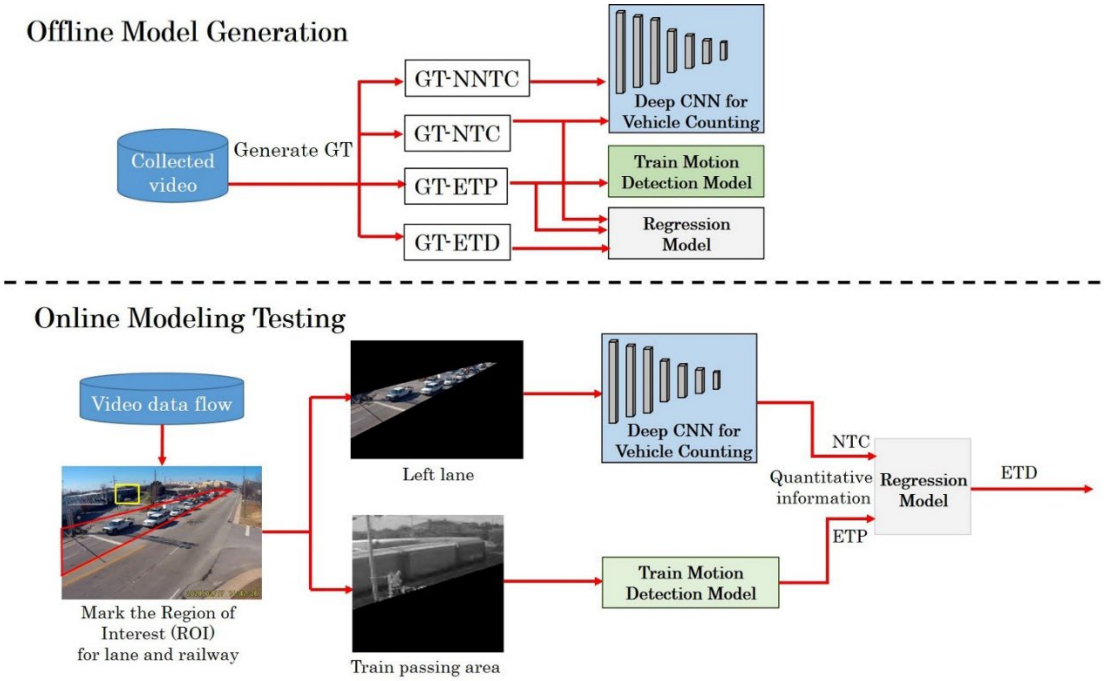
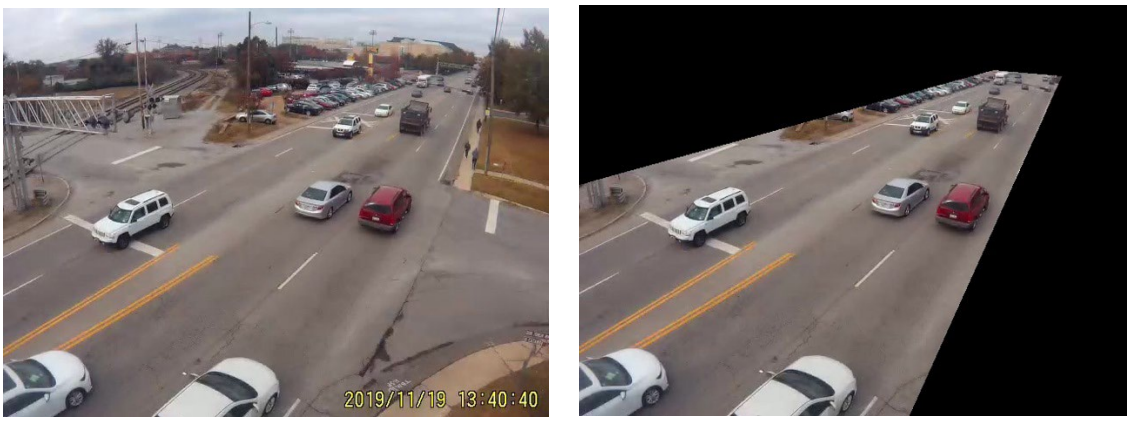


Figure 17. Proposed learning-based method for decongestion time prediction

4.2.1 Image Preprocessing

As shown in Figure 15, the images captured from the designated grade crossing contains a variety of information, including the lanes, railroad, trees, buildings, and parking area. The irrelevant background could degrade the model performance without preprocessing on the original imagery data. For example, the railroad is located at the upper left corner of the image, and will be analyzed by the motion detection algorithm to evaluate ETP, which might be influenced by moving vehicles on the lane. Moreover, changes in illuminating conditions and shadows of buildings and/or trees would also compromise the detection performance. Thus, an image preprocessing step that marks the ROIs and appropriately processes the raw images is necessary. Figure 18(a)(b) presents an example of the captured raw frames at the grade crossing and its corresponding ROI in the lane area, respectively. Note that the whole lane area is labeled as the ROI for the offline training of the deep CNN, since as many instances as possible are needed to improve the model. While extracting NTC for decongestion time estimation in the online stage, the ROI is limited to the left lane because it is adequate for studying the congestion event.



(a) (b)

Figure 18. Image preprocessing: (a) Raw images of the grade crossing from camera feed; and (b) Region of Interest (ROI) for the offline training at the grade crossing

Another preprocessing step is to mark the railroad area as the ROI, which only contains the trains without other moving objects and is then used to evaluate ETP. Figure 19 illustrates two candidate ROIs for the railroad area, although there might be other choices. In this work, ROI 2 is selected since it minimizes the effect of background changes on train detection. In ROI 1 there are other moving objects, such as vehicles and pedestrians, and the motion detector might falsely count them as the moving train, leading to overestimation of ETP. In contrast, the background of ROI 2 is free of external disturbance and is stable.

Figure 19. Two candidate ROIs for train passing time estimation

4.2.2 Vehicle Counting for NTC Estimation

Vehicle counting is an important component of the proposed framework, and it produces a key quantitative feature (i.e., the NTC), in a given frame for modeling and predicting the ETD. As mentioned, the deep CNN-based vehicle crowd-counting has three options: counting by detection, by regression, and by estimation of the density map. In this work, the density map approach was adopted.

(a) **(b)**

Figure 20. Example of a grade crossing frame and its corresponding density map

Density map estimation needs a set of annotated images, where all the vehicles are marked by dots. In this scenario, the GT density map D_I for an image I, is defined as a sum of Gaussian functions centered on each dot annotation,

$$D_I^{GT}(p) = \sum_{\mu \in K_I} N(p; \mu, \sigma), \tag{6}$$

where $N(p; \mu, \sigma)$ represents a normalized 2D Gaussian function, with standard deviation σ, K_I, is the set of 2D points annotated for image I, and p is the pixel position. Unlike the geometry-adaptive kernels used in the literature, the research team used the fixed standard deviation value $\sigma = 5$ considering the relatively small number of objects in the scenes. With a density map D_I^{GT}, the total object count M_I^{GT} can be directly obtained by integrating the density map values in D_I^{GT} over the entire image:

$$M_I^{GT} = \sum_{p \in I} D_I^{GT}(p). \tag{7}$$

Figure 20(a)(b) show an example image at the grade crossing and its corresponding density map. Given the input image and the density map, a deep CNN architecture is constructed to learn the nonlinear mapping relationship H. Once trained, model H takes an image I as an input, and returns an object density map prediction D_I^{pred},

$$D_I^{pred} = H(I|\omega), \tag{8}$$

where ω is the set of model parameters.

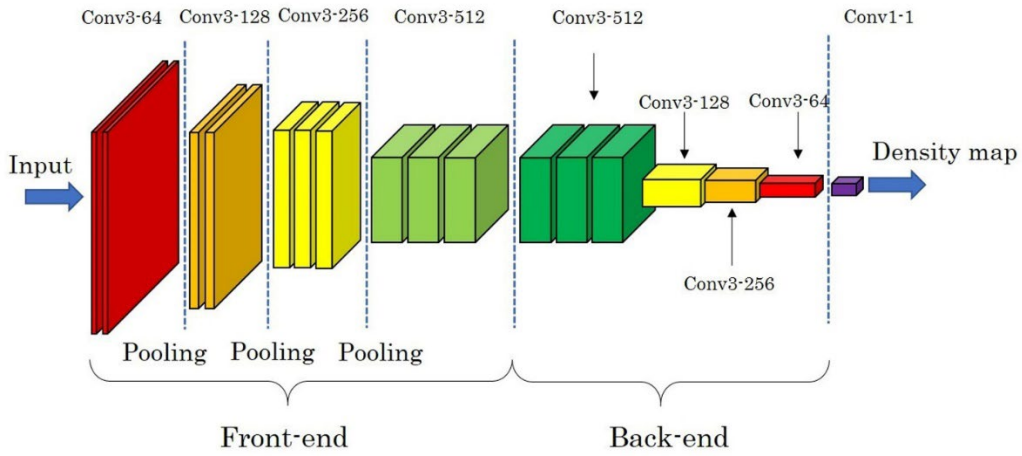


Figure 21. Architecture of deep CNN for density map estimation

The CSRNet was adopted for density map estimation, and its architecture is shown in Figure 21. The convolutional layers' parameters are denoted as "Conv(kernel size)-(number of filters)." The pooling layers are conducted over a 2 × 2 pixel window with stride 2. The front-end part of the model uses normal convolution, the back-end part use dilated convolution with a dilation rate of $r = 2$. The CSRNet contains a front-end network and a back-end network. The VGG-16 network was chosen as the front-end because of its strong transfer learning ability and its flexibility for easily concatenating the back-end to generate the density map. Similar to the configuration of the CSRnet, the front-end network in this work has 10 convolutional layers, and each convolutional layer has a 3 × 3 kernel. Note that there are three max-pooling layers in the front-end, and therefore the output size of the front-end network is 1/8 the original input size. To avoid further shrinkage of the density map size, CSRNet employed dilated convolutional layers as the back-end for extracting deeper information of saliency and maintaining the output resolution. A 2D dilated convolution can be defined as follows:

$$O(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i, n + r \times j)\, f(i,j), \tag{9}$$

where $O(m,n)$ is the output of dilated convolution from the input and a filter $x(m,n)$ and the filter has a length and a width of M and N, respectively. The parameter r is the dilation rate, where $r = 1$ indicates that dilated convolution turns into normal convolution. $r = 2$ is chosen in our work. In the end, a 1 × 1 convolutional layer is added as an output layer to generate the density map.

The Euclidean distance is chosen to measure the difference between the GT and the generated density map. The loss function is given as follow:

$$L(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left|D_i^{Pred} - D_i^{GT}\right|_2^2 \tag{10}$$

where N is the size of a training batch and D_i^{pred} is the density map predicted by the DNN model, and D_i^{GT} is the true density map of the ith input image. To train this network, the pre-trained VGG-16 model was used to initialize the front-end network. For the rest of the layers, the

initial values came from a Gaussian initialization with a standard deviation value of 0.01. Stochastic gradient descent was applied with a fixed learning rate of e^{-7}.

4.2.3 Estimated Time of Train Passing

Given the marked ROI of the railroad, the next task was to extract another important feature: the ETP. Due to the limited data of congestion events, it would be challenging to adapt or retrain a DNN for train detection and tracking for two reasons: first, it normally requires a diverse set of annotated data associated with different congestion events; and second, the well-trained models that could be fine-tuned to solve the ETP estimation problem in this scenario is indeed scarce. Therefore, project researchers proposed an economic motion detection model to estimate the ETP, which has a lower data requirements and circumvents the need for data annotation or a training process.

The acquired videos recorded the duration of the train presence during a congestion event, including the train approaching, passing, and leaving the scene. Hence, the research team defines the ETP as the time interval between the two instants the train appeared and vanished in the ROI. As discussed above, the selected ROI had a relatively stable background compared to the road, and the train presence could be detected by measuring the difference between two frames, which then could be used to estimate ETP. The idea was to construct a background model, and then identify pixels in the current frame that differed from those in the background to indicate the presence of the train. Therefore, the quality of the background model dictated the accuracy of the ETP estimation. The analysis shows that the use of a single stationary image as the background yielded poor performance due to the changes in illuminating conditions at two different time scales and the extremely slow-moving trains. In the short time scale, i.e., during the period when the train passed through the grade crossing, the instantaneous changes in the pixel intensity between two neighboring frames, e.g., signal impulse and drift in pixels could be caused by the varying illuminating conditions due to reflection and refraction of vehicles on the road and the shade of an approaching train. In the long term, the illuminating conditions vary with the time of the day, due to the sun orientations, cloud, and other conditions. Therefore, image preprocessing and dynamic background models need to be developed to address these issues.

To mitigate the effect of the illuminating changes in the short term, the captured video frames were first converted from three channels to a single channel, since one channel frame would be sufficient for motion detection. Then a Gaussian filter was applied on the grayscale patches in the ROI to smooth out the impulse signal caused by the instantaneous illuminating changes:

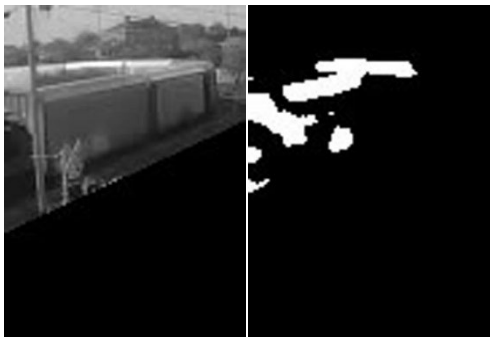
$$G_{2D}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \qquad A_g(t) = A(t) * G_{2D}, \tag{11}$$

where the Gaussian filter G_{2D} has a 9×9 kernel, the A, and A_g is the corresponding input and output image. This process smooths the ROI patch and improves the image quality for motion detection. Then, a dynamic background based on the running average of the previous frames was employed to further tackle the problem due to the short-term drift of light intensity, i.e.,

$$R(t) = (1 - \alpha)R(t - 1) + \alpha A_g(t), \tag{12}$$

where $R(t)$ is the running average, $A_g(t)$ is the frame being added to the running average. The current running average at the time (i.e., \$t\$) is the weighted sum of the running average at $t-1$

and the current frame $A_g(t)$. $R(t)$ might look similar to the current frame since it incorporates the current frame into the background model. Different from the Gaussian filtering, Eq. (11) smooths the data in the time to further reduce the drift in illumination when the train is passing through the grade crossing. Thus, $R(t)$ represents a short-term, dynamic background updating, which reflects the instantaneous image characteristics near the current frame. α is a parameter that decides how much weight should be assigned to the new frame: a large α value will reduce the difference between the current frame and running average. The parameter α was set to 0.1 in the experiments.



(a) **(b)**

Figure 22. (a) Gray-level image of the frame and (b) threshold the absolute difference between the smoothed frame and the running average to yield the binary map

Pixels in the current frame that show distinction from the background are identified by

$$E(t) = \left|A_g(t) - R(t)\right|, \tag{13}$$

Where $E(t)$ is the difference between the current frame $A_g(t)$ and the running average $R(t)$. A threshold value of 20 is applied on the $E(t)$ to classify the pixels into the moving object and the background, as shown by the binary motion map in Figure 22. Figure 23(a) shows the ratio of the motion pixels in ROI over time. The original fractions (blue curve in the Figure 23) are the ratio of the detected motion pixels to the overall pixels in the ROI. The smoothed fractions (orange curve in the Figure 23) are the result of averaging the original fraction value with five previous fraction values, aiming to reduce the effect of impulse noises. The research team denoted the smoothed fractions as t. As shown in Figure 23, the GT of the train passing time of the example is from frame No. 100 to No. 717. A threshold line in green with a value of 0.025 is drawn in the figure to separate the frames with the low and the high ratio of motion pixels. Note that the threshold value is an empirical parameter and dependent on specific grade crossings. Thus, the ETP can be estimated by counting the number of frames, whose fraction of motion pixels is above the threshold line. However, the example in Figure 23(a) clearly shows that this fraction value for frame No. 150 to No. 650 is much lower than the threshold line, although the train is indeed present in the image. In real application scenarios, the train might move extremely slowly or even stop at the grade crossing. Then the detector that only consider a few previous frames would not work since no moving objects are detected in the ROI.

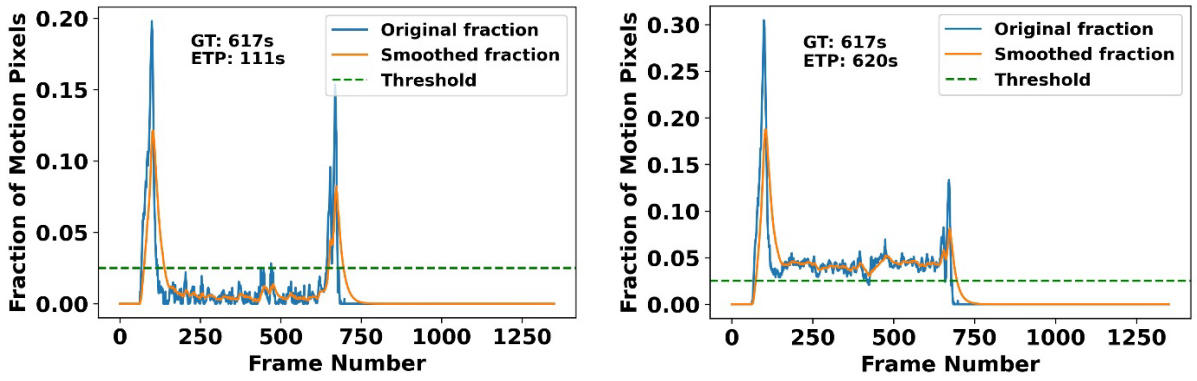


Figure 23. Fraction of motion pixels over time. GT and ETP mean the ground truth and estimated train passing time; (a) motion difference is calculated by Eq. 10, and (b) motion difference is calculated by Eq. 12

An explicit approach to resolve this problem is to compare the current frames $F_g(t)$ with another background image that does not contain the train, and this background is termed the offset background hereafter to distinguish from the short-term background above. However, the offset background varies with the time of day and the weather condition, such as sun orientation, cloud, etc., i.e., the long time-scale effect of the illuminating conditions. Therefore, to accommodate this, the offset background $B(t)$ also needs to be updated by the running average around the clock when the scene does not contain a train. As an initial condition $t = 0$, assume there is no train in the ROI and set $B(0) = A_g(t)$, and then $B(t)$ is updated by adding the current frame $A_g(t)$:

$$B(t) = [B(t) + A_g(t)]/2, \tag{14}$$

Note that operation in Eq. (11) will be performed only if the previous fraction $T(t-1) < 0.01$, where 0.01 is an empirical parameter. One special circumstance is that the train is not present in the frame at $t-1$, but appears in the frame t, then the train-contained ROI patch will be added to $B(t)$. In fact, this would not affect the offset background significantly, since the faction $T(t)$ would become larger than 0.01, and no more train-contained patch after t will be added to $B(t)$. In other words, the offset background is only updated during NTCs without congestion, and becomes "frozen" when a train is passing. Compare to the background $R(t)$ in Eq. (12), which only considers the short-term changes in illuminating conditions near the current frame; the offset background $B(t)$ considers the long-term environmental variation before the train arrives. Eventually, those two backgrounds are combined for calculating the motion difference. The new difference for motion detection is written as:

$$E(t) = (1 - \beta)|A_g(t) - R(t)| + \beta|A_g(t) - B(t), \tag{15}$$

where β is a weight parameter ranging from 0 to 1. Note that β is an empirical coefficient and depends on the specific grade crossing and camera location and orientation relative to the rail, which needs to be determined during the calibration stage of the system. Here researchers chose $\beta = 0.3$.

Figure 24(b) shows the fraction ratio of the motion pixels in each frame over time. In this figure, the offset background patch was applied to detect the motion pixels. The value of the fraction ratio in Figure 24(b) was larger than the fractions in Figure 24(a), indicating the moving train

was detected in the ROI. By incorporating the offset ground, the frames even with slowly moving trains could be easily distinguished from the background frames subjected to the threshold value. Specifically, the ETP time predicted by the proposed method was 620 seconds—very close to the ground truth value 617 seconds.

Algorithm1: Proposed algorithm for the train passing time estimation.

1. Mark the ROI for railroad area.
2. For {t = 1:N} do \%\% N is the number of frames
   a. Convert the frame to one channel (gray scale).
   b. Smooth the gray scaled image with a Gaussian kernel using Eq. 11.
   c. Develop a model of background by running average of the previous frames and current frame; see Eq. 12.
   d. Develop an offset background model by updating the frames during NTCs; see Eq. 14.
   e. Identify the overall difference using Eq. 15, then calculate the original fraction value.
   f. Smooth the original fraction value to obtain the averaged fraction T.
3. End For
4. Threshold the smoothed fraction curve, and calculate the train passing time.

Algorithm 1 summarizes the entire process of the ETP estimation. Lines 3 and 4 describe the preprocessing step for each input frame. Lines 5 and 6 establish the background model $R(t)$ and the offset background model $B(t)$, where both the short-term illumination changes and long-term environmental variations are taken into consideration. Line 7 and 8 identify the motion difference and calculate the smoothed fraction of motion pixels. In the end, line 10 calculates the ETP by counting the number of frames above the threshold line.

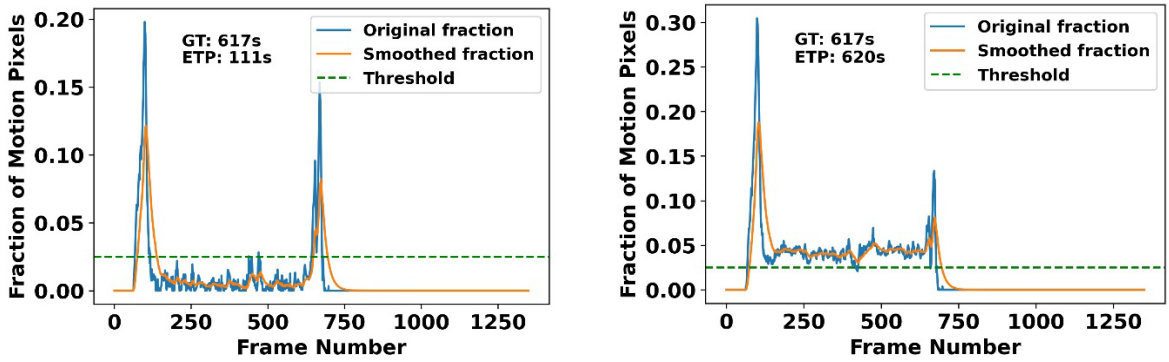


Figure 24. Fraction of motion pixels over time. GT and ETP mean the ground truth and estimated train passing time; (a) motion difference is calculated by Eq. 10, and (b) motion difference is calculated by Eq 15

4.2.4 Estimated Time of Decongestion

As shown in Eq. 2, the ETD depends on the estimated number of vehicles in the NTC before a train arrives (i.e., the NTC) and the ETP. A model that can learn the relationship F between these

three parameters needs to be built. There are plenty of techniques to determine F, such as the regression model and the neural network. In this report, the regression model is selected for the following reasons. First, the mapping relationship F is relatively simple and only involves two inputs (NTC and ETP) and one output, while neural networks are normally used for more complex problems. Second, the regression features a simple model structure and algorithm (i.e., least squares) that requires less data for modeling and is more effective for mitigating the data deficiency issue associated with a limited number of the congestion events in the present study. Third, the neural network has more hyperparameters to determine before training, such as the number of layers and the number of neurons in each layer, in order to mitigate the risk of overfitting and improve model accuracy, which in general is not required for the regression model of a low-order polynomials with fewer inputs. Last, the computing requirements for the regression model is much lower, rendering it well-suited for estimation and prediction in real-time on cost-effective edge computing platforms. More specifically, a quadratic polynomial regression form that takes the ETP and NTC as the inputs and the ETD as the output is adopted to capture the mapping relationship F, i.e.,

$$ETD = a{\cdot}NTC^2 + b{\cdot}ETP^2 + c{\cdot}NTC{\cdot}ETP + d{\cdot}NTC + e{\cdot}ETP + f \tag{16}$$

where the variables from a to f are regression parameters, which are determined by the least squares methods. Once fitted, Eq. (16) can be used for the ETD prediction. Note that due to the lack of the information of the train operation, it cannot predict the ETD if the train is still passing the grade crossing. Therefore, as indicated by Figure 16, the ETD should be evaluated at the moment the crossing gate raises up or the train just passes through the railroad ROI. Thus, the ETP can be estimated accurately using the motion detection algorithm in Section 4.2.3.

4.3 Results and Discussion

This section presents the experimental details and evaluation results using the collected imagery dataset at the designated grade crossing. The performance of the deep CNN-based vehicle crowd-counting algorithm for the NTC assessment is first evaluated, and the quantitative error is measured using a standard mean absolute error (MAE) metric. Secondly, the train motion detection algorithm to estimate the ETP is examined by comparing the prediction results with the ground truth. Finally, the prediction of the ETD in the congestion events using the regression model and the extracted quantitative features is verified.

4.3.1 Experimental Results of Vehicle Crowd Counting

The performance vehicle crowd counting is characterized in this section. The dataset contains 1,000 annotated images for nearly 20,000 vehicles. MAE was used as the metric for assessing the performance:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|M_i^{Pred} - M_i^{GT}\right| \tag{17}$$

where M_i^{pred} is the model-predicted vehicle number, and M_i^{GT} is the vehicle count from human-labeled annotations. Note that M_i^{pred} predicted by the DNN can take a continuous number while the ground truth is always an integer. MAE is an indicator of the accuracy of the predicted vehicle count across the test frames. Extensive experiments are conducted, and the dataset is divided into three parts: training, validation, and testing sets. The training set has 700 images,

with the validation set and the testing set containing 150 images each. The pre-trained VGG-16 model is used to initialize the front-end network, while the remaining part of the model is initialized by a Gaussian kernel with a standard deviation value of 0.01. A stochastic gradient descent optimizer with a fixed learning rate $1 \times e^{-7}$ is employed to update the model. The mask of the ROI is applied to each image in the dataset.

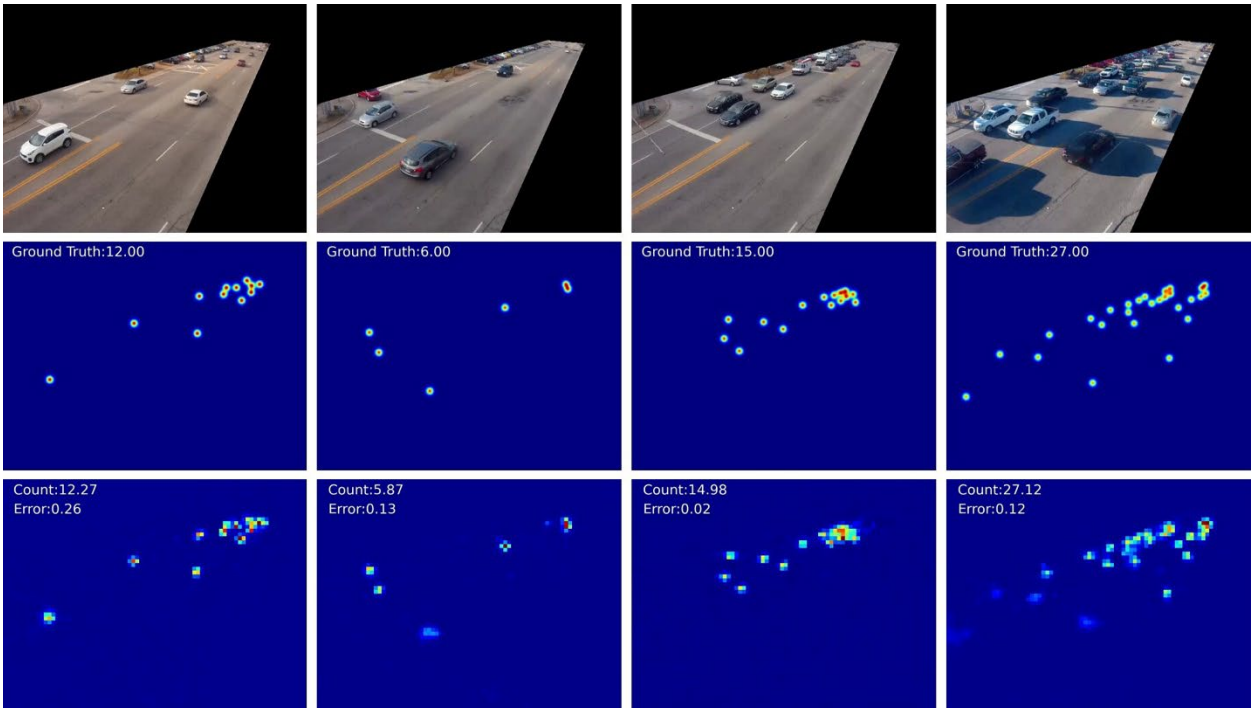


Figure 25. Comparison of density map between model prediction and GT. Top row: Sample input image from the grade crossing dataset with ROI. Second row: GT. Third row: Estimated density map

Figure 25 illustrates four example images with the increasing number of vehicles in the scene (from the left to the right of the figure), and the comparison between the ground truth (the second row) and the predicted density map (bottom row). Note that the CSRNet has three max-pooling layers, which means that the output size of the front-end network is 1/8 of the original input size. Thus, the target density map needs to be resized to keep consistent with the original image, leading to the less smooth density distribution in the output image. The model-predicted density maps followed the spatial distribution of the vehicles visually, and matched the ground true very well. The MAE errors of the four examples were all less than one (and 1.16 vehicle across the test set), confirming that the model based on the CSRNet was accurate for vehicle crowd counting in the grade crossing being considered.

4.3.2 Experimental Results of ETP Estimation

The proposed motion detection algorithm to evaluate ETP was examined using test dataset collected during 30 congestion events. The captured video clips had a frame rate of 24 frames per second. Due to the slow-moving nature of the train in several congestion events, the difference between the adjacent frames was too small to detect. Moreover, running the algorithms on every single frame required more computing power and might not meet the

requirement of real-time processing in our future efforts. Thus, video clips were downsampled by extracting 1 out of every 24 frames. In other words, only one frame was processed in each second—greatly reducing computational loads and improving the detection efficiency without compromising the predication accuracy because the ETP was typically at the magnitude of minutes and even tens of minutes. Figure 26 presents six examples of the testing result with GT and ETP provided in each case. The ETP value was calculated by counting the frames of the smoothed fraction curve that exceeded the threshold value of 0.025 in all the cases. The ground truth and estimated ETP are provided in each sub-figure, and the average MAE of the ETP in these six examples was 5.83 seconds (and 8.42 s across the entire testing dataset), indicating that the proposed algorithm produced fairly good estimations. In Figure 26(c)and (f), impulse noise is observed near frame No. 50 and frame No. 25, respectively, which may be attributed to the sudden change of the illumination intensity or the vibration of the camera. Fortunately, the duration of the impulse noise was very short, and smoothing the original fraction curve could dramatically alleviate its effect on ETP estimation.

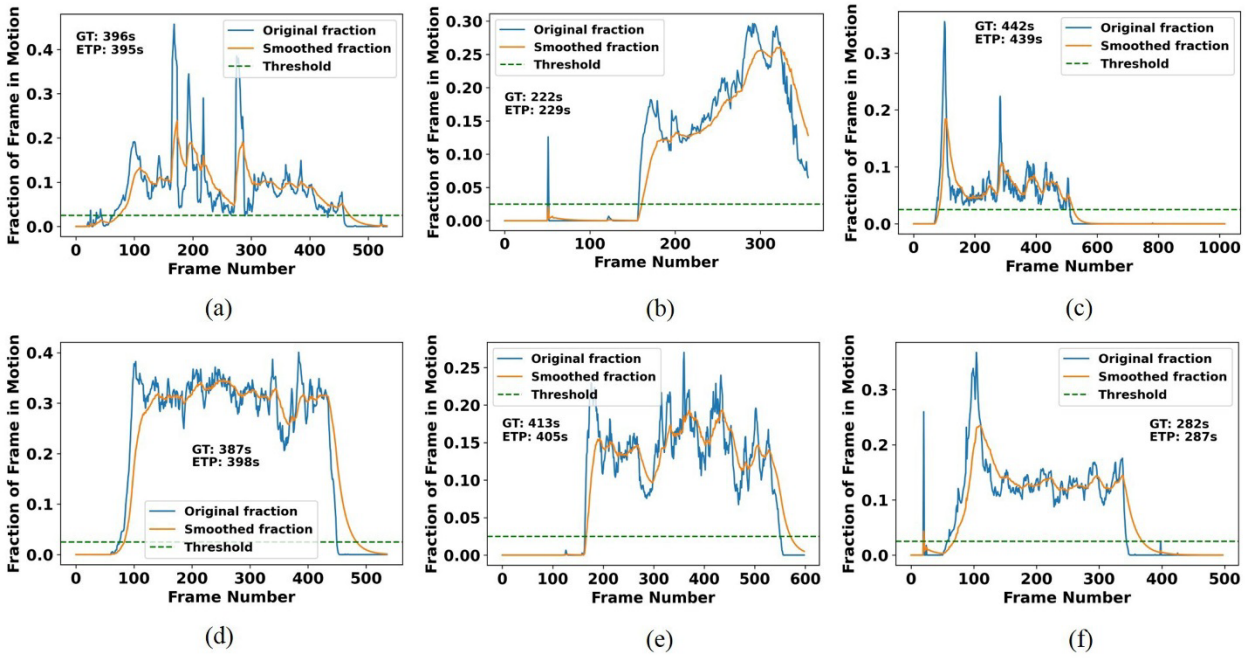


Figure 26. Fraction of motion pixels within the ROI over time for evaluating ETP

4.3.3 Experimental Results of ETD Prediction

In this section, the traffic behavior of the vehicle numbers during the congestion events and its dependence on the ETP is analyzed. The results of extracting the ETD are presented, and the performance of the proposed framework for ETD regression and prediction using the two quantitative features, i.e., the NTC and the ETP, is characterized.

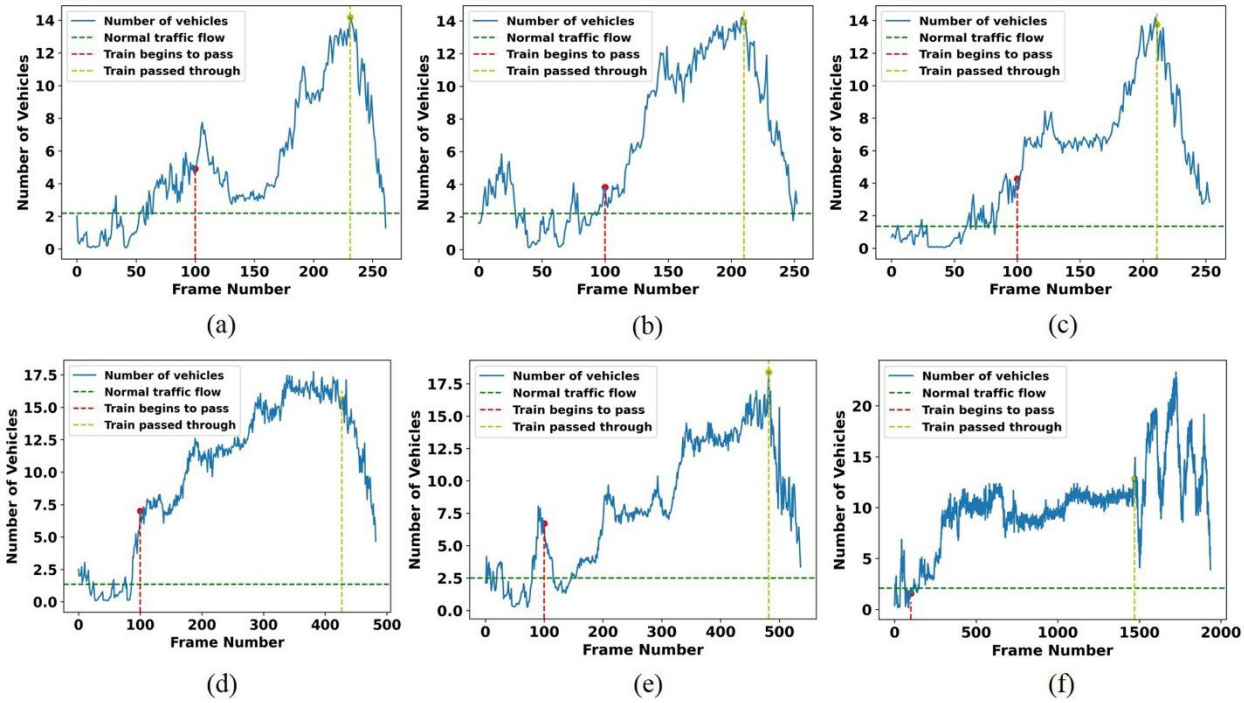
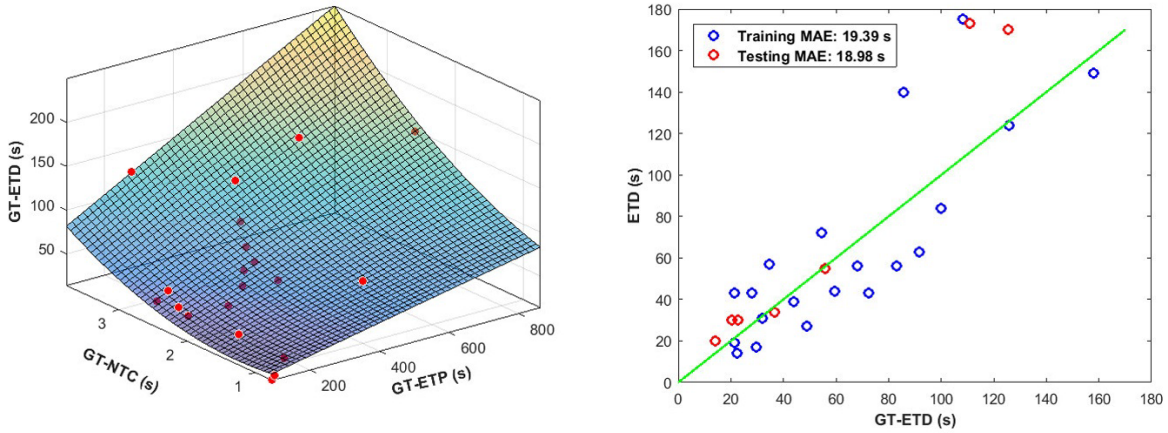


Figure 27. Variation of vehicle numbers against the number of frames

Prior to discussing the results, the hypothesis proposed in Figure 27 to assess the traffic behavior during a congestion event was examined first using the collected imagery data and the deep CNN vehicle crowd counting. Six representative congestion events were selected for analysis and visualization, among which three had a regular ETP around 120 s, two had a longer ETP over 300 s, and one extreme case had an ETP over 1,370 seconds. The selected cases had a large range of the ETP, which covered almost all the congestion events that occurred at this specific grade crossing. Figure 27 describes the variation of the number of vehicles versus the number of frames (corresponding to the time) for these six events. The green horizontal dash line demonstrates the normal traffic flow obtained by averaging the vehicle numbers predicted by the density map before train arrival. The red vertical dash line indicates the moment the train began to pass the grade crossing, and the yellow vertical dash line represents the complete transit of the train through the grade crossing. The curves in Figure 27(a)–(e) follow the traffic pattern as proposed in Figure 27 during the congestion event. The number of the vehicle increases during the time while the train is passing (i.e., between the red and the yellow dash line), and peaked near the moment when the gate raised up. Note that the number of vehicles did not increase monotonically, indicated by many local variations in the curves. This is mainly because the motorists might have taken an alternative route rather than waiting until the train passed through.

Once the traffic behavior was examined, next the research team analyzed the pattern of the ETD and its dependence on the ETP and NTC. Recall that the ETD was defined as the period between the moments when the train completely passes through the grade crossing and the traffic returns to the normal condition. The team expected it would take a longer ETD to clear the blocked traffic if the time of train passing is longer. For the three cases in Figure 26(a)–(c), the ETPs range from 100 to 150 seconds, and the corresponding ETD was 31, 40, and 41 seconds, respectively. In Figure 27(d)(e), the ETPs increase to nearly 350 s, and as a result, the ETD also rose to 48 and 47 seconds, respectively. Those results confirm that the ETD was strongly related

to the ETP. However, Figure 27(f) shows a very special scenario, and its traffic behavior did not follow the proposed pattern in Figure 27, which can be attributed to the prolonged period of the ETP. The train took around 1,370 s to cross the grade crossing, leading to more serious congestion that needed 260 s to be cleared. The drastic oscillation in the number of detected vehicles reflected the slow-moving traffic and "move-stop-move" behavior. As a result, the decongestion time was much longer than in other cases.



(a) **(b)**

Figure 28. ETD surface plot and regression result: (a) 3D surface plot using GT-ETP, GT-NTC, and GT-ETD; (b) regression analysis of ETD

The data collected in 30 congestion events were separated into training data (23 instances) and testing data (7 instances). Given the GT values of the training data (GT-ETD, GT-NTC and GT-ETP), the regression model in Eq. (13) can be fitted using the training data in the offline stage, shown as the 3D surface plot in Figure 28(a). Red dots in the figure is the congestion data point for 3D surface fitting. In the online stage, the quantitative features NTC and ETP extracted automatically by the deep CNN vehicle crowd-counting and the train motion detection model, which are entered as inputs to the regression model for ETD prediction. Figure 28(b) shows the comparison of the GT-ETD and the predicted ETD, the training data and testing data are both plotted in the figure. The ETD values are obtained by entering the ETP and NTC extracted by the proposed system to the regression model. The blue dots are training data for fitting the regression model, the testing data is marked in red dots. The prediction performance can be evaluated by measuring the distance between the dots and GT line. Note that the unusual cases like that in Figure 27(f) with the extremely long ETP and ETD, which cannot be accommodated by the proposed hypothesis of the congestion behavior or accurately captured by the model formulation in Eq. 16, were not included in the regression model in order not to compromise the model accuracy. For instance, MAE of training data, including all the congestion data is 67.23 seconds, in which the special case in Figure 27(f), yielded an absolute error of 380.4 s. After removing it, the MAE of training data dropped to 19.44 seconds. The fitted regression model was then examined with the testing data. As shown in Figure 28(b), the data points are at the vicinity of the GT line, quantitatively, corresponding to an MAE of 18.98 seconds, indicating good agreement between the predicted and GT results. These observations confirm that the proposed framework was able to reliably estimate NTC, ETP and ETD during the online model utilization stage. In the future, a more sophisticated model formation and structures including additional

factors needs to be developed with more congestion event data to consider the special cases. Figure 28 also provides valuable insights to understand the characteristics of decongestion at the grade crossing. Most importantly, the ETD increased almost linearly with the ETP. This observation is consistent with the common knowledge that more vehicles accumulate in front of the crossing gate if it takes more time for the train to pass. Second, the number of vehicles before the congestion, obtained by averaging the vehicle counts predicted by the deep CNN model, accurately represents the NTC at the grade crossing. Similarly, the ETD also depends on the number of vehicles during the NTC, but in a strong and nonlinear fashion, and the nonlinearity becomes more apparent at the larger NTC and ETP values, where the ETD rapidly increases.

4.4 Improve the Existing EDT Model

In this section, the research team introduces the techniques developed to improve the current decongestion time prediction method (Section 4.2) and the specific deep learning models used to assist the EDT method to enhance accuracy. The main improvement includes two aspects: automatic road ROI generation and anomaly detection-based ETP estimation.

4.4.1 Automatic Road ROI Generation Using Deep Learning with Attention

In Section 4.2.2, a deep learning-based method is introduced for NTC value estimation. In this method, the road ROI of input images need to be manually labeled before applying vehicle counting algorithms. However, it would pose extra challenges for deploying the vehicle-counting model at a grade crossing that has been trained. For example, any small vibration of the camera caused by bad weather or changes in a camera's positions after battery replacement requires recalibration of the road ROI to ensure correct NTC estimation. To solve this problem, the team proposes a U-net architecture with an attention mechanism to estimate the road ROI. The architecture of the proposed method can be found in Figure 29.

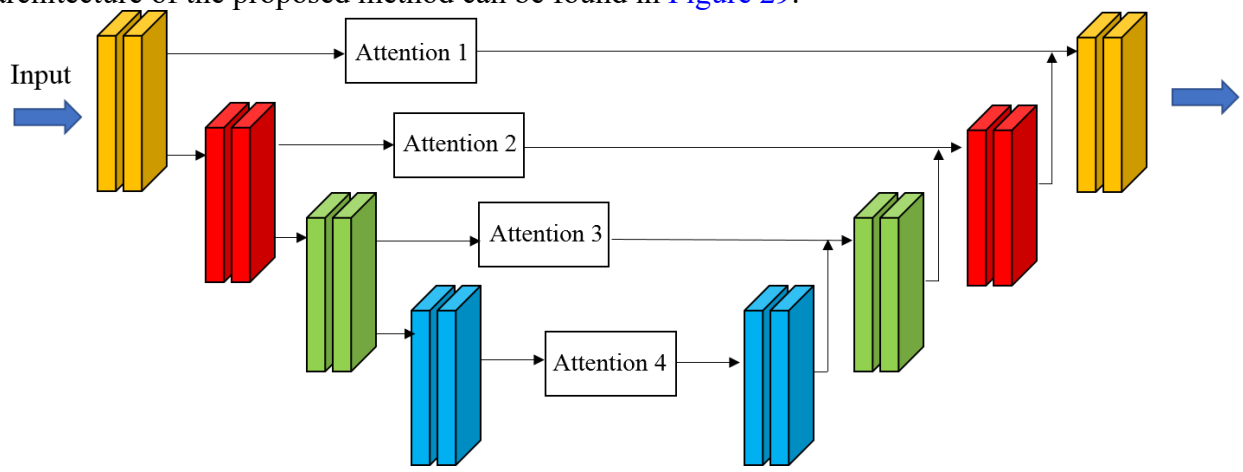


Figure 29. U-Net architecture with attention mechanism for current frame prediction and attention map estimation

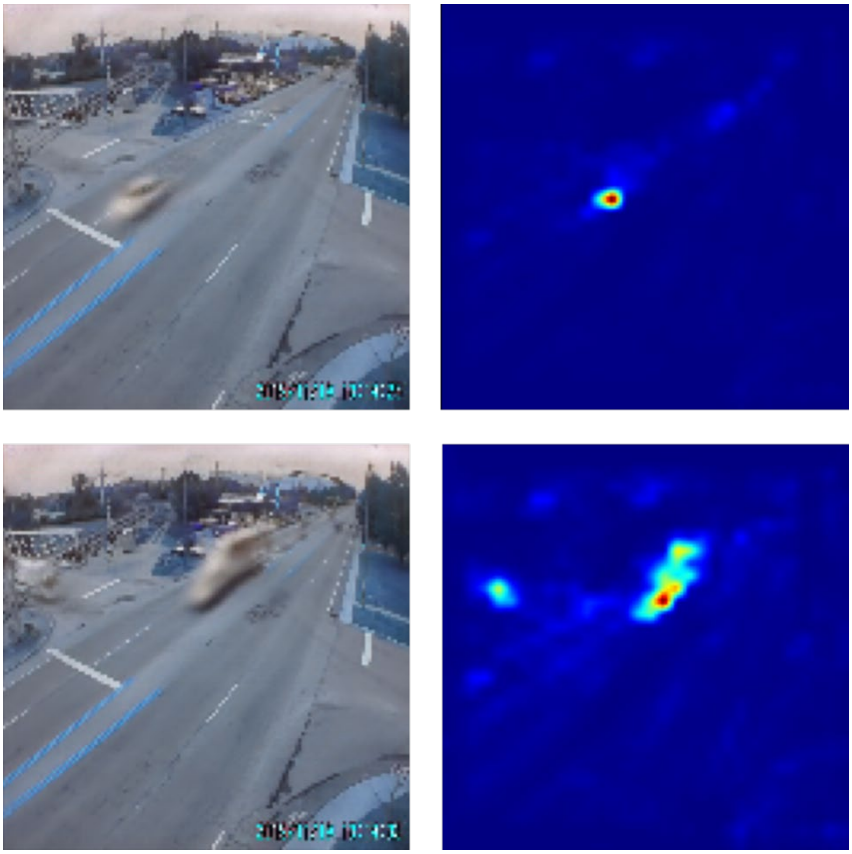
The key goal of U-net is to use attention maps to identify and exploit the effective spatial information extracted by CNN to predict the current frame image. This approach is premised on the hypothesis that it would be beneficial to identify ROI in the image and enlarge their influence, while suppressing the irrelevant and potentially confusing information in other regions. Thus, the team studied a trainable attention estimator and developed an approach to integrate it into U-Net to improve output prediction.

The road ROI generation takes advantage of the prediction of the current frame. For a given input image sequence $I = \{I_{n-1}, I_{n-2}, I_{n-3}, I_{n-4}\}$, the U-Net model output the current frame I_n. That is, four previous frames are sent to the network as inputs to predict the current frame. The set of feature map $L^s = \{l_1^s, l_2^s, \ldots, l_k^s\}$ denote as the feature vectors extracted at a given convolutional layer $s \in \{1,\ldots S\}$. Here, each l_i^s is the vector of output activations at the spatial location i of k total spatial location in a layer. For each of one or more layers s, the set of compatibility scores $C(L^s, g^s) = \{c_1^s, c_2^s, \ldots, c_k^s\}$, where g is a global image descriptor which is derived from input image and passed through a fully connected layer to obtain prediction probabilities. The compatibility scores are then normalized by a soft-max operation:

$$a_i^s = \frac{exp(c_i^s)}{\Sigma_j^k exp(c_j^s)}, i \in \{1, \ldots, k\} \tag{15}$$

The normalized compatibility scores $A^s = \{a_1^s, a_2^s, \ldots, a_k^s\}$ are corresponding to the attention as defined. The attention blocks are used after each convolutional blocks, but before their corresponding max-pooling operation. Note also that the use of the softmax function in normalizing the compatibility scores enforces $0 \leq a_i \leq 1 \; \forall i \in \{1, \ldots, n\}$ and $\Sigma_i a_i = 1$, that is, the combination of feature vectors is convex.

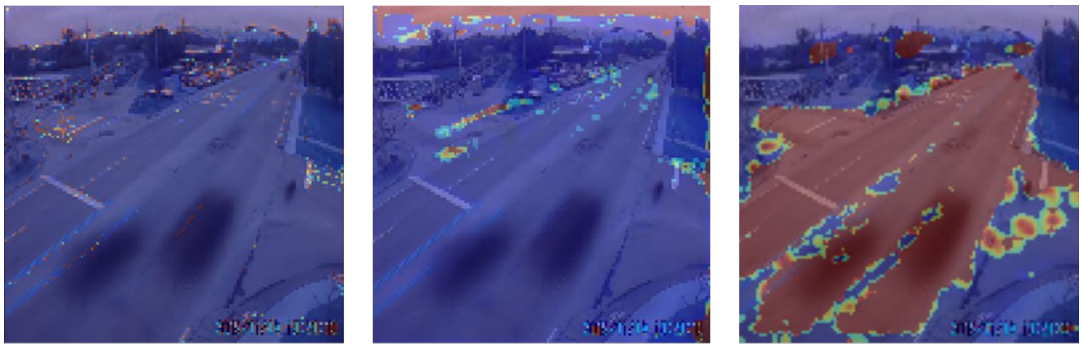
To incorporate attention into the model, the max-pooling layers were moved after each attention block to ensure that the local layers used for estimating attention have a higher resolution. The model has 16 convolutional layers, includes 8 encoder layers and 8 decoder layers; the feature map 2, 4, 6, and 8 are selected for generating attention, where g^s is calculated by sending L^s to another convolutional layer for mapping the feature to the given dimensionality. Figure 30 shows the results of prediction and the corresponding attention map generated from attention block 3. The attention was mainly localized in the vehicle areas of the input image, indicating that a high probability was assigned to those spots via soft-max operation. This is consistent with the knowledge that the moving vehicles in the input frames contain a lot of details that makes the prediction error higher than other areas—and hence, needs to receive more attention.



(a) (b)

Figure 30. Predicted image and the corresponding output of attention estimator 3: (a) the predicted images and (b) attention map

The experimental results above demonstrate that the attention block can localize the moving vehicles in the predicted frame. By accumulating all the attentions across the frames into a single map, a comprehensive road ROI will be generated according to the relative position of attention in each frame. Figure 31 shows the generated road ROI map by super imposing the attentions at each attention estimator across all the input frames. The road ROI maps become more comprehensive with the increase of convolutional layers, and the ROI estimated by attention 3 can cover almost all the road area. The automatically generated road ROI (from attention 3) can be easily combined with NTC model for vehicle counting.



(a) (b) (c)

Figure 31. Road ROI map generation: (a) ROI road map on attention estimator 1; (b) ROI road map on attention estimators 2; (c) ROI road map on attention estimator 3

4.4.2 ETP Estimation Using Anomaly Detection

In Section 4.2.3, the ETP estimation using motion detection is sensitive to the change in background. Thus, for the ROI labeling to evaluate ETP, it is necessary to exclude as much background as possible. Considering that the image data at the NTCs (without the train-induced congestion) is available, and therefore, a deep anomaly detection model was developed for passing train detection.

ETP estimation was formulated as a problem of anomaly detection based on the fact that the number of train-contained video clips (less than 5 percent of all the videos) were significantly less than the normal traffic data. For a given input image L_t, the model provided an output image $\widehat{L_t}$, which was the reconstructed version of the input image. The key idea of anomaly detection is that the model trained on normal data would generate large reconstruction error if used on abnormal data. Thus, the team defines the NTC (without train) as a normal event and a congested condition (train passing) as abnormal. An eight-layer convolutional neural network was developed for anomaly detection, which contains four encoders and four decoders, as shown in Figure 32.

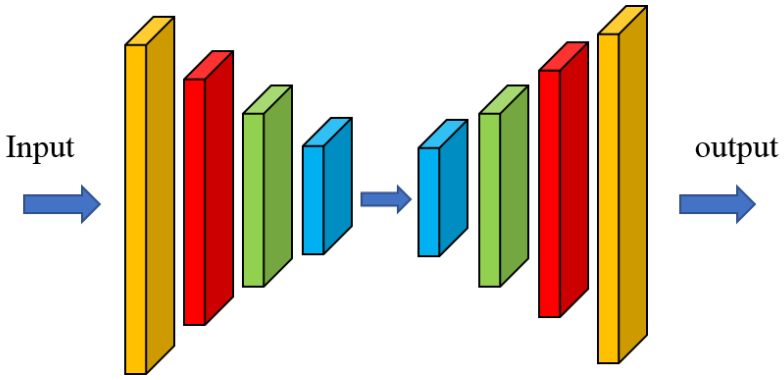


Figure 32. ROI patch for anomaly detection model for ETP estimation

An appropriate image preprocessing step was required for better detection results. Unlike the image preprocessing step introduced in Section 4.2.1, where backgrounds were excluded in the ROI, the ROI for anomaly detection-based ETP prediction can include more background areas

without influencing the ETP prediction result. Figure 32 shows the ROI patch used for anomaly detection.

To evaluate the reconstruction performance, the MSE was calculated first for each frame, then the peak-signal-to-noise ratio (PSNR) was used to measure the normality score:

$$PSNR = 20log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right) \tag{16}$$

Where Max_f is the maximum value of mean square error. Figure 33 gives the normality score for four cases where a train enters or leaves the ROI at specific moment. By observing the normality curve and ground truth value, researchers selected the threshold value 0.4 to distinguish the normal and abnormal frames. They found anomaly detection-based ETP prediction were comparable with the motion detection-based method, while the anomaly detection method was more stable and robust with the changing of ROI.

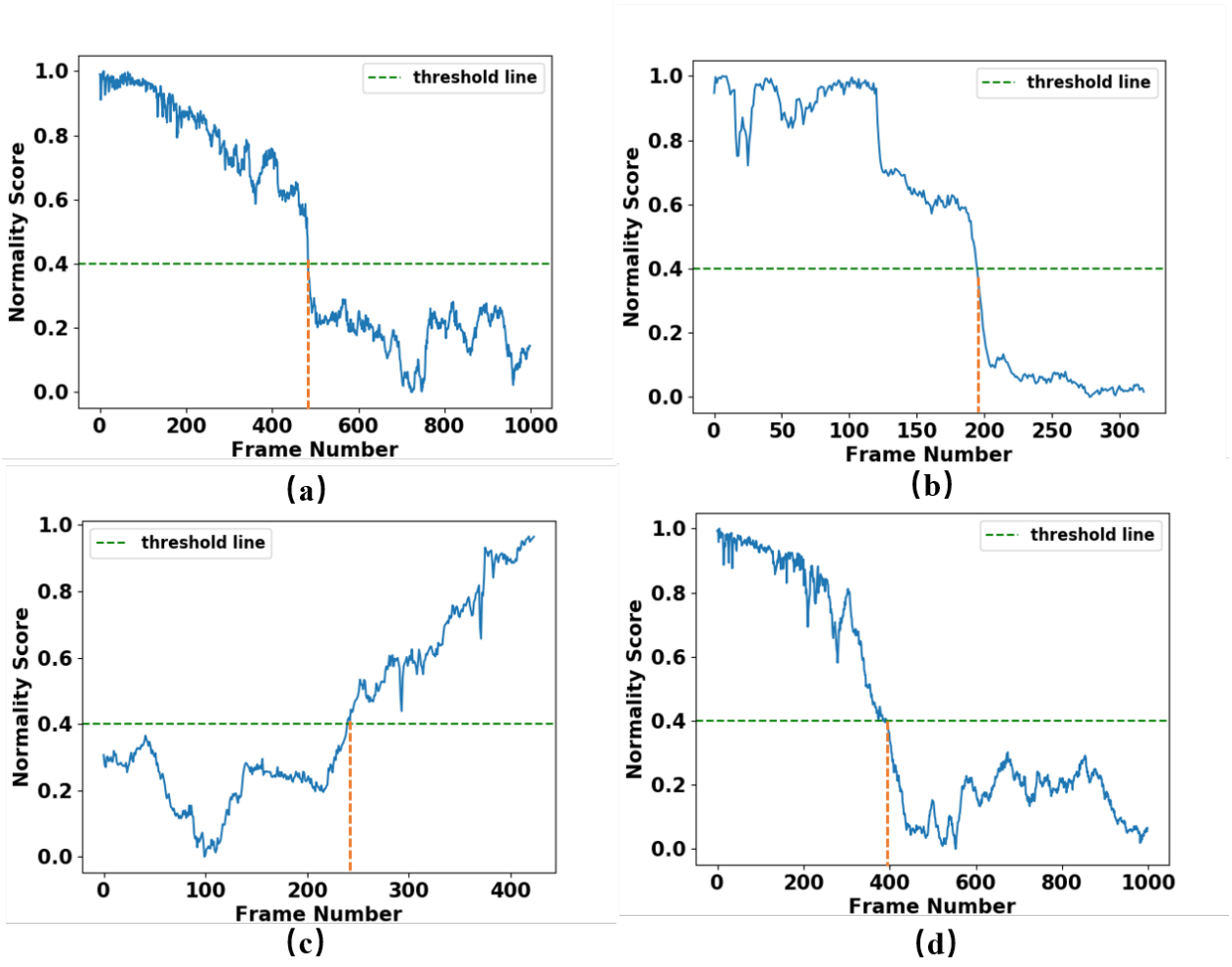


Figure 33. Normality score for four test cases where train enters or leaves the ROI at specific frame: (a) the ground truth value is (a) train enters ROI at frame 486; (b) train leaves ROI at frame 196; (c) train leaves ROI at frame 241; (d) train leaves ROI at frame 400

5. Conclusion

Railroad crossings may cause unexpected blockages and considerable traffic delays, especially in urban areas. There is no established system available to share the potential crossing blockage information with the public. Unexpected railroad crossing blockages may cause significant difficulties to first responders while they respond to emergencies. This research project aims to develop an affordable and field-deployable i-CATSS that can detect and predict highway-rail blockages at grade crossings and provide real-time information of traffic conditions to first responders. The key findings of the study are:

1. An information sharing channel has been established between the railroads and the system to receive train operation information from the industry partner. The shared train information, including GPS locations, train speed, and train length, were used to calculate the ETA and the ETD for the monitored railroad crossing.
2. A survey has been performed with the inputs of the first responders in Columbia, SC, to better understand the issues with unexpected congestion at railroad crossings and how this type of study can assist them. The crossing of interest in this study was selected based on the responses from first responders.
3. A novel AI model was developed based on video recorded at the crossing of interest to automatically detect the number of vehicles waiting in the queue during crossing blockages. The AI model operation was automatically triggered by an approaching train detected from the surveillance camera.
4. An adaptive model was developed to predict the total delay time due to potential railroad crossing blockages. This information can be shared with the first responders to assist with vehicle dispatching and rerouting for rapid situational assessment and decision-making during an emergency.

This study established an information-sharing mechanism between the railroad and first responders that depends on the railroads to update their train operation information. It is possible to extend this study in the future to directly read PTC signals from the wayside to make the system fully automated.

This study focuses on predicting the total delay time for the grade crossings under monitoring. However, to make it more practical, it was necessary to extend the predictions to different crossings along the railroad track. First responders could use delay time information collected at the blocked crossings en route to the destination to enhance the responsiveness.

6. References

Arnott, R., De Palma, A., & Lindsey, R. (1991). Does providing information to drivers reduce traffic congestion? *Transportation Research Part A: General, 25*(5), 309–318.

Association of American Railroads. (2021). *Freight Railroads & Positive Train Control.* Retrieved from AAR.

Babu Sam, D., Sajjan, N. N., Babu, R. V., & Srinivasan, M. (2018). Divide and Grow: Capturing Huge Diversity in Crowd Images with Incrementally Growing CNN. *Proceedings of the IEEE conference on computer vision and pattern recognition.* Bangalore, India.

Babu Sam, D., Surya, S., & Babu, R. V. (2017). Switching Convolutional Neural Network for Crowd Counting. *Proceedings of the IEEE conference on computer vision and pattern recognition.* Bangalore, India: Indian Institute of Science.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. (1995). Dynamic model of traffic congestion and numberical simulation. *Physical Review E, 51*(2), 1035.

Barkouk, H., En-Naimi, E.M., & Mahboub, A. (2017). Overview of Positive Train Control (PTC) and Railway Collision Avoidance System (RCAS). *ICCWCS'17: Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems*, (pp. 1–6).

Baron, W., & da Silva, M. (2019). *Effects of In-Pavement Lights on Driver Compliance with Grade Crossing Safety Equipment.* Technical Report No. DOT/FRA/ORD-19/10, Washington, DC: U.S. Department of Transportation, Federal Railroad Administration.

Bauza, B., & Gozalvez, J. (2013). Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. *Journal of Network and Computer Applications, 36*(5), 1295–1307.

Brabb, D. C., Vithani, A., & Martin, K. (2017). *In-Depth Data Analysis of Grade Crossing Accidents Resulting in Injuries and Fatalities.* Technical Report No. DOT/FRA/ORD-17/04, Washington, DC: U.S. Department of Transportation, Federal Railroad Administration.

Cambridge Systematics, Inc., & Meyer, M. D. (2008). *Crashes vs. Congestion What's the Cost to Society?* Bethesda, MD: AAA.

Chan, A. B., & Vasconcelos, N. (2009). Bayesian Poisson Regression for Crowd Counting. *2009 IEEE 12th International Conference on Computer Vision.* Kyoto, Japan.

Chan, A. B., Liang, Z.-S. J., & Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition.* San Diego, CA.

Dabiri, S. (2019). *Application of Deep Learning in Intelligent Transportation Systems.* Blacksburg, VA: Virginia Polytechnic Institute and State University.

Eluru, N., Bagheri, M., Miranda-Moreno, L. F., & Fu, L. (2012). A latent class modeling approach for identifying vehicle driver injury severity factors at highway-railway crossings. *Accident Analysis & Prevention, 47*, 119–127.

Federal Railroad Administration. (2019). *Highway-Rail Grade Crossings Overview*. Retrieved from U.S. Department of Transportation, Federal Railroad Administration.

Federal Railroad Administration. (2020). *PTC Interoperability Status.* Retrieved from U.S. Department of Transportation, Federal Railroad Administration.

Ge, W., & Collins, R. T. (2009). Marked Point Processes for Crowd Counting. *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition.* University Park, PA: The Pennsylvania State University.

Gephardt, M., & Poe, M. (2018). *Trains blocking roadway creates dangerous situations, first responders say.* Retrieved from 2KUTV.

Gruden, C., Otković, I. I., & Šraml, M. (2020). Neural Networks Applied to Microsimulation:. *sustainability, 12*(13), 5355.

Haleem, K. (2016). Investigating risk factors of traffic casualties at private highway-railroad grade crossings in the United States. *Accident; analysis and prevention, 95 Pt A*, 274–283.

Hartong, M., Goel, R., Farkas, C., & Wijesekera, D. (2007). PTC-VANET Interactions to Prevent Highway Rail Intersection Crossing Accidents. *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, (pp. 2550–2554). Dublin, Ireland.

Hensher, D. A., & Mannering, F. L. (1994). Hazard-based duration models and their application to transport analysis. (1994). *Transport Reviews, 14*(1), 63–82.

Himsoon, T. K., & Siriwongpairat, W. P. (2011). Design and Analysis of 220 MHz RF Communications for Interoperable Positive Train Control System. *Proceedings of the ASME/IEEE Joint Rail Conference.*

Hoel, C.-J., Driggs-Campbell, K., Wolff, K., Laine, L., & Kochenderfer, M. J. (2019). Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, *5(2)*, pp. 294–305.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, *15(5)*, pp. 2191–2201.

Hwang, J.-J., Jung, Y.-H., Cho, B.-H., & Heo, M.-S. (2019). An overview of deep learning in the field of dentistry. *Imaging Science in Dentistry, 49*(1).

IEEE 802. (n.d.). *IEEE 802.15 WPAN™ Task Group 15.4p Positive Train Control*. Retrieved from IEEE 802.15.

Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., & Shao, L. (2019). Crowd counting and density estimation by trellis encoder-decoder networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

Jin, J., & Ma, X. (2019). A multi-objective agent-based control approach with application in intelligent traffice signal system. *IEEE Transactions on Intelligent Transportation Systems*, *20(10)*, pp. 3900–3912.

Khan, A. M. (2010). Prediction and display of delay at road border crossings. *The Open Transportation Journal, 4*(1).

Komatsu, T. S., & Sasa, S.-i. (1995). Kink soliton characterizing traffic congestion. *Physical Review E, 52*(5), 5574.

Kurniawan, J., Syahra, S. G., & Dewa, C. K. (2018). Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network. *Procedia computer science*, 291–297.

Leduc, G. (2008). *Road Traffic Data: Collection Methods and Applications. Working Papers on Energy, Transport and Climate Change.*

Lee, K., Hong, B., Jeong, D., & Lee, J. (2014). Congestion pattern model for predicting short-term traffic decongestion times. *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC).*

Lempitsky, V., & Zisserman, A. (2010). *Learning To Count Objects in Images.* Oxford, UK: Visual Geometry Group, University of Oxford.

Li, M., Zhang, Z., Huang, K., & Tan, T. (2008). *Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection.* National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Li., Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. *Proceedings of the IEEE conference computer vision and pattern recognition.*

Lin, Y., Dai, X., Li, L., & Wang, F.-Y. (2018). An Efficient Deep Reinforcement Learning Model for Urban Traffic Control. *arXiv preprint arXiv:1808.01876*.

Liu, C., Weng, X., & Mu, Y. (2019). Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1217–1226).

Liu, W., Salzmann, M., & Fua, P. (2019). Context-Aware Crowd Counting. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne (EPFL).

Lv, Y., Duan, Y., & Kang, W. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 865–873. Retrieved from

Ma, X., Yu, H., Wang, Y., & Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one, 10*(3), e0119044.

Marchesini, P., & Weijermars, W. A. M. (2010). *The relationship between road safety and congestion on motorways.* Netherlands: SWOV Institute for Road Safety Research Leidschendam.

McCollister, G., & Pflaum, C. (2007). A model to predict the probability of highway rail crossing accidents. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *221*, pp. 321–329.

Minge, E., Kotzenmacher, J., & Peterson, S. (2010). *Evaluation of Non- Intrusive Technologies for Traffic Detection.* Research Report No. MN/RC 2010-36, St. Paul, MN: Minnesota Department of Transportation.

Njus, E. (2019). *How long can trains block railroad crossings? (Commuting Q&A)*. Retrieved from The Oregonian/OregonLive.

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2019). Deep Learning vs. Traditional Computer Vision. *Proceedings of the Science and Information Conference.* Tralee, Ireland: IMaR Technology Gateway, Institute of Technology Tralee.

Ogden, B. D., & Cooper, C. (2019). *Highway-Rail Crossing Handbook, 3rd Edition.* Report No. FHWA-SA-18-040/FRA-RRS-18-001, Washington, DC: U.S. Department of Transportation, Federal Highway Administration, Office of Safety Design.

Ogden, B. D., Korve Engineering, a Division of DMJM + Harris. (2007). *Railroad-Highway Grade Crossing Handbook - Revised Second.* Report No. FHWA-SA-07-010, Washington, DC: U.S. Department of Transportation, Federal Highway Administration, Office of Safety Design.

Paselk, T. A., & Mannering, F. L. (1994). Use of duration models for predicting vehicular delay at a US/Canadian border crossing. *Transportation, 21*(3), 249–270.

Polson, N. G., & Sokolov, V. O. (2017). Deep Learning for Short-Term Traffic Flow Prediction. *Transportation Research Part C: Emerging Technologies, 79*, 1–17.

Ranjan, V., Le, H., & Hoai, M. (2018). Iterative Crowd Counting. *Proceedings of the European Conference on Computer Vision (ECCV).* Department of Computer Science, Stonybrook University.

Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M.-M., & Zheng, G. (n.d.). Crowd Counting with Deep Negative Correlation Learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (p. 2018).

U.S. Department of Transportation. (2020a). *Tracking Toward Zero: Improving Grade Crossing Safety and Addressing Community Concerns.* Retrieved from U.S. Department of Transportation.

U.S. Department of Transportation. (2020b). *Tracking Toward Zero: Improving Grade Crossing Safety and Addressing Community Concerns.* Retrieved from U.S. Department of Transportation.

Veres, M., & Moussa, M. (2019). Deep learning for intelligent transportation systems: A survey of emerging trends. *IEEE Transactions on Intelligent Transportation Systems*, *PP(99)*, pp. 1–17.

Walach, E., & Wolf, L. (2016). *Learning to Count with CNN Boosting.* Tel Aviv, Israel: The Blavatnik School of Computer Science, Tel Aviv University.

Wan, J., Yuan, Y., & Wang, Q. (2017). Traffic congestion analysis: a new perspective. *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Shaanxi, China: School of Computer Science and Center for

OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University.

Wang, C. (2010). *The Relationship Between Traffic Congestion and Road Accidents: An Econometric Approach Using GIS.* Loughborough, England: Loughborough University.

Wang, J., Gu, Q., Wu, J., Liu, G., Xiong, Z. (2016). Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, (pp. 499–508). Barcelona, Spain.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems, 48*, 144–156.

Wang, Q., Wan, J., & Yuan, Y. (2018). Locality Constraint Distance Metric Learning for Traffic Congestion Detection. *Pattern Recognition, 75*, 272–281.

Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., & Bayen, A. M. (2017). Flow: A Modular Learning Framework for Mixed Autonomy Traffic. *arXiv preprint arXiv:1710.05465*.

Xiong, F., Shi, X., & Yeung, D.-Y. (2017). Spatiotemporal Modeling for Crowd Counting in Videos. *Proceedings of the IEEE International Conference on Computer Vision.* Hong Kong: Department of Computer Science and Engineering, Hong Kong University of Science and Technology.

Yan, X., Richards, S., & Su, X. (2010). Using hierarchical tree-based regression model to predict train-vehicle crashes at passive highway-rail grade crossings. *Accident Analysis & Prevention*, 64–74.

Yi, H., Jung, H., & Bae, S. (2017). Deep neural networks for traffic flow prediction. *Proceedings of the 2017 IEEE international conference on big data and smart computing (BigComp).*

Zhang, S., Yao, Y., Hu, J., Zhao, Y., Li, S., & Hu, J. (2019). Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks. *Sensors, 19*(10), 2229.

Zhang, Z., Liu, X., & Holt, K. (2018). Positive Train Control (PTC) for railway safety in the United States: Policy developments and critical issues. *Utilities Policy, 51*, 33–40.

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term forecast. *IET Intelligent Transport Systems, 11*(2), 68–75.

Appendix A.
Survey Questionnaire To First Responders

Part I: Background information

1. Which department are you serving?

Police ☐
Fire ☐
EMD ☐
Other ☐

If "other," please specify: _____

2. How long have you served?

Less than 6 months ☐
6 months to 1 year ☐
1 years to 2 years ☐
2 years to 3 years ☐
3 years to 5 years ☐
5 years to 10 years ☐
more than 10 years ☐

3. Have you ever experienced congestion at the railroad crossings due to trains?

Yes ☐
No (skip 3-8) ☐

4. How many times on a monthly base?

Less than 5 ☐
5-10 ☐
10-20 ☐
20-30 ☐
More than 30 ☐

5. Typically, how long did you have to wait on average?

Less than 5 mins ☐
5-10 mins ☐
10-20 mins ☐
20-30 mins ☐

6. The longest time you had to wait:

_____mins

7. Do you think congestions at railroad crossing is a problem?

Yes ☐

No ☐

8. What would you do while experiencing congestion at grade crossing?

Make a U-turn ☐

Wait the clearance ☐

Don't know what to do ☐

Part II: Please answer question 1-15 based on the experience <u>WHILE YOU ARE ON DUTY</u>

1. Have you ever experienced congestion at the railroad crossings WHILE ON DUTY?

Yes ☐

No (skip 2-4) ☐

2. Approximately how many times?

Less than 5 ☐

5-10 ☐

10-20 ☐

20-30 ☐

More than 30 ☐

3. Typically, how long did you have to wait on the average?

Less than 5 mins ☐

5-10 mins ☐

10-20 mins ☐

20-30 mins ☐

4. The longest time you had to wait:

___mins

5. Do you think congestions at railroad crossing is a problem for first responders?

Yes □

No □

6. Do you have any way to know the expected delay time due to moving or stopped trains?

Yes □

No □

If Yes, please specify: _____

_____

7. What would you do while you are waiting at grade crossing?

Make a U-turn □

Wait for clearance □

Don't know what to do□

8. Which crossing gave you the most trouble (please give the street name)

_____

No → END

9. Do you think it would be helpful to provide first responders the potential crossing congestion information while they respond to an emergency?

Yes □

No □

10. Would you like to receive the information of potential crossing congestion from the dispatching center while responding to an emergency?

Yes □

No □

11. What specific information do you want to receive from the dispatching center?

_____

_____

_____

12. Would you like to install an app on your phone that can automatically send information of potential crossing congestion while you are on duty?

Yes ☐

No ☐

13. Do you think it would be helpful to also notify the public of the crossing congestion condition at the crossing during your response to an emergency?

Yes ☐

No ☐

14. What specific features do you want to have in this app?

_____

_____

15. Do you have any comments/suggestions on the development of real time traffic information sharing system for grade crossing?

_____

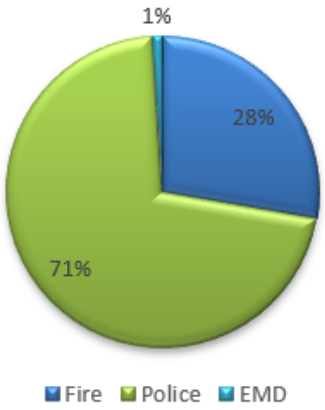
_____

_____

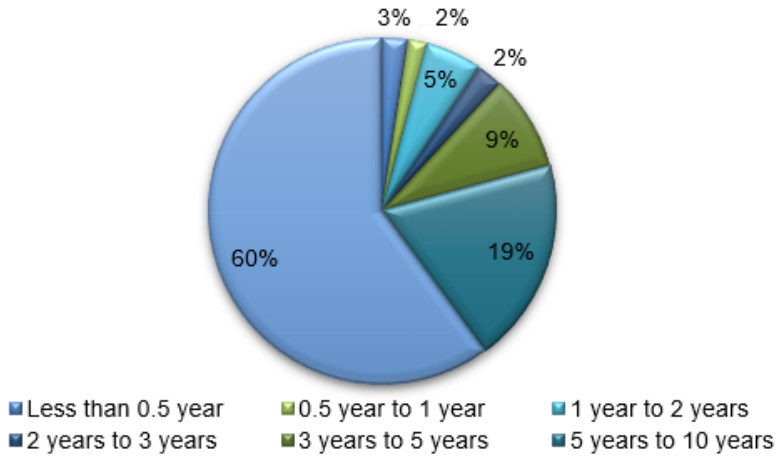
_____

Appendix B.
Survey Results

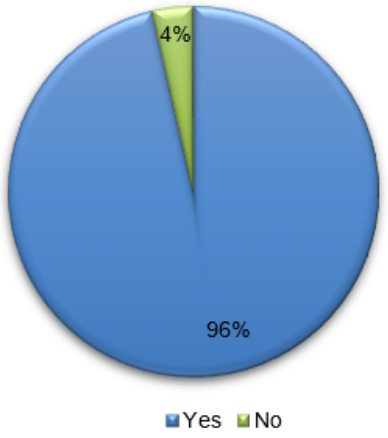
Q1	
Fire	27.72%
Police	71.29%
EMD	0.99%
	100.00%



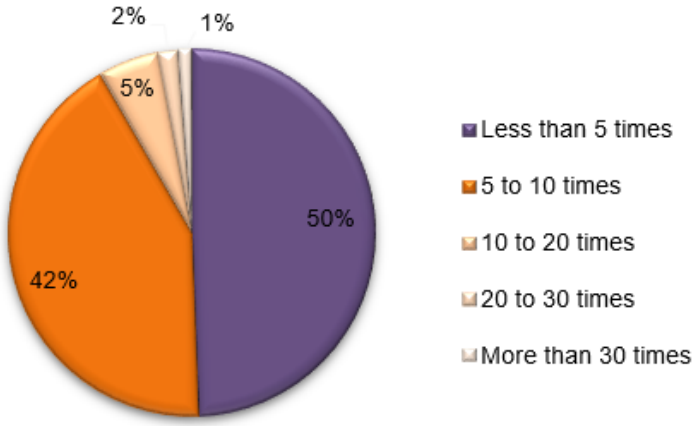
Q2	
Less than 0.5 year	2.38%
0.5 year to 1 year	1.79%
1 year to 2 years	5.36%
2 years to 3 years	2.38%
3 years to 5 years	8.93%
5 years to 10 years	19.05%
More than 10 years	60.12%
	100.00%



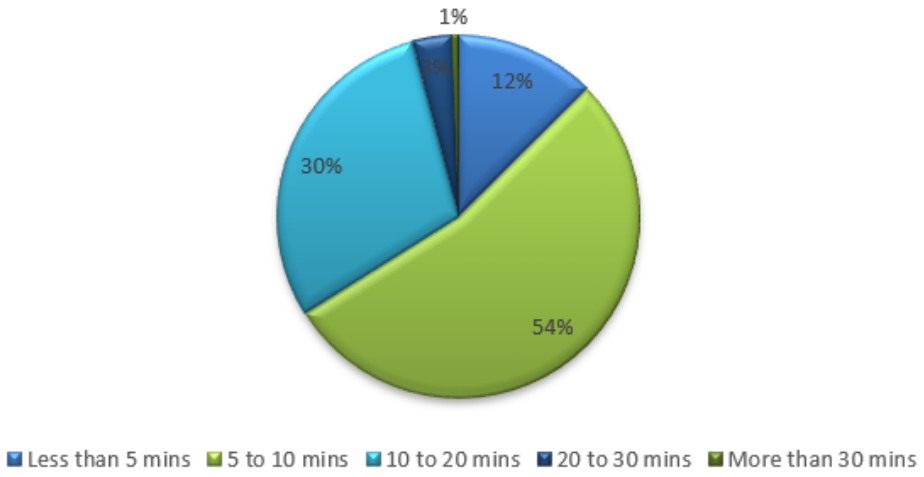
Q3	
Yes	96.43%
No	3.57%
	100.00%



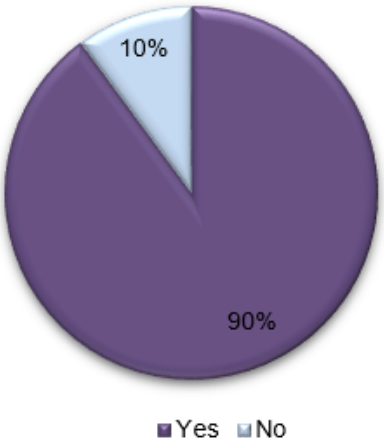
Q4	
Less than 5 times	49.40%
5 to 10 times	42.26%
10 to 20 times	5.36%
20 to 30 times	1.79%
More than 30 times	1.19%
	100.00%



Q5	
Less than 5 mins	12.50%
5 to 10 mins	53.57%
10 to 20 mins	29.76%
20 to 30 mins	3.57%
More than 30 mins	0.60%
	100.00%

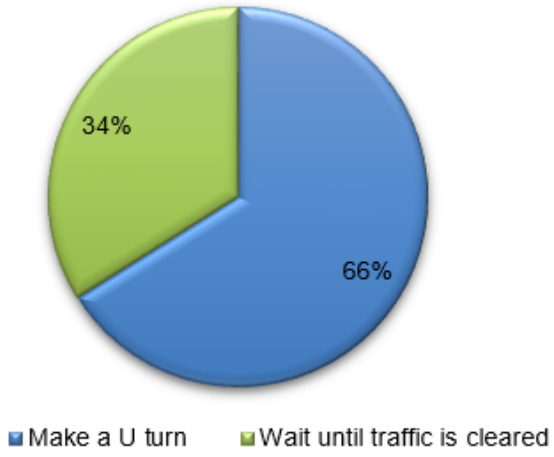


Q7	
Yes	89.88%
No	10.12%
	100.00%



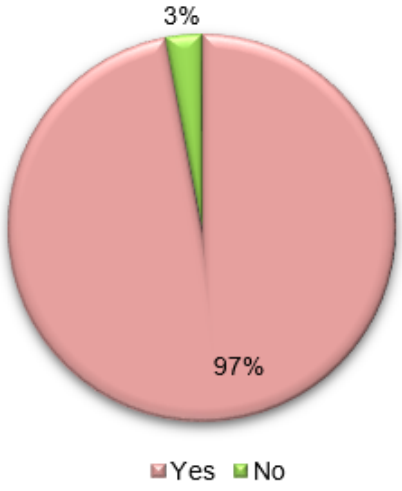
Q8

Make a U turn	66.07%
Wait until traffic is cleared	33.93%
	100.00%

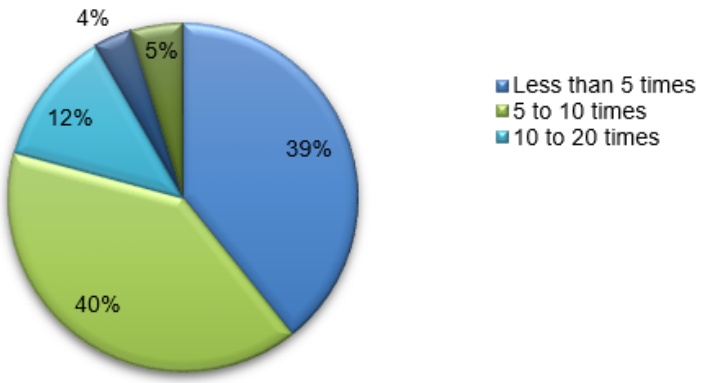


Q9

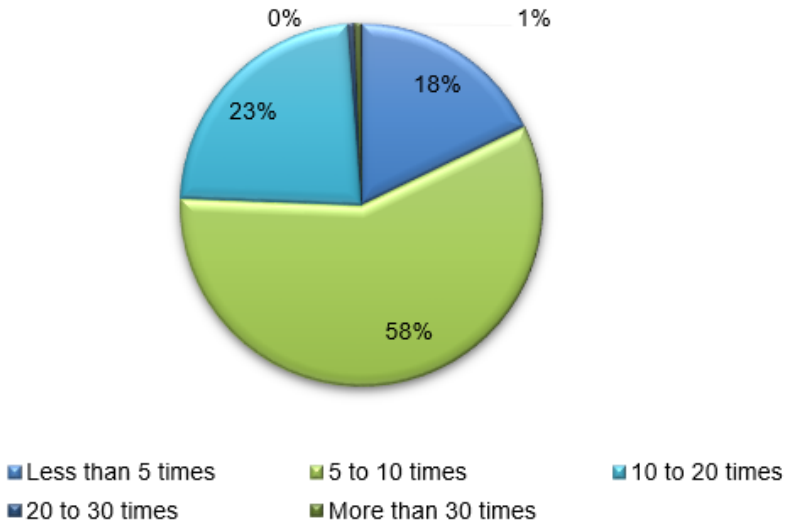
Yes	97.02%
No	2.98%
	100.00%



Q10	
Less than 5 times	39.29%
5 to 10 times	39.88%
10 to 20 times	12.50%
20 to 30 times	3.57%
More than 30 times	4.76%
	100.00%

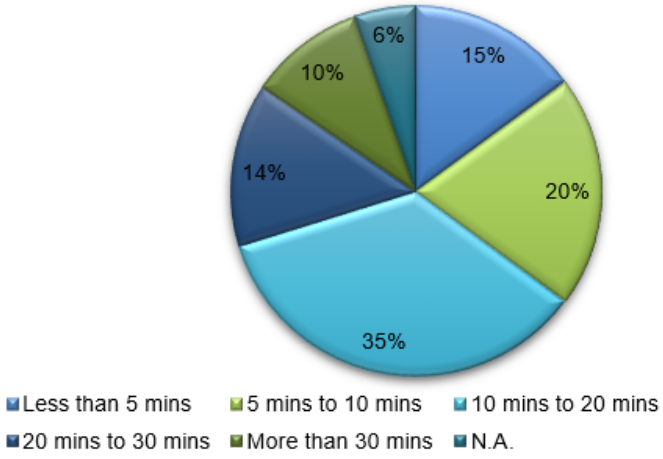


Q11	
Less than 5 times	17.86%
5 to 10 times	57.74%
10 to 20 times	23.21%
20 to 30 times	0.60%
More than 30 times	0.60%
	100.00%



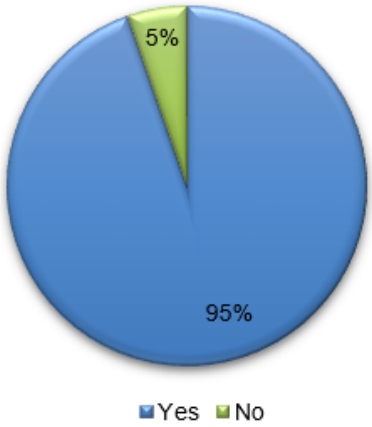
Q12

Less than 5 mins	14.88%
5 mins to 10 mins	20.24%
10 mins to 20 mins	35.12%
20 mins to 30 mins	14.29%
More than 30 mins	10.12%
N.A.	5.36%
	100.00%

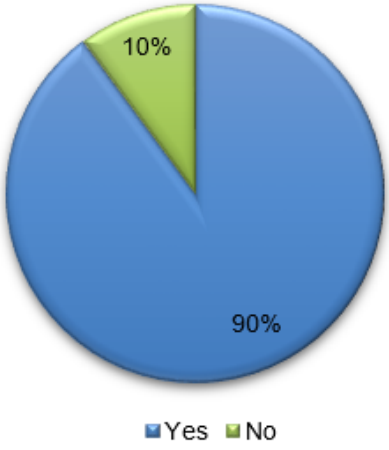


Q13

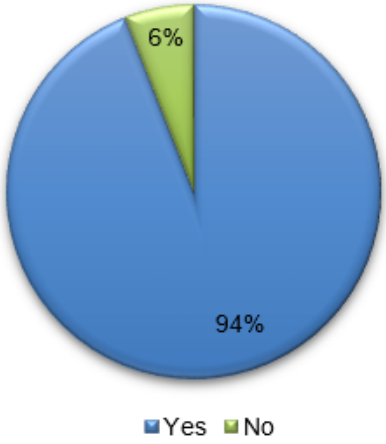
Yes	94.64%
No	5.36%
	100.00%



Q14	
Yes	89.88%
No	10.12%
	100.00%

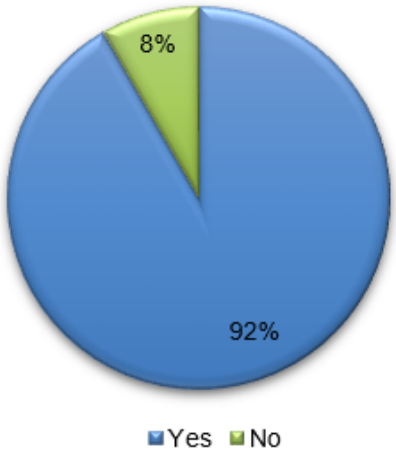


Q17	
Yes	94.05%
No	5.95%
	100.00%



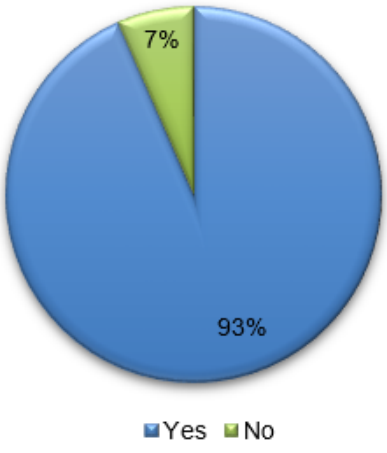
Q18

Yes	91.67%
No	8.33%
	100.00%

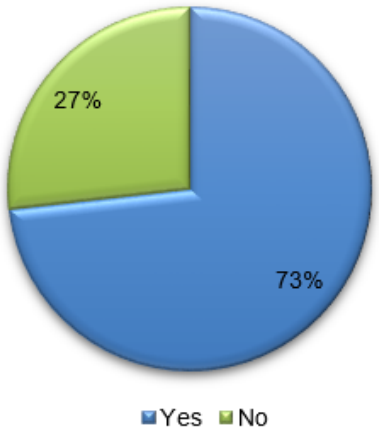


Q20

Yes	93.45%
No	6.55%
	100.00%



Q21	
Yes	73.21%
No	26.79%
	100.00%



Abbreviations and Acronyms

ACRONYMS	EXPLANATION
AI	Artificial Intelligence
ANN	Artificial Neutral Networks
ATR	Automatic Target Recognition
CBM	Condition-based Maintenance
CNN	Convolutional Neural Network
CV	Computer Vision
CSX	CSX Transportation, Inc.
DBN	Deep Brief Network
ePRO	Electronic Police Reports Online
EMD	Emergency Medical Department
ETA	Estimated Time of Arrival
ETC	Estimated Time of Clearance
ETD	Estimated Time of Decongestion
ETT	Estimated Total Time
ETP	Estimated Time of Passing
FRA	Federal Railroad Administration
GIS	Geological Information System
GPS	Global Positioning System
GT	Ground Truth
GUI	Graphic User Interface
i-CARES	Intelligent Camera Aided Railway Emergency System
i-CATSS	Intelligent Crossing Assessment and Traffic Sharing System
LSTM	Long-term Short-term Memory
MAE	Mean Absolute Error
MTL	Multiple-task Learning
NNTC	Non-normal Traffic Condition
NTC	Normal Traffic Condition
PSNR	Peak-signal-to-noise Ratio
PTC	Positive Train Control
RSIA	Rail Safety Improvement Act of 2008

ACRONYMS	EXPLANATION
RNN	Recurrent Neutral Network
ROW	Rights-of-Way
USDOT	United States Department of Transportation