ITRE
Institute for Transportation Research and Education

Pedestrian Incident Detection in the Rail Right-of-Way using Artificial Intelligence

Prepared By
Institute for Transportation Research and Education (ITRE) @ North Carolina State University

Principal Investigator
Chris Cunningham, MSCE, PE
NCSU Campus Box 8601
Raleigh, NC 27695-8601
cmcunnin@ncsu.edu
Tel. 919-515-8562, Fax 919-515-8898

Team Members:
Sarah Searcy
Azhagan Avr
Chris Vaughan
Daniel Coble

Prepared For
North Carolina Department of Transportation (NCDOT)

Submission Date
July 12, 2019

RESEARCH & DEVELOPMENT

Pedestrian Incident Detection in the Rail Right-of-Way using Artificial Intelligence

Chris Cunningham - PI
Azhagan Avr
Sarah Searcy
Chris Vaughan, PE
Daniel Coble

Institute for Transportation Research and Education (ITRE)
North Carolina State University

NCDOT Project 2019-50
FHWA/NC/2019-50
July 2019

TECHNICAL DOCUMENTATIONS PAGE

Report No. **FHWA/NC/2019-50**	Government Accession No.	Recipient's Catalog No.
Title and Subtitle **Pedestrian Incident Detection in the Rail Right-of-Way using Artificial Intelligence**		Report Date **July 12, 2019** Performing Organization Code
Author(s) **Chris Cunningham, Azhagan Avr, Sarah Searcy, Chris Vaughan, Daniel Coble,**		Performing Organization Report No.
Performing Organization Name and Address **Institute for Transportation Research and Education** **North Carolina State University** **Centennial Campus Box 8601** **Raleigh, NC**		Work Unit No. (TRAIS) Contract or Grant No.
Sponsoring Agency Name and Address **North Carolina Department of Transportation** **Research and Development Unit** **1020 Birch Ridge Dr.** **Raleigh, North Carolina 27610**		Type of Report and Period Covered **Final Report** **January 2019 to June 2019** Sponsoring Agency Code **2019-50**
Supplementary Notes:		

Abstract

This research project builds on NCDOT RP 2015-18 ("Reduction in Railroad Right-of-Way Incidents") and 2017-15 ("Rail Corridor Trespass Severity Assessment"). This effort focused on the development of a working prototype train-mounted camera system that will capture trespassing events in the nearby vicinity of moving or stopped trains. This dynamic system captures real-time trespassing data along any rail line, which will be used to better define trespassing issues. In the short term, the tools explored as part of this project will allow rail personnel to explain the extent of trespassing to municipal and law enforcement personnel, as well as the public.

Prototype machine learning algorithms, sometimes referred to as "artificial intelligence", were developed as a part of this project. The algorithms developed showed a lot of promise, even with a very limited library of thermal imagery in its database. Future research efforts should look to increase the image database to continue to increase the confidence in the algorithms ability to capture pedestrian events. Even with such a limited database, the team was able to capture a significant number of events on its test track.

Key Words **Railroad Trespassing, Pedestrians, Safety**	Distribution Statement

Security Classif. (of this report) Unclassified	Security Classif. (of this page) Unclassified	No. of Pages 18	Price

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

DISCLAIMER

The contents of this report reflect the views of the authors and not necessarily the views of the Institute for Transportation Research and Education or North Carolina State University. The authors are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the North Carolina Department of Transportation or the Federal Highway Administration at the time of publication. This report does not constitute a standard, specification, or regulation.

Table of Contents

INTRODUCTION AND RESEARCH NEED DEFINITION

Trespassing is the leading cause of rail-related deaths in the United States. In 2017, the Federal Rail Administration (FRA) held a Grade Crossing Research Needs Workshop that established five research needs focus areas.[1] The top recommended action for the Community Outreach and Education focus area is trespasser identification, motivation, and messaging.[2] The goal of this action is to provide communities with tools for deterring trespassing, including better targeting of messaging based on demographics, geography, and reasons for trespassing. Achievement of this goal requires identifying types of and reasons for trespassing along with developing modes and methods to test messaging aimed at trespassers. Limited previous research has been conducted that successfully deploys machine learning (a.k.a. "artificial intelligence", or "A.I.") to gather information about the number of unsafe trespassing events (i.e. near misses). This key data point provides a much clearer picture of the actual problem than train-trespasser strikes alone – a number that is low in comparison to crash statistics along highways for pedestrians and motorists. In addition, the data on strike events is often inconclusive in determining the causation of the strike in the first place. Supplementing actual strike events with actual near-miss events in real-time provides a resolution not possible currently for rail safety proponents who need physical data to explain the extent of the rail trespassing problem.

The Institute for Transportation Research and Education (ITRE) is involved in ongoing research that seeks to develop a more complete understanding of the extent of trespassing within the 174-mile Piedmont corridor from Raleigh to Charlotte, NC. A major component of the research involves thermal video data collection at a sample of trespassing hot spots along the corridor (**Figure 1**).

Figure 1. Individual playing guitar on the tracks and recording their performance in Greensboro, NC (left); large groups of people crossing the tracks near Elon University, NC (right)

[1] Alibrahim, Sam. (2017). **FRA Grade Crossing Safety Research.** Retrieved from https://www.fra.dot.gov/conference/2017/rnw/pdf/Presentations/Other%20Sessions/FRA%20Accomplishments.pdf.

[2] Federal Rail Administration. (2017). **Working Group Summary of Top Recommended Actions.** Retrieved from https://www.fra.dot.gov/conference/2017/rnw/pdf/Presentations/Other%20Sessions/Working%20group%20Summaries.pdf.

Data that greatly enhances the actual strike data is a key piece of information currently missing from ITRE's ongoing research effort. This research project will help fill this need by closing the gap with parallel efforts already underway – giving a more complete picture of the extent of the trespassing problem and how agencies should consider dealing with this issue through engineering, educating, and/or enforcement. This data, supplemented with contextual data at the case study level, can be used by the FRA, local communities, and the NCDOT to construct a better picture of trespassing activities that do not result in injury or loss of life. In turn, these data can inform practical safety initiatives and countermeasures to reduce overall trespassing events.

Current research at both the state and national level has sought to determine the causation of pedestrian-train strikes to other factors with very little success. Recent research by ITRE looked at census data paired with FRA-reported strikes and train crew surveys. While this database helps identify hotspots, models are still limited in their ability to accurately predict where near-strike events are taking place along the corridor. This project builds on current research efforts already underway by this team to help predict the extent of the rail trespassing problem to inform appropriate countermeasures and safety initiatives at the municipal, state, or national level. This includes, but is not limited to, working with municipal and law enforcement personnel to put in practice targeted education campaigns, installing countermeasures to deter trespassing, and even explore potential uses of detection technology in the future to better inform rail personnel prior to a possible strike.

In addition to safety, maintenance and operational concerns are very real when trains must apply emergency brakes or when the train strikes a trespasser. Trains that strike a pedestrian must wait many hours to file the necessary reports required before allowing the train to continue operation. Detour routes do not exist for trains like they do for vehicles, making delays very costly. In addition, maintenance of train wheels and strike plates is very costly and time consuming.

In the not-so-distant future, technologies such as the ones proposed in this proposal, offer significant promise for saving lives as well as money at a price-point that those in the rail industry can likely accept should they choose to adopt such technologies. These technologies are already being deployed by other industries such as car manufacturers instrumenting connected and autonomous vehicles. Understanding the capabilities of similar systems and how they can improve safety and drive down cost is a worthwhile cause that should pay dividends in the not-so-distant future.

RESEARCH OBJECTIVES

The objective of this research effort is to develop a working prototype train-mounted camera system that will capture trespassing events in the nearby vicinity of moving or stopped trains. This dynamic system will capture real-time trespassing data along any rail line, which will be used to better define trespassing issues. In the short term, the tools explored as part of this project will allow rail personnel to explain the extent of trespassing to municipal and law enforcement personnel, as well as the public.

HARDWARE CONSIDERATIONS AND TESTING

The team researched and studied about the use of a dynamic camera system that can be installed on a train in push-pull operation for front view detection of trespassing events. The research involved two different setups to collect video data from the locomotive. The first setup consisted of a standard digital camera, while the second setup involved a thermal camera. Following are the details of the hardware involved in these setups.

Setup 1. Full System Configuration
The system concept, shown in Figure 2, is composed of a Raspberry Pi that allows several plug-in devices to provide or store data.

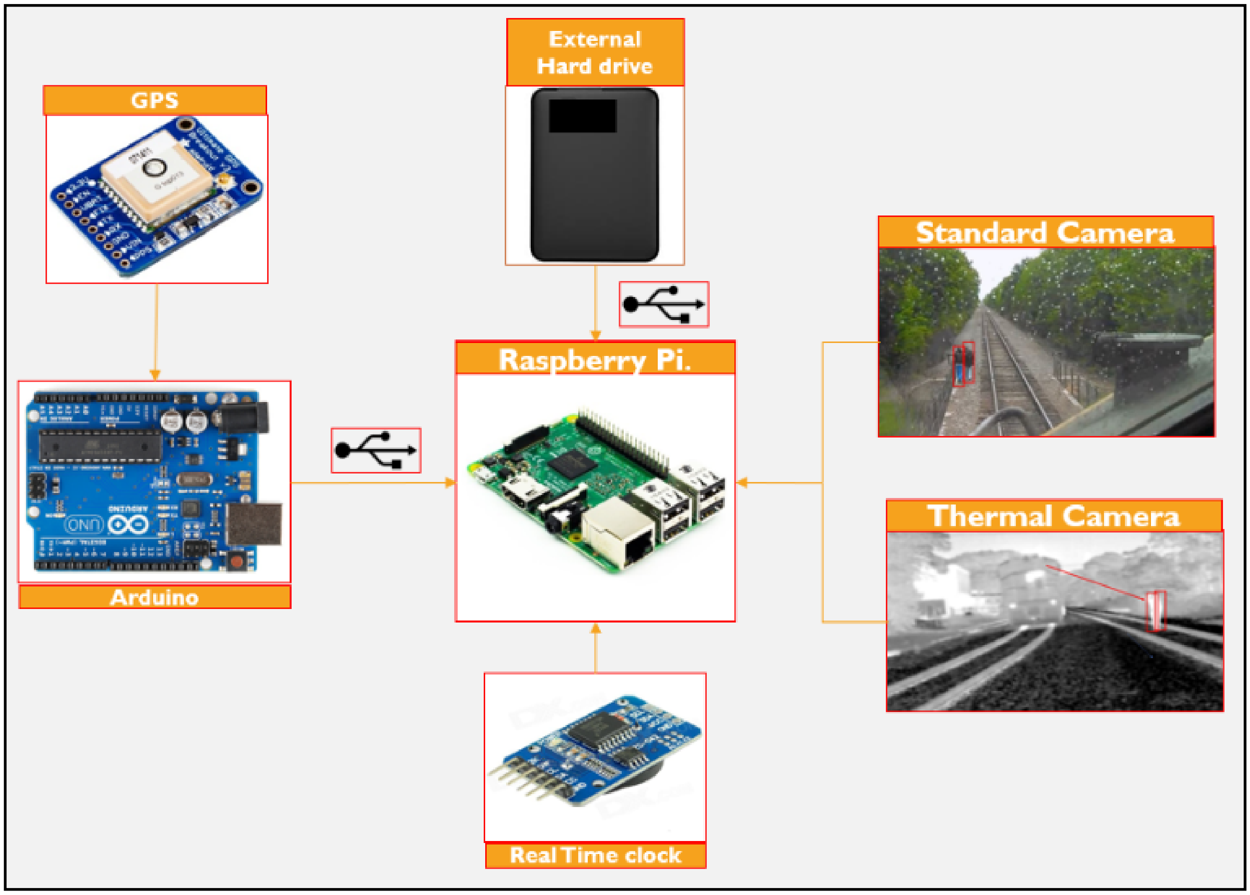

Figure 2. Dynamic camera system components

The Pi is a small microcomputer, approximately 4"x 4" x 1," which can be easily housed in a small storage container and powered by AC or DC power. A thermal and standard camera are plugged into the device to provide video inputs, and GPS and digital timestamp overlays are applied to the video to provide spatial and temporal data for future analysis. A hard drive is utilized for external storage in our current test that will allow our team to post-process video using various prototype algorithms; however, it is possible that data will be stored in a cloud-based system at the end of this project. An Arduino is used to transfer the GPS signal into usable data that is then overlaid onto the video. The various hardware components are described in more detail in Table 1.

Table 1. System component descriptions

Equipment	Specifications	Description
Raspberry Pi	• Quad Core 1.2 GHz processor with 1GB RAM • Wireless LAN and Bluetooth Low Energy (BLE) on board • 4 USB 2.0 ports, Ethernet and an HDMI port • Memory size depends on SD card • Can be powered by DC or AC power	• It is a mini computer • It can perform parallel processing • Multiple computer peripherals can be connected such as keyboard, mouse, monitor, etc. • The Pi receives the GPS data through the Arduino and the video data from the camera and stores them in the hard drive
Arduino	• 14 digital I/O pins and 6 Analog I/O pins • Memory size is 32KB • Operates at 5y and powered by Pi	• Different sensors can be connected to these digital and analog I/O pins • For example, the GPS mobile is connected to the digital pins
GPS	• 10 samples per second (10hz) • Provides date, time, latitude, longitude, speed in real time • Connects directly to satellites	• The GPS module connects to the Arduino • Used for determining the location of the train and trespassing event • The GPS data is overlaid on the video in real time
Real Time Clock	• Powered by 3V button cell • It is a timekeeper	• Used to keep track of the Pi time when it is switched off • Time and date is also overlaid on the video real-time
Hard Disk	• Memory size is 1TB	• Connected to the Pi to store video
Camera	• 8MP camera • Wide viewing angle • Maximum resolution of 1080p	• Used to capture trespassing events

At this time, our research team has deployed a prototype system several times from the cab of a passenger vehicle for pilot testing purposes. The system components have linked together nicely and are very reasonably priced (total less than $500). The biggest problem with the standard camera is that, it has poor visibility in the dark, despite the headlights of the locomotive. Hence, we decide to change to a thermal camera, which can capture during the day and during the night, even at the darkest of situations.

Setup 2. Thermal Camera Considerations

The variation in thermal camera cost currently ranges from $50-$5,000. The research team recognizes cost is a substantial factor in determining how deployable such a device would be; however, for the purposes of prototyping, we do not want to limit our ability to detect pedestrians in any way using machine learning and recognize that cost ultimately will be driven down as the technologies continue to improve. For the purposes of prototyping and proof-of-concept, we recently moved forward with a higher-end thermal camera arrangement that will not limit the team in field-of-view, will not limit pixel quantity, and will not cost the team significant time. For our purposed, we used the Axis Q1941-E thermal imaging camera with two options – 15mm and 25 mm lenses. In the end, the wider angle 15mm lens was preferred as they can capture the shoulders of the tracks comfortably. Below is an image of the thermal camera that was ultimately used for deploying machine learning algorithms in the following sections.

Figure 3. Axis Q1941-E thermal imaging camera

MACHINE LEARNING/ALGORITHM DEVELOPMENT

Pedestrian detection involves four primary thrusts. The first requires understanding the problem and writing the code accordingly to solve the problem. The second requires acquiring a training database of images of interest. For this project, the primary focus is on pedestrians; however, it could involve other objects such as bicycles, cars and so on. Currently, the most widely used image database is the Caltech Pedestrian detection database, but it is used for digital color images. For our purposes, this database was not useful as we require thermal image in gray scale. Therefore, we must create our own database. The third thrust requires us to train the pedestrian detection algorithm (i.e. "machine learning") with the image training database. The final thrust requires that the algorithm classify whether the extracted features are a pedestrian or not based on the database.

Thrust 1. Algorithm Development

Before actual pedestrian events could be tested, an experiment protocol and test track were needed. The test track that was utilized was located in Star, NC where the research team was able to utilize the resources of Aberdeen Carolina and Western Railway Company (ACWRC). A representative from the ACWRC provided a hi-rail (short for highway and railroad) maintenance vehicle which was capable of driving on the road and railroad tracks. The research team deployed several thermal and video camera setups

Figure 4 shows the thermal camera mounted to the rail maintenance truck.

on top of this vehicle facing forward and backward to capture staged pedestrian events in the front and rear of the train as it approached.

With the vehicle and test bed acquired, the team defined an experiment protocol which staged four pedestrians at four separate locations of a half mile section of track. Next, our team performed specific predefined trespassing scenarios, using four pedestrians (A, B, C, D) as shown in Figures 5 and 6. Pedestrians A and B were purposely staged at curves in two varying directions and pedestrian C and D were in straight sections. The vehicle went back and forth capturing various trespassing events perpendicular and parallel to the tracks, each performed at distances of 200ft and 600ft from an approaching vehicle.

Subject	Location Type	Start Direction	Cross Track (ft)
A	Curve	Perpendicular	600
B	Curve	Perpendicular	600
C	Straight	Perpendicular	600
D	Straight	Perpendicular	600

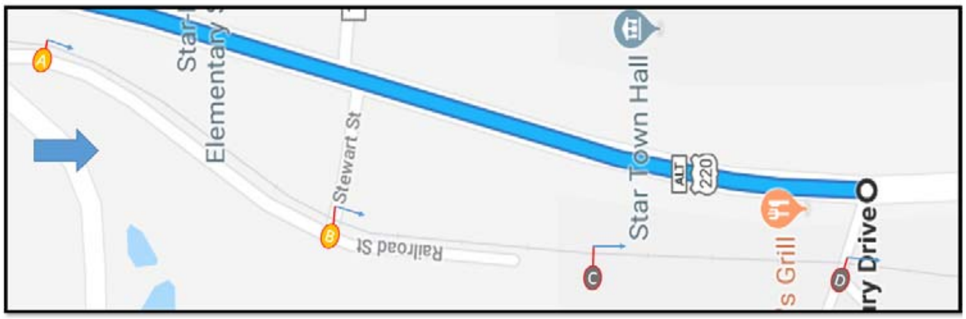

Figure 5. Hi-rail vehicle moving left to right and pedestrians A through D making perpendicular movements 600' in front of the vehicle.

Subject	Location Type	Start Direction	Cross Track (ft)
D	Straight	Perpendicular	200
C	Straight	Perpendicular	200
B	Curve	Perpendicular	200
A	Curve	Perpendicular	200

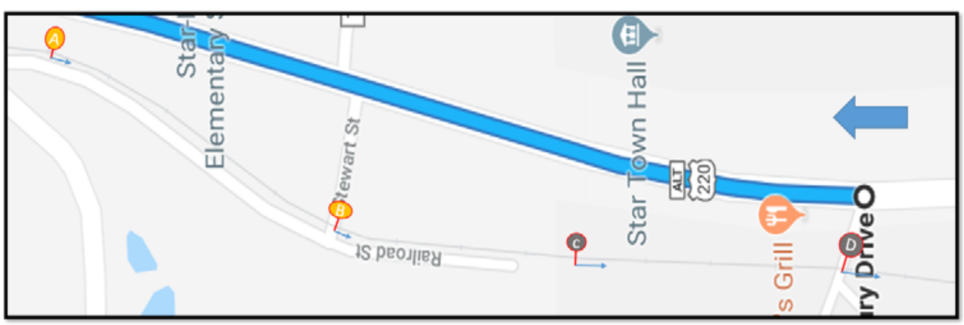

Figure 6. Hi-rail vehicle moving right to left and pedestrians A through D making perpendicular movements 200' in front of the vehicle.

Once video of each trespassing event was captured and stored, the team began the development of code to perform the detection task. To detect pedestrians accurately, our team decided on a region of interest (ROI) that would avoid any unnecessary false detection of pedestrians. This region of interest, shown in Figure 7, was tested at a height that would attempt to capture

pedestrians at 200 and 600 feet. This provided an opportunity to test false detections the further the pedestrian was from the train as well as determine how far away features could be detected at the resolution used in our video during the experiment (480x 320). This resolution was chosen considering the tradeoff between file size and the quality of the image; however, dependent on the findings from the experiment, it could be increased or decreased. As an example, an ROI which captures a pedestrian at 200 feet (but not 600 feet) is shown below in Figure 7.

Having established the ROI, coding was done in three main areas to capture the events of interest. First the code had to initialize so that the data could be captured based on desired ranges including where the ROI was drawn, output filename, confidence levels to be used to determine a positive event capture, how many frames before and after event that are kept, etc. The second section of code imports the video and splits it into frames which are used by the machine learning algorithm. Last, the detection algorithm performs detection on each frame and marks the pedestrian it has identified based on the ROI and confidence interval set in the first stage. The final output file used by the team captures the entire scene around the flagged pedestrian event, places a yellow box around the pedestrian, and displays the confidence level for each detected pedestrian. For example, if the if 65% is displayed, the algorithm considers the object to match 65% of the image features in the database. A well-trained machine learning algorithm will have a confidence level in the range of 90+%. Hence it is important to create a database consisting of a wide variety of pedestrian images.

Thrust 2. Creating the Database

To create the database, we needed a significant number of pedestrian thermal images. Through a parallel research effort with NCDOT, we had collected many days of static video at known hot spot locations which had a variety of rail trespassing events. An example from Elon, NC is shown below in Figure 8 from two separate trespassing events.

Figure 8. Images from the static database that show pedestrian trespassing.

After several iterations of testing the algorithm with static images, we determined that the static video data was not suitable for dynamic object detection. The primary reason was that the pedestrian images used to form the database had fewer dominant features (they were far away from the camera) and were more like stick figures with no detail. Therefore, we chose to collect imagery from a moving vehicle along Hillsborough Street adjacent to NC State University's main campus. Our research team mounted the camera on a car and made several trips back and forth on Hillsborough street where the average pedestrian volume is high throughout the day due to the presence of students. A total of five hours of pedestrian data were collected during differing times of day between 07:30 to 09:30, 12:30 to 14:00 and 15:30 to 17:00. Examples of imagery collected from the drive are provided in Figure 9.

Figure 9. Images from the dynamic database that show pedestrian data.

The collected videos are then reduced to images, which forms the database. Only images with pedestrians are chosen.

Thrust 3. Training the Algorithm

The third thrust requires training the algorithm. MATLAB has an inbuilt algorithm called Aggregate Channel Features (ACF) object detector. The algorithm processes the images from the database. While training the algorithm learns to identify the positive instances of objects (pedestrians) in images and negative instances of objects (trees, cars, lamp posts, etc.) in the images during training. Once the algorithm is trained a variable or parameter is created this parameter can be saved in ". mat" file format through MATLAB. Once the ". mat" file is created the database is not required, as all the necessary information will be stored in the file. Later the information in this file is used to detect pedestrians. For more details on the object detector use the following link; https://in.mathworks.com/help/vision/ref/trainacfobjectdetector.html.

Thrust 4. Providing Individual Event Output Files

Once the ". mat" file is created based on the imagery database, a code was written to import videos previously collected from the high rail setup in Star, NC in Thrust 1. The algorithm was overlaid onto this previously retrieved video to detect rail trespassers. Figure 10 shows a snapshot from the recorded video which is captured as a video output that consists of only 50 frames before and after a detection. You can see that the pedestrian, although not marked, is captured on the left side of the image. This video is manually checked to determine if it was a true or false positive and recorded in a database by the research team for future analysis.

Figure 10. Forward facing view of the thermal camera in Star, NC (full video found at https://www.youtube.com/watch?v=qZqtBTqH1pE)

TESTING THE ALGORITHM

Once the development of the algorithm and outputs were in place, the research team began testing the algorithm to see where improvements needed to be made. This was done through a series of pilot test using the Star, NC video. It is important to note that the database created in Thrust 2 was not created from any of the video collected on this corridor. In the development stages of the machine learning algorithm, the video from Star, NC was only used for testing the output files in Thrust 4. An explanation of the pilot testing completed-to-date is provided below.

Pilot Test 1. Using the Static Camera Database

As noted earlier in Thrust 2, we originally used the static video imagery to create the initial database since our team had a significant amount of static video data. Based on this static imagery, the algorithm was used to create the trained object (Thrust 3). Once the object was created, the dynamic video from the Star, NC camera deployment was used to test (i.e. Thrust 4) if the algorithm had a sufficient database to detect objects (i.e. pedestrian trespassers). The initial results were poor to say the least. As shown in Figure 11, many false positive detections were captured which had a significantly high confidence level (i.e. non-pedestrian objects were given high scores). In layman terms, objects which are not pedestrians were classified as pedestrians.

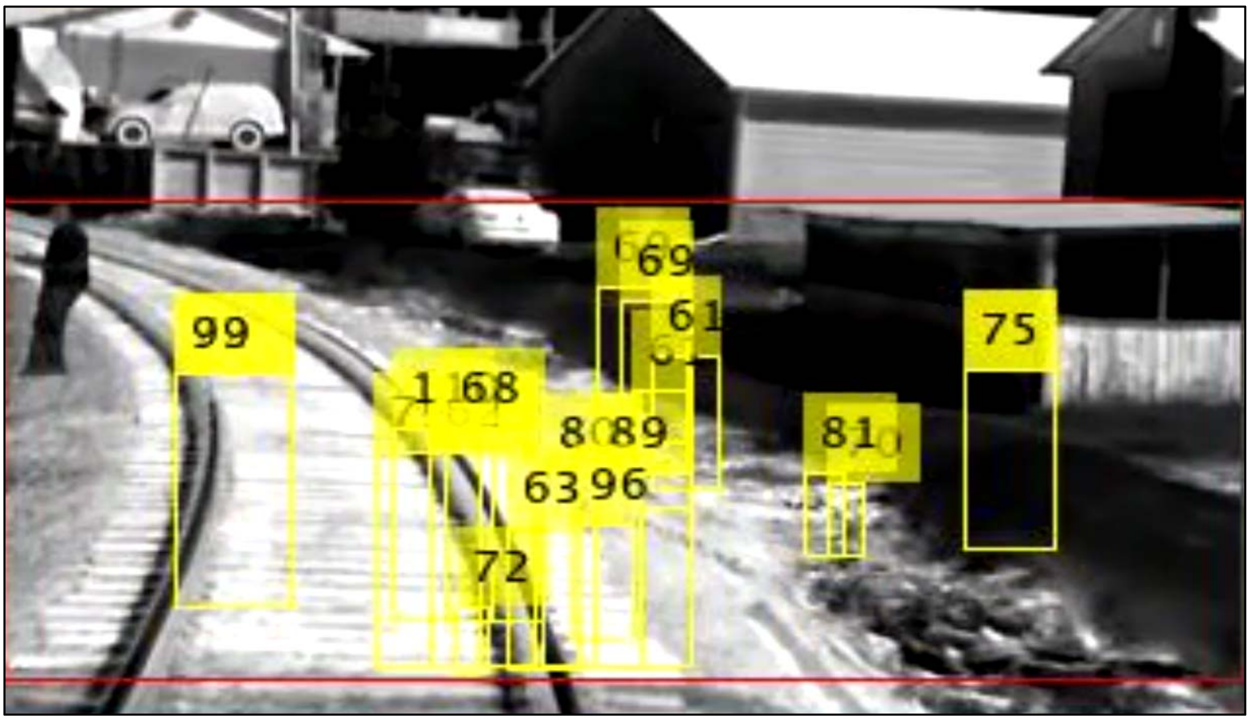

Figure 11. Pilot Test 1 results using static camera imagery only which shows a high number of false positive detections with high confidence

Pilot Test 2. Using the Dynamic Camera Database

From our initial test using the static camera imagery, it was evident that the algorithm was producing a significant number of false positive objects that were considered "confident". Therefore, we created a new thermal image database with pedestrians that had more features (i.e. perpendicular and parallel paths, different walking gaits and arm movements etc.) which helps the trained object developed in Thrust 3 better detect pedestrians. The new pedestrian database (Thrust 2) was created from Hillsborough Street 5-hour database collected from a moving vehicle. Last, the same dynamic video from the Star, NC deployment was used to test the algorithm. The results were much more promising. From Figure 12 it is evident that the false-positive detections were only shown with low confidence intervals, meaning most of these could be filtered out using a minimum confidence threshold.

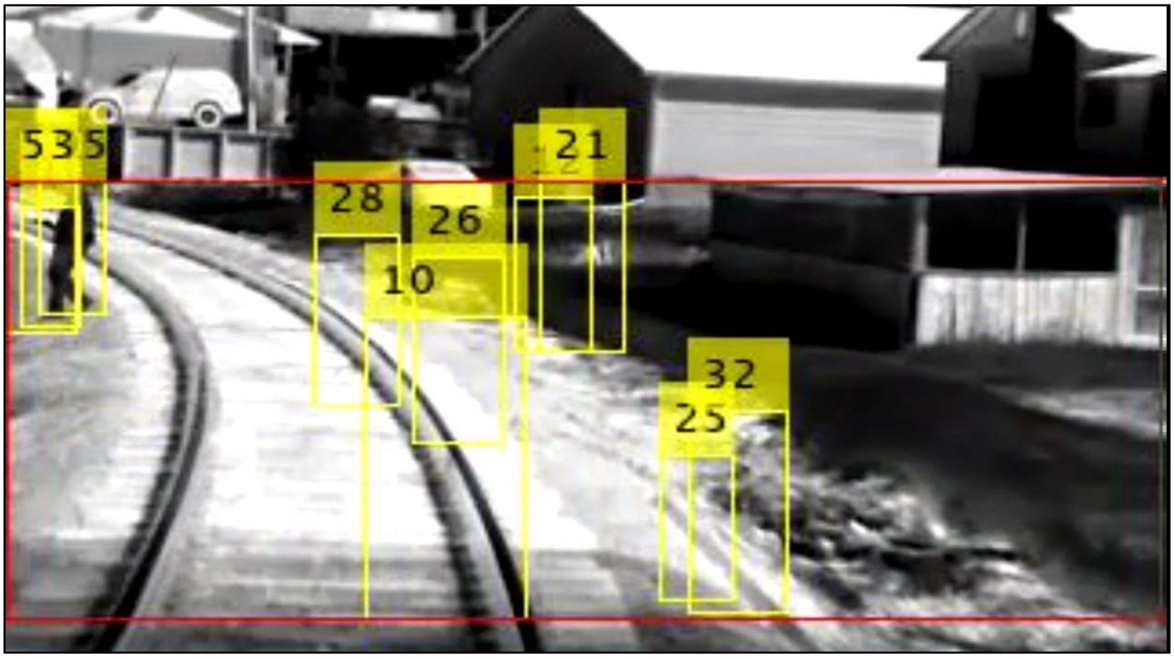

Figure 12. Pilot Test 2 results using dynamic camera imagery from Hillsborough Street database shows false positive detections with low confidence interval scores

Based on this first test, a filter of 60 percent confidence was employed, and the results were much better. In this example, only the pedestrian is detected; however, when running the entire Star, NC test track, a few false positives are still detected. A discussion of what is needed to improve the prototype is discussed in the following sections.

Figure 13. Pilot Test 2 results using a filter of 60 percent confidence shows that the most detected objects are removed and by only showing the more confident detections.

CONCLUSION

The aim of this project was to develop a prototype algorithm that will allow NCDOT to capture near miss trespassing events along rail corridors. The prototype software developed through this research effort shows a lot of promise in detecting the true extent of the trespass problem and not just reported strikes. In addition, the software will greatly reduce the number of man hours needed to manually check videos for near miss events by automating the process of pedestrian detection.

FUTURE WORK AND DISCUSSION

Although the prototype machine learning algorithm developed as a part of this effort shows a lot of promise, it does not mean that all false positive detections are removed. It is possible that there are few false-positive detections with high confidence scores, which means there is still improvement that can be made to reduce the number of false positives that would need to manually be observed. Most false positives can be avoided through updates in two of the four thrust described earlier and expounded on below.

1. Thrust 1: The region of interest (ROI) could be segmented instead of a large frame of reference, allowing unnecessary regions to be avoided. This is especially true when a train goes around a curve as the outer portion of the curve often looks at areas not of interest. Figure 13 shows an example of how the ROI can be implemented where the two small lower boxes track the rail heads to capture when the train is entering a curve and the three upper boxes track the regions of interest. The two upper/outer boxes would respond to curves by shrinking horizontally on the outside section of a curve.

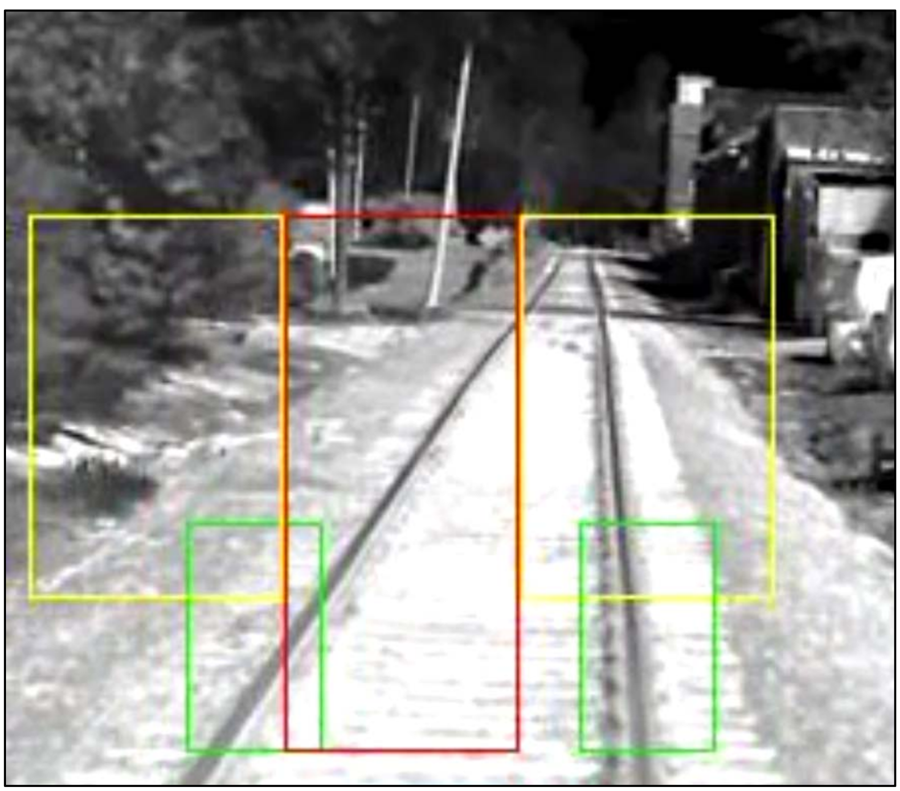

Figure 13. ROI implementation

2. Thrust 2: The imagery from more dynamic camera deployments should be used to greatly enhance the image database. The team estimates that a database in the size of 1GB or more would be enough. For reference, the current database consists of 609 images which is approximately 120 MB.

3. To capture images at distances further than 200ft the resolution can be increased at the cost of size of the file