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

NCDOT Project 2020-50
FHWA/NC/2020-50
November 2023

Christopher M. Cunningham, MSCE, PE
Edgar Lobaton, PhD
Christopher L. Vaughan, MSCE, PE
Daniel Coble, MSCE, PE
James Poslusny
Kyra Stash
Institute for Research and Education
North Carolina State University

NC STATE UNIVERSITY
ITRE
Institute for Transportation Research and Education

RESEARCH & DEVELOPMENT

RESEARCH & DEVELOPMENT

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

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

Christopher M. Cunningham, MSCE, PE – PI
Edgar Lobaton, PhD – Co-PI
Christopher L. Vaughan, MSCE, PE
Daniel Coble, PE
James Poslusny
Kyra Stash

NCDOT Project 2020-50
November 2023

TECHNICAL DOCUMENTATIONS PAGE

1. Report No. **FHWA/NC/2020-50**	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle **Pedestrian Incident Detection in the Rail Right-of-Way using Artificial Interlligence**		5. Report Date **November 16, 2023** 6. Performing Organization Code
7. Author(s)		8. Performing Organization Report No.
9. Performing Organization Name and Address **Institute for Transportation Research and Education** **North Carolina State University** **Centennial Campus Box 8601** **Raleigh, NC**		10. Work Unit No. (TRAIS) 11. Contract or Grant No.
12. Sponsoring Agency Name and Address **North Carolina Department of Transportation** **Research and Development Unit** **104 Fayetteville Street** **Raleigh, North Carolina 27601**		13. Type of Report and Period Covered **Final Report (11/1/19 – 01/31/23)** 14. Sponsoring Agency Code **RP2020-50**
15. Supplementary Notes:		

16. Abstract

In 2017, the NCDOT began funding vital research observing trespassing on railroads from fixed positions. Two rounds of this research showed the magnitude of this problem across the state of North Carolina, which is a microcosm of the trespassing picture all over the United States. With several trespassers actually being struck by trains over the course of that research, it became apparent that the frequency of trespassing was not the only issue, but the number of near miss incidents was also of extreme importance to truly paint the entire picture of trespassing in the state. As such, the NCDOT funded this study, which would fund cameras on trains to observe the frequency of near miss activity between trains and trespassers, as well as the proximity of trespassing pedestrians to moving trains. Researchers attached thermal and RGB (red, green, blue – i.e., color video) cameras to trains and captured video while these trains were commuting between workstations. However, one major hurdle to completing this research was being able to accurately and efficiently identify trespassers on and near train tracks when moving trains were close by. This research was able to prove that the problem of trespassing in close proximity to moving trains exists on a large scale while also creating a machine learning algorithm that could not only identify trespassers, but could do so with much greater accuracy than even manual observation of the captured video.

17. Key Words **Rail trespassing, trespassing, trespass, artificial intelligence, machine learning**	18. Distribution Statement		
19. Security Class. (of this report) Unclassified	20. Security Class. (of this page) Unclassified	21. No. of Pages 30	22. Price $182,560

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

DISCLAIMER

ACKNOWLEDGEMENTS

This project could not have been accomplished without the support of Aberdeen and Carolina Western Railway (ACWR) and Blue Ridge Southern Railroad (BLU). Their willingness to outfit their trains in a way that would allow ITRE technicians to install and maintain equipment on their locomotives is what made this project possible.

1. BACKGROUND AND RESEARCH NEED

Addressing trespassing along railroad rights-of-way (ROW) was a top priority for the Federal Railroad Association (FRA). Pedestrian trespassing remained the leading cause of rail-related deaths in the United States, reporting 539 fatalities in 2019 and 528 in 2020.[1] Trespass-related fatalities exceeded 500 since 2017, contrasting with highway-rail fatalities, which stayed below 300 in 2019 (294) and 2020 (198).[2] The reported trespass-related fatalities for 2019 and 2020 marked the highest numbers in the past 15 years.

In 2020, FRA statistics ranked North Carolina as the 12th state with the most pedestrian rail trespass casualties, reporting 12 deaths and 12 injuries out of 1,099 total casualties for the nation.[3] Over the 5-year period from January 2015 to December 2020, 97 pedestrians were killed while trespassing along the railroad right-of-way in North Carolina.[4]

Recent NCDOT-funded research delved into pedestrian trespassing on and around railroad ROW, including two studies observing, quantifying, and modeling trespassing activity across North Carolina[5,6]. This research allowed ITRE to accurately model expected trespassing activity based on factors like the percentage of people without cars walking to work, and business densities in various sectors such as retail, grocers, and low-income housing. While the previous research shed light on trespassing influences, it did not provide a clear picture of the frequency and severity of near-misses, as stationary thermal cameras only captured a fraction of the phenomenon. These near-miss events, not resulting in casualties, were not included in FRA's incident reports on trespassing.

To address this gap, NCDOT funded a study to capture near-misses observed from trains, determining a cost-effective method for accurate quantification. This research offered a broader geographical representation of trespassing compared to static location studies, understanding hardware requirements for observing behavior from trains. Additionally, it created a machine learning algorithm capable of observing pedestrians near trains more accurately than human observers. The AI developed was crucial for handling the vast video data from train cameras, as manual observers couldn't effectively capture all trespassing activity. As detailed in this report, manual observation fell short in accurately documenting trespassing events around trains.

2. LITERATURE REVIEW

2.1. Railroad Trespasser Detection

As discussed earlier, trespassing emerged as the primary cause of rail-related deaths in the United States, with over 500 trespass-related fatalities occurring each year since 2017.[2] The Federal Railroad Administration (FRA) designated this issue as a high-priority concern during their 2017 Grade Crossing Research Needs Workshop, establishing five focus areas related to their research needs.[7] In the focus area of Community Outreach and Education, the top recommendation was to enhance trespasser identification, motivation, and messaging.[7]

In previous studies, stationary camera footage was employed at hotspot locations for trespassing along a railroad right-of-way (ROW) to aid in trespasser identification. A study led by the FRA Office of Research, Development, and Technology (RD&T), in collaboration with the U.S. Department of Transportation (USDOT) and the John A. Volpe National Transportation Systems

Center (Volpe), utilized stationary cameras at a ROW in Worcester, Massachusetts, identifying 115 trespassers in a 30-day period in 2017. Data analysis revealed that at this specific site, the majority of trespassers were loitering under or around the bridge ROW, with most of them also engaging in drug use according to video analysis. Recommendations from this study included ongoing outreach to the homeless population, increased installation of NO TRESPASSING signs, and heightened law enforcement patrols in the area.[8] While this study contributed to understanding the types of trespassers in the local area of Worcester, MA, it only represents a small portion of the trespasser types found along railroad ROW across the nation.

Beyond ITRE research, Rutgers University established itself as a leading railroad trespassing research group nationwide. In 2018, Zaman et al. published research regarding the use of artificial intelligence (AI) on video recorded from cameras already in place on trains, along tracks, at grade crossings, and in train stations.[9] This research focused on a grade crossing in New Jersey, developing an algorithm that successfully identified near-miss events between pedestrians and trains. Similarly, Zhang et al. focused on railroad tracks immediately adjacent to and across at-grade crossings, considering the presence of vehicles or pedestrians in correlation with the position of the crossing gate and the presence of a train.[10] This research identified thousands of trespassers over several weeks, assigning demographic information to those trespassing events and noting the surrounding environment for future countermeasure selection. This research also highlighted the need for AI compared to human observation of trespassing behavior due to observer fatigue, a crucial conclusion of this report. A more recent article by the same Rutgers researchers discusses using AI to send real-time alerts to railroad or public safety personnel regarding trespassing events detected.[11] This research includes motor vehicles as well as pedestrians and, like previous Rutgers research, utilizes existing camera infrastructure without requiring new installations. Notably, the Rutgers researcher papers emphasize the AI's ability to capture all trespassing events while producing minimal or no false positives, a promising aspect.[9,10,11]

Apart from previous research by ITRE and Rutgers researchers, machine learning has not been widely applied in research efforts for rail-trespass prevention, particularly on video captured from moving trains.[12] Machine learning has found success in other areas of transportation research, such as applications with autonomous vehicles and Advanced Driver Assistance Systems (ADAS). In a study published in the Journal of Advanced Transportation, researchers used mounted cameras in vehicles and machine learning technology to enhance driving style recognition technology with vehicle trajectory data.[13] The aim of developing this technology is for machine learning to assist autonomous vehicles and ADAS in preventing rear-end collisions.[13]

2.2. Pedestrian Detection Using AI/ML

Pedestrian detection and trajectory tracking witnessed significant progress in recent years due to the development of deep learning techniques. This application is part of object detection and finds extensive use in areas such as automatic driving[14,15,16] and video surveillance. In the domain of automatic driving, a camera captures environmental information akin to human eyes, while a lidar provides position data, and a wireless sensor network aids in additional sensing and communication. Similarly, train security monitoring utilizes cameras on moving trains to enhance safety through real-time pedestrian detection.

The inception of deep learning for pedestrian detection occurred in 2014 when Girshick et al. introduced RCNN**Error! Reference source not found.**. Subsequent advancements included Fast RCNN[18] and Faster RCNN[19], which aimed to simplify the algorithm and reduce computational complexity. Building upon Faster RCNN, Mask RCNN[20] emerged to address general object detection and pedestrian detection challenges.

The aforementioned methods follow a two-stage detection process involving region suggestion and object detection. In pursuit of improved efficiency and real-time applicability, one-stage detection was introduced. Unlike its two-stage counterpart, one-stage detection establishes a series of anchors on the feature map to predict the object center and bounding box directly. The YOLO series algorithms[21] embody one-stage detection and prove versatile in various object detection tasks. While one-stage detection might lead to missed detections in crowded pedestrian scenes, its high detection speed proves valuable for real-time applications, particularly in the field of intelligent driving. To further enhance efficiency, Wei Liu et al. proposed an anchor-free object detection mode in 2019[22], garnering attention from various works[23,24,25].

Several public datasets contribute to pedestrian detection training and validation, including Caltech[26], KITTI[27], and CityPersons[28]. Commonly used datasets in the broader object detection field, such as COCO[29], also aid in building pedestrian detection models. Despite these resources, there remains limited data covering diverse contents. In 2020, Guofa Li et al. enriched their database by collecting pedestrian images in hazy weather and employing six augmentation strategies[24]. In 2021, Matteo Fabbri et al. introduced MOTSynth[30], a synthetic dataset generated by a rendering game engine to simulate various aspects of the multi-person tracking problem. This extensive and diverse dataset significantly improves detection performance.

3. METHODOLOGY

3.1. Dynamic Trespassing Detection System

3.1.1. Technology Evaluation

The research team recognized that video would be the appropriate method to capture trespassing events along railroads, but they faced a challenge in determining the most suitable technology – thermal video or simple RGB ("red, green, blue," or color) video. Thermal cameras offered an advantage at night and during severe weather, but they came with higher costs and generally lower resolution compared to standard RGB cameras. The team understood that resolution could impact machine learning algorithms, which tend to perform better with higher resolution images.

Regardless of the chosen camera system, the team needed to develop an algorithm for both thermal and color video images. This algorithm had to effectively identify potential humans and distinguish them from other animals or objects within the camera view. Developing such algorithms required numerous images for training, both with and without the subjects of interest. To achieve this, cameras needed to be installed on trains. Staged events with both types of cameras were also conducted to ensure the team could identify video clips containing humans.

3.1.2. Description of Selected System

Initially, thermal cameras seemed promising due to their ability to detect heat signatures, especially at night and in adverse weather conditions. However, during pilot data collection, the team discovered a potential issue with thermal cameras. Heat signatures from rail lines, animals, pavement, and even the ground could be detected, diminishing the perceived advantage of thermal technology. Discussions with rail crews and train conductors revealed that the bright headlights on trains could effectively illuminate as far as RGB cameras could see, reducing the advantage of thermal cameras.

To address this, the team decided to test both thermal and RGB cameras during all pilot testing and future data collection events, developing algorithms for both. The team leveraged their experience with AXIS thermal cameras and AXIS Companion, the AXIS video management software, gained from previous NCDOT-sponsored rail trespassing research at static locations. Additionally, ITRE owned several AXIS thermal cameras, as depicted in Figure 1, attached to a train. shows an image of one of the AXIS thermal cameras attached to a train. The research team also had prior experience with and a small inventory of Hikvision RGB cameras, also featured in Figure 1.

Figure 1. Example of Camera Systems Installed on a Train

3.1.3. Pilot Testing

Before engaging in camera installations or testing on actual railroads, trains, or other rail equipment, the research team conducted preliminary tests on thermal and RGB cameras attached to an NCSU-owned minivan. The cameras were mounted on a steel plate with strong magnets on the opposite side. This setup was magnetically affixed to the front of the minivan. NCSU students employed by ITRE drove the minivan around the NCSU campus, capturing pedestrian activity typical of large college campuses. It's important to note that, at this stage, an algorithm was written and tested for this configuration but was in an early stage and to a limited extent. This testing was conducted solely for experimental purposes, and the research team recognized the need to limit time spent on this effort, understanding that it wouldn't translate well to the algorithm for video captured from trains on railroads. Additionally, this effort predated the funding of this research by the NCDOT and served as an added value to previous static rail trespass research.

3.2. Experiment Design with Controlled Environment

3.2.1. Site Selection

For the controlled rail-trespass experiment, the research team selected two sites owned by the Red Springs and Northern Railroad Foundation (RSN). These sites encompass inactive rail sections acquired by RSN in 2004, now maintained by the foundation for local fundraising and public service events. An RSN member supported the field team throughout the research by providing and operating a hi-rail vehicle for controlled studies. Two separate trips to the RSN railroad were conducted to enhance the database of controlled-trespassing incidents. Two sites in the area were utilized. The first site, situated in the town of Parkton along the Red Springs and Northern Railroad, is depicted in Figure 2, while the second site is just outside of Red Springs, shown in **FIGURE 3**. Research activities were conducted at both sites during the first trip, while the second trip focused solely on the Red Springs site to minimize travel time between locations.

3.2.2. Data Collection

3.2.2.1. Preparation

To prepare for field trials, the research team created a map using Google MyMaps to identify potential sites along the tracks. A week before conducting trials, the team visited the site to place stakes for crossing points. For each subject, five stakes were positioned: a center point where they stood at the trial's outset, two stakes 200 feet in each direction from the center, and two stakes 600 feet in each direction. These stakes served to guide subjects on where to stand and when to perform the prescribed maneuver based on their distance from the approaching hi-rail vehicle. The team chose two curved locations of tracks—one in the sun and one in the shade—and two straight locations, also with one in the sun and the other shaded, to introduce varying lighting conditions. As illustrated in the experiment design figures below, a fifth crossing point beyond subject four was added at both locations.

Figure 2 and Figure 3 depict the experiment design slides printed for each subject. Each slide displayed the trial number, maneuver type, and train direction. Additionally, the slides featured each subject's center point as a pink marker and their 600-foot stake as an orange place marker.

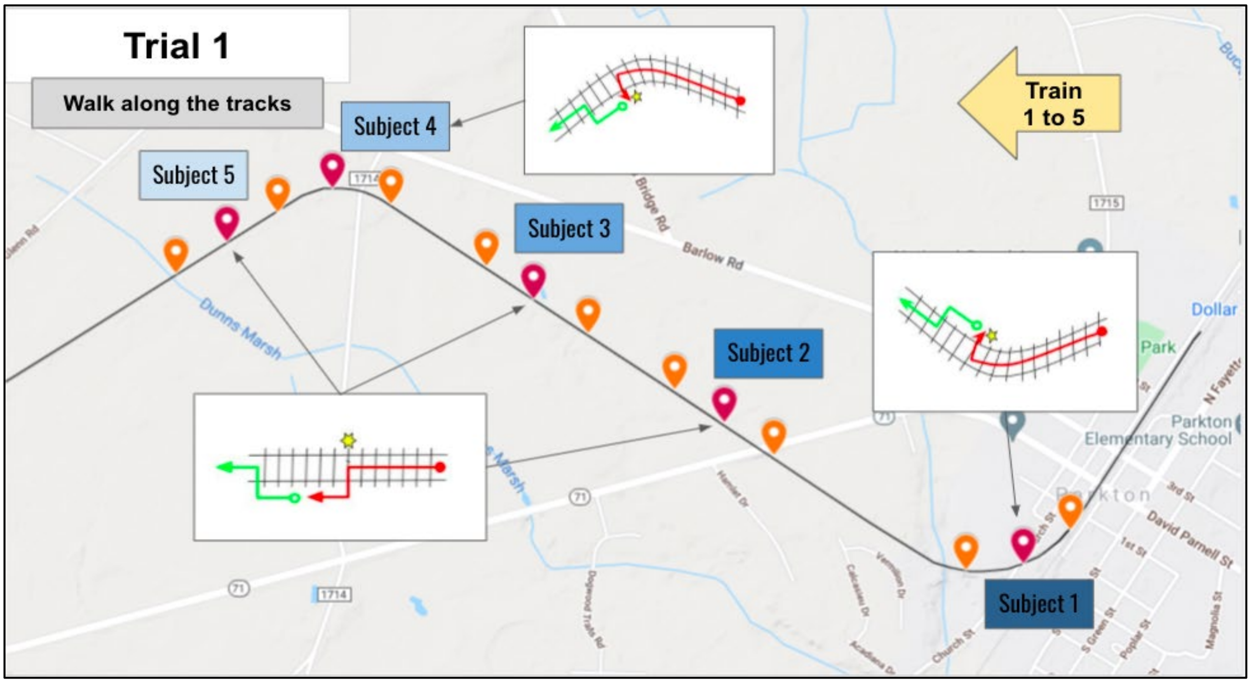

Figure 2. Parkton Experiment Design

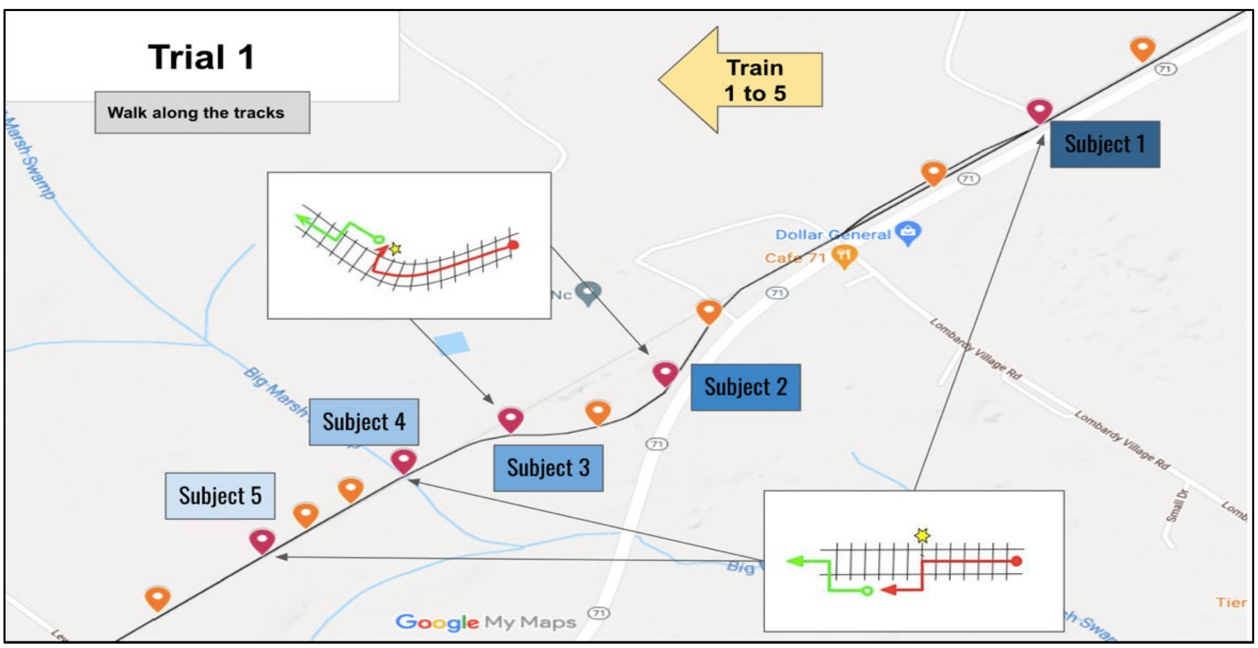

Figure 3. Red Springs Experiment Design

3.2.2.2. Field Trial 1

The first field trial was conducted on Wednesday, June 16th, 2021. In order to establish a diverse database for the machine learning algorithm, the research team included various crossing behaviors, capturing subjects from different angles, distances, and postures. Below are examples of the crossing behaviors incorporated into the trials.

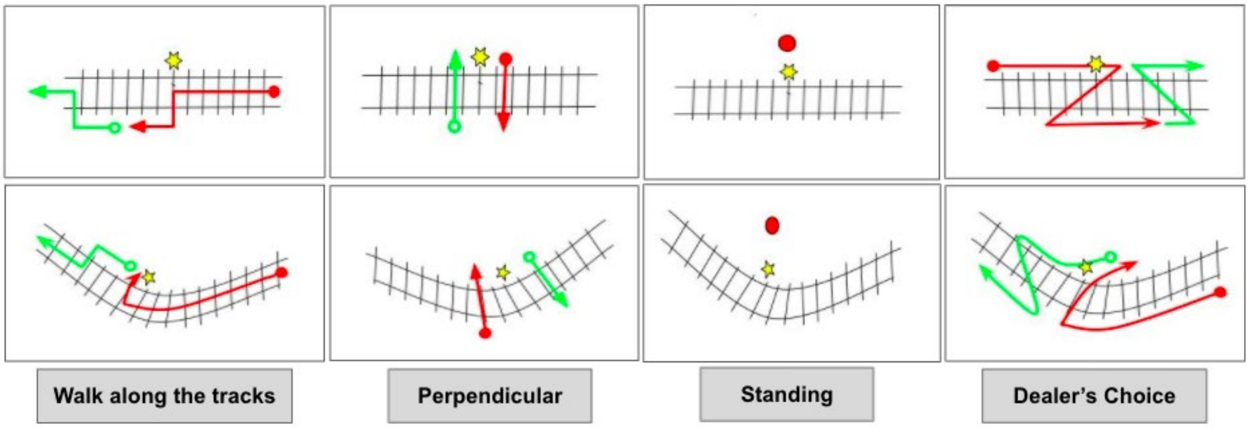

Figure 4. Crossing Behaviors

For each crossing behavior, the red lines indicate movement in front of the approaching hi-rail vehicle, while green represents movement after the hi-rail has passed. The star in the crossing behavior images indicates the center point, marked with a stake at the research sites. Stakes were also positioned at 200 ft and 600 ft from each crossing point in both directions. During the experiment trials, subjects were instructed to initiate their movement when the hi-rail reached the 600-foot or 200-foot mark, depending on the trial. Similarly, they were instructed to move as the hi-rail receded after passing, enabling the collection of movement data at various distances with the hi-rail approaching and receding from the subject. Additionally, these movements were conducted individually initially and then in groups, providing the algorithm with video footage of both individual trespassers and groups, mirroring real-world scenarios.

In the last two examples in Figure 4, "Standing" and "Dealer's Choice," subjects were encouraged to be creative, introducing a variety of postures (standing, crouching, waving their arms, etc.) and movements (running across the tracks, crossing multiple times in front of the hi-rail, etc.).

3.2.2.3. Field Trial 2

The second field trial was conducted on Tuesday, July 13th, 2021. The research team conducted field trials in three sessions, starting with the morning session at the Parkton location, followed by afternoon and evening sessions at Red Springs. For each session, they captured data on individuals and groups performing the crossing behaviors described earlier. At each location, nine trials were conducted, including a test trial at the beginning of each session after positioning subjects at their sites. Two bullet RGB-Infrared cameras and two thermal cameras were installed on the front and back of the hi-rail vehicle, as shown in Figure 5 and Figure 6, to collect data both in front of and behind the hi-rail. Figure 5 displays the camera installation on the hi-rail for Field Trial 1.

Figure 5. Camera Installation on Hi-Rail for Field Trial 1

3.2.2.4. Field Trial 3

The third field trial was conducted on Wednesday, November 10, 2021. The second trip to the Red Spring and Northern Railroad replicated the procedure from the first trip, except for the morning session in Parkton. Field Trial 3 consisted of sessions conducted in Red Springs, one in the afternoon and one in the evening. Video data were captured for individuals and groups performing crossing behaviors, totaling nine trials in the afternoon and nine at night, with a test trial at the beginning after setting subjects in place. The same cameras and setup from Field Trial 1 were used. The primary difference was the use of a different hi-rail vehicle, depicted in Figure 6, in the hope that it would provide smoother camera footage. The first trial test resulted in unstable video imagery, usable but not ideal for training the algorithm. Figure 6 illustrates the camera installation on the hi-rail for Field Trial 2.

Figure 6. Camera Installation on Hi-Rail for Field Trial 2

3.2.2.5. Footage of Subjects Trespassing from Field Trials 1 and 2

The following photos are screenshots from the downloaded footage collected during Field Trials 1 through 3. These images represent various types of imagery with differing distances, lighting, and movements used to calibrate the machine learning algorithm.

Figure 7. Daytime Footage of Individual Walking Along the Tracks

Figure 8. Daytime Footage of Perpendicular Group Crossing

Figure 9. Nighttime footage of Individual Walking Along the Tracks

Figure 10. Nighttime footage of Group Crossing

3.2.2.6. Summary of Controlled Data Collection Effort

In the initial testing of cameras installed on the hi-rail vehicles, the team quickly observed that the images tended to be jumpy and unstable due to vibrations caused by metal wheels on metal rails. This issue was partly due to the small and light nature of the hi-rail vehicles, which lacked suspension. Additionally, the cameras were being used atypically, as they are typically installed as security cameras on buildings, not on moving objects, and lacked image stabilization technology like GoPros.

The team attempted to address the vibration issue by using stronger and tighter straps to attach the cameras to the hi-rail vehicles, which helped to some extent, but did not eliminate the problem. Complete elimination of vibration proved difficult due to the size and weight of the hi-rail vehicles. However, video collected from train engines did not face the same issue, being much more stable and smoother.

Early on, the team anticipated that only an RGB camera might be necessary for collecting trespassing data from trains due to the brightness of the headlights and the resolution issues with thermal cameras affecting the machine learning algorithm's video processing abilities. Testing continued with both thermal and RGB cameras throughout the data collection process, but the team recommends that future data collection efforts only utilize RGB cameras, not thermal cameras.

3.3. Machine Learning and Modeling with AI

The research team trained a pedestrian detection algorithm using a training dataset and validated it with test data. Experiments conducted in Red Springs and Parkton, NC, aimed to gather additional datasets similar to the data collected in Star, NC, to validate the developed algorithms. The collected dataset for experiments in Star, Red Springs, and Parkton examined factors and associated levels (Table 1) using both forward and rear-facing thermal and RGB camera arrangements with five subjects along a 2-mile section of railroad. After calibrating and validating the algorithm in a controlled environment, the research team tested it against real-world video data from active rail scenarios.

Factor	Levels
Direction of travel	Away from Train, Towards Train, Stationary
Distance to redirect from travel path	200’, 600’
Time frame to continue after train passes	1 second, 5 seconds
Direction of travel after train passes	Away from Train, Towards Train, Cross Perpendicular
Section type	Straight or Curve
Lighting type	Direct sun, shade, nighttime (not included in the Star, NC dataset from 2017)

Table 1. Experiment Design Factors and Associated Levels

3.4. Experiment Design with Active Rail

3.4.1. Site Selection

After controlled environment testing, the team needed to test the algorithm on trains commuting on active rail lines. Despite concerns about rail companies' hesitancy to allow research on their private property, the research team identified forward-thinking rail companies that welcomed and encouraged this research: Aberdeen-Carolina Western Railroad (ACWR) in south-central North Carolina and Blue Ridge Southern Railroad (BLU) in western North Carolina.

3.4.2. Camera Installation and Data Collection

Aberdeen-Carolina Western Railroad

The ACWR offices in Candor, North Carolina, were visited by the research team in March 2020 to determine potential installation locations on their train. ACWR staff provided assistance, determining suitable locations inside the train for camera installation without obstructing the train engineer. Due to the COVID-19 pandemic, the actual camera installation occurred in June 2020. The research team collaborated with the ACWR train maintenance crew to install ITRE thermal and RGB cameras on one train, ACWR 6909. Data were collected from this train between June 2020 and January 2022.

The cameras were strategically installed near the top of the train between the front windshields, as shown in Figure 11, to capture trespassers and near-miss events. The power connection for the cameras was activated only when the train was in operation, preventing data collection during "dead" periods when the train was not running.

a) ACWR Installation

b) ACWR Camera Field of View

Figure 11. Installation on ACWR Train with Field of View

Blue Ridge Southern Railroad

BLU, located in Canton, North Carolina, about 20 miles west of Asheville, allowed the research team to install thermal and RGB cameras on three different trains: BLU 4202, BLU 3932, and BLU 4204. The BLU maintenance staff played a crucial role in ensuring a clean and effective installation that did not interfere with train engineers.

a) BLU Installation

b) BLU Camera Field of View

Figure 12. Installation on BLU Train with Field of View

The team conducted an initial installation trip in March 2021 for one BLU train, followed by additional installations on two more trains in May 2021. Data were collected from these trains between March 2021 and January 2022, as shown in Table 2.

BLU Train	Beginning	End
4202	March 2021	January 2022
3932	May 2021	January 2022
4204	May 2021	January 2022

Table 2. BLU Train Recording Periods

4. ALGORITHM DEVELOPMENT

4.1. Data Annotation

Two different annotation methods were employed—one for thermal videos and another for RGB videos. For thermal videos, the team initially used the video annotation toolbox in Matlab, requiring manual frame-by-frame label checking. However, this process was time-consuming and resource-intensive. Subsequently, the team transitioned to Labelbox[31] for RGB video annotation, an online tool supporting automated labeling. Labelbox incorporates Easy tracker to assist in object tracking, allowing labelers to adjust box size and position when the tracker is ineffective, eliminating the need for manual modifications per frame. Labelbox also facilitates convenient data storage directly from Google Cloud, addressing the limitations of Matlab, which mandates saving data locally.

4.2. Model Development

4.2.1. System development

The research team tackled the railroad trespassing issue by concentrating on reliable pedestrian detection based on proximity to the railroad. The approach involved training accurate pedestrian detection, robust railroad tracking, and integrating these modules to remove false detections. A three-phased system was designed, as illustrated in Figure 13.

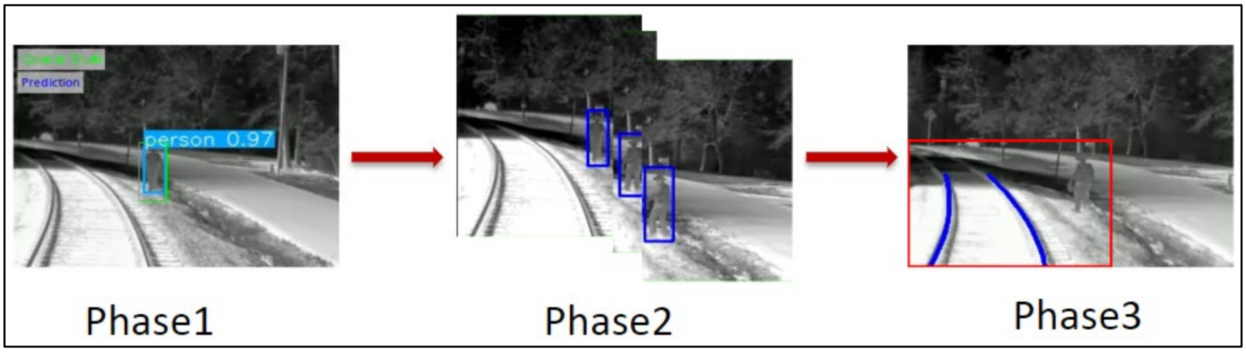

Figure 13. System Work Flow

In phase one, an object detection model was trained using YOLO variants (v3, v4, and YOLOX) and Detectron2. The models were evaluated on thermal and high-resolution RGB videos. In phase two, the tracking algorithm DeepSort was integrated with YOLO variants, enhancing prediction accuracy and reducing processing time. In phase three, a rail detection algorithm utilizing polynomial curve fitting was developed. This algorithm, capturing two rails more precisely than template-matching, concentrated on the focus area, optimizing computational resources.

4.2.2. Algorithms

In phase one, the popular object detection algorithm Yolo[21] was tested, along with a popular object detection library Detectron2[32] which includes many fine-tuned models. Yolo, which stands for "You Only Look Once," is a highly efficient object detection algorithm for real-time processing. The basic idea is that an image can be evenly divided into several grid cells and a neural network backbone can be used to do predictions on each cell simultaneously. The outputs of the neural network are bounding box locations and the corresponding boxes' probabilities for each class, which are the probabilities of seeing a person in this case. This dividing process can be done multiple times with different sizes of cells due to object size diversity. This produces many overlap bounding boxes for each object and some of them are not accurate enough to label the object precisely. The non-max suppression mechanism is implemented to solve this problem by keeping the bounding box with the highest probability over several overlapping boxes with the same prediction object.

Because of the success of the YOLO architecture, there have been numerous YOLO variants released over the years. Three of those have been implemented in our work: YOLO v3, YOLO v4, and YOLOX. YOLO v3 was a commonly used version published in 2018 by Joseph Redmon and Ali Farhadi[33] based on the Dark-53 backbone. This is also the last YOLO version created by the original author. The two main improvements are 1) replacing the softmax activation function with independent logistic classifiers and 2) predicting on three different image scales to enhance the ability to recognize small objects. YOLO v4 was released by Alexey Bochkovskiy et al. in April 2020[34]. YOLO v4 outperforms YOLOv3 by 10% in average precision and 12% in speed on the public dataset COCO[35] through several new features including Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation, Mosaic data augmentation, DropBlock regularization, and CioU loss. YOLOX was proposed by Zheng Ge et al. in August 2021[36]. In YOLOX, the detector was rebuilt in an anchor-free manner and other advanced detection techniques, i.e., a decoupled head, and the leading label assignment strategy SimOTA were conducted to surpass all old YOLO models across a large-scale range. YOLO v3 and YOLO v4 were evaluated and compared on thermal videos collected as a part of this project. Next, the light version YOLOX was evaluated on high-resolution RGB videos collected likewise as a part of this project.

Detectron 2[32] is a library developed by the Facebook AI Research team. This provides a number of state-of-the-art detection and segmentation algorithms including Mask R-CNN[20] published in 2018 by KaiMing He et al.. Mask R-CNN 101 version was tested on the new RGB videos. Mask R-CNN was developed based on Faster R-CNN[37] by adding Rbinary mask outputs. This architecture is more generalizable to other tasks, e.g., estimating human poses and object segmentation, than YOLO architecture, but with a heavier computational load. This algorithm was treated as the baseline model for this research and it was compared with YOLOX on the collected RGB videos.

In the second phase, the most popular object tracking algorithm, DeepSort[38], is used. This algorithm is widely used due to its practicality. DeepSort has a core component called Kalman filtering which used a simple linear velocity model. This model has two variables, absolute position and velocity, used for predicting the next state. Therefore, this mechanism could not be used to remove the wrong prediction in the object detection model based on the prior state. To do that, DeepSort learns a deep association metric associated with the relation between the prediction by detection algorithm and Kalman filter on a large-scale dataset. During the online application, this association metric could speed up the algorithm by utilizing nearest neighbor queries in visual appearance space. In this application, this mechanism was added to YOLO v3 and YOLO v4 on the thermal dataset and the research team investigated the potential gains.

The railroad detection algorithm was developed specifically for this application and it uses polynomial curve fitting of two parallel lines on the physical space through some basic calibration of the image view. This algorithm can capture two rails at the same time more precisely than the template-matching approach[39] that was used as a baseline, especially on curves. Some examples are shown below.

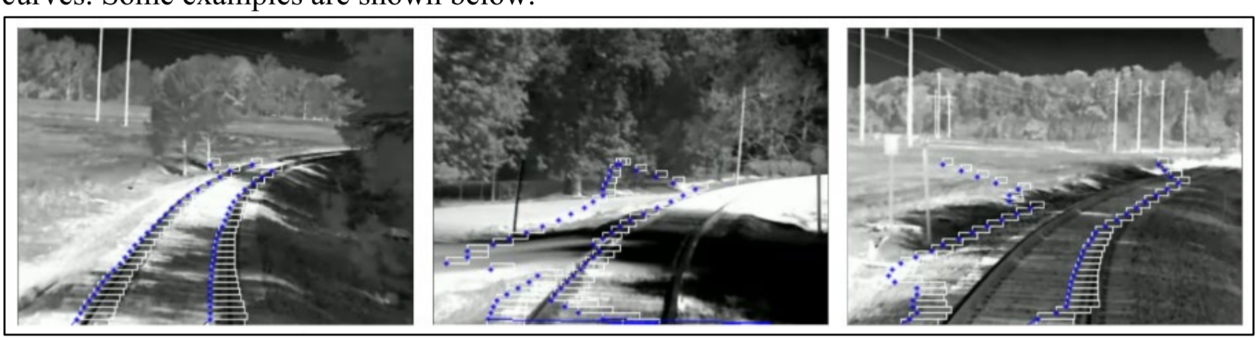

Figure 14. Template Matching Approach Struggling to Identify Rails

Figure 15. Proposed Rail Identification Method

A visualization of incorporating pedestrian detection and rail detection is shown in the image below. The blue lines are rails detected by the polynomial curve fitting algorithm and the red lines indicate the boundaries of our Region of Interest (ROI) near the railroad. Different thresholds were set for pedestrian detection in the ROI and on the outside. For example, the red and green boxes are both detections of people by YOLO v4. According to the different thresholds, the red color in the middle box indicates that this object is highly likely to be a person, while the detection on the green box is showing lower confidence.

Figure 16. Example of How the System Works

4.2.3. Evaluation Metrics

To assess the performance of the system, basic standard metrics for a binary classification problem, including Precision and Recall, were employed. In the context of pedestrian detection, each object appearing in a given time frame was treated as a sample, with the total samples encompassing all objects across all time frames. Detections of pedestrians were designated as the positive class, while the remaining instances constituted the negative class.

For every prediction, the correctness was determined by assessing the overlap between the predicted bounding box and the ground truth. If this overlap exceeded a specified threshold, such as 0.6, the prediction was deemed a true detection. Conversely, predictions failing to meet this criterion were classified as false detections. Consequently, the four combinations of prediction results were identified as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

Precision gauges detection accuracy, expressed as the ratio of true predictions to all predictions. Recall, on the other hand, reflects the system's ability to retrieve relevant instances, denoted by the ratio of true predictions to all positive samples. The equations for precision and recall are presented below.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

In addition to standard metrics, the evaluation incorporates frame-based and event-based metrics. Frame-based metrics assess algorithms in time frames where at least one pedestrian is considered a positive sample. Similarly, event-based metrics regard each continuous period of individuals' presence as a positive sample, referred to as an event (as illustrated in Figure 17).

Given this framework, precision and recall are computed using the same equations as standard metrics, with separate calculations for frame-based and event-based metrics. Unlike predicting the precise location of pedestrians, the algorithm's task becomes more manageable when determining whether pedestrians are present in the frame or not.

Frame-based and event-based metrics showcase the algorithm's capability to function as an annotation assistant, facilitating the acceleration of the labeling process. To assess the algorithms' runtime efficiency, the Frame Per Second (FPS) metric is employed. FPS indicates the number of frames the algorithm can process in one second, serving as a measure of its real-time processing capability. A higher FPS signifies a faster processing speed.

4.3. Implementation and Results

4.3.1. Results for thermal images

Due to the limited data available from the thermal camera, training for YOLO v3 and YOLO v4-based architectures utilized the FLIR dataset[40], accessible online. To assess the DeepSort tracker's effectiveness, the performance of the detector-only model was compared to the combined model using standard metrics. The analysis employed the small version of YOLO v3, and the numeric results are presented in Table 3.

In the absence of integration with the tracker, YOLO v3 achieved only 32.2% precision and 10.49% recall, primarily due to numerous false detections. DeepSort contributed to a notable improvement, enhancing precision to 75.00% and recall to 12.60% by eliminating unassociated detections, often mistakenly identified by the detector. This suggests that the system excels at providing accurate individual predictions but faces challenges in capturing all subjects, as evidenced by the relatively low recall values.

The limitation arises when subjects are situated far away from the camera, resulting in their size being too small for recognition by the system due to the thermal camera's low resolution. Addressing this challenge could be achieved through hardware improvements, primarily by enhancing camera resolution, or by expanding the dataset with images of pedestrians positioned farther away. Such enhancements would contribute to the algorithm's ability to overcome this specific issue.

Detector	Tracker	Precision	Recall
YOLO V3-small	None	32.20%	10.49%
YOLO V3-small	DeepSort	75.00%	12.60%

Table 3. Comparison of Detector with and without Tracker

Various versions of the YOLO v4 architecture with DeepSort were tested, and the results are detailed in Table 4. YOLO v4-original represents the original size of YOLO v4, while YOLO v4-small and YOLO v4-tiny are derived from YOLO V4-original with reduced weights.

Comparing YOLO v3-small and YOLO v4-small systems, the upgrade from YOLO v3 to YOLO v4 yielded a notable improvement, increasing precision by 15.75% and recall by 5.3%. Notably, the small version of YOLO v4 did not significantly decrease precision compared to the original version, but it experienced a 5.46% drop in recall. The tiny version exhibited the fastest processing speed at 43.41 mean FPS but at the expense of a substantial reduction in precision and recall.

In the context of pedestrian detection, the failure to identify objects poses a risk of traffic accidents. The challenges contributing to low recall include the small size of individuals and the low resolution of thermal images, as previously mentioned. To enhance performance, blurred images were introduced into the training data to instruct the model in recognizing challenging cases. This approach, applied to the YOLO v4-small system, resulted in approximately a 6% improvement in recall. However, the speed also increased to 24.07 FPS, accompanied by a decrease in precision to 72.65%. The decline in precision suggests that the available data may be insufficient for the model to learn comprehensively.

System	Precision	Recall	Mean FPS
YOLO V4-small + DeepSort	**90.78%**	17.90%	21.55
YOLO V4-original + DeepSort	90.40%	22.66%	15.09
YOLO V4-tiny + DeepSort	74.36%	7.27%	**43.41**
YOLO V4-blur + DeepSort	72.65%	**23.78%**	24.07

Table 4. Comparison of Different Versions of YOLO V4

4.3.2. Results for RGB images

With the availability of RGB videos, the object detection algorithm in the existing pipeline underwent an upgrade to the latest version of YOLO, specifically YOLOX. The evaluation utilized two new datasets recorded at The Red Springs and Northern Railroad in Parkton, NC. These datasets, recorded on three different dates, are denoted as Red Springs 1, Red Springs 2, and Red Springs 3.

For the initial testing of YOLOX, the data from Red Springs 1 and 2 were labeled by identifying timestamps with any person present. Reliable detections of individuals were observed, particularly when they were in close proximity to the camera. However, it was noted that during the approach of the hi-rail to a person, there is a period when the algorithm becomes unstable. Figure 17 illustrates this instability, with the red line representing prediction results. High values indicate the algorithm's detection of a pedestrian in the frame, and vice versa. Peaks and valleys occur at the beginning of an event, such as when the camera is approaching a subject.

To address this instability, a median filter was introduced after the prediction process to smooth out the erratic areas. The green line in Figure 17 depicts the filtered predictions. The median filter operates frame by frame, replacing each frame's result with the median or majority of neighboring results. After analyzing various filter sizes, it was determined that a filter size of 15 frames yielded the best results for both Red Springs 1 and 2 datasets. The corresponding results are presented in in Table 5.

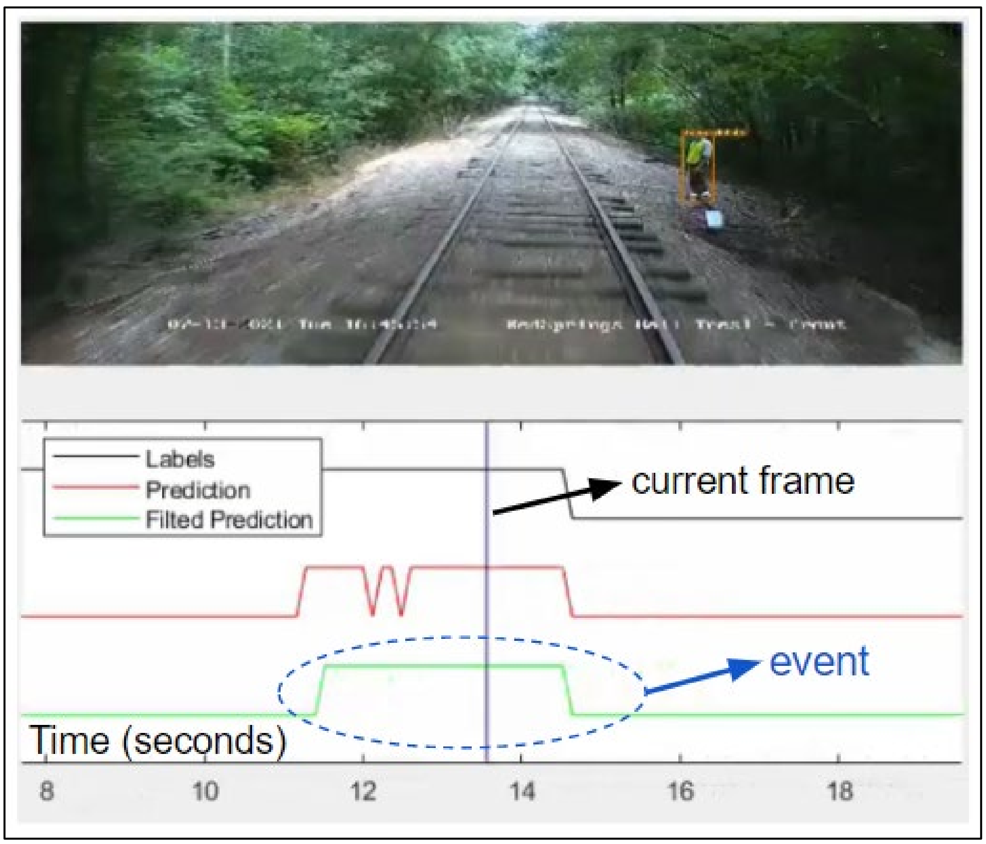

Figure 17. Event Example from Front-Facing Camera

The algorithm's performance was assessed using two metrics: (1) frame-based and (2) event-based. The frame-based approach treats each frame as an independent observation, revealing a high precision of >93% (indicating accurate detection) but a low recall of <50% (attributed to difficulties in detecting individuals further away from the camera). The event-based approach, considering the interval a person is visible as a single event, showed both high precision and recall, both exceeding 90%. These results suggest that the detector can be valuable in annotating people by signaling the occurrence of an event in the dataset.

Dataset	Frame Precision	Frame Recall	Event Precision	Event Recall
RedSpring 1	94.23%	37.19%	90.48%	90.48%
RedSpring 2	93.20%	44.32%	100.00%	95.12%

Table 5. Frame and Event Level Analysis with YOLOX architecture

Upon examining Red Springs 1 & 2, two labeling assistant candidates were evaluated for use on Red Springs 3 data: YOLOX and Mask R-CNN. YOLOX achieved a higher frame precision of 83.58%, while Mask R-CNN achieved a frame recall of 93.68%. Given the potential frustration caused by false detections triggering false alerts when the train is in motion, the high-speed YOLOX is deemed superior to Mask R-CNN. However, high recall is beneficial in the labeling process when real-time response is unnecessary, making Mask R-CNN a viable choice for AI-assisted labeling.

Detector	Frame Precision	Frame Recall
YOLOX	83.58%	74.49%
Detectron2 (Mask R-CNN)	41.22%	93.68%

Table 6. YOLOX and Mask R-CNN Analysis on Red Springs 3 with frame-based metrics

AI-assisted labeling serves to reduce the human resources dedicated to labeling tasks. In practical scenarios, where videos can span several hours, manual annotation becomes time-consuming and cumbersome. Mask R-CNN's filtered predictions were imported into the Labelbox online annotation tool. For a 2.5-minute video, AI-assisted labeling using Mask R-CNN significantly accelerated the process, reducing the labeling time from 2 minutes and 41 seconds to 20 seconds. This substantial time reduction highlights Mask R-CNN's potential as a powerful assistant in video labeling. Further testing is required to refine conclusions on the tool's overall impact.

4.4. Summary of Algorithm Findings

In conclusion, state-of-the-art (SOTA) object detection algorithms, specifically the YOLO series, demonstrate superior precision in rail pedestrian detection challenges. The integration of tracking algorithms further enhances the ability to track and retrieve individuals. While this algorithm may sacrifice a small percentage of performance by occasionally missing trespassers, the notable improvement in processing speed is a valuable trade-off, especially in the context of real-time detection and notification for train engineers.

The latest iteration of the YOLO algorithm was successfully implemented on the most recent RGB datasets, incorporating a median filter. This combination yielded impressive results, achieving 90% event precision and recall on the RedSpring 1 dataset and 100% event precision with 95% event recall on the RedSpring 2 dataset.

As a potential avenue for future work, rail detection could be integrated into the system to filter out unimportant objects distant from the rails, allowing the algorithm to allocate more attention to significant objects in close proximity to the rails. Additionally, these cutting-edge object detection algorithms can play a crucial role in AI-assisted labeling tools, accelerating the annotation process and generating more labeled datasets. This, in turn, can contribute to the ongoing development of automatic pedestrian detection systems.

5. DISCUSSION AND REAL-WORLD DATA

The installation of the YOLOX algorithm on a new machine running Ubuntu Operating Software posed challenges for researchers at ITRE in preparation for receiving real-world data. The algorithm required a specific setup, known as an environment, which proved to be a complex task due to constant updates in various packages. The algorithm specifically functioned with environment packages specified to a particular version. Despite the difficulties, the algorithm was eventually installed correctly and underwent testing on randomly selected test videos. The initial attempt to process the data through the algorithm involved copying one terabyte of the four terabytes of data locally onto the machine. The machine was left running for several days to process this substantial amount of data.

However, an unknown error occurred during this process, leading to the corruption of the operating system on the machine. The cause of the failure remained undetermined, necessitating a complete wipe of the machine and a fresh reinstallation of the operating system. After this reinstallation, the environment was rebuilt, and the algorithm was once again installed. This time, the algorithm failed its initial quality check tests, detecting no humans at all. Despite multiple attempts at tweaking and fixing, it was concluded that the researcher who designed and calibrated the algorithm would need remote access to the machine, a challenging process for a Linux machine. The IT department at ITRE revealed that the machine was incompatible with the university's software architecture, prompting another complete operating system reinstallation.

Faced with numerous delays, student interns were assigned the task of manually identifying trespassers and recording key characteristics from all four terabytes of video. The process took longer than expected, and by the time the algorithm was ready, the students had already completed their task. Consequently, a decision was made to run the algorithm only on the videos where trespassers had been identified by students, reducing the workload from tens of thousands of video clips to just under three hundred.

After the second operating system reinstallation and the third installation of the environment and corresponding algorithm, videos identified with trespassers were successfully processed through the algorithm, generating output videos. Student interns were then tasked with evaluating the algorithm by going through the output videos and determining key characteristics.

Across all observed days, the students identified a total of 128 trespassers, with 53 having been inside the tracks at some point. Notably, the cameras capturing these trespassers were mounted on the front of a moving train. Twenty-nine of the trespassers were in groups of two or more, while the remaining 99 were trespassing alone. Fortunately, no pedestrian strikes were recorded, but a low-speed collision with a trailer occurred, failing to clear the tracks in time.

Despite the challenges in installation, the algorithm proved more thorough than the initial pass by student interns. The algorithm detected 252 true positive trespassers in the same time period, with 121 of them having been in the tracks within sight of the train. However, the algorithm cannot completely replace human labor, as it flagged 3,427 objects as pedestrians, of which only 786 (22.9%) were actual humans. Note: 534 of the 786 "trespassers" were identified as railroad workers when the train was turned on in the rail yard and were deemed "non-trespassers", leaving 252 "true trespassers" for evaluation.

Category	Student Pass (all data)	Algorithm Pass (Trespasser Sample)	Difference
Total Trespassers	128	252	124
Trespassers in Tracks	53	121	68

Table 7. CompariSON OF AI TO HUMAN ANALYSIS OF VIDEO

The implementation of the algorithm has proven to be crucial, as it prevented the loss of 49.2% of trespassers and 56.2% of those within the tracks, representing individuals engaged in the highest risk behavior. This finding underscores the significant impact of the algorithm on detecting and mitigating potential risks. Further research is warranted to delve into this phenomenon, understanding the nuances and implications of these results. Investigating the specific circumstances surrounding the detected and undetected trespassers could provide valuable insights for refining and improving the algorithm's performance. Additionally, assessing the algorithm's efficacy in different scenarios and conditions would contribute to a more comprehensive understanding of its potential applications and limitations. Last, although not discussed in great detail, removal of the train hood and increasing the pixilation should further increase the accuracy and precision of the algorithm.

6. CONCLUSIONS

The research team successfully identified 252 trespassers captured by cameras mounted on the front of trains, with over a hundred (121) found within the tracks while in the line of sight of the train. In comparison, manual methods only captured 128 trespassers, with only 53 found to be within the tracks. While the frequency of such trespassing behavior is certainly concerning, the fact that AI found significantly more trespassing behavior aligns with the research's objective of capturing and understanding dangerous actions by humans around trains.

Further research is deemed necessary for enhancing the efficacy of the detection system. One avenue is exploring alternative camera placements that would avoid occluding areas directly in front of the train. This might involve mounting cameras in a way that conceals the front of the train from their view. Additionally, capturing videos in higher resolution and/or at a higher frame rate could contribute to improving the detection algorithm's performance.

Efficient observation of all captured videos is another aspect that warrants exploration. Research into methods for managing and reviewing extensive video datasets, such as testing the detection algorithm on a sufficiently powerful computer or assessing smaller video samples at a time, could streamline the process. Furthermore, considering video capture in a time-lapse format may present another avenue for improvement in this regard.

In summary, ongoing research and exploration of various enhancements, including alternative camera placements, improved video quality, and more efficient observation methods, will contribute to the refinement and optimization of the detection system.

7. REFERENCES

1. Federal Rail Administration Office of Safety Analysis. Trespasser Causalities. Trespasser fatality data for 2017/2022. Retrieved from https://safetydata.fra.dot.gov/officeofsafety/publicsite/query/castally4.aspx. Accessed August 4, 2021.
2. Federal Rail Administration Office of Safety Analysis. Highway-Rail Incidents. Highway-Rail fatality data for 2018/2021. Retrieved from https://safetydata.fra.dot.gov/OfficeofSafety/publicsite/query/gxrtally1.aspx. Accessed August 4, 2021.
3. Operation Lifesaver. Trespassing Casualties by State. Retrieved from https://oli.org/about-us/news/statistics/trespassing-fatalities-by-state. Accessed August 5, 2021.
4. Federal Rail Administration Office of Safety Analysis. Trespasser Causalities. Trespasser fatality data for 2015/2020. Retrieved from https://safetydata.fra.dot.gov/officeofsafety/publicsite/query/castally4.aspx. Accessed August 5, 2021.
5. Searcy, S., Cunningham, C., and Vaughan, C. (2018). Rail Corridor Trespass Severity Assessment. Retrieved from https://connect.ncdot.gov/projects/research/Pages/ProjDetails.aspx?ProjectID=2017-15.
6. Searcy, S., Cunningham, C., and Vaughan, C. (2020). Rail Network Trespass Statewide Severity Assessment and Predictive Modeling. Retrieved from https://connect.ncdot.gov/projects/research/Pages/ProjDetails.aspx?ProjectID=2019-08.
7. USDOT Federal Railroad Administration. (2017). 2017 Grade Crossing Research Needs Workshop. DOT/FRA/ORD-19/0. Retrieved from https://railroads.dot.gov/sites/fra.dot.gov/files/fra_net/18353/2017%20Grade%20Crossing%20Workshop%20Report.pdf
8. John A. Volpe National Transportation Systems Center. (2021). Trespasser Prevention Research Study – Worcester, MA (2016-2018). Federal Railroad Administration, USDOT, https://railroads.dot.gov/sites/fra.dot.gov/files/2021-04/TPRS%20II-A.pdf.
9. Zaman, A., Liu, X., & Zhang, Z. (2018). Video Analytics for Railroad Safety Research: An Artificial Intelligence Approach. Transportation Research Record, 2672(10), 269–277. https://doi.org/10.1177/0361198118792751
10. Zhang, Z., Trivedi, C., and Liu, X. "Automated detection of grade-crossing-trespassing near misses based on computer vision analysis of surveillance video data," Safety Science, Volume 110, Part B, 2018, Pages 276-285, ISSN 0925-7535, https://doi.org/10.1016/j.ssci.2017.11.023.
11. Zaman, A., Ren, B., & Liu, X. (2019). Artificial Intelligence-Aided Automated Detection of Railroad Trespassing. Transportation Research Record, 2673(7), 25–37. https://doi.org/10.1177/0361198119846468
12. Institute of Transportation Research and Education. (2019). Pedestrian Incident Detection in the Rail Right-of-Way using Artificial Intelligence. NCDOT. Retrieved from https://connect.ncdot.gov/projects/research/RNAProjDocs/2019-50%20Final%20Report.pdf
13. Qingwen Xue, Ke Wang, Jian John Lu, & Yujie Liu. (2019). Rapid Driving Style Recognition in Car-Following Using Machine Learning and Vehicle Trajectory Data.

Journal of Advanced Transportation. Retrieved from https://doi.org/10.1155/2019/9085238
14. Combs, T.S., Sandt, L.S., Clamann, M.P. and McDonald, N.C., 2019. Automated vehicles and pedestrian safety: exploring the promise and limits of pedestrian detection. American journal of preventive medicine, 56(1), pp.1-7.
15. Chen, L., Lin, S., Lu, X., Cao, D., Wu, H., Guo, C., Liu, C. and Wang, F.Y., 2021. Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 22(6), pp.3234-3246.
16. Tian, D., Han, Y., Wang, B., Guan, T. and Wei, W., 2021. A review of intelligent driving pedestrian detection based on deep learning. Computational intelligence and neuroscience, 2021.
17. Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
18. Jiang, Y., Tong, G., Yin, H. and Xiong, N., 2019. A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters. IEEE Access, 7, pp.118310-118321.
19. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
20. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
21. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
22. Liu, W., Liao, S., Ren, W., Hu, W. and Yu, Y., 2019. High-level semantic feature detection: A new perspective for pedestrian detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5187-5196).
23. Zhang, S., Chi, C., Yao, Y., Lei, Z. and Li, S.Z., 2020. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9759-9768).
24. Li, G., Yang, Y. and Qu, X., 2019. Deep learning approaches on pedestrian detection in hazy weather. IEEE Transactions on Industrial Electronics, 67(10), pp.8889-8899.
25. Zhu, C., Chen, F., Shen, Z. and Savvides, M., 2020, August. Soft anchor-point object detection. In European conference on computer vision (pp. 91-107). Springer, Cham.
26. Dollar, P., Wojek, C., Schiele, B. and Perona, P., 2011. Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence, 34(4), pp.743-761.
27. Geiger, A., Lenz, P. and Urtasun, R., 2012, June. Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3354-3361). IEEE.
28. Zhang, S., Benenson, R. and Schiele, B., 2017. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3221).

29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
30. Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L. and Cucchiara, R., 2021. Motsynth: How can synthetic data help pedestrian detection and tracking?. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10849-10859).
31. Labelbox, "Labelbox," Online, 2022. [Online]. Available: https://labelbox.com
32. https://github.com/facebookresearch/detectron2
33. Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
34. Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
35. https://cocodataset.org/#home
36. Ge, Z., Liu, S., Wang, F., Li, Z. and Sun, J., 2021. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.
37. Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
38. Wojke, N., Bewley, A. and Paulus, D., 2017, September. Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649).
39. Brunelli, R., 2009. Template matching techniques in computer vision: theory and practice. John Wiley & Sons.
40. https://www.flir.com/oem/adas/adas-dataset-form/