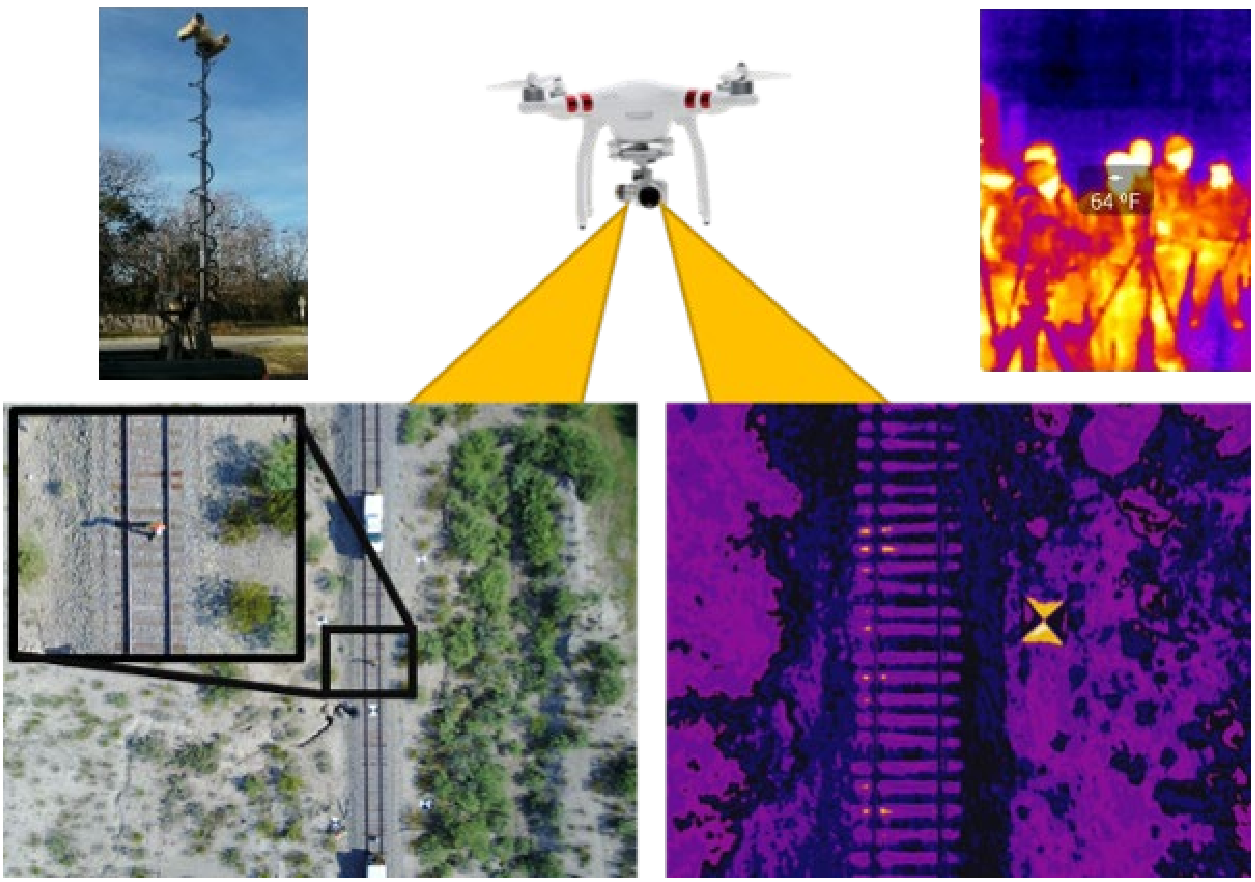
U.S. Department of Transportation
Federal Railroad Administration

Office of Research,
Development and Technology
Washington, DC 20590

Railroad Artificial Intelligence Intruder Learning System (RAIILS)



DOT/FRA/ORD-24/38

Final Report | September 2024

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
9/19/24	Technical Report	March 21, 2019–September 12, 2021

4. TITLE AND SUBTITLE
Railroad Artificial Intelligence Intruder Learning System (RAILS)

5a. CONTRACT NUMBER
693JJ619C000001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Erick Vega, ORCiD: 0000-0002-2400-7720
Pasi Lautala, ORCiD: 0000-0001-5552-5590
David Nelson, ORCiD: 0000-0002-5082-6340
Charlotte Weinstein, ORCiD: 0000-0001-9636-2919
Colin Brooks, ORCiD: 000-0003-4544-2569
Richard Dobson, ORCiD: 0000-0002-5699-154X

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Michigan Technological University
1400 Townsend Dr
Houghton, MI 49931-1200

8. PERFORMING ORGANIZATION REPORT NUMBER
USTRA-MTTI-RAIILS-09-12

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-24/38

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

13. SUPPLEMENTARY NOTES
COR: Francesco Bedini Jacobini

14. ABSTRACT
This report summarizes the research conducted as part of Phase 2 of the Railroad Artificial Intelligence Intruder Learning System (RAILS), led by a team of researchers at Michigan Technological University (Michigan Tech) and sponsored by the Federal Railroad Administration (FRA) from March 2019 to September 2021. The project had two primary objectives: (1) to help FRA understand what industry with respect to developing Artificial Intelligence (AI) models that could be applied to railroad trespass detection, and (2) to assist North American railroads in developing machine learning capabilities to detect railroad trespassers in real time. The Phase 1 report served primarily as a literature review and focused on addressing the first of these goals.

15. SUBJECT TERMS
Artificial intelligence, AI, machine learning, railroad, crossing, security, UAV, UAS, drone, automated

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		29	Erick Vega
U	**U**	**U**			**19b. TELEPHONE NUMBER** *(Include area code)* 734-994-7239

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC

LENGTH (APPROXIMATE)

1 inch (in)	=	2.5 centimeters (cm)
1 foot (ft)	=	30 centimeters (cm)
1 yard (yd)	=	0.9 meter (m)
1 mile (mi)	=	1.6 kilometers (km)

AREA (APPROXIMATE)

1 square inch (sq in, in^2)	=	6.5 square centimeters (cm^2)
1 square foot (sq ft, ft^2)	=	0.09 square meter (m^2)
1 square yard (sq yd, yd^2)	=	0.8 square meter (m^2)
1 square mile (sq mi, mi^2)	=	2.6 square kilometers (km^2)
1 acre = 0.4 hectare (he)	=	4,000 square meters (m^2)

MASS - WEIGHT (APPROXIMATE)

1 ounce (oz)	=	28 grams (gm)
1 pound (lb)	=	0.45 kilogram (kg)
1 short ton = 2,000 pounds (lb)	=	0.9 tonne (t)

VOLUME (APPROXIMATE)

1 teaspoon (tsp)	=	5 milliliters (ml)
1 tablespoon (tbsp)	=	15 milliliters (ml)
1 fluid ounce (fl oz)	=	30 milliliters (ml)
1 cup (c)	=	0.24 liter (l)
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 yard (cu yd, yd^3)	=	0.76 cubic meter (m^3)

TEMPERATURE (EXACT)

[(x-32)(5/9)] °F = y °C

METRIC TO ENGLISH

LENGTH (APPROXIMATE)

1 millimeter (mm)	=	0.04 inch (in)
1 centimeter (cm)	=	0.4 inch (in)
1 meter (m)	=	3.3 feet (ft)
1 meter (m)	=	1.1 yards (yd)
1 kilometer (km)	=	0.6 mile (mi)

AREA (APPROXIMATE)

1 square centimeter (cm^2)	=	0.16 square inch (sq in, in^2)
1 square meter (m^2)	=	1.2 square yards (sq yd, yd^2)
1 square kilometer (km^2)	=	0.4 square mile (sq mi, mi^2)
10,000 square meters (m^2)	=	1 hectare (ha) = 2.5 acres

MASS - WEIGHT (APPROXIMATE)

1 gram (gm)	=	0.036 ounce (oz)
1 kilogram (kg)	=	2.2 pounds (lb)
1 tonne (t)	=	1,000 kilograms (kg)
	=	1.1 short tons

VOLUME (APPROXIMATE)

1 milliliter (ml)	=	0.03 fluid ounce (fl oz)
1 liter (l)	=	2.1 pints (pt)
1 liter (l)	=	1.06 quarts (qt)
1 liter (l)	=	0.26 gallon (gal)
1 cubic meter (m^3)	=	36 cubic feet (cu ft, ft^3)
1 cubic meter (m^3)	=	1.3 cubic yards (cu yd, yd^3)

TEMPERATURE (EXACT)

[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION

Inches	0			1			2			3		4		5
Centimeters	0	1	2		3	4	5	6	7		8	9	10	11	12	13

QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO

°F	-40°	-22°	-4°	14°	32°	50°	68°	86°	104°	122°	140°	158°	176°	194°	212°
°C	-40°	-30°	-20°	-10°	0°	10°	20°	30°	40°	50°	60°	70°	80°	90°	100°

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

Acknowledgements

Francesco Bedini of the Federal Railroad Administration provided valuable technical direction on the project. In a series of interviews, industry experts provided important overviews on their deployment of technologies and methods to detect and deter trespassers. Their help is being acknowledged here and included:

- Association of American Railroads: members of the Grade Crossing and Trespass subcommittee including Jeff Moller
- Amtrak: Douglas Balk
- Brunswick (Maine) Police Department: Patrol Commander Thomas Garrepy and Sergeant Greg Mears
- Chicago Police Department: Mark Harmon
- Iowa Department of Transportation (DOT): Phillip Meraz
- Maine DOT: Nathan Moulton
- Michigan DOT: Jeannine Cleveland, Tina Hissong, Ryan Hoensheid, Al Johnson, Rob Lippert

Contents

Illustrations

Tables

Executive Summary

This report summarizes the research conducted as part of Phase 2 of the Railroad Artificial Intelligence Intruder Learning System (RAIILS), led by a team of researchers at Michigan Technological University (Michigan Tech) and sponsored by the Federal Railroad Administration (FRA) from March 2019 to September 2021. The project had two primary objectives: (1) to help FRA understand how Artificial Intelligence (AI) models could be applied to railroad trespass detection, and (2) to assist North American railroads in developing machine learning capabilities to detect railroad trespassers in real time. The Phase 1 report served primarily as a literature review and focused on addressing the first of these goals.

In Phase 2, the research team developed new knowledge, insights, and implementation tools to assist FRA in improving trespass detection systems using predictive analytics. Because AI concepts and predictive analytics are quite broad and often ill-defined, the particular AI concepts of interest were deep convolutional neural networks operating on resource-constrained robotic platforms. While recognizing that the fields of AI and analytics extend well beyond this class of models, the unique capabilities afforded by high-detection accuracy and hardware-enabled, real-time operation makes the use of neural networks an attractive tool for trespasser prevention efforts. In addition, this project focused on the systems engineering portion of the problem – namely how one goes from a collection of algorithmic tools that solve a very specific problem (i.e., detecting a person in an image), to a useful and integrated solution that addresses a discrete social problem: mitigating or eliminating trespassing.

The Michigan Tech team developed a robot-mounted system for automated trespasser detection. This system, consisting of synergistic sensing, algorithms, and communications, was integrated into both ground and air-based robotic platforms, with detections taking place in real time via an onboard computer. Upon detection, an email and/or text message containing an image of the potential trespassing event was automatically transmitted via cell networks. By explicitly constructing this proof-of-concept demonstration, the team showed that AI-enabled technologies have the potential to save lives by (1) automating detection of trespassing events in real time and (2) reducing the time required to respond to trespassing events by using existing communications infrastructure. This research effort resulted in a cost-effective tool that, with appropriate maturation, could serve to complement existing trespasser mitigation strategies.

1. Introduction

The use of unmanned vehicles is becoming increasingly common in the fields of remote sensing and transportation research. These platforms afford researchers the opportunity to collect data from vantage points that optimize the utility sensors such as cameras and light detection and ranging (LiDAR).

This report summarizes the research conducted as part of Phase 2 of the Railroad Artificial Intelligence Intruder Learning System (RAIILS), led by a team of researchers at Michigan Technological University (Michigan Tech) and sponsored by the Federal Railroad Administration (FRA) from March 2019 to September 2021. The project had two primary objectives: (1) to help FRA understand how Artificial Intelligence (AI) models could be applied to railroad trespass detection, and (2) to assist North American railroads in developing machine learning capabilities to detect railroad trespassers in real time.

1.1 Background

1.1.1 Trespassing

In 2018, FRA prepared a detailed report [2], in response to a congressional request on the causal factors that led to trespassing incidents on railroad property. This report found that rail trespass accidents cost society about $43 billion from 2012–2016. In the U.S., 42 percent of railroad-related deaths are a result of trespassing, as compared to 17 percent in the European Union [1]. Note that this number encompasses only the direct monetary costs associated with trespass events and does not account for emotional distress or productivity losses. Trespassing continues to be a serious problem for the rail industry, and the number of casualties having increased significantly since 2011. In 2020, the FRA Office of Safety Analysis reported 1,091 trespass casualties, with 536 of them fatalities [5]. Thus, there is a clear need for technologies that can alleviate both the human and financial costs of trespassing-related accidents.

1.1.2 AI and Autonomous Technology

There has been increased interest from Federal agencies on the use of AI and autonomous technologies in the railroad industry. In September 2021, FRA supported additional research on AI-enabled static cameras as well as the use of UAVs for rail grade monitoring. The present work fills research gaps on (1) real-time detection, (2) automated notification, and (3) integration into robot platforms.

The authors note that although there are several studies that feature the use of UAVs for several rail-related applications, the present work is unique in that the processing is taking place "at the edge."[1] This edge computing framework allows for minimal delay between data collection and data analysis, potentially improving the likelihood of timely intervention in trespassing events.

1.2 Objectives

As noted above, the overall project had the long-term objective to investigate technologies with a high potential of helping to reduce the number of trespasser deaths and injuries. The guiding

[1] Edge computing is a technical term that describes bringing computation and data storage closer to the data sources.

objective during Phase 2 was to assist in the development of machine learning techniques for real-time trespasser detection. In particular:

(a) Implement AI/analytics based techniques that had the potential to address the trespassing problem.

(b) Demonstrate the feasibility of adapting these techniques to an edge-computing framework by executing them on a mobile robot platform.

(c) Test the proposed solution in relevant real-world scenario.

In addition, the team developed two additional capabilities in support of these objectives – (1) the ability to transmit compressed video data to a server for additional processing and (2) partially autonomous navigation via a web-interface to the robots' navigation board. Figure 1 shows a schematic representation of the trespasser detection and alerting system.

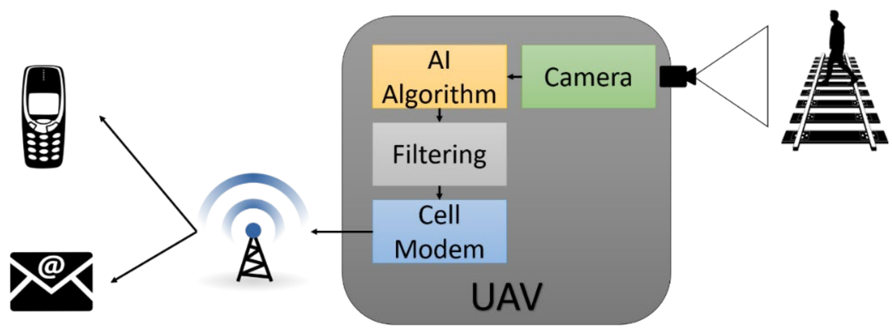


Figure 1. Component diagram of trespasser detection and alert system

1.3 Overall Approach

The Phase 1 literature review, including an overview of technical expertise within MTRI and existing technologies that might be applied to the problem, informed the Phase 2 approach. Initial steps taken toward development of a prototype system in Phase 1 also proved useful during Phase 2. Researchers started with benchtop testing with initial algorithms and sensors. They then moved to porting these algorithms and sensors on mobile platforms, including a unmanned ground vehicle (UGV) and unmanned air vehicle (UAV). They then performed real-world scenarios to test and improve upon these platforms.

The team considered the following trade-space variables when exploring initial design concepts for the AI-enabled UAV-based trespasser detection system:

1. Real-time detection
2. Asynchronous communication
3. Ease of use
4. Flexibility
5. Regulatory considerations

The sections that follow detail the rationale behind the design choices to address each of these requirements.

1.4 Scope

The objective of this research project was to develop a prototype system capable of detecting trespassing activities on the right of way of any given railroad using artificial intelligence capabilities, and by using a unmanned aerial vehicle.

1.5 Organization of the Report

Researchers presented the work performed in this report as follows: Section 1 introduces the work. Section 2 describes experiments performed and then an analysis took place in Section 3. Section 4 documents the real-world testing that occurred in Oscoda, MI, and Milford, MI. Section 5 summarizes the work conducted with concluding results.

2. Initial Benchtop Testing

2.1 Baseline Experiments

This section presents the baseline experiments used in this work.

2.1.1 Raspberry Pi

Baseline experiments made use of a popular single-board computer called the Raspberry Pi. These computers have a number of advantages including cost, widespread community support, and a large number of out-of-the-box compatible sensors. This computer was paired with an 8MP standalone camera module to support streaming data with minimal power draw. Although researchers found the actual computational capability of the Pi somewhat lacking for high-throughput video processing, the size, weight, and power (SWaP) considerations made the Raspberry Pi an attractive platform for possible deployment on a UAV. Table 1 shoes a summary of the typical specifications for the onboard hardware needed to perform the required tasks.

Table 1. Raspberry Pi Technical Specifications

Feature	Specification
SoC	Broadcom BCM2711
CPU	4x Cortex A72 @1.8GHz
Memory	Up to 8GB
Connectivity	Gigabit Ethernet, 2.4/5 GHz WiFi
I/O	28 × GPIO supporting either 1.8v or 3.3v
Power rating	600 mA (3 W) idle, 1.25 A (6.25 W) under stress

As part of the initial testing, researchers used the Darknet neural network architecture [1]. The Darknet architecture was one of the breakthrough algorithms in real-time person detection. In addition, with the software being written in C, the algorithm was amenable to deployment on a wide range of platforms – including those without a powerful graphics processing unit (GPU) such as the Raspberry Pi. Although this model has been superseded by new architectures tailor-made to run on resource-limited devices, Darknet architecture remains popular as a feature-extraction tool for a wide range of machine-vision applications.

Initial validation of the algorithm was completed offline; video data was collected and post-processed on the Raspberry Pi. Figure 2 shoes the Raspberry Pi and camera module used for the initial work. Video data was collected from atop a one-story building to simulate the viewing geometry from the drone. Researchers deliberately captured images with minimal stabilization to account for possible motion-induced blur during flight collections. Results from these experiments are shown in Figure 3.

Figure 2. Raspberry Pi and camera module used for initial benchtop testing

Figure 3. Results from initial testing with rooftop data collection: left – correctly detected people; middle – missed detection; right – incorrect detection

These initial experiments led to the following insights:

- Camera blur would not present a significant problem for the detection algorithm.
- The Raspberry Pi would likely have insufficient resources for real-time operation.
- Algorithm performance deteriorated as the viewing geometry approached a nadir look angle.

In addition, researchers observed empirically that front-facing individuals were generally better detected, as were those whose full, unobscured bodies were in the frame. They also observed a small but persistent false alarm rate. Prior experience with neural network models suggest that diagnosing the root cause of these algorithmic behaviors is generally quite difficult. The lesson here was the need for additional real-time post processing to mitigate these false alarms.

2.1.2 Jetson Nano

The initial experiments with the Raspberry Pi quickly highlighted the need for a more powerful computer to process the data. In particular, researchers found that to run in real time, the algorithms required a dedicated GPU. This requirement essentially limited the team to the Nvidia Jetson family of products. The Jetson line is a collection of single-board computers targeting Internet-of-Things and edge computing applications. Although there are other products that offer potentially similar performance such as field-programmable gate arrays, these are typically more difficult to debug and interface, making development slow and inefficient.

Researchers ultimately chose the Jetson Nano board as the target computing platform based on cost, and ran Ubuntu Linux natively. Software development was relatively straightforward as they could develop algorithms on a desktop Linux environment, and then simply copy the source files to the Nano. This rapid iteration was invaluable when they shifted to integrating the algorithms on the mobile platforms (see Figure 4).

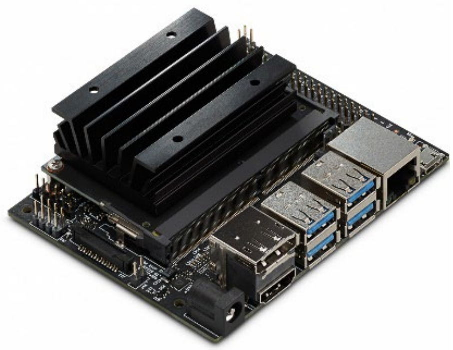


Figure 4. Jetson Nano board used for UxV experiments

Note that the specifications of the Jetson Nano are quite similar to those of the Raspberry Pi – with the key difference being the more powerful GPU on the Nano (see Table 2). For CPU-bound tasks, the Raspberry Pi is likely the better option. This distinction is relevant here as there are other non-neural network-based processing techniques (described in the next section) which might benefit from the Pi's higher clock speed.

Table 2. Jetson Nano Technical Specifications

Feature	Specification
SoC	Tegra
CPU	ARM Cortex-A57 MPCore Processor
GPU	NVIDIA Maxwell with 128 NVIDIA CUDA® cores
Memory	4GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Connectivity	Gigabit Ethernet
I/O	GPIO, I^2C, I^2S, SPI, UART
Power rating	10W @ 921 MHz GPU Clock

2.1.3 Detection Algorithm

Whereas the Raspberry Pi experiments used Darknet (sometimes also called YOLO) model as the backend for detection, when using the Jetson Nano, researchers chose a different model that was better able to use the GPU resources of the board. The MobileNet2 model that they used is a more modern single-shot detection algorithm with improved detection capabilities compared to the Darknet model [1]. The particular model that they used was trained using the COCO dataset and deployed using Nvidia's Jetson Inference API. This model also occasionally reported spurious/incorrect detections. As incorrect detections were typically not persistent across frames they chose to only report detections that were consistently present over 2 seconds' worth of data. This threshold yielded an acceptable tradeoff between detections and false alarms.

2.2 Thermal Camera Experiments

In parallel with the optical camera experiments, researchers also carried out small data collections to determine whether existing AI algorithms would function with thermal camera inputs. Thermal imagery has a number of phenomenological differences from conventional color imagery. The longer wavelengths of infrared radiation limits the maximum effective resolution of thermal imagery. In addition, most commercial thermal cameras capture only a single channel of data, making the data functionally incompatible with object detection algorithms that expect 3-channel color imagery. The team therefore expected the algorithm to altogether fail with thermal camera inputs.

As a simple test, researchers replicated the single-channel imagery to generate synthetic 3-channel imagery (i.e., with three identical channels). They then fed this false-color imagery into the Darknet architecture and found that (surprisingly) the algorithm could still detect people in the images (shown in Figure 5). One possible reason for this unexpected outcome is that many of the salient features that are useful for detecting people—such as facial features, limbs, and hair—were still visible in the thermal imagery. The team did not carry out an exhaustive investigation of this as the detection rate was generally lower than with the color image inputs. The comparatively lower data rate of thermal camera video, however, makes the modality a good match for other non-neural network-based processing.

Figure 5. Thermal camera data processed using Darknet architecture

2.2.1 Robust PCA and Thermal Camera Video

Robust principal component analysis (RPCA) is a numerical algorithm that originates from the field of compressed sensing [7]. Given an input matrix $\boldsymbol{X}$, the goal of RPCA is to write $\boldsymbol{X}$ as

$$\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$$

Where L is a low-rank matrix and S is a sparse matrix (that is, a matrix where only a small number of entries are nonzero) (see Figure 6).

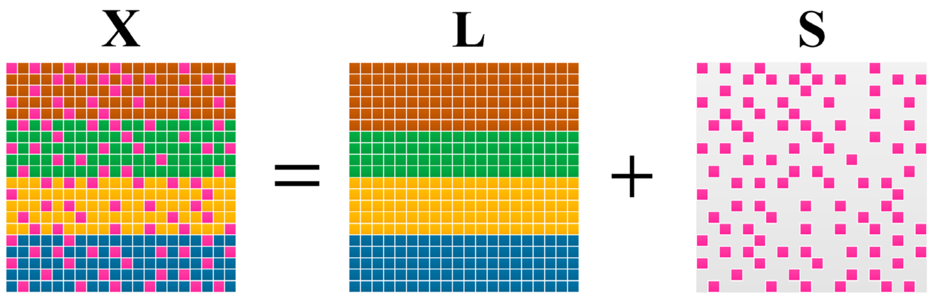


Figure 6. RPCA decomposition which writes an input matrix *X* as the sum of low rank and sparse matrices

In the context of video processing, RPCA and related algorithms are commonly used for background subtraction [7]. In this setting, one typically considers each frame of a video as corresponding to a single column of the matrix X. In other words, each frame is "flattened" into a long vector by lexicographically listing the color values of each pixel in the image. For the problem of trespasser detection, the team considered RPCA as a tool for working with static thermal cameras. In particular, for a camera observing a locally stationary scene, potential trespassers will stand out against the stationary background of the scene.

Note that although this technique lacks the granularity of the neural-network based approach in that it can only detect that there are foreground objects, it is considerably better understood and has significantly fewer tuning parameters. This technique has many benefits over classical computer vision techniques both in terms of performance and computational complexity. In particular, recent algorithmic developments for computing the RPCA decomposition make it suitable for real-time processing of video data (see Figure 7).

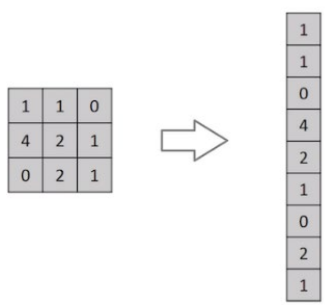


Figure 7. Flattening procedure for processing video data with RPCA

For project experiments, the team used the iterative soft-thresholding algorithm for computing RPCA. This algorithm, while not the fastest, admits a straightforward implementation from which researchers could easily validate the usefulness of the technique. An example of using this technique to process a video containing a simulated trespassing event is shown in Figure 8. The image on the right corresponds to a single column of the sparse S matrix reshaped back into an image. The sparse matrix was thresholded to yield a logical mask that could then be used to index into the original image. Note that it is possible to apply morphological operations to the mask to obtain larger "blobs" that fully contain the potential trespasser.

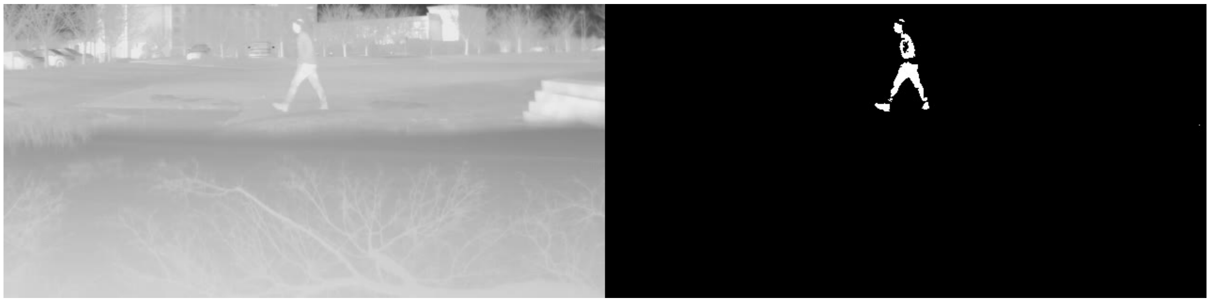


Figure 8. Left – thermal camera input; right – sparse component of RPCA decomposition of video data

One possible use case for this technique is for monitoring multiple areas of interest using static cameras. As this technique does not require the use of high-end GPUs, it could be readily adapted to run on existing computing resources. This technique does require a human-in-the-loop, however, as output masks are not labeled with an object class. Lastly, note that this technique is also usable with static color/optical cameras, which makes multimodal collections synergistic with this approach.

3. Transition to Mobile Platforms

3.1 Trade Space Analysis

Armed with knowledge of the performance of the detection algorithms, researchers next moved to the systems engineering problem of creating a usable product that used these on the backend. As part of their initial planning, they considered the following trade space analysis shown in Table 3.

Table 3. Design considerations and proposed solutions for UAV-based trespasser detection system

Design Consideration	Proposed Solution
Real-time detection	GPU-enabled onboard computer
Asynchronous communication	Detection-driven events, cell-network modem
Ease of use	Require user to only fly UAV, no additional task-switching
Flexibility	Use primarily open-source, COTS components
Regulatory considerations	Use federally-vetted platform, optional OS-level hardening

The first of these was driven from the very practical consideration that minimizing response time is critical for producing favorable outcomes in trespass events. Asynchronous communication here means that one would only want to alert the operator(s) when a potential issue arises (in other words, event-driven messages from the UAV). In addition, the system must be easy to use to have any chance of being adopted by non-expert users (in particular users who are not experts in AI/predictive analytics). Such a system should also be flexible to allow for on-the-fly repairs and extensions. Lastly, the system ought be designed in accordance with local, State, and Federal regulations.

3.2 Implementation

The specific hardware/software choices that the team ultimately settled on is given in Table 4.

Table 4. Specific component choices

Component	Chosen Instantiation	Cost
UAV Platform	Uvify IFO-S	$6,000
Onboard computer	Nvidia Jetson Nano	$110
Camera	Intel RealSense D435	$220
Operating System	Ubuntu Linux	No Cost
Communications	Verizon Wireless cell network modem	$250

Each component was deliberately chosen and was meant to address one or more of the design constraints from Table 3. The UAV platform, for example, is one used by the Defense Advanced Research Projects Agency as part of its Offensive Swarm Enabled Tactics Program – meaning it was authorized by the U.S. Department of Defense. In practical terms, this has the effect of making it easier to gain approval for its use within other State/Federal agencies. The onboard computer has among the best SWaP characteristics, is cost-effective, and enables the execution of deep neural network architectures in real time. The use of Linux makes the system quite flexible and offers a high degree of re-configurability. Lastly, the use of a standalone cell-network modem means the system can function wherever there is cell phone coverage. (Note: the communications can also function over a local network, such as a Wi-Fi mesh.). The images in Figure 9 capture the above discussion of the components used.

This particular set of components is well-suited for the specific problem of detecting trespassers from a moving platform in real time. When some of these constraints are relaxed by, for example, specifying the camera will be static or the video data will be post-processed not necessarily in real time, alternative choices may offer competitive advantages.

Figure 9. Left – Uvify IFO-S UAV; right – wireless cell dongle for transmission of trespasser detections

3.3 UGV Implementation

Near the end of the project, researchers also ported the algorithms to an Aion Robotics R4 UGV (Figure 10). The R4 comes standard with a Jetson Nano computer, so porting the algorithms to this platform was relatively straightforward. They designed and 3D printed a mount for the camera and attached the cell modem to the outside of the robot chassis.

Figure 10. Aion Robotics R4 UGV

The UGV has the advantage of a longer run time, as the power draw is significantly lower when the robot is not actively in motion. As such, the potential use cases for the UGV over the UAV are situations where the region of interest can be adequately surveyed from the ground and for which the dynamics evolve slowly over the time scale of, e.g., 1 hour. These requirements would allow for the robot to be piloted into the region of interest and surveil/report (with minimal energy expenditure) – while allowing the operator to adjust the view as needed. The drawback, of course, is that the viewing geometry is not ideal for surveilling large areas. Additionally, the UGV requires somewhat benign terrain, although as discussed in the subsequent section, researchers operated the UGV on the track shoulder without difficulty.

4. Real-World Testing

4.1 Oscoda, MI

The first test site to validate the UAV mounted system (the UGV had not yet been fully integrated at the time of this test) was in Oscoda, MI. This site was chosen in consultation with project collaborators from Lake State Railway. The site featured a portion of decommissioned rail line owned by Lake State Railway – the fact that the line was decommissioned afforded researchers a great deal of freedom in safely testing different viewing geometries which would not be possible on active lines. An overhead view of the site as well as the relative location of the site is shown in Figure 11.

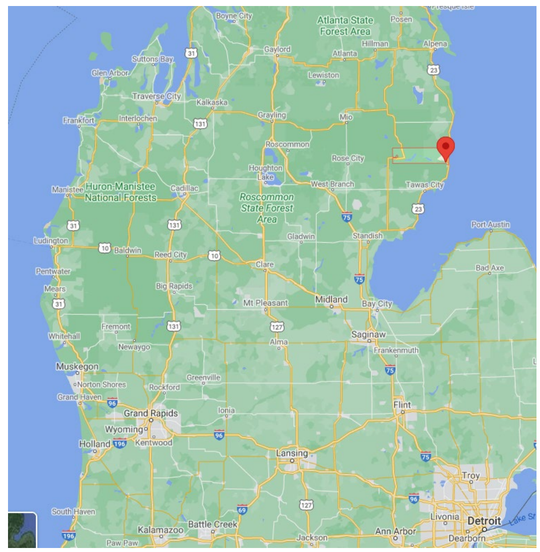


Figure 11. Left – aerial view of Oscoda testing site; right – relative geographic location of site

4.1.1 Regulatory Considerations

Certain geographic locations have more stringent permitting requirements than others when it comes to operating UAVs. This particular site had a small municipal airport in the vicinity, which meant that the altitude limit of the UAV was slightly reduced. Although not strictly a requirement, the research team contacted the airport to let them know when and where they would be operating the UAV. Multiple members of the research team were FAA certified to pilot the UAV. For a production trespasser detection system, additional flight regulations may apply. Although not of a technical nature, these types of hurdles can have a significant impact on the success of emerging technologies.

4.1.2 Testing Framework

The testing team consisted of three individuals designated as the pilot, trespasser, and lookout. The pilot had not been on the team that developed any of the software for the UAV and so could act as something of an impartial judge on the usability of the system developed. Actions the trespasser was asked to perform were:

1. Walking across the track

2. Walking along the track
3. Loitering in the middle of track
4. Walking beside the track

For each of these scenarios, the pilot was asked to have the UAV:

1. Hover in-place.
2. Turn in-place.
3. Approach the trespasser.

The combination of these trespasser-UAV actions provided a qualitative sense of the performance envelope of the system.

4.1.3 Results

Researchers generally observed favorable detection performance despite a relatively low-resolution sensor. The system could also successfully transmit images with the detected trespasser over the cell network. Detection was generally most effective when the height of the UAV was lower than 15 feet. Representative images, as generated by the onboard computer, are shown in Figure 12.

Figure 12. Representative outputs from UAV-based trespasser detection system

Researchers did, however, observe a few undesired characteristics of the system, which they sought to correct before the next deployment. First, when a string of detections was made in quick succession, the system was programmed to send out a correspondingly large number of emails. In this situation, there was at times a large delay before the last image in the collection was received. Second, collecting the data for post-analysis was not very efficient, as it required researchers manually download the image attachments from each individual email. Third, as the data was not being recorded on the UAV itself, they had no quick means of verifying the integrity of the data when there were large delays in the email transmission.

4.1.4 Algorithmic Revisions

Rather than transmitting the data to an email address, the team determined that sending the data directly to a dedicated storage server would allow for reduced latency. To this end, they used one of their public-facing MTRI servers as the client for the incoming images. The authors note that one could easily set up a small, lightweight computer (even a Raspberry Pi) to act as the logger for the incoming data. The advantage of using an on-site logger is that server-side latency is reduced, meaning that the person in charge of responding to the trespassing incident receives an alert faster.

Researchers also adapted the software so that the transmitted data would also be locally saved on the onboard storage. The data was compressed and stored as a video file, as this allowed for reduced CPU usage and also preserved storage on the local SD card.

4.2 Milford, MI

The Milford, MI, site was identified as a well-known trespassing hotspot by Lake State Railway. Before testing, the team visited the site and confirmed that trespassers were indeed quite common. In the team's brief 20-minute survey, they observed no fewer than four groups of people cross the tracks. Somewhat disturbingly, although there was a "No Loitering" sign next to the track, there was also a set of conveniently placed stairs for anyone crossing the tracks (see Figure 13.

Figure 13. Trespassing hotspot in Milford, MI, featuring easy-access stairs beside a no loitering notice

4.2.1 Testing Framework

Because this test was conducted on an active line, the Michigan Tech team had two representatives from Lake State Railway present to ensure adherence to appropriate safety protocols. Again, members of our team acted as trespassers while one person piloted the UAV and another monitored the incoming data transmission from an internet-connected tablet (see Figure 14). For these tests, the team flew the same UAV that was used for the Oscoda collection, as well as the newly outfitted UGV, and a DJI Mavic UAV with thermal sensing capabilities.

Figure 14. Left – simulated trespassers cross tracks at known hot-spot location; right – MTRI team demonstrates real-time receipt of data to Lake State Railway staff member

4.2.2 Results

Detection performance was comparable to what was observed in Oscoda, suggesting a degree of uniformity across sites. Researchers consistently verified detections from the UAV with a delay typically on the order of 90 seconds. They verified that the UGV could operate on the shoulder of the track and collect data suitable for the identification of trespassers. An example image collected from UGV is shown in Figure 15.

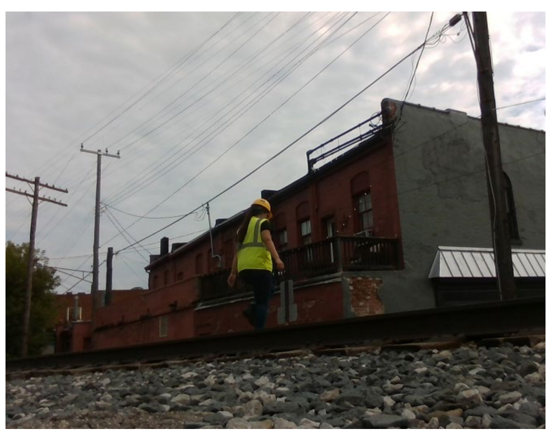


Figure 15. Simulated trespasser captured using UGV-mounted camera

Thermal imagery was collected at heights of 15, 25, 50, and 100 feet (see Figure 16). The qualitative characteristics of the data suggest that the thermal imagery collected at 15 and 20 feet are of potential use for automated trespasser detection. The data collected at the higher altitudes might still be of use to human analysts, but likely bump into the current limits of what is algorithmically possible for automated detection. In particular, at high altitudes, even using a high resolution camera a person in an image might only occupy tens of pixels. With so few pixels, current machine vision algorithms have difficulty making correct classifications.

Figure 16. Simulated trespassers captured using thermal camera on UAV operating at an altitude of 25 ft.

5. Conclusion

This project demonstrated that UAV/UGV platforms with onboard computing capabilities can be used to detect potential trespassers. The system developed as part of this effort sends an email or text message to a monitoring authority within 90 seconds of detection from the mobile platform. The detection algorithm has a neural network backend, which operates in real time and transmits information about detections via cell networks. In addition, the technology developed as part of this effort is also suitable for deployment with static camera inputs.

5.1 Roadmap to a Production System

The technology developed as part of this effort provides a proof-of-concept demonstration. To deploy this technology as part of a production system, significant development effort is required to estimate a comprehensive performance envelope. A non-exhaustive list of issues to address includes:

1. Sensitivity to lighting/environmental conditions
2. Tuning of detection parameters
3. Resolution of edge cases
4. Optimizing transmission latency
5. Data warehousing/logistics

Some of these issues are partially relevant to the research realm, while others are simply software/engineering design. There is also a higher-order trade analysis to be done that ought to weigh the benefits and drawbacks of UxV-based remote sensing with static camera analysis to find the optimal pairing of the two. In addition, one might also consider how to make this technology more broadly available, by, for example, providing “kits” to any interested parties.

6. References

[1] Redmon, J. (2016). "Darknet: Open Source Neural Networks in C."

[2] Federal Railroad Administration (2018). "National Strategy to Prevent Trespassing on Railroad Property."

[3] Bouwmans, T., Serhat Aybat, N., &Zahzah, E.-h. (2016). Handbook of Robust Low-Rank and Sparse Matrix Decomposition. New York: Chapman and Hall/CRC.

[4] Topel, K., Savage, I., Havârneanu, G. M., Gabree, S., DaSilva, M., Martino, M., &Warner, J. (2019). "A Literature Review of Rail Trespassing and Suicide Prevention Research," Transportation Research Circular (Report No. E-C42). Washington, DC.

[5] Federal Railroad Administration Office of Safety Analysis (n.d.). "2.0 Trespasser Casualties," Washington, DC: Federal Railroad Administration.

[6] Sandler, M., Howard, A., Zhu, M., Zhmoginov A., & Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings o f the IEEE Conference on Computer Vision and Pattern Recognition*.

[7] Candès, E., Xiaodong, L., Yi, M., & Wright, J. (2011). "Robust principal component analysis?," *Journal of the ACM (JACM),* p. 1–37.

Abbreviations and Acronyms

ACRONYM	DEFINTION
AI	Artificial Intelligence
BNSF	Burlington Northern Santa Fe Railway.
BTS	Bureau of Transportation Statistics
BVLOS	Beyond Visual Line of Sight
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
DOT	Department of Transportation
FAA	Federal Aviation Administration
FRA	Federal Railroad Administration
IR	Infrared
ML	Machine Learning
NAS	National Academies of Science
NR	Network Rail
PD	Police Department
RESTRAIL	Reduction of Suicides and Trespasses on Railway Property
RPN	Region Proposal Network
SA	Situational Awareness
TRID	Transportation Research International Documentation
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UK	United Kingdom
UP	Union Pacific