Smart and Resilient Transportation

Artificial intelligence aided detection and analysis of vehicular stop on track at railroad grade crossings

Huixiong Qin, Asim Francis Zaman, Sarthak Jain and Xin Wang
Department of Civil and Environmental Engineering, Rutgers The State University of New Jersey, Piscataway, New Jersey, USA, and
Xiang Liu
Department of Civil and Environmental Engineering, Rutgers University, New Brunswick, New Jersey, USA

Received 28 July 2025
Revised 17 September 2025
Accepted 17 September 2025

Abstract

Purpose – This paper aims to address this rail safety issue through collection and analysis of video data at grade crossings using artificial intelligence. Railroad safety is vital to transportation and the economy, particularly in the USA, where highway-rail grade crossings are hotspots of accidents, 9% of which involve vehicles stopping on tracks.

Design/methodology/approach – This research develops a robust artificial intelligence (AI)-assisted system to detect stopped-on-tracks incidents by overcoming unique challenges in the scene of grade crossings via neural architecture search and dynamic trajectory filtering.

Findings – The system achieves 94% precision in stopped-on-tracks detection and 95% recall in traffic counting. The findings suggested that road markings in the first case study facilitate incident reduction up to 50%, while a targeted intervention could potentially decrease stopped-on-tracks incidents by 80% in the second case study.

Social implications – This AI-based approach will potentially enhance rail safety and enable more informed decision-making in rail infrastructure management by providing insight on stopped-on-tracks behavior.

Originality/value – The model's accuracy was evaluated using the test set. Its effectiveness was further validated through two case studies: one assessing the impact of road markings (e.g. dynamic envelopes) on reducing stopped-on-tracks incidents and another examining the correlation between different types of traffic congestion and these incidents.

Keywords Artificial intelligence, Object detection and tracking, Highway-rail grade-crossing, Railroad safety, Trajectory analysis

Paper type Research paper

1. Introduction

Highway-rail grade crossing incidents can lead to fatalities, train derailments, train delays, increased traffic congestion, damage to infrastructure and spillage of hazardous materials. For example, in July 2024, a taxi driver was killed by an oncoming train at a railroad crossing in Manorville, NY, while the taxi was stopped in the middle of the track after driving through

Smart and Resilient Transportation
Emerald Publishing Limited
e-ISSN: 2632-0495
p-ISSN: 2632-0487
DOI 10.1108/SRT-07-2025-0022

a lower crossing gate (Zanger *et al.*, 2024). Similarly, in Moorpark, CA, in June 2023, an Amtrak train derailed after it struck a public works water truck, leading to the hospitalization of 16 people (Olson, 2023). Stopped-on-tracks incidents continue to be a leading cause of grade-crossing accidents. A Federal Railroad Administration (FRA) safety database indicates that 4,698 (9%) of 52,630 grade-crossing accident records with narrative descriptions were attributed to stopped-on-tracks incidents (Department of Transportation – Data Portal, 2024).

A stopped-on-tracks incident is defined as an occurrence where one or more vehicles stop within approximately 15 feet of the outer rail within a highway rail grade crossing and remain there for a period of 5 seconds (Assembly, 2024). Such incidents are often caused by traffic lights, construction or traffic congestion. Stopped-on-tracks vehicles are also at high risk of being struck by a train. According to the National Highway Traffic Safety Administration (NHTSA), a freight train can "take up to a mile or more – the length of 18 football fields – to stop" and a light rail train can "require about 600 feet – the length of two football fields" (NHTSA, 2024). It is for this reason that the NHTSA enforces stopping 15 feet away from flashing red lights, lowered gates or signaling flagmen or at a stop sign at a grade crossing.

Grade-crossing accidents may be prevented through the implementation of engineering, enforcement or education campaigns. The installation of video cameras can facilitate data analysis and provide the means for before-and-after evaluation to verify the effectiveness of implemented solutions. The Fixing America's Surface Transportation Act of 2015 mandated the installation of cameras around passenger rail lines to enhance safety, and in 2016, the FRA expanded this effort by encouraging states to equip traffic lights connected to railroad crossing with camera systems. This would allow the states to obtain valuable information during inspections that could be used to improve safety. If followed, these measures would likely prove to be beneficial as there are more than 5,000 railroad crossings interconnected with traffic lights (Railroads, 2024).

Currently, videos from camera systems are reviewed manually. This process can be labor-intensive and expensive, potentially leading to missed stopped-on-tracks incidents. Fortunately, with recent rapid advancements in artificial intelligence (AI), computer vision (CV) technology has significantly progressed, allowing for the rapid and automatic completion of repetitive tasks. Collecting this data manually would require prohibitively large amounts of labor.

This research presents a new system with improved performance of grade-crossing surveillance, especially in detecting stopped-on-tracks incidents, enabling the use of advanced techniques to reduce manual effort. Through two case studies, the system demonstrates its potential to identify patterns and risk factors associated with such incidents, contributing to the transportation network topologies optimization in rail systems, such as Akopov and Beklaryan, (2025).

2. Literature review

The team conducted a comprehensive literature review to understand the recent progress in stopped-on-tracks detection. While traffic monitoring at roadway intersections shares some techniques with grade-crossing surveillance, the latter poses more demanding requirements because it involves both roadway and railroad elements. Therefore, given the length constraints, the literature review concentrates on studies specifically related to grade-crossing surveillance.

In the field of railroad stationary surveillance, several previous studies have focused on the scenario of grade-crossing traffic monitoring. One study (Amin *et al.*, 2024) uses YOLO variants for object detection and UNet for segmentation, achieving promising accuracy at a cost of latency. Since this data set only consisted of images from 10-h videos in a single location, models are overfitted to the recurring scene. Other studies address data challenges tailored to specific use cases. For example, Guo *et al.* (2022) implement object detection using CNN-based models enhanced with an attention mechanism, improving accuracy for dense traffic scenes. However, their generalization capability is uncertain, as both the training and evaluation processes rely on a proprietary data set containing 2,358 images. Tang *et al.* (2023) adapts segmentation trained with weak supervised learning and enhanced with background generation to detect grade-crossing trespassers. But the enhancement in accuracy comes at the expense of increased latency. Some researchers have devoted themselves to the scenario of right-of-way obstacle detection (Gong *et al.*, 2022; Klammsteiner *et al.*,2023; Rahman *et al.*, 2022). Because the scene captured by stationary surveillance cameras is fixed, obstacle extraction is easy. For instance, background differences can separate the background and the target. However, monitoring the entire line using stationary cameras requires many devices and is subject to the obvious limitations in transmission, processing and analysis of massive amounts of data and equipment maintenance. A comparison of these previous studies on both scenarios is presented in Table 1.

Based on the literature review, previous studies have not focused on detecting and analyzing stopped-on-tracks behavior. In addition, many prior works claim contributions by addressing specific challenges in the transportation field through the adoption of emerging neural components from the CV field, without providing justification for the selection of that component. Furthermore, few existing solutions have been validated through real-world

Table 1. Previous studies in vision-based grade-crossing monitoring

Reference	Methodology	Limitations
Amin *et al.* (2024)	Combine various YOLO models with UNet to detect traffic participants and the grade crossing area for enhanced safety	1) The use of a data set with repeating images for both training and testing raises concerns about potential overfitting of the model; 2) the combination of YOLO and UNet is computationally expensive
Gong *et al.* (2022)	Propose an enhanced few-shot learning solution for intrusion detection to mitigate data imbalance	1) Although suggested solutions achieved high accuracy, they are tested using limited data, which may affect their usefulness in the long run; 2) model iterations are based on manual work, which makes it time-consuming and labor-heavy; 3) the suggested solutions are only evaluated on proprietary data sets or in a single location, raising concerns about their extensibility
Klammsteiner *et al.* (2023)	Propose a railway track monitoring system based on Yolov5 for detection and BiSeNetv2 for segmentation to detect obstacles on various track areas	
Rahman *et al.* (2022)	Explore MobileNetv2 for obstacle detection at rail crossings through data augmentation and transfer learning	
Guo *et al.* (2022)	1) Propose a transformer-based model for traffic detection in the occluded scenes; 2) provide performance under bad video quality such as extreme weather condition	The proposed solutions are evaluated solely on proprietary data sets with 2,358 images which are all from a single location, raising concerns about its generalizability
Tang *et al.* (2023)	1) Adapt segmentation enhanced with SuBsense for background generation to detect grade-crossing trespassers; 2) training the model with weak supervised learning to mitigate the lack of annotated data	The solution combines background generation with semantic segmentation, leading to high latency

Source(s): Created by authors

applications. This research advanced the detection of stopped-on-tracks incidents at grade crossings by adapting existing DL techniques and incorporating post-processing of vehicle trajectories. The main contributions of this paper are as follows:

- A coarse-to-fine neural architecture search (NAS) algorithm is proposed to efficiently identify the most suitable existing models through early pruning and bandit-based optimization.
- A data-driven thresholding algorithm is developed to determine whether a vehicle has stopped within the track area, even under imperfect trajectory data.
- Two case studies are presented to demonstrate the potential application of stopped-on-tracks detection supported by insightful analysis and safety measurement recommendations.

3. Methodology

To detect vehicles stopped on tracks, it is necessary to detect cars, track them and analyze their trajectories to determine whether they stopped within the grade crossing. As presented in Figure 1, the video stream is initially processed by an object detector, which identifies objects of interest and outputs their bounding boxes. These bounding boxes are then linked across consecutive frames using an object tracker to generate object trajectories. Finally, these trajectories are analyzed using a data-driven thresholding algorithm to determine whether any represent stopped-on-tracks incidents.

In this section, the theoretical derivation of these models is briefly explained to support the discussion of NAS. Then, a coarse-to-fine NAS framework for identifying the most effective model among existing state-of-the-art options is outlined. Building on the selected

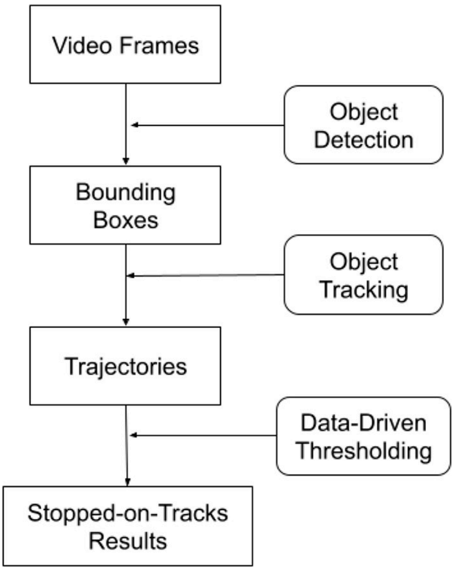


Figure 1. Stopped-on-tracks inference detection framework
Source(s): Created by authors

detector and tracker, a custom algorithm is proposed to robustly detect stopped-on-tracks incidents by addressing the challenges of imperfect trajectories through a data-driven thresholding approach.

3.1 Object detection

Based on the literature review and public benchmarks, detectors using the vision transformer (ViT) architecture demonstrate superior performance on open-source data sets, outperforming models based on other architectures by up to 10 points in commonly used evaluation metrics, such as average precision. Therefore, this paper focuses on exploring ViT-based models.

The classic ViT architecture for object detection is the detection transformer (DETR). As shown in Figure 2, DETR approaches object detection as a direct set prediction problem, where the model predicts a fixed number of objects and directly outputs their bounding boxes and class labels.

The input image I is first fed to a convolutional neural network CNN to extract the low-level feature map with C channels. Then the feature map is flattened to a two-dimensional feature sequence X using equation (1):

$$X = flatten(CNN(I)) \tag{1}$$

where $X \in R^{N \times C}$, N is the length of the feature sequence.

A positional encoding P is added to the feature sequence to retain spatial information, as shown in equation (2):

$$Z_0 = X + P \tag{2}$$

where $P \in R^{N \times C}$.

The feature sequence is then passed through a transformer encoder *Encoder* for l rounds [equations (3)–(6)], which consists of a self-attention layer *Self Attention*, a feed-forward network (*FFN)* and normalization layers *norm*:

$$Z_i = Encoder(Z_{i-1}) \;\; i = 1, 2, \ldots, \;\; l \tag{3}$$

$$Encoder = norm \cdot FFN \cdot norm \cdot Self\ Attention \tag{4}$$

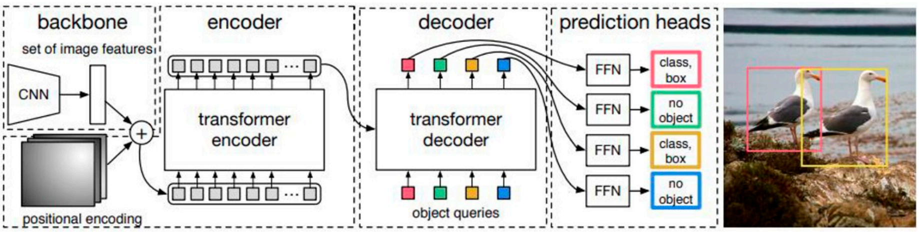


Figure 2. DETR model architecture
Source(s): Figure courtesy of Carion *et al.* (2020)

$$FFN(Z) = activation(W_{FFN}Z + b_{FFN}) \tag{5}$$

$$Self\ Attention(Z) = softmax\left(\frac{W_Q Z (W_K Z)^T}{\sqrt{d}}\right) W_v Z \tag{6}$$

where $Q = W_Q Z$, $K = W_K Z$, $V = W_v Z$. W_Q, W_K, W_V and $W_{FFN} \in R^{d \times N}$ are learnable weight matrices. d is the embedding dimension.

After the sequence is encoded, it is delivered to the decoder *Decoder* to fuse with a fixed set of objects embedding $Y_0 \in R^{C \times M}$ for k rounds [equations (7)–(9)]. The decoder consists of a cross-attention layer *Cross Attention*, *FFN* and normalization layers *norm*:

$$Y_j = Decoder(Y_{j-1})\ \ j = 1, 2, \ldots, \ \ k \tag{7}$$

$$Decoder = norm \cdot FFN \cdot norm \cdot Cross\, Attention \tag{8}$$

$$Cross\, Attention(Y) = softmax\left(\frac{YW_Q (W_K Z_l)^T}{\sqrt{d}}\right) W_v Z_l \tag{9}$$

The output of the last decoder layer is passed through a FFN to produce the final class labels equation (10) and bounding box equation (11) – a four-digit vector (x, y, w and h) representing the x coordinate of the center, the y coordinate of the center, the width and the height, respectively:

$$BBox = FFN_{bbox}(Y_k) \in \mathbb{R}^{M \times 4} \tag{10}$$

$$Class = FFN_{class}(Y_k) \in \mathbb{R}^{M \times (K+1)} \tag{11}$$

where K is the number of classes of interest. *Class* contains the class probabilities for each object, including a “no object” class.

3.2 Object tracking

Once objects of interest are detected as bounding boxes, the next step is to match instances of the same object across frames, a process known as data association. The fundamental architecture of detection-based trackers involves extracting appearance descriptors, predicting object-wise future trajectories and then matching detections in the current frame with the predicted trajectories.

Let $t = 1, 2, \ldots, T$ denote the time steps (where each step corresponds to a new frame) in the video, $Track_t = \{Track_t^0, \ Track_t^1, \ \ldots, \ Track_t^m\}$ denotes the set of trajectories at time t. Upon initiation, a statistical state estimator, such as Kalman Filter, is used to predict the $BBox_{t+1}$ based on $Track_t$ equation (12):

$$\widehat{Track}_{t+1} = KalmanFilter(Track_t) \tag{12}$$

During the data association phase, the estimated trajectories $\widehat{Track}_{t+1}$ are combined with the newly detected bounding boxes $BBox_{t+1}$ to form an assignment problem via equations (13) and (14). By solving this assignment problem, using such as the Hungarian algorithm, the updated trajectories $Track_{t+1}$ are obtained:

$$Track_{t+1} = \text{Hungarian}(\text{C}) \tag{13}$$

$$C = Cost\left(\widehat{Track}_{t+1},\quad BBox_{t+1}\right) \in \mathbb{R}^{n \times m} \tag{14}$$

where n is the number of trajectories in time t and m is the number of detections in time $t + 1$.

Some advanced tracking algorithms modify the assignment problem to achieve theoretically superior performance. For instance, DeepSORT (Wojke *et al.*, 2017) incorporates deep features, extracted by a selected CNN, into the cost function *Cost*, facilitating the object re-identification after occlusions. In addition to incorporating deep features, ByteTrack (Zhang *et al.*, 2022) performs data association twice to maintain object identities even in crowded or occluded situations. It first matches $\widehat{Track}_{t+1}$ with high-confidence detections and then retries any unmatched trajectories using low-confidence detections.

3.3 Coarse-to-fine neural architecture search

Given the variety of object detection and tracking algorithms, each with its own set of advantages, it is essential to select the best one for a specific use case. NAS algorithms are typically used to enumerate all combinations of model architecture, hyperparameters and training configurations to identify the optimal model. For instance, Nvidia applies NAS to optimize its large language model, Llama-Nemotron (Bercovich *et al.*, 2025). To avoid conducting an enumeration-based search, the proposed NAS algorithm incorporates early pruning and bandit-based algorithm to reduce computational complexity.

As illustrated in Figure 3, the NAS process consists of two stages:

(1) model selection for coarse filtering; and

(2) hyperparameter tuning and training configuration for fine filtering.

Selecting the best model from available options is the key to achieving optimal performance. Commonly tuned hyperparameters include batch size, dropout rate, image size and learning rate – each of which may or may not have a significant impact on performance depending on the task and data set. Default values commonly used in these hyperparameters are applied during the initial stage. Optimization is an important training configuration. While stochastic gradient descent and Adam (Kingma and Ba, 2017) are commonly used. Adam (Loshchilov and Hutter, 2019) has emerged as a promising alternative due to its improved regularization properties.

In the phase of coarse filtering, if a tentative model fails to meet handcrafted requirements with these default settings, further stages of the search are pruned to accelerate the searching process. Handcrafted requirements are defined based on specific use cases. For instance, in the context of stopped-on-tracks detection, the model must not be overly sensitive to data imbalance. In transportation surveillance, most objects of interest are typically pedestrians and family vehicles, while stopped-on-tracks incidents are often caused by less common objects such as heavy trucks and trailers. Therefore, a pruning line should be established based on class-ratio-irrelevant metrics such as mean average precision *mAP* equation (15):

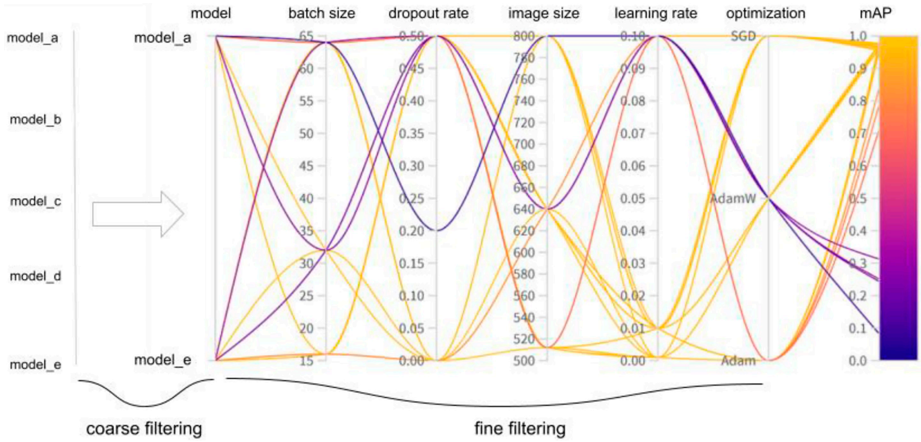


Figure 3. Illustration of the NAS
Source(s): Created by authors

$$mAP = \frac{1}{C}\sum_{k=1}^{C} AP_k \tag{15}$$

where C is the number of classes of objects of interest. Average precision (AP) is the area under the precision-recall curve, calculated by equation (16). It is often computed by using discrete recall levels and averaging the maximum precision value at each recall level:

$$AP = \frac{1}{2s}\sum\nolimits_{i\in\{0.5,\ \ 0.5+s, 0.5+2s,\ \ \ldots, 1-s\}} (Recall_{i+1} - Recall_i)Precision_i \tag{16}$$

where s is the stride and $Recall_i$ and $Precision_i$ are recall and precision at the ith intersection over union threshold.

Second, the surveillance system needs to be run in real-time with a standard server during the inference phase. Therefore, models with performance below a certain frame rate and GPU memory threshold will be excluded. Third, the ability to address occlusion is required, as congested vehicles often stop on track areas while waiting at red lights. This ability can be evaluated using identification metrics such as ID switch ratio $IDSwR$ equation (3):

$$IDSwR = \frac{IDSw}{M} \tag{17}$$

where $IDSw$ is the number of times a track's identity is incorrectly assigned to a different object and M is the total number of correct associations between predicted and ground truth objects across all frames.

After filtering model candidates with handcrafted requirements, a bandit-based heuristic tunning algorithm, hyperband (Li *et al.*, 2018), is applied to optimize hyperparameters and training configurations in parallel to achieve the best performance for each model. Hyperband begins by sampling a wide range of hyperparameter configurations and allocates limited resources (e.g. training epochs or iterations) to each. It then iteratively eliminates underperforming configurations based on intermediate results, reallocating resources to the most promising candidates. This adaptive strategy enables efficient exploration of the hyperparameter space while reducing the computational cost. The best-performing model is selected upon completion of the fine filtering phase.

3.4 Data-driven thresholding for stopped-on-tracks detection

Determining whether a vehicle stops in the track area based on its trajectory is straightforward if the detection and tracking are accurate. It involves calculating the distance of a sequence of trajectory points and checking whether the distance reduces to zero within the defined track area. However, inaccuracies exist even when the best open-source models are used.

False positives, where nonexistent objects are detected, can lead to erroneous alerts. Conversely, false negatives, where actual objects on the track are missed, pose serious safety risks and can potentially lead to accidents. False localization of detected objects may result in incorrect judgments about whether an object is within the track area. In addition, issues like ID switching and drifting in tracking can disrupt the analysis of an object's trajectory, causing incorrect calculation of the distance. Furthermore, failing to account for occlusions can cause the system to lose track of objects, missing critical stop events. These inaccuracies significantly undermine the effectiveness of a straightforward stopped-on-tracks detection approach.

Aiming at mitigating the inaccuracies of detection and tracking, a data-driven thresholding algorithm is proposed. It assumes that a fraction of trajectories includes the pattern of vehicle stopping, such as the trajectories of vehicles stopped while they are waiting at a red light. As outlined in the pseudo-code (Figure 4), it first calculates L2 distances *dist* between consecutive points in each trajectory and filters out the trajectory points that lie out of three standard deviations. Theoretically, this step can remove the false trajectory point caused by ID switching or drifting if the sample size is large enough. Second, it uses a percentile threshold M to determine if the trajectory point matches the pattern of vehicle stopping, which is usually set to $\frac{red\_time}{day\_time}$. Because stopped-on-tracks incidents are defined as vehicles stopped on the track area for more than 5 s, another threshold C for checking stop duration is usually set to $5 \times$ *frame rate*. Once enough stopping signals are accumulated, this trajectory is defined as a stopped-on-tracks incident.

After obtaining the stopped-on-tracks results, a comprehensive data analysis will be conducted to extract valuable insights. Temporal distribution analysis will reveal correlations between time of day and the likelihood of stopped-on-tracks incidents, providing insight into patterns and trends. Spatial distribution analysis will identify areas where vehicles are more prone to stopping on the track, potentially highlighting design flaws in those locations. In addition, comparing data between two nearby intersections will offer evidence of how traffic congestion impacts stopped-on-tracks incidents, thereby substantiating common-sense factors that contribute to these occurrences.

In conclusion, object detection and object tracking are fundamental techniques for assessing the states of objects of interest in grade-crossing surveillance video footage. A systematic and feasible approach to identifying the most appropriate solution from various publicly available models and numerous settings is through a domain-specific NAS. Although determining stopped-on-tracks incidents from a batch of trajectories is affected by the inaccuracies of detection and tracking, a data-driven thresholding method is designed to mitigate false alarms and missed detections. The insights gleaned from analyzing stopped-on-tracks incidents are invaluable for improving safety and informing infrastructure design.

Algorithm 1 Stop-on-tracks detection

```
Input:
    trajs ← A list of bounding-box trajectories
    ROI ← A list of coordinates, representing the polygon track area
    M ← Minimum distance required to classify as stopped
    C ← Minimum frame required to classify as stopped
Output:
    Boolean values indicating each trajectory whether stopped on tracks
dists ← [ ], stops ← [ ]
for traj in trajs do
    dist ← [ ]
    for i=1,2,...|traj| − 1 do
        (x_{i+1}, y_{i+1}, w_{i+1}, h_{i+1}) ← traj[i + 1]
        (x_i, y_i, w_i, h_i) ← traj[i]
        d = sqrt((x_{i+1} − x_i)^2 − (y_{i+1} + h_{i+1}/2 − y_i − h_i/2)^2)
        dist.push(d)
    end for
    dists.push(dist)
end for
bound ← (mean(dist)-3std(dist), mean(dist)+3std(dist))
dists′ ← dists where dists is within bound
D ← percentile(dists′, M)
for dist in dists′ do
    c ← 0, flag ← false
    for i=1,2,...|traj| − 1 do
        d ← dist[i]
        (x_i, y_i, w_i, h_i) ← traj[i]
        if d < D && ((x_i − w_i/2, y_i + h_i/2) or (x_i + w_i/2, y_i + h_i/2) is within
ROI) then
            c++
            if c > C then
                flag ← true
                break
            end if
        else
            c ← 0
        end if
    end for
    stops.push(flag)
end for
return stops
```

Figure 4. Pseudocode of stopped-on-tracks detection via data-driven thresholding
Source(s): Created by authors

4. Experiment

Several experiments were conducted to evaluate the effectiveness of the coarse-to-fine NAS and the data-driven thresholding. This section includes data set construction, technical implementation details of the NAS and the performance of the selected model in detecting stopped-on-tracks incidents.

4.1 Data set

A total of 10,346 images are extracted from grade-crossing footage captured at 19 different locations over the course of a year. Of these, 9,692 images are annotated for object detection and 654 images are annotated for object tracking. The data set encompasses nine classes of objects of interest, which are exemplified and detailed in the Appendix. Note that the largest class accounts for 46.4% of the detection annotations, while the smallest class comprises only 0.2%. To address this notable imbalance among classes, the designated *mAP* metric is used during the NAS phase to ensure that the model is capable of handling disproportion in class distribution.

A key difference between our data set and typical open-source object detection data sets is the presence of repeated backgrounds, as all images are extracted from fixed-camera footage. In such settings, static background elements (or say no object of interest) can be mistakenly detected as objects of interest, potentially leading to high false positive rates during live monitoring. To mitigate this, an annotation guideline is proposed to minimize the inclusion of occlusions caused by static background elements within bounding boxes. Specifically, if an occlusion covers more than half of an object, the object is excluded from annotation. If the occlusion divides the object centrally, the team evaluates whether the remaining visible parts are sufficiently representative: as illustrated in Figure 5(a) and (c), , if the occluded region is smaller than both the left and right segments of the object, the entire object is annotated. Conversely, as shown in Figure 5(b) and (d), , if the occlusion is larger than the smaller segment, only the larger visible segment is annotated.

In addition, the team introduced several customized annotation guidelines to further enhance object detection and tracking performance, such as minimizing overlap between objects of interest and annotating individual train cars separately. Collectively, these efforts contribute to the development of a diverse and comprehensive data set, serving as a foundation for the subsequent NAS.

4.2 Neural architecture search results

According to criteria such as accessibility, replicability and performance public benchmarks (Sanchez-Matilla *et al.*, 2025; Papers with Code, 2024), 12 models for object detection and four models for object tracking were considered during the model selection stage. Among them, YOLOv5 (Jocher, 2024) and DeepSORT (Wojke *et al.*, 2017) serve as baseline models for object detection and object tracking respectively due to their widespread adoption in prior studies (Amin *et al.*, 2024; Klammsteiner *et al.*, 2023; Rahman *et al.*, 2022). The experiments were conducted on a host equipped with a 3.40 GHz Intel Xeon E-2236 CPU, a Nvidia RTX 3080 GPU with 10.24 GB memory and running Ubuntu 20.04 as the operating system.

For object detection, given that the live streams for case studies are 15 FPS in a resolution of 1080 × 1920 pixels, the minimal inference speed is 20 FPS with 5 FPS as redundancy. The maximum GPU memory a model can use is 9.24 GB, with 1 GB as redundancy. The minimal requirement for accuracy is set at 0.35 mAP@0.5:0.95 by referring to the COCO benchmark (Papers with Code, 2024). In the context of object tracking, the pruning line was set at an IDSwR of 0.15 by referring to the MOT17 benchmark (Sanchez-Matilla *et al.*, 2025).

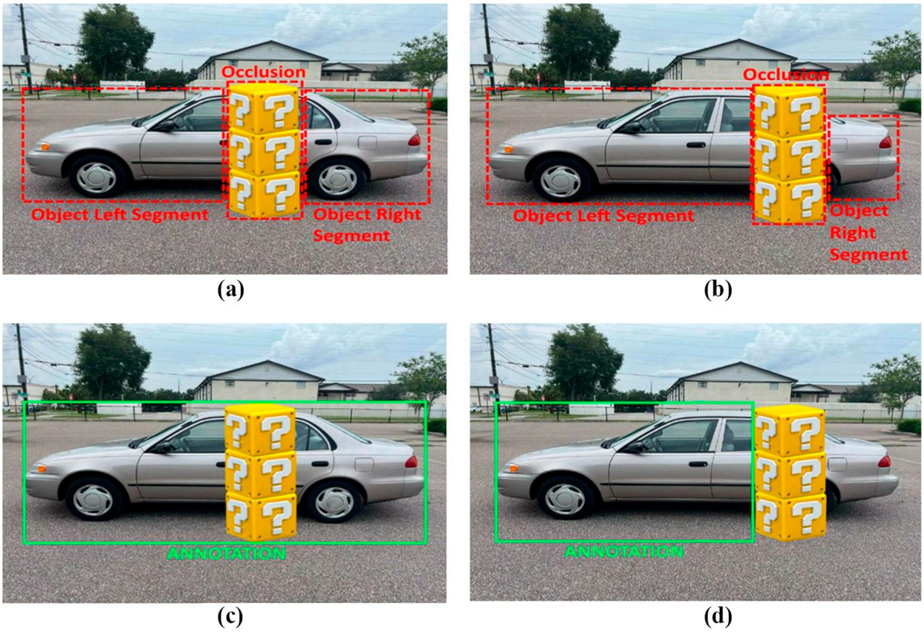


Figure 5. Illustration of annotating occluded objects
Source(s): Created by authors

The data set is divided into three subsets: 70% for training, 15% for validation and 15% for testing. For each training scenario, five independent trials are conducted, each using a randomly shuffled split of the data set. In each trial, the model is trained on the training set, hyperparameters are tuned using the validation set and performance is evaluated on the test set. The final accuracies reported in Table 2 represent the average test accuracies across five trials. As a result, five detection models meet the handcrafted requirements and three tracking models proceed to the fine-filtering stage. The performances of the best trials for each model are presented in Table 2, DAB-DETR and ByteTrack achieve the best performance for object detection and for object tracking. This choice is well-justified given the frequent localization errors observed in the baseline detection model and the prevalence of short-term occlusions. DAB-DETR improves localization accuracy and accelerates convergence by representing queries as dynamically updated anchor boxes. Meanwhile, ByteTrack uses a two-step association strategy that effectively mitigates issues such as short-term occlusions, motion blur and temporarily missed detections.

4.3 Real-world evaluation

To further validate the proposed system, it is deployed in eight grade-crossing locations for counting traffic and detecting stopped-on-tracks incidents for three months. As the ground-truth, the team manually reviews 3 h of video in each location during the peak hour and reviews detected all stopped-on-track incidents during the three-month period.

As shown in Table 3, 907 stopped-on-tracks incidents were detected, with 852 of those being actual stopped-on-tracks incidents. This demonstrates that the system achieves a precision rate of up to 94%. Six percent of falsely detected incidents consisted of objects

Table 2. NAS results: the italiced row indicates the best-performing model, while the italicized row represents the baseline model

	Coarse filtering				Fine filtering		
Detection model	Memory (GiB)	Frame rate (FPS)	Accuracy (mAP@0.5:0.95)	Evaluation	Best hyperparameters	Best training configuration	Best accuracy (mAP@0.5:0.95)
CRPN	11.2	23	0.399	Pruned because of high memory usage			
Conditional DETR	9.1	30	0.411	Passed	Backbone: Resnet50	Epoch: 150 Batch size: 16	0.445
DAB-DETR	*9.1*	*26*	*0.423*	*Passed*	*Backbone: Resnet50*	*Epoch: 50* *Batch size: 16*	*0.456*
DETR	7.2	34	0.399	Passed	Backbone: Resnet50	Epoch: 150 Batch size: 16	0.433
DINO	11.1	9	0.490	Pruned because of high memory usage and low inference speed			
EfficientNet	11.1	5	0.404	Pruned because of high memory usage and low inference speed			
FreeAnchor	4.9	30	0.387	Passed	Backbone: ResNeXt 101	Epoch: 300 Batch size: 128	0.419
RetinaNet	5.0	19	0.365	Pruned because of low inference speed			
RTMDet	8.9	16	0.446	Pruned because of low inference speed			
SSD	9.9	30	0.255	Pruned because of low accuracy			
YOLOv5	*4.7*	*29*	*0.338*	*Pruned because of low accuracy*			
YOLOX	7.6	24	0.405	Passed	Backbone: CSPDarknet	Epoch: 150 Batch size: 64	0.440

	Coarse filtering			Fine filtering	
Tracking model	Accuracy (MOTA)	Occlusion handling (IDSwR)	Evaluation	Best neural architecture	MOTA
ByteTrack	*0.781*	*0.102*	*Passed*	*Detector: DAB-DETR*	*0.786*
DeepSORT	*0.620*	*0.159*	*Pruned because of low occlusion handling*		
OC-SORT	0.778	0.041	Passed	Detector: DAB-DETR	0.781
Tracktor	0.641	0.061	Passed	ReID: Resnet50 Detector: Faster-RCNN	0.693

Source(s): Created by authors

Table 3. Real-world evaluation results

Task	Duration	Performance
Stopped-on-tracks incidents detection	3 months	The system detected 907 incidents, of which 852 are true based on manual review, resulting in *94% precision* rate
Traffic counting	24 h	The manual review collected 15,657 traffic instances, of which 14,884 were successfully detected, resulting in a *95% recall* rate

Source(s): Created by authors

slowly crossing the track area without coming to a complete stop (4%) and people stood at the edge of the ROI while waiting for the red light (2%). Relocating the camera to record from a bird’s-eye view might make it easier to capture slow movements. To verify the system’s ability in real-world object tracking, we manually count traffic volume for 24 h. The system counts 15,657 traffic instances and 14,884 of them are correct. As such, the recall of the traffic counting is more than 95%. The remaining 5% of traffic identifications are attributed to video glitches, which resulted from intermittent packet loss. Transitioning from cellular data to satellite data might provide a more reliable network. In summary, the proposed solutions demonstrate promising performance across two tasks over an extended testing period, highlighting their practical applicability and effectiveness as an evaluation tool. However, they do not directly contribute to improving safety.

5. Stopped-on-tracks case studies

Using the best-performing models, two case studies were conducted at different grade crossings. The first study evaluated the effectiveness of methyl methacrylate (MMA) road marking in reducing stopped-on-tracks incidents. MMA road marking is known for its durability and high visibility and may help reduce stopped-on-tracks incidents by providing clearer lane demarcation, especially under adverse weather conditions or low-light environments. The second study explored the correlation between stopped-on-tracks incidents and nearby traffic congestion. Video footage was analyzed to extract vehicle trajectories, which were then assessed using the proposed algorithm to determine whether any vehicles stopped within the track area.

5.1 Case study I

The first crossing is on an airline highway and was chosen due to local officials reporting frequent stopped-on-tracks incidents. In October 2023, MMA road markings were installed to prevent stops on tracks at this grade crossing. Two rounds of data collection and video analytics were conducted, one before and one after the MMA road markings were painted. Only daytime results were processed due to inadequate illumination at night. All video data were captured with a resolution of 720×1080.

The traffic volume in the videos before the MMA road marking was applied was 230,244 vehicles, with 1,489 stopped-on-tracks incidents recorded. In contrast, the traffic volume in the videos after the MMA road marking was applied was 132,315 vehicles, with 808 stopped-on-tracks incidents. The videos collected before the MMA road marking application were taken during the summer of 2021, a period of higher daily traffic. The before-and-after comparison is presented in Table 4. Note that the AADT derived from the detected traffic is lower than the one published by the department of transportation (DOT). This discrepancy arises because the installed camera does not support night vision, making it hard to capture any traffic data from sunset to sunrise. Besides, the official AADT accounts for traffic in both

Table 4. Results of stopped-on-tracks analysis before and after the MMA road marking

Metric	Before	After
Hours of video	301	329
Analyzed periods	2021-06-09 to 2021-06-23	2023-10-12 to 2023-10-20, 2024-1-29 to 2024-1-31, 2024-2-1 to 2024-2-4
Vehicles stopped on tracks	1,489	808 (−45.7%)
Stopped-on-tracks rate/hour	4.94	2.45 (−50.4%)
Vehicle traffic	230,244	132,315
Stopped-on-tracks rate/1,000 vehicles	6.4	6.1 (−4%)

Source(s): Created by authors

the north-south and east-west directions, while our grade crossing system only monitored traffic in the north-south direction.

The findings indicates a 50.4% reduction in the stopped-on-tracks rate per hour and a 45.7% decrease in the overall number of stopped-on-tracks incidents. This observation suggests that MMA road markings show promise in preventing such incidents; however, long-term results are needed to draw a definitive conclusion. The findings align with the intuition that drivers are more likely to avoid stopping on the tracks when the area is clearly marked and highlighted by the MMA road markings. However, there was no significant improvement in the stopped-on-tracks rate per 1,000 traffic instances. This discrepancy suggests that factors other than the introduction of MMA road markings might have contributed to the overall reduction, such as seasonal or yearly variations between the two periods.

Figure 6 shows the distributions of traffic volume and stopped-on-tracks ratio vs the hour of day. It is observed that vehicles tend to stop on tracks during off-peak hours. There are some possible reasons for this phenomenon: first, during off-peak hours, drivers may be less vigilant or feel less pressure due to lower traffic volumes, leading to a diminished awareness of the potential dangers of stopping on tracks. Second, there might be less enforcement of traffic regulations during off-peak hours; drivers may feel more comfortable breaking the

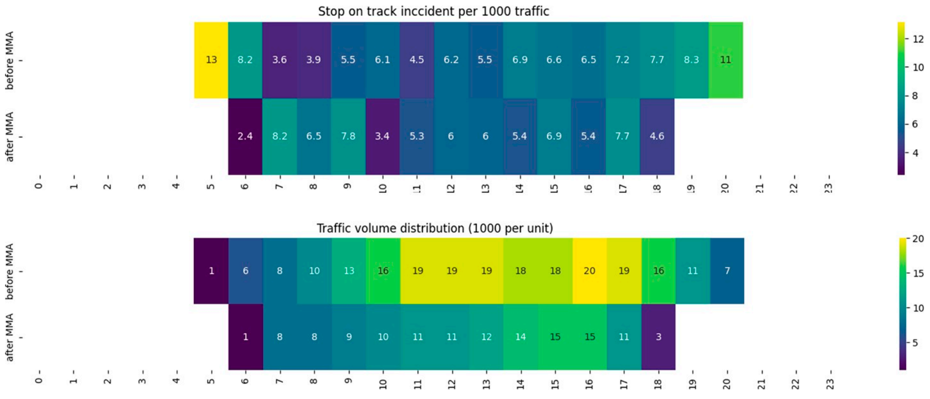


Figure 6. Time-of-day temporal heatmap, before and after the MMA road marking
Source(s): Created by authors

rules and stopping on tracks. Third, drivers might believe that train traffic is less frequent during off-peak hours, leading them to underestimate the risk of stopping on tracks. Therefore, further safety measures can target stopped-on-tracks incidents during off-peak hours. For instance, the DOT can collaborate with local media and community organizations to spread awareness about the risks of stopping on tracks, particularly during off-peak hours.

Given the safety improvement, it is recommended to deploy the MMA road markings at multiple locations for a minimum of one year. This approach will not only help reduce incidents of vehicles stopping on tracks but also facilitate a more comprehensive before-and-after evaluation. Currently, the study only analyzes daytime footage from a single location.

5.2 Case study II

Located between two state highways, Midland Ave hosts one of the busiest grade crossings, with an annual average daily traffic exceeding 300,000 vehicles. The nearest southbound and northbound intersections are just 200 and 60 feet away from the grade crossing, respectively. According to complaints from local commuters, traffic congestion often forces them to stop on the tracks. These passive stops, where vehicles are blocked and unable to move, pose significant safety risks and motivated the case study presented below.

A long-term online camera system was used to provide live stream feed at a resolution of 1080 × 1920 for the grade crossing. During the stopped-on-tracks analyzed period, a short-term offline camera system was set up in the nearest intersection along the southbound direction. It collected videos to determine whether the traffic queue was backed up to the grade crossing. Note that the nearest intersection along the northbound direction can be seen in the online camera system. Because the camera in this location supported night-vision mode and there was sufficient roadside illumination, footage from both daytime and nighttime were processed.

Based on the results shown in Table 5, the algorithm detected 53 vehicles stopping on the tracks, resulting in an average stopped-on-tracks rate of 0.31 vehicles/h. With a total of 78,289 vehicles passing through the area, the stopped-on-tracks rate was approximately 0.67 per 1,000 vehicles.

To better understand the reasons for these stops, the team manually reviewed and categorized the incidents. According to the results summarized in Figure 7, the highest proportion of stops were caused by southbound left-turn traffic (41%), closely followed by southbound through traffic (39%). Northbound left-turn traffic and northbound through traffic were responsible for 5% and 3% of stops, respectively. Active stops, such as U-turns, account for 11% of the incidents. While these incidents are infrequent compared to overall traffic volume, 89% of passive stops still require immediate mitigation.

Table 5. Results of stopped-on-tracks analysis

Metric	Value
Hours of video	168
Analyzed periods	2023-09-01 to 2023-09-06
Vehicles stopped on tracks	53
Stopped-on-tracks rate/hour	0.31
Vehicle traffic	78,289
Stopped-on-tracks rate/1,000 vehicles	0.67

Source(s): Created by authors

Figure 7. Illustration of stopped-on-tracks attributions
Source(s): Created by authors

Although vehicles stopping on the track area should be more likely to be caused by backups from the closer northbound intersection, more stopped-on-tracks incidents occurred in the farther southbound direction. There are three case-specific factors that could have led to this phenomenon. First, the southbound direction experienced higher traffic volume, leading to more frequent traffic congestion. Second, the signal design at the intersections along the southbound direction might be inappropriate, such as shorter green times or no protected left turns, causing vehicles to back up on the track. Third, the local environment, such as the presence of commercial areas, schools or other high-traffic destinations in the southbound direction, could lead to more frequent stops on the tracks.

Based on the discussion, potential mitigations for stopped-on-tracks incidents at this location include eliminating the left turn at the first intersection and increasing the green light duration at the second intersection for southbound traffic. These measures could lead to an 80% reduction in stopped-on-tracks incidents. However, several caveats should be noted. In addition, the limited period may not capture variations that occur over different weeks or months. Therefore, more extensive data collection over a broader period is necessary to confirm these conclusions.

6. Conclusion

In this paper, the team proposed a practical AI-assisted system designed to detect stopped-on-tracks incidents. The system achieved 94% precision across 2,727 detected stopped-on-tracks incidents and 95% recall over counting 15,657 traffic instances, using a customized NAS algorithm that identifies the most suitable models for detection and tracking. These models were optimized to handle imbalanced data sets, achieve real-time inference speeds and mitigate issues related to object occlusion. In addition, a data-driven thresholding algorithm for vehicular stoppage determination was designed to further reduce the impact of imperfect AI detections.

Two case studies were conducted to demonstrate the system's benefits. The first case study revealed that the MMA road marking improved stopped-on-tracks incident prevention, while the second case study indicated that most stopped-on-tracks incidents were primarily caused by southbound traffic, suggesting potential areas for mitigation.

Overall, AI-based stopped-on-tracks detection was proven to be a powerful tool for harnessing the potential of massive video data to gain a deeper understanding of real-world stopped-on-tracks behavior. Moreover, data-driven decision-making enhanced the safety of train crews, rail passengers and road users while also contributing to congestion relief.

Nevertheless, the proposed methodology has several limitations. It may struggle in nighttime conditions without adequate illumination and during extreme weather events, such as strong winds, heavy rain and snow. The system should be aware of these adverse situations and warn users of low-confidence results. To address environmental challenges, the system can use enhanced sensors like infrared cameras, LiDAR and radar, paired with improved lighting and weather-resistant casings, while using algorithmic approaches such as sensor fusion and confidence estimation to ensure reliable performance and warn users of low-confidence results. In addition, it will be insightful to compare the system's performance under various situations, including crossings with different layouts, lighting conditions and traffic patterns. Furthermore, the before-and-after analysis is based on the video data collected in different seasons, which could introduce seasonal biases into the insights. To address this, extended data collection is planned to study long-term trends and provide more robust insights.

Acknowledgements

This research was partially funded by the Federal Railroad Administration (FRA) of the US Department of Transportation (USDOT). The authors also thank Julius Codjoe, Raju Thapa, Milhan Moomen and Ashifur Rahman from the Louisiana Transportation Research Center and Bill Shrewsbury and Shawn Luke for their efforts in the deployment of cameras and collection

of video data for this study. However, the authors are solely responsible for all views and analyses presented in this paper.

References

Akopov, A.S. and Beklaryan, L.A. (2025), "Evolutionary synthesis of high-capacity reconfigurable multilayer road networks using a multiagent hybrid clustering-assisted genetic algorithm", *IEEE Access*, Vol. 13, pp. 53448-53474.

Amin, A., Chimba, D., Hasan, K., *et al*. (2024), "Intelligent railroad grade crossing: leveraging semantic segmentation and object detection for enhanced safety", *Epub Ahead of Print*, doi: 10.48550/arXiv.2403.11060.

Assembly (2024), "No. 1353 state of new jersey 219th legislature", available at: https://pub.njleg.gov/bills/2020/A1500/1353_I1.HTM (accessed 20 October 2024).

Bercovich, A., Levy, I., Golan, I., *et al*. (2025), "Llama-Nemotron: efficient reasoning models", *Epub Ahead of Print*, doi: 10.48550/arXiv.2505.00949.

Carion, N., Massa, F., Synnaeve, G., *et al.* (2020),. "End-to-end object detection with transformers", available at: http://arxiv.org/abs/2005.12872 (accessed 23 August 2024).

Department of Transportation – Data Portal (2024), "Highway-rail grade crossing accident data (form 57)", available at: https://data.transportation.gov/Railroads/Highway-Rail-Grade-Crossing-Accident-Data-Form-57-7wn6-i5b9/about_data (accessed 26 September 2024).

Gong, X., Chen, X., Zhong, Z., *et al.* (2022), "Enhanced few-shot learning for intrusion detection in railway video surveillance", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23 No. 8, pp. 11301-11313.

Guo, F., Jiang, Z., Wang, Y., *et al*. (2022), "Dense traffic detection at highway-railroad grade crossings", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23 No. 9, pp. 15498-15511.

Jocher G. (2024), "Ultralytics/yolov5: - bug fixes and performance improvements – NASA/ADS", available at: https://ui.adsabs.harvard.edu/abs/2020zndo…4154370J/abstract (accessed 14 April 2024).

Kingma, D.P. and Ba, J. (2017), "Adam: a method for stochastic optimization", *Epub Ahead of Print*, doi: 10.48550/arXiv.1412.6980.

Klammsteiner, M., Döller, M., Golec, P., *et al.* (2023),. "Vision based stationary railway track monitoring system", *2023 33rd Conference of Open Innovations Association (FRUCT)*, pp. 325-330.

Li, L., Jamieson, K., DeSalvo, G., *et al.* (2018), "Hyperband: a novel bandit-based approach to hyperparameter optimization", *Epub Ahead of Print*, doi: 10.48550/arXiv.1603.06560.

Loshchilov, I. and Hutter, F. (2019), "Decoupled weight decay regularization", *Epub Ahead of Print*, doi: 10.48550/arXiv.1711.05101.

NHTSA (2024), "Railroad crossing | NHTSA", available at: www.nhtsa.gov/campaign/railroad-crossing (accessed 24 August 2024).

Olson, E. (2023), "16 people are hospitalized after an Amtrak train crashes into a water truck", NPR, available at: www.npr.org/2023/06/29/1184989807/moorpark-california-train-derailment-amtrak (accessed 24 August 2024).

Papers with Code (2024), "COCO test-dev object detection benchmark", available at: https://paperswithcode.com/sota/object-detection-on-coco (accessed 30 August 2024).

Rahman, F.U., Ahmed, M., Hasan, M., *et al.* (2022), "Real-time obstacle detection over railway track using deep neural networks", *Procedia Computer Science*, Vol. 215, pp. 289-298.

Railroads (2024), "FRA to states: verify traffic lights connected to railroad crossings function correctly", available at: https://railroads.dot.gov/elibrary/fra-states-verify-traffic-lights-connected-railroad-crossings-function-correctly (accessed 24 August 2024).

Sanchez-Matilla, R., Poiesi, F. and Cavallaro, A. (2025), "Online multi-target tracking with strong and weak detections", in Hua, G. and Jégou, H. (Eds) *Computer Vision – ECCV 2016 Workshops*, Springer International Publishing, Cham, pp. 84-99.

Tang, Y., Wang, Y. and Qian, Y. (2023), "Railroad crossing surveillance and foreground extraction network: weakly supervised artificial-intelligence approach", *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2677 No. 9, pp. 525-538.

Wojke, N., Bewley, A. and Paulus, D. (2017), "Simple online and realtime tracking with a deep association metric", *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645-3649.

Zanger, J., DeAngelis, J. and Houlis, K. (2024), "1 dead after LIRR train hits taxi on Long Island - CBS New York", available at: www.cbsnews.com/newyork/news/lirr-train-hits-car-on-long-island/ (accessed 24 August 2024).

Zhang, Y., Sun, P., Jiang, Y., *et al.* (2022), "Bytetrack: multi-object tracking by associating every detection box", available at: http://arxiv.org/abs/2110.06864 (accessed 25 August 2024).

Corresponding author
Xiang Liu can be contacted at: xiang.liu@rutgers.edu

Appendix

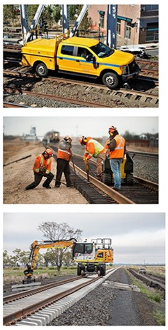
Name	Specification	Example	Accumulated Instances
Person	Pedestrian, police, military, child, etc.		4,764 (12.7%)
Bike	Bicycle, scooter, motorcycle, moped		620 (1.7%)
Family vehicle	Sedan, SUV, pickup truck, mini van		17,431 (46.4%)
Legal occupier	People who wear high-visualized vests and helmet, construction vehicles, high-rail vehicles, pickup trucks with railroad logo		8,245 (21.9%)
Bus	Commuter bus, minibus, coach, RV		190 (0.5%)

Figure A1. Objects of interest in detection
Source(s): Created by authors

			
			
Train	Locomotive, passenger train cars, freight railcars, light rail train		2,834 (7.5%)
			
Truck	Semi-truck (including semi-trailer), pickup truck, RV, van, and another commercial vehicle		3,046 (8.1%)
			
			
School bus	U.S.-style yellow school buses, short school buses		375 (1%)
Trailer	Unpowered vehicle towed by another (excluding semi-trailer)	 	84 (0.2%)

Figure A1. Continued