U.S. Department of Transportation
Federal Railroad Administration

Office of Research, Development and Technology
Washington, DC 20590

Intelligent Abnormal Situation Awareness Platform (i-ASAP)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
18-07-2023	Technical Report	08/2020-08/2023

4. TITLE AND SUBTITLE

Intelligent Abnormal Situation Awareness Platform (i-ASAP)

5a. CONTRACT NUMBER
693JJ620C000021

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Yu Qian – ORCiD: 0000-0001-8543-2774
Yi Wang – ORCiD: 0000-0002-5750-3181
Youzhi Tang – ORCiD: 0000-0002-4222-4139
Ge Song – ORCiD: 0000-0001-5679-0329

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of South Carolina
300 Main St.
Columbia, SC 29208

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Department of Transportation
Federal Railroad Administration
1200 New Jersey Ave SE
Washington, DC 20590

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
DOT/FRA/ORD-26/24

12. DISTRIBUTION/AVAILABILITY STATEMENT

This document is available to the public through the FRA website.

13. SUPPLEMENTARY NOTES

COR: Shala Blue, Ph.D.

14. ABSTRACT

This report introduces the Intelligent Abnormal Situation Awareness Platform (i-ASAP), a cutting-edge system developed to mitigate trespassing incidents at railroad crossings. The platform employs advanced computer vision and artificial intelligence technologies to monitor and identify trespassers and obstructions in real-time. Using AlphaPose and OpenPose models within a deep learning anomaly detection system, i-ASAP tracks pedestrian behavior to pinpoint anomalies. A newly developed weakly supervised foreground segmentation network extends the platform's detection capabilities to static and moving objects. Validation through custom dataset testing and real-time field trials demonstrate i-ASAP's effectiveness, setting the stage for potential deployment on edge computing devices.

15. SUBJECT TERMS

Crossing safety, crossing monitoring, pedestrian behavior, computer vision, artificial intelligence

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		62	19b. TELEPHONE NUMBER *(Include area code)*

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

Acknowledgments

The authors thank Mr. Francesco Bedini, Dr. Shala Blue, Mr. Michael Jones, and Dr. Starr Kidda from the Federal Railroad Administration for their invaluable insights and guidance. Sincere gratitude also extends to the City of Columbia, particularly the dedicated teams at the Columbia Fire Department, Department of Transportation, and the 911 Dispatching Center. Furthermore, appreciation extends to CSX, Brightline, Florida East Coast Railway, and the South Carolina Railway Museum for their tremendous support and assistance.

Contents

Executive Summary 1

1. Introduction 2
   1.1 Background 2
   1.2 Objectives 2
   1.3 Overall Approach 3
   1.4 Scope 3
   1.5 Organization of the Report 3

2. Literature Review 4
   2.1 Video Anomaly Detection Using Deep Learning 4
   2.2 Human Pose Estimation and Tracking 6
   2.3 CNN-based Foreground Segmentation Algorithm 7

3. Pedestrian Abnormal Behavior Detection 10
   3.1 Data Collection 10
   3.2 Gated Recurrent Unit (GRU)-Based Anomaly Detection System 12
   3.3 Analysis of Abnormal Pedestrian Behaviors at Grade Crossings Based on Semi-Supervised Generative Adversarial Networks (GAN) 23
   3.4 Implementation in Real-time Testing 35

4. Outliers Monitoring based on Foreground Segmentation 39
   4.1 Advantage of Foreground Detection 39
   4.2 Methodology 40
   4.3 Results and Discussion 45

5. Conclusion 51

6. References 52

Tables

Table 1. Comparison of the present method with other video-based models 23
Table 2. Comparison of methods on custom dataset 35
Table 3. The comparison of supervised learning and weakly supervised learning among CDnet2014 dataset 46
Table 4. F-measure comparison of different foreground algorithms among CDnet2014 dataset 48
Table 5 The comparison of RC-SAFE and Mask-RCNN railroad crossing dataset 49

Illustrations

Figure 1. Overview of the AE structure in Nguyen and Meunier (2019) 5
Figure 2. The pipeline of video frame prediction network in Liu et al. (2018) 6
Figure 3. Framework of pose estimation and tracking in AlphaPose (Fang et al., 2022) 7
Figure 4. Pipeline of OpenPose (Cao et al., 2017) 7
Figure 5. Three types of CNN-based foreground segmentation algorithms 8
Figure 6. Examples of the railroad grade crossing scenes 10
Figure 7. Examples of collected abnormal and normal behaviors 11
Figure 8. The proposed method for detection and localization of a pedestrian with abnormal behavior 12
Figure 9. A message-passing encoder-decoder recurrent neural network 14
Figure 10. Two example frames in Case Study 1 17
Figure 11. Performance of anomaly detection method in Case Study 1 18
Figure 12. Reconstruction errors of the pedestrian's behaviors in the test video 19
Figure 13. Anomaly localization results in Case Study 1 20
Figure 14. An example of a frame with anomaly in Case Study 2 21
Figure 15. Results of anomaly detection and localization in Case Study 2 22
Figure 16. The ROC curve of the proposed method 22
Figure 17. Overview of the anomaly detection pipeline 24
Figure 18. Model architecture of the generator 26
Figure 19. Model architecture of the discriminator 27
Figure 20. Anomaly detection results for the abnormal behavior in Case Study 1 29
Figure 21. Anomaly detection results for the normal behavior in Case Study 1 30
Figure 22. Video sequences of two examples of abnormal behaviors in Case Study 2 31
Figure 23. Anomaly detection results for the abnormal behavior in Case Study 2 31
Figure 24. Anomaly detection results for the normal behavior in Case Study 2 32
Figure 25. Video sequences of two examples of abnormal behaviors in Case Study 3 32
Figure 26. Anomaly detection results for the abnormal behavior in Case Study 3 33
Figure 27. Anomaly detection results for the normal behavior in Case Study 3 33
Figure 28. Real-time monitoring video capture module hardware implementation 35
Figure 29. The grade crossing at 305 Country Club Dr, Titusville, FL 36
Figure 30. The visual detection result in the case of single pedestrian 37

Figure 31. The visual detection result in the case of two pedestrians 38
Figure 32. Differences between foreground segmentation and object detection (from top to bottom: foreground detection, object detection) 39
Figure 33. Data pipeline of the proposed RC-SAFE network 40
Figure 34. Example of the first and the last background images generated with SuBSENSE 41
Figure 35. The U-Net-based network architecture of RC-SAFE 42
Figure 36. The detailed network architecture of RC-SAFE 42
Figure 37. Details of the residual blocks at each scale 43
Figure 38. Foreground and background masks of the weakly supervised learning 44
Figure 39. Example detection results on CDnet 2014 dataset (from left to right: input frame, ground-truth image, weakly supervised network (RC-SAFE), supervised network, SuBSENSE) 47
Figure 40. Example detection results on the railroad crossing dataset (from top to bottom: RC-SAFE, Mask-RCNN) 49
Figure 41. The grade crossing at 300 Main Street, Columbia, SC 50
Figure 42. Field testing illustration of RC-SAFE 50

Executive Summary

Railroad crossings are often hotspots for trespassing incidents, with nearly three quarters of all incidents occurring within 1,000 ft of a crossing. Grade crossings serve primarily as a nexus where both pedestrians and vehicles traverse the railway tracks, so prioritizing the mitigation of trespassing within grade crossing is important. Recognizing this challenge, the Federal Railroad Administration (FRA) sponsored a research team from the University of South Carolina to develop an Intelligent Abnormal Situation Awareness Platform (i-ASAP). This innovative solution uses cutting-edge computer vision and artificial intelligence (AI) technologies to monitor and identify trespassing pedestrians, vehicles, and unforeseen obstructions at rail crossings in real-time. i-ASAP is a substantial advancement in railroad crossing safety protocols, with the potential to curtail railroad-related mishaps, injuries, and fatalities. This research took place between 2021 and 2023 at multiple railroad crossings in South Carolina.

The i-ASAP contains a deep learning-based anomaly detection system that identifies and tracks pedestrians exhibiting unusual behaviors at grade crossings. This system is built on pose estimation models, such as AlphaPose and OpenPose, to detect and monitor key points of each pedestrian appearing in the crossing area, and subsequently generating corresponding trajectories. These trajectories input to either a message-passing recurrent neural network or a generative adversarial network to understand normal behavioral patterns. As a result, anomalous movements are pinpointed as outliers by the model. Broadening the scope of detection, the research team also developed a weakly supervised foreground segmentation network. This novel network accurately identifies and tracks both static and moving objects within the railroad crossing area. Trial runs on a custom dataset have shown the feasibility and effectiveness of the proposed platform, and real-time testing has authenticated the suitability of this system for deployment.

1. Introduction

The Federal Railroad Administration (FRA) sponsored a research team from the University of South Carolina to develop an Intelligent Abnormal Situation Awareness Platform (i-ASAP). This innovative solution uses cutting-edge computer vision and artificial intelligence (AI) technologies to monitor and identify trespassing pedestrians, vehicles, and unforeseen obstructions at rail crossings in real-time. i-ASAP is a substantial advancement in railroad crossing safety protocols, with the potential to curtail railroad-related mishaps, injuries, and fatalities, thereby boosting overall railroad safety. This research took place between 2021 and 2023 at multiple railroad crossings in South Carolina.

1.1 Background

Accidental intrusions at rail crossings are among the leading causes of crossing-related accidents. On June 27, 2022, a catastrophic incident occurred near Mendon, Missouri, when a train collided with a dump truck at a public crossing. This tragic event resulted in 4 fatalities and approximately 150 injured passengers. On June 13, 2023, another collision happened in Lower Richland County, South Carolina, between a train and a box truck, resulting in one death and two injuries. These unfortunate incidents underscore the urgent need for enhanced safety measures at rail crossings.

According to FRA data, approximately 210,000 public and private grade crossings span the United States, the majority of which are accessible to the public (Ogden and Cooper, 2019). Since the 1980s, consistent and significant reductions in the number of collisions at these grade crossings have been reported (Operation Lifesaver, 2023). This positive trend is attributable to safety improvements, including the installation of signaling units and gate arms at grade crossings. Grade crossing collisions, however, remain a leading cause of railroad-related fatalities. The U.S. recorded an annual average of about 2,144 collisions at these crossings between 2018 and 2022, with accidents resulting in an estimated 250 deaths and 775 injuries each year (Operation Lifesaver, 2023). These figures reveal a significant societal burden, with disruptions to highway and rail operations and substantial adverse impacts on local economies and communities.

In recent years, there has been an increase in accidents and fatalities on railroad rights-of-way (ROW) in North America, with trespassing incidents accounting for about 70 percent of these cases. Over 60 percent of collisions occur at crossings equipped with automatic warning systems, and 34.7 percent take place at crossings with flashing lights and gates (FRA, 2019). Flashing lights and gate arms in current systems only indicate an approaching train's presence. They do not detect trespassing or track fouling incidents involving pedestrians, vehicles, or unforeseen obstructions on crossings.

An advanced, comprehensive warning system capable of offering real-time monitoring information at grade crossings may alleviate a risk associated with grade crossings.

1.2 Objectives

The primary objective was to develop a cost-effective, easy-to-deploy, intelligent platform to identify pedestrians displaying abnormal behaviors (e.g., lingering or squatting) and foreign objects at grade crossings in real-time.

1.3 Overall Approach

Field experiments were conducted to gather behavioral data from pedestrians at various crossings under diverse environmental and operational conditions. Pedestrian behavior within the grade crossing area was analyzed by representing movements as trajectories of skeleton joint points detected by AI models. Advanced deep-learning algorithms processed these skeleton trajectories to identify anomalies in real time. Additionally, a foreground detection network was developed to identify foreign objects, including improperly moving or stationary vehicles and pedestrians. An alert system was then implemented to notify pedestrians exhibiting abnormal behavior while saving the corresponding video footage for further analysis.

1.4 Scope

The scope of this study consisted of three tasks.

Task 1 – The team established a comprehensive dataset that includes video clips with both normal walking and abnormal behaviors. This dataset was used for AI model development. To ensure generalization, the video clips were captured at different grade crossings and under different operational conditions (e.g., camera angle, distance to the grade crossing, and camera location).

Task 2 – The team developed self-learning AI models to process and analyze pedestrians in the video clips. The models used a popular automatic human skeleton detection technique to recognize pedestrians and quantify motion patterns. Anomaly detection networks were then used to assign anomaly scores to pedestrians to identify abnormal behaviors.

Task 3 – The team deployed the developed software on an edge computing device and combined it with a video capture system, including a camera, tripod, and speaker, to complete anomaly detection tasks in real-time.

1.5 Organization of the Report

This report is organized into five chapters. Chapter 1 provides an introduction and background to the project. Chapter 2 offers a literature review and discusses current developments. Chapter 3 presents the work of pedestrian abnormal behavior detection. Chapter 4 introduces a method for outliers monitoring based on foreground segmentation. Chapter 5 presents the project's conclusions.

2. Literature Review

AI and machine learning technologies have stimulated new grade crossing research. For instance, Zaman et al. (2019) employed Mask R-CNN for the detection of intrusion events on railroads. Sikora et al. (2020) introduced a railway crossing surveillance system, using YOLO and SSD networks to monitor vehicles and pedestrians. Building on the principle of swift object detection, Guan et al. (2022) devised a high-speed obstacle detection algorithm for railway images, based on a streamlined YOLO-Tiny network and a swift region proposal. Wang and Yu (2021) unveiled an innovative neural network, rooted in the SSD framework, for railway intrusion detection. Additionally, Zhang et al. (2022) developed a YOLO-based framework for the automatic identification of railroad trespassing incidents. These methodologies leverage object detection techniques for rail-related monitoring. While they deliver essential functions for intrusion detection, they do not provide a comprehensive solution for rail crossing monitoring.

Recognizing this limitation, the research team developed i-ASAP. This platform combines object detection techniques and analyzes pedestrian anomalies and foreground segmentation. By fusing multiple monitoring elements, i-ASAP provides a surveillance solution to detect unsafe behaviors at grade crossings.

2.1 Video Anomaly Detection Using Deep Learning

Convolutional neural network (CNN)-based anomaly detection methods developed specifically for video data have surpassed traditional anomaly detection techniques in addressing complex, real-world problems. Autoencoder (AE), an unsupervised anomaly detection technique, uses video data to identify abnormalities, deviations, or outliers markedly different from most data instances (Pang et al. 2021). AE and its variations are extensively employed to learn feature representations of normality. The conventional encoder-decoder scheme iassesses the divergence of test data from a set of normal training videos (Nguyen and Meunier, 2019). As depicted in Figure 1, the model comprises two processing streams. The first one uses a Convolutional AE to learn the common spatial structures in normal events. The second stream establishes an association between each input pattern and its corresponding motion, which is represented by a three-channel optical flow (*xy* displacements and magnitude). The skip connections in the U-Net are advantageous for image translation as they directly transform low-level features (e.g., edge, image patch) from the original domains to the decoded ones.

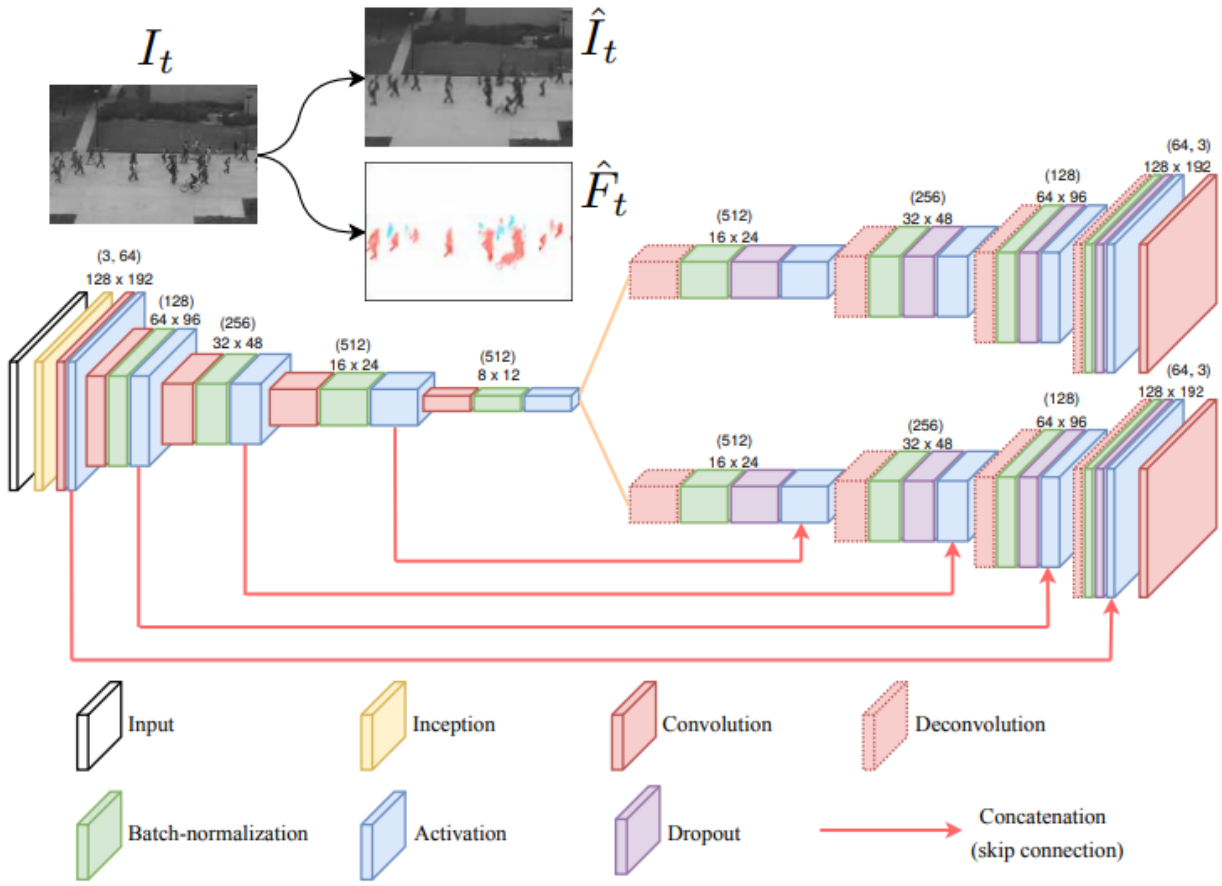


Figure 1. Overview of the AE structure in Nguyen and Meunier (2019)

Enhancements to AE, such as convolutional AE (Zhang et al., 2019), spatial-temporal AE (Chong and Tay, 2017), 3D convent AE (Zhao et al., 2017), and recurrent neural network AE (Malhotra et al., 2016), have further enhanced its performance. Specifically, Liu et al. (2018) suggested employing the prediction of optical flow for anomaly detection (see Figure 2).

In addition to spatial information, temporal information is also regarded as a vital feature for generating high-quality frames in videos. To ensure motion consistency for normal events in the training dataset, an optical flow in the temporal domain was added. It is notable that abnormal events can be detected based on either appearance or motion. This work uses both appearance and motion loss for normal events, making the identification of abnormal events straightforward by comparing the prediction with the ground truth. In a different approach, Pang et al. (2020) proposed an unsupervised algorithm that combines feature extraction and anomaly scoring, thus enabling end-to-end optimization of an anomaly detection model.

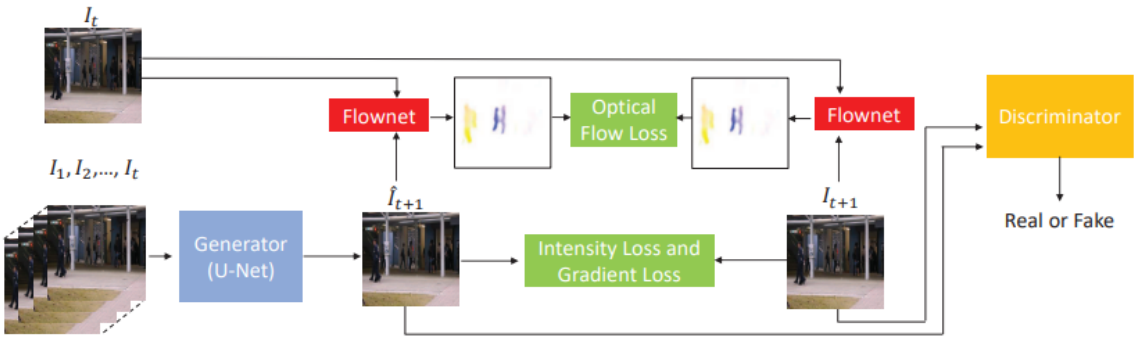


Figure 2. The pipeline of video frame prediction network in Liu et al. (2018)

2.2 Human Pose Estimation and Tracking

Human trajectories within video scenes provide critical insights for behavior pattern analysis. These trajectories have been leveraged in various computer vision applications, primarily in feature learning tasks such as understanding motion (Gupta et al., 2018), recognizing actions (Piergiovanni and Ryoo, 2019), and predicting trajectories (Bera and Manocha, 2018). Typically, these works represent pedestrian motion as a one-dimensional trajectory sequence, where input data are configured as *xy* coordinates of the whole-body center, while overlooking local body postures and relative movements. Addressing this gap, skeleton joint coordinates have been employed for dynamic human behavior modeling (Villegas, et al., 2017). Skeleton trajectories contain the positions of various body joints in the spatiotemporal domain of video sequences. As opposed to appearance-based representations, skeleton features are compact, heavily structured, semantically rich, and offer detailed descriptions of human action and movement, all of which are essential for detecting behavioral anomalies.

The creation of trajectory data representing behaviors is the initial step in anomaly detection. The quality of data representing normal behavior directly affects the training of the anomaly detection model and the overall performance of the pipeline. However, manually extracting and labeling ground truth trajectory data is impractical due to the vast volume of collected data.

Fortunately, reliable pose estimation and tracking models can aid in this process. Human pose estimation and tracking, core tasks in computer vision, involve detecting and localizing human body joints (i.e., keypoints) and tracking their positions over time. These tasks are instrumental in various applications, including action recognition, human-computer interaction, surveillance, sports analysis, and augmented reality. By using pose estimation algorithms, it is possible to generate high-quality skeleton trajectories that closely mimic the ground truth data, eliminating the need for manual labeling.

Two popular and widely adopted pose estimation algorithms, AlphaPose (Fang et al., 2022) and OpenPose (Cao et al, 2017), have been developed to accurately estimate human poses from images or videos. They employ deep learning techniques to detect and localize key body joints, resulting in a detailed depiction of human body poses.

2.2.1 AlphaPose

The AlphaPose framework consists of three components: object detector, pose estimation model, and tracking networks (see Figure 3). Given an input image, off-the-shelf object detectors such as Yolov3 or EfficientDet are designed to generate human detections. Each detected human was

cropped, resized, and forwarded through pose estimation and tracking networks to obtain full body human pose and ID features.

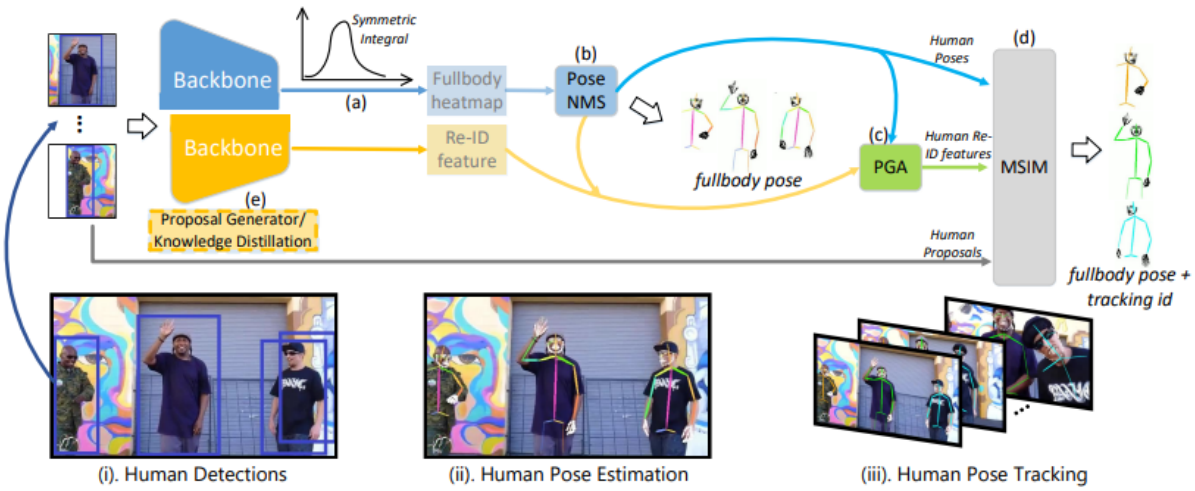


Figure 3. Framework of pose estimation and tracking in AlphaPose (Fang et al., 2022)

2.2.2 OpenPose

The overall pipeline of OpenPose is illustrated in Figure 4. This system uses a color image as input and produces 2D locations of anatomical keypoints for each person in the image. First, a feedforward network predicts a set of 2D confidence maps of body part locations and a set of 2D vector fields of part affinity fields (PAFs), which encode the degree of association between parts. Finally, the confidence maps and PAFs are parsed by greedy inference[1] to output the 2D keypoints for all people in the image.

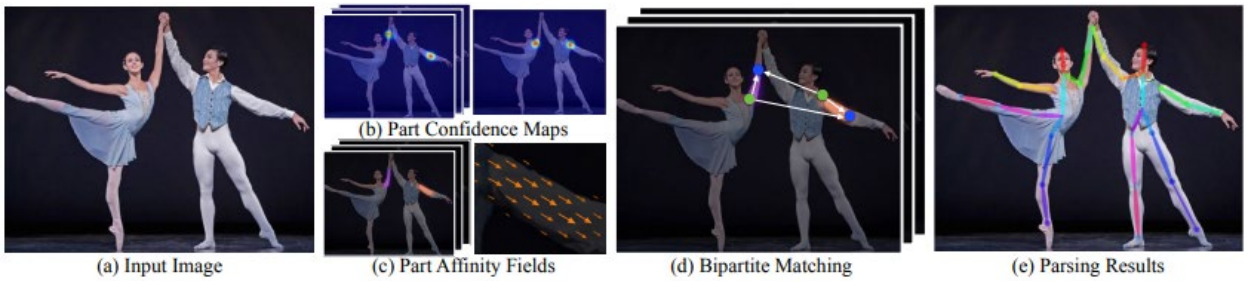


Figure 4. Pipeline of OpenPose (Cao et al., 2017)

2.3 CNN-based Foreground Segmentation Algorithm

Earlier studies have attempted to use CNNs to segment the frames into foreground and background regions. As shown in Figure 5, according to the input type of the network input, these methods can be categorized into single frame-based, multiple frames-based, and background frame(s)-based models.

[1] A greedy inference follows the problem-solving heuristic of making the locally optimal choice at each stage.

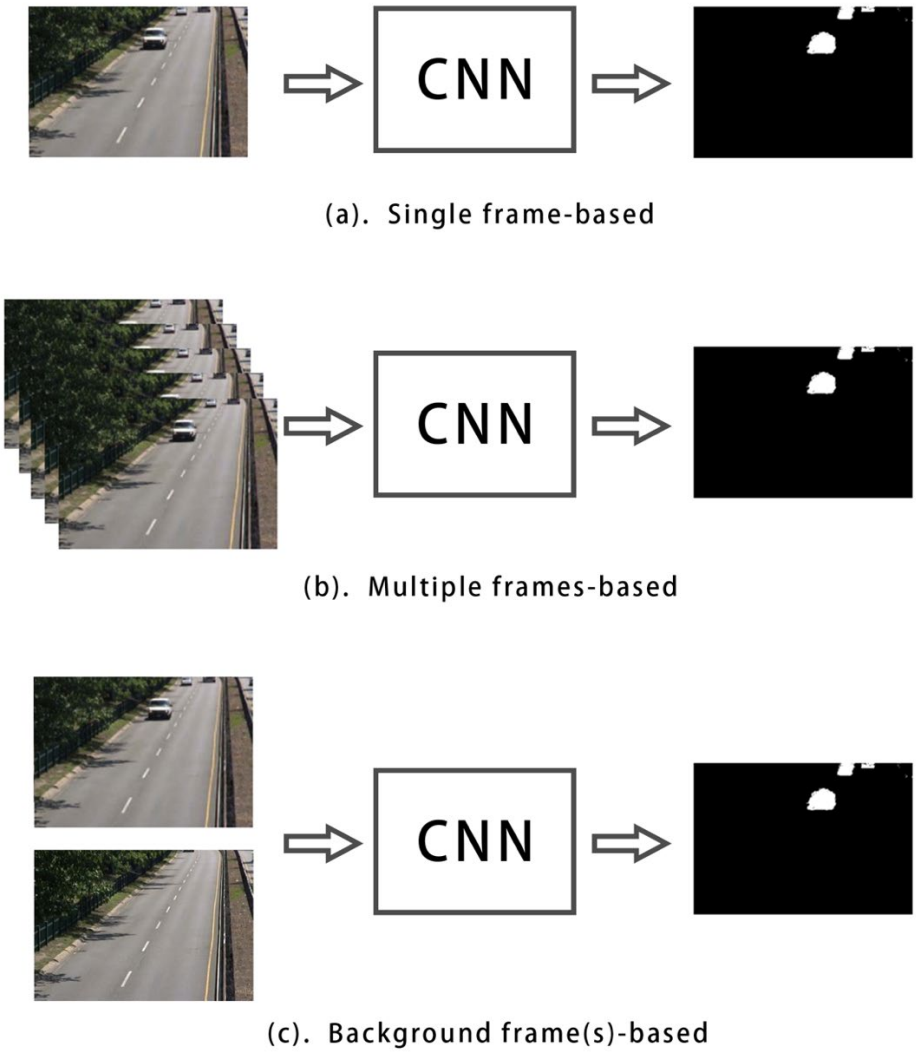


Figure 5. Three types of CNN-based foreground segmentation algorithms

Single-frame-based surveillance models output the foreground mask based solely on the current frame input. Lim and Keles (2020) proposed a multi-scale architecture using this single-frame input method. Similarly, Rahmon et al. (2021) used the current frame as input for their proposed motion U-Net, also incorporating foreground masks generated by traditional foreground segmentation algorithms. Xu et al. (2022) introduced a cascaded feature-mask fusion network with single-frame input. However, these single-frame-based models typically perform well only on scenes for which they were trained, often showing subpar performance when introduced to novel scenes.

Models that use multiple frames as input, in addition to the current frame, form another category of surveillance models. These models compare differences between the current and previous or future frames, subsequently outputting the foreground mask. Yang et al. (2017) used a Fully Convolutional Network (FCN) for foreground segmentation, incorporating multiple frames as input. Meanwhile, Hou et al. (2021) proposed a lightweight, fast 3D CNN for foreground segmentation – a model naturally suited for processing video sequences. Valipour et al. (2017) proposed a recurrent CNN for foreground segmentation that inputs one frame at a time, using the

hidden state to correlate successive frames. These models can handle new scenes but may fail to detect stationary outliers if these outliers remain static in the reference frames.

Background reference-based models use a traditional background generation model to create the background frame(s) and compare the current frame with the background frame(s) to output the foreground mask. Lin et al. (2018) proposed a foreground segmentation FCN with both background reference and current frame as inputs, employing SuBSENSE for background generation. Vijayana et al. (2021) generated the background reference based on the segmented mask of WeSamBE, also inputting an optical flow image to enhance the recognition rate of moving objects. These models can overcome the shortcomings of single and multiple frame-based models, but their performance heavily depends on the quality of the background frame(s).

Background frame(s)-based models may be an optimal choice forgrade crossing monitoring due to their robust performance with novel scenery, negating the need for continuous network retraining. Detecting static objects is as important as tracking moving ones in grade crossing monitoring. Given that the background at grade crossings is typically stable and uncomplicated, traditional background generation models can rapidly create high-quality background frames, thereby ensuring the performance of background frame(s)-based models. Thus, this study employs background frame(s)-based models for the grade crossing monitoring task.

3. Pedestrian Abnormal Behavior Detection

The detection of abnormal behavior at grade crossings is highly context-dependent based on the specific locale and surrounding environment. In this study, the team primarily focused on identifying specific anomalies commonly observed at crossings, including lingering near the rails (an often-noted occurrence) as well as individuals sitting, squatting, or lying on or in proximity to the tracks. In certain situations, these behaviors might indicate a potential for trespassing or suicide attempts, thereby posing a significant risk of accidents or fatalities. Accordingly, the goal of this study was to develop a system capable of effectively detecting and pinpointing these specific behaviors.

Note that this research focused on pedestrian-related anomalies, as most fatalities at grade crossings are linked to human activities (FRA, 2019). The i-ASAP system is specifically designed to target these pedestrian-related anomalies.

Additionally, the i-ASAP framework is not reliant on location-specific data, thereby allowing for its application across various railway grade crossing locations. This feature demonstrates the framework's versatility and its potential for broader implementation.

3.1 Data Collection

Figure 6 presents two U.S. grade crossing scenes investigated in this study. These sites are at 300 Main Street and 697 Devine Street in West Columbia, South Carolina, and are designated as Spot 1 and Spot 2, respectively. A Canon EOS 700D camera capable of continuous video recording for up to four hours was positioned adjacent to the tracks for data collection. It should be noted that different camera positions and angles were used at these two grade crossings to ensure clear visuals and foster data diversity. In total, the team collected 50 hours of video recordings from which over 2000 trajectories were extracted. The bulk of these collected videos capture normal pedestrian behaviors and movements, primarily serving for model training and validation. For model testing and performance assessment, the team used seven additional videos captured from various camera angles featuring anomalous behaviors at both locations.

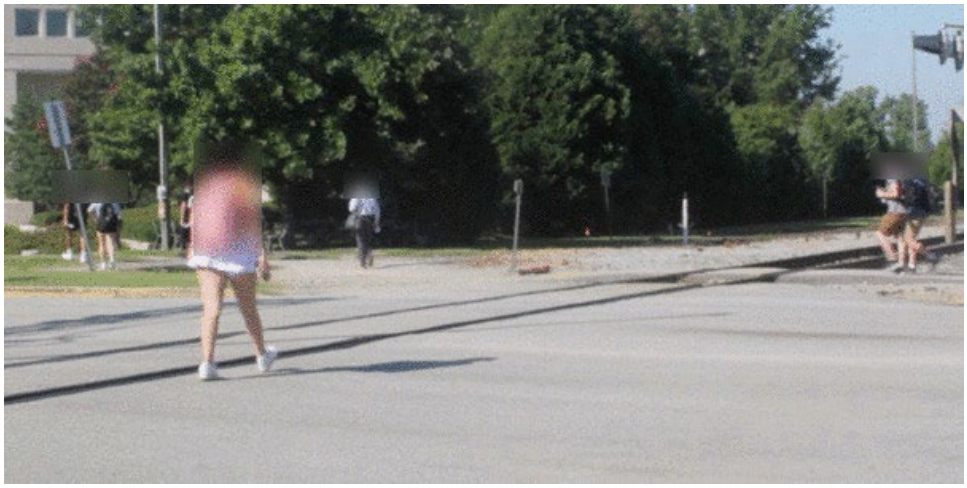
Note: To protect privacy, no Personally Identifiable Information (PII, as defined in OMB Memorandum M-07-1616) was collected for this study (e.g., facial features were not collected or used in the entire system). All the detected pedestrians were digitized as several key points only to analyze their gestures and trajectory.

(a) 697 Devine Street, Columbia, SC, USA (b) 300 Main Street, Columbia, SC, USA

Figure 6. Examples of the railroad grade crossing scenes

Figure 7 a and b illustrate examples of normal and abnormal behaviors documented at the two grade crossing locations. Figure 7a presents a typical scenario at Spot 1, where pedestrians traverse the grade crossing continuously at a relatively consistent speed and direction. In contrast, Figure 7b displays an anomalous event at Spot 2, where an individual is seated on the rail - a position that could pose a serious hazard if a train were to approach.

The classification of abnormal behaviors depends on the location and environmental context of the grade crossing scenes. The research team was primarily interested in abnormal occurrences such as lingering around the rail - one of the most frequently observed unusual behaviors at grade crossings - and actions like sitting, squatting, or lying on or near the rails. Researchers focused on creating a system capable of detecting and pinpointing these behaviors. Notably, this research focuses exclusively on pedestrian anomalies as reported by FRA (FRA, 2021) which indicate that most fatalities at grade crossings are attributable to human actions.



(a)

(b)

Figure 7. Examples of collected abnormal and normal behaviors

3.2 Gated Recurrent Unit (GRU)-Based Anomaly Detection System

The overview of the first system for anomaly detection and localization is shown in Figure 8.

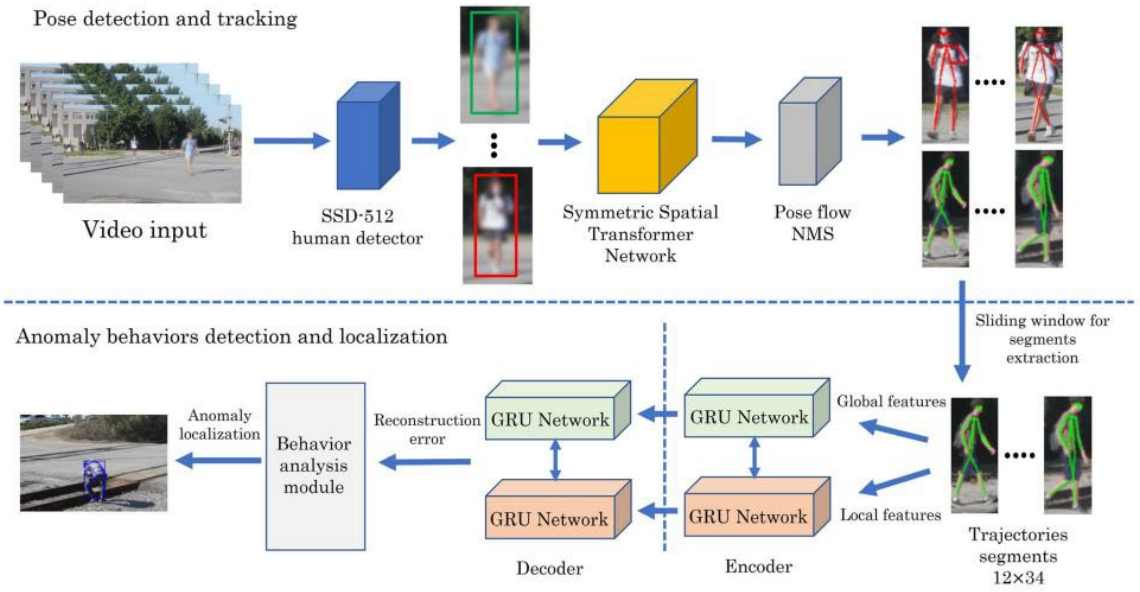


Figure 8. The proposed method for detection and localization of a pedestrian with abnormal behavior

The system includes two stages:

1. *Generation of 2D skeleton trajectories*. This process involves detecting and tracking the skeleton of each pedestrian in each video frame using AlphaPose, producing time-dependent trajectories. This tool incorporates a VGG-based Single Shot MultiBox Detector (SSD) for human detection, facilitating effective and efficient object recognition. Subsequently, the human detection proposals are sent to a Symmetric Spatial Transformer Network (SSTN) for pose generation. SSTN employs ResNet-18 (Li, et al., 2019) to localize individuals and a 4-stack hourglass network to concurrently estimate their pose proposals. These pose proposals undergo processing via pose flow Non-Maximum Suppression (NMS) to eliminate redundancies and facilitate pose tracking (Xiu, Li, Wang, Fang, & Lu, 2018). This produces a time series of 17 skeleton trajectories associated with detected key points, which are then used to characterize pedestrian movement patterns and behaviors in the video. Unlike the traditional method (Bera and Manocha, 2018) which models a pedestrian's trajectory as a 1D temporal sequence, treating them as a single point based on skeleton trajectories (the proposed method) captures detailed local body movements of a pedestrian for behavior analysis.
2. *Anomaly detection and localization*. The collected data from the estimated pose trajectories are categorized into normal and abnormal trajectories. The normal trajectories are then used to train an encoder-decoder network that can accurately reconstruct the normal trajectories but cannot do so with the abnormal ones. Specifically, a normal pedestrian's trajectory segments are further broken down into global and local coordinates. An interactive gated recurrent unit (GRU) network is then used to model the trajectory through both the global and local features. The process of anomaly detection and localization involves evaluating the anomaly score at both the frame and pedestrian

levels. This anomaly score, connected with the reconstruction error predicted by the encoder-decoder network, reflects a pedestrian's trajectory similarity to the normal trajectories used in training. In essence, behaviors deviating from the norm will yield a high anomaly score due to poor reconstruction by the encoder-decoder network. This anomaly score is then examined at both the frame and pedestrian levels. The anomaly score of a frame is assigned the maximum score of all pedestrians within that frame and is used to identify the presence of any abnormal behavior in relation to a threshold value. Any detected anomalous behavior is localized at the individual pedestrian level, with a bounding box assigned to the pedestrian whose anomaly score exceeds a certain threshold, indicating their position within the frame.

For online testing, the two components above (i.e., the skeleton trajectory generation model and then trained anomaly detection and localization model) are sequentially linked to form the entire pipeline.

3.2.1 Data Preprocessing

Pedestrian abnormal behavior within a video clip can be identified by understanding the characteristics of regular movement patterns at the grade crossing. This is achieved by reconstructing the skeleton trajectories of the subjects using the features of typical dynamic motion. By analyzing the discrepancy in this reconstruction, it becomes possible to detect and pinpoint anomalous behaviors. Assume the generated trajectories of the i^{th} detected pedestrian with the normal behavior appears in the frame t_1 and leaves in the frame t_n. They can be represented as the trajectory $\text{traj}_i = \left\{\left(x_{t_1}^j, y_{t_1}^j\right)_{j=1\cdots k}, \cdots, \left(x_{t_n}^j, y_{t_n}^j\right)_{j=1\cdots k}\right\}$, where k is the number of skeleton key points (17 in this case study). This set of skeleton trajectory sequences is the training data for the anomaly detection model.

Ideally, all pedestrians in a video would exhibit uniform skeleton scales and engage in similar types of activities. Skeleton scales depend on pedestrian location, direction of movement, and activity. However, pedestrians may appear in various locations within the scene, leading to significant variations in scale. Furthermore, the differing movement directions of pedestrians present an additional challenge to model convergence. While some efforts have been made to solve this problem, such as categorizing skeleton keypoints into multiple groups (Du, Fu, and Wang, 2016), this study employed a motion decomposition method. This method divides trajectory features into global and local features (Morais, et al., 2019), providing a more nuanced approach to account for the variability inherent in real-world scenarios.

In general, the global feature f_g contains the bounding box information of all skeleton keypoints for a pedestrian: $f_g = \left(c_x, c_y, w, h\right)$, where c_x and c_y are the center of the bounding box and w and h are the width and height of the bounding box. The local feature f_t is the skeleton trajectories normalized by the global feature, that is, $f_t = \left((x - c_x)/w\,, \left(y - c_y\right)/h\right)$, where x and y are the original coordinates of skeleton trajectories.

3.2.2 Deep Learning-Based Feature Learning Model

In the motion modeling of pedestrians displaying normal behavior, global and local features exhibit strong patterns and correlations, and deviations from these patterns can indicate anomalies. Therefore, instead of independently learning global and local features, an interactive

encoder-decoder network based on a GRU was adopted to simultaneously learn the features within a message-passing scenario (Morais et al., 2019). The GRU, as proposed by Gao and Glowacka (2016), allows each recurrent unit to adaptively capture temporal dependencies within the model at different scales. The GRU comprises two gates, an update gate and a reset gate, which control the inflow and outflow of information, respectively. Like the Long Short-Term Memory (LSTM) unit, the GRU has gating units that regulate the flow of information within the unit, but without a separate memory cell. This design aids in the efficient learning of both short- and long-term dependencies, crucial for anomaly detection. Mathematically, the relationship between the input and the output of the GRU is defined by a set of equations (shown below), where z_t and r_t are the update gate and the reset gate, $\tilde{h}_t$ and h_t are the internal state and output of the GRU, and W is the corresponding weight matrixes:

$$z_t = \sigma(W_{hz}h_{t-1} + W_{xz}x_t) \quad (1)$$

$$r_t = \sigma(W_{hr}h_{t-1} + W_{xr}x_t) \quad (2)$$

$$\tilde{h}_t = \tanh\left(W_{chx}x_t + W_{chr}(r_t \cdot h_{t-1})\right) \quad (3)$$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (4)$$

Figure 9 shows the complete structure of the Message-Passing Encoder-Decoder Recurrent Neural Network (MPED-RNN). It comprises two recurrent encoder-decoder pathways: the global component (in orange) and the local component (in blue). These two pathways are interconnected by the exchange of global and local information temporally. In essence, the cell state from one pathway at a previous time step serves as an additional input to the GRU block in the other pathway at the current time step. More specifically, at each time step, the GRU unit in one pathway receives the cell state information from the previous time step generated by the other pathway. This information is combined with the global/local features at the current time step, which are then used as the input to the GRU unit at the current time step.

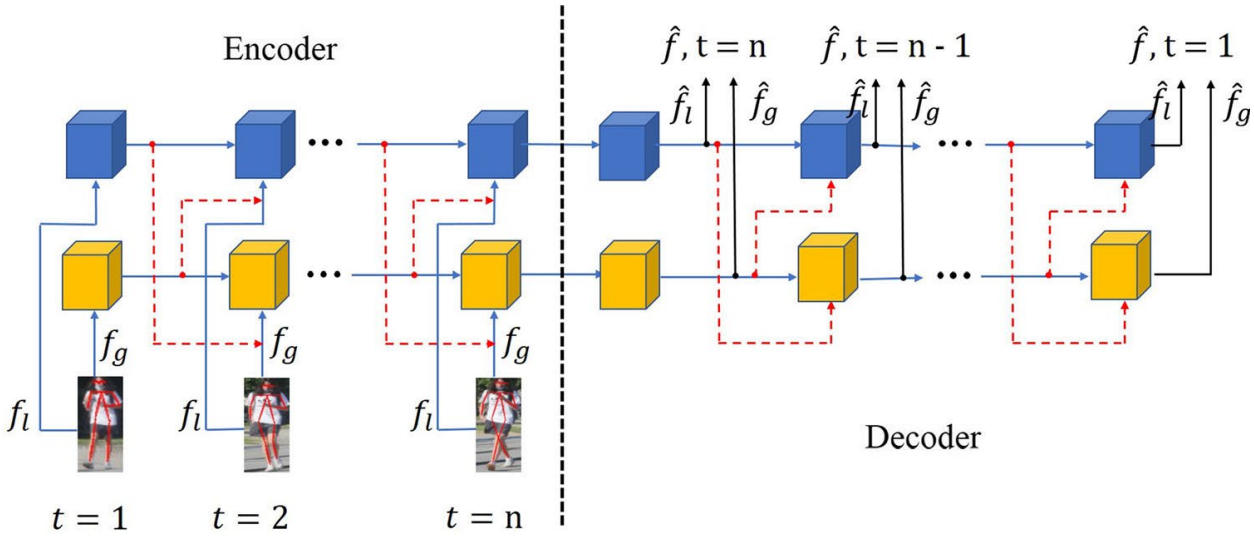


Figure 9. A message-passing encoder-decoder recurrent neural network

This procedure is similarly applied to the GRU unit in the other pathway. This mechanism enables a rich exchange of global and local features, thereby enhancing the overall model's effectiveness in capturing nuanced pedestrian behaviors. The global branch can be used as an

example. For $t = 1,2,\cdots,n$, where n is the length of the input time steps, the GRU operation in the encoder can be written as:

$$h_g^t = \mathrm{GRU}_{\mathrm{encoder}}\left(\left[f_g^t, M_{l2g}^t\right], h_g^{t-1}\right) \tag{5}$$

where h is the output of the GRU unit and t is the time step. M_{l2g}^t is the cell state information exchanged from the local channel to the global channel:

$$M_{l2g}^t = \sigma(W_l \cdot h_l^{t-1} + b_l) \tag{6}$$

The GRU operation and feature extraction along the local channel are performed in the same manner except that the cell state information will be fed from the global channel to the local channel, i.e., M_{g2l}^t.

After encoding the input segment, the global and local channels reconstruct the original input segment following the same procedure but in a reversed temporal direction, i.e., $t = n, n-1, \cdots 1$. The cell state at $t = n$ in the decoder is initialized with the encoder values at $t = n$ in Equation 5. Thus, the cell state of the global channel in the decoder is given by:

$$h_g^{t-1} = \mathrm{GRU}_{\mathrm{decoder}}\left(M_{l2g}^t, h_g^t\right) \tag{7}$$

Given global and local cell states in the decoder, the projected global and local features of the decoder $\hat{f}_g^t$ and $\hat{f}_l^t$ can be obtained by:

$$\hat{f}_g^t = F_c\left(h_g^t\right);\ \hat{f}_l^t = F_c(h_l^t) \tag{8}$$

where F_c is a fully connected layer. Then the segment of the skeleton trajectories $\hat{f}$ is reconstructed through another fully connected layer by linking the independent output $\hat{f}_g^t$ and $\hat{f}_l^t$:

$$\hat{f}_{rec}^t = F_c\left(\hat{f}_g^t, \hat{f}_l^t\right) \tag{9}$$

The loss function for model training contains the global loss L_{global}, local loss L_{local} and reconstruction loss L_{rec}, and can be calculated by a mean square error loss function:

$$L_* = \frac{1}{T}\sum_{t=i}^{j}\left|\hat{f}_*^t - f_*^t\right|_2^2 \tag{10}$$

where $*$ represents *global*, *local*, or *rec*. The overall loss function is a combination of those three losses:

$$L = \lambda_g L_g + \lambda_l L_l + \lambda_{rec} L_{rec} \tag{11}$$

where the parameters $\left(\lambda_g, \lambda_l, \lambda_{rec}\right) \geq 0$ are the corresponding weights to the losses. In the MPED-RNN model, the combined loss function is minimized, enabling the encoder to learn a set of features that provide a compact representation of normal skeleton trajectories in the latent space. Simultaneously, the decoder reconstructs the original input segment. During the testing phase, abnormal skeleton trajectories yield a significant reconstruction error, as their patterns deviate from the norm. The prediction decoder and the reconstruction decoder could be merged to form a single encoder-dual decoder structure (Morais et al., 2019); however, this research opted for the exclusive use of the reconstruction decoder, as the goal was to detect and localize

anomalies through the reconstruction error of trajectories. This approach also offers the added benefit of reducing computational time.

Using the skeleton trajectories output from AlphaPose, a preprocessing step divides the trajectory sequence into multiple overlapped segments. Each segment has a sliding window of size T, and the overlap length is $T-1$ (i.e., the offset between two successive segments is one frame). For example, a single trajectory sequence with a length m is converted into $m-T+1$ segments with each segment length of T. Global and local coordinates are extracted from each segment and fed into the MPED-RNN for reconstruction. The reconstructed results for global, local, and overall coordinates are computed using Equations 8 and 9. The combined losses for each segment are then aggregated using Equation 11.

3.2.3 Behavior Analysis and Anomaly Identification

A behavior analysis is conducted based on the reconstruction error of each pedestrian, which serves to identify anomalous frames and localize abnormal behaviors. As mentioned earlier, the trajectory sequence of a pedestrian is reformatted into overlapping segments. Let S_p denote a set of segments containing a specific pedestrian p. For the pedestrian-level anomaly analysis, the reconstruction error of pedestrian p at frame t is the average error of all the segments containing pedestrian p:

$$R_{ft} = \frac{\sum_{S_p} L_{rec}(p)}{|S_p|} \tag{12}$$

This shows that the pedestrian-level reconstruction error at frame t is the averaged results of all the segments in the set S_p. Eventually, anomalous events are localized by placing bounding boxes around pedestrians whose anomaly scores exceed the predefined threshold value.

The frame-level reconstruction error R_v is calculated by performing a max-pooling operation on the reconstruction loss of all pedestrians' trajectories $p_1, \cdots, p_k$ present in that frame v. The max-pooling operation suppresses the influence of regular trajectories, and thus the anomaly at the frame level is governed by the pedestrian contribution to the largest reconstruction error of trajectories. Specifically, the frame-level error R_v is given by:

$$R_v = max\left(R_{p_1}, \cdots, R_{p_k}\right) \tag{13}$$

where $R_{p_i}, i \in 1, \cdots, k$ denotes all the pedestrians' reconstruction errors present in the video.

3.2.4 Results and Discussion

Experimental details and evaluation results are presented based on the image dataset collected from the target grade crossings. More than 2,000 standard pedestrian trajectories were used to train the anomaly detection model, with 80 percent serving as training data and the remaining 20 percent as validation data. The model testing involved seven instances of abnormal behavior. The model was trained with a learning rate of 0.001 and a batch size of 256. The evaluation of the trained model was conducted using test data from the same location as the training data, but with varying camera view angles. Using training and testing data captured from different angles allowed the team to further examine the robustness and generalization capability of the proposed model and pipeline. Two case studies corresponding to two different grade crossings were investigated in this study, with results presented for both pedestrian-level and frame-level

anomaly detection. All experiments were conducted on an NVIDIA GeForce RTX 2080 Ti GPU computing workstation, equipped with 4,352 CUDA cores and 14 Gbps memory speed. Training the proposed anomaly detection model took approximately 3 hours and 35 minutes to complete 300 epochs with a batch size of 32.

3.2.4.1 Case Study 1

The testing data used in Case Study 1 (Figure 10) was collected from the first location depicted in Figure 6a. This data maintained the same resolution and image quality as the training data. Figure 10 presents two examples of unusual pedestrian behavior, with manually added bounding boxes for focus. In Figure 10a, a pedestrian lingers near the rail and places their feet on it. In Figure 10b, another pedestrian squats on the track. Such abnormal behaviors on the railway may indicate serious risks, such as potential suicide attempts, and hence need to be detected and localized to prompt timely alerts.

(a)

(b)

Figure 10. Two example frames in Case Study 1

The frame-level reconstruction error and ground truth were used to evaluate the performance of the proposed framework and the trained model (Figure 11). The fluctuation in the reconstruction error throughout all the video frames was computed via Equation 13, as shown in Figure 11a. To mitigate the high-frequency local variation and noise, a moving average (MA) operation was performed on the original curve. Subsequently, an empirical threshold value of 0.15 was set to differentiate abnormal behavior, with frames that have an MA-reconstruction error above the threshold line being classified as abnormal. It is important to note that while the threshold value is uniform for detecting all anomalous events in a specific location, it might differ across various grade crossing spots. Nevertheless, in this study, the same threshold was employed for different grade crossing spots post-model retraining. Figure 11b shows the ground truth of the anomalous frames, comprising two distinct abnormal pedestrian behaviors: 1) lingering from frame No. 340 to No. 900; 2) squatting from frame No. 900 to No. 1250; and 3) lingering around the railway from frame No. 1250 to No. 1450 until departure. Upon comparing the reconstruction error with the ground truth presented in Figure 11, anomalies were identified in frames ranging from No. 342 to No. 1493. This closely aligns with the manually labeled ground truth (frame No. 330 and No. 1450). Given that the testing video operates at 30 frames per second, this equates to a discrepancy of less than 2s between the model's prediction and the ground truth. This indicates that the proposed framework is proficient at detecting frame-level anomalies.

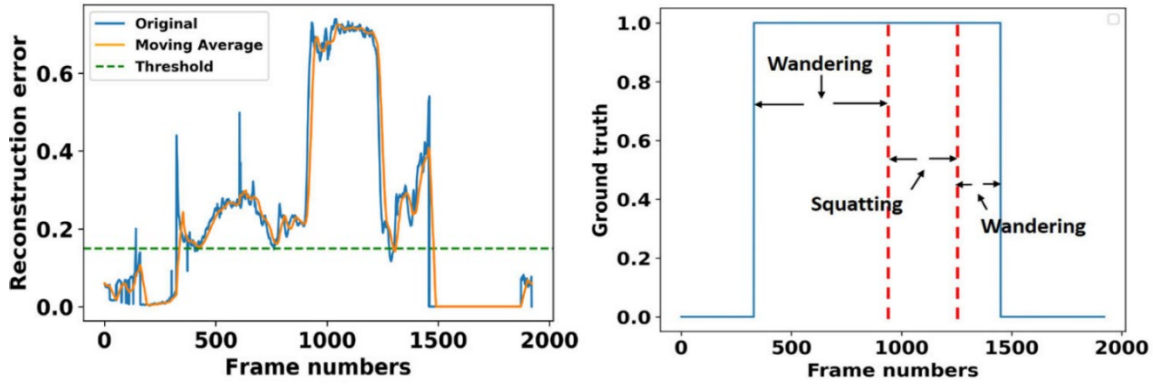


Figure 11. Performance of anomaly detection method in Case Study 1

An analysis was then performed on pedestrian-level reconstruction errors, which are associated with each individual pedestrian in every frame. Given that a single frame may encompass multiple pedestrians, the reconstruction error of each skeleton trajectory was examined to identify pedestrians exhibiting abnormal behaviors. Figure 12 illustrates the reconstruction error for the skeleton trajectories of six pedestrians, with a comparative threshold line added at a value of 0.15.

Figure 12a shows the error curve of the first pedestrian exhibiting abnormal behavior. It is apparent that the reconstruction errors in frames from No. 900 to No. 1250, which correspond to squatting behavior, are significantly higher than those in the remaining frames. The diagram also reveals that the reconstruction error for lingering behavior falls between that of squatting and regular walking. This is consistent with expectations since the skeletal coordinates and temporal variations associated with the squatting position are notably different from regular walking, more so than those associated with lingering. Additionally, as squatting trajectories were not present in

the training dataset, the decoder was unable to effectively reconstruct them using the dynamic patterns derived from regular walking.

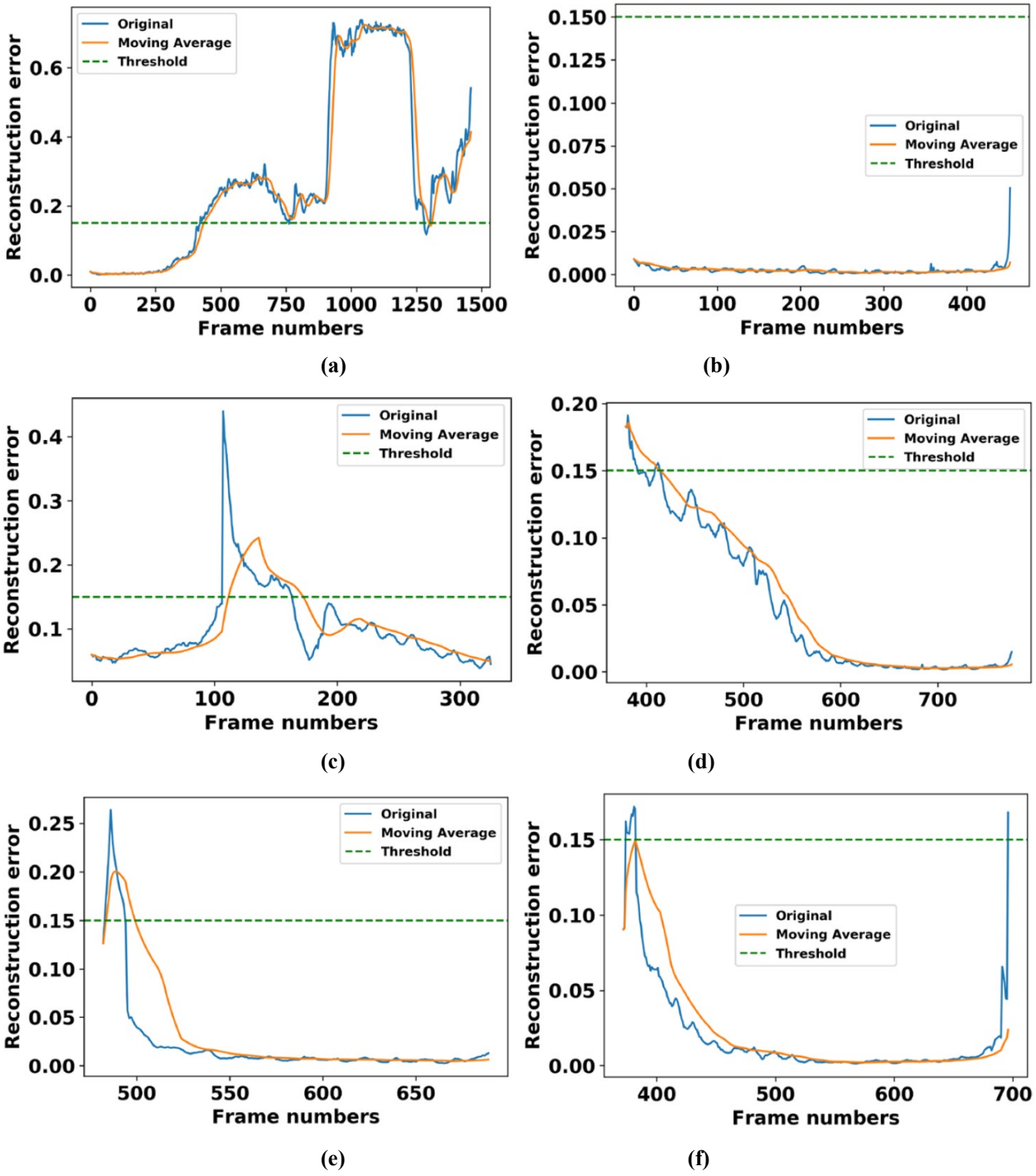


Figure 12. Reconstruction errors of the pedestrian's behaviors in the test video

The other error curves presented in Figure 12b-f correspond to pedestrians exhibiting normal behavior. It is important to note that a pedestrian's skeleton might not be detectable if it is obstructed by other pedestrians or objects, which could result in a discontinuous reconstruction curve. To ensure curve continuity, frames where the pedestrian does not appear were omitted.

Moreover, when two pedestrians overlap, it can cause tracking loss. The skeleton key points of the two pedestrians can become entangled; for example, the hand of one pedestrian may be misidentified as the torso of the other, resulting in inaccurate reconstruction errors. This can be observed in Figure 12c, where several frames of normal behavior were misclassified as anomalies. The application of an MA operation significantly reduced the occurrence of false positives, with fewer than 100 frames exceeding the threshold line in Figure 12c.

To localize anomalies, a bounding box was assigned to those pedestrians in each frame whose reconstruction error exceeded the predetermined threshold. In the scope of this study, researchers initially identified the numbers assigned to pedestrians displaying abnormal behaviors. The team then retrieved their skeleton points from the input data (as opposed to the reconstructed values) to generate bounding boxes that encapsulated these pedestrians within the corresponding video frames. Figure 13 shows example results of anomaly localization. Note that the curve depicted in Figure 13 was derived from the moving average of the frame-level error presented in Figure 11a. As seen in Figure 11a, the shape of curve above the threshold line corresponds closely to the shape observed in Figure 12a. This is due to the frame-level error being an amalgamation of pedestrian-level errors, assembled through a max-pooling operation as defined in Equation 13. As there was only one pedestrian displaying abnormal behavior in this case study, their reconstruction error as seen in Figure 12a was significantly greater than those of the other normal pedestrians shown in Figure 12b-f, thereby dominating the frame-level error. The computational time required by the proposed framework comprises two components: 1) generating the skeleton trajectories of pedestrians using AlphaPose, and 2) detecting and localizing the anomalies within these trajectories. In this case study, AlphaPose took approximately 40 seconds to process close to 1,900 frames. In terms of anomaly detection, the proposed method required about 23 seconds to detect and localize the anomalies across all these frames.

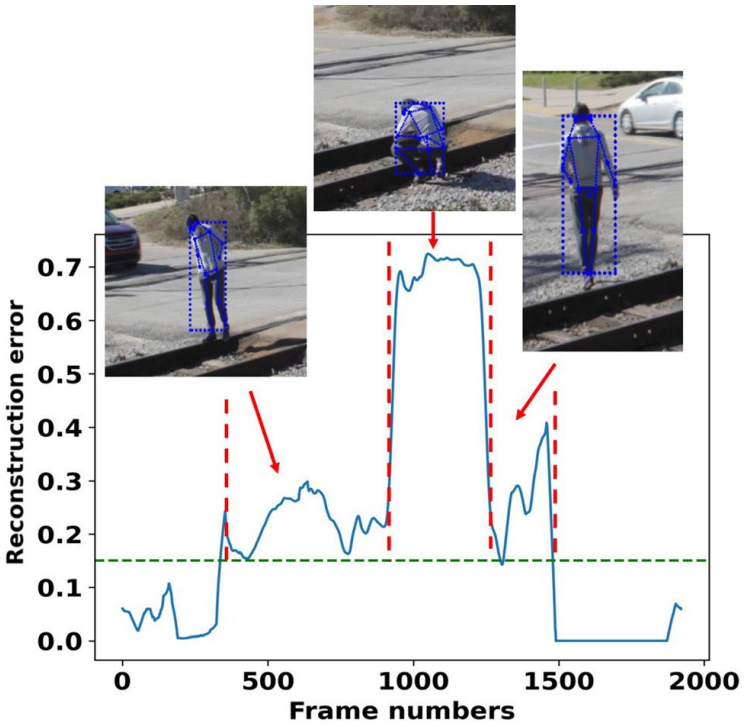


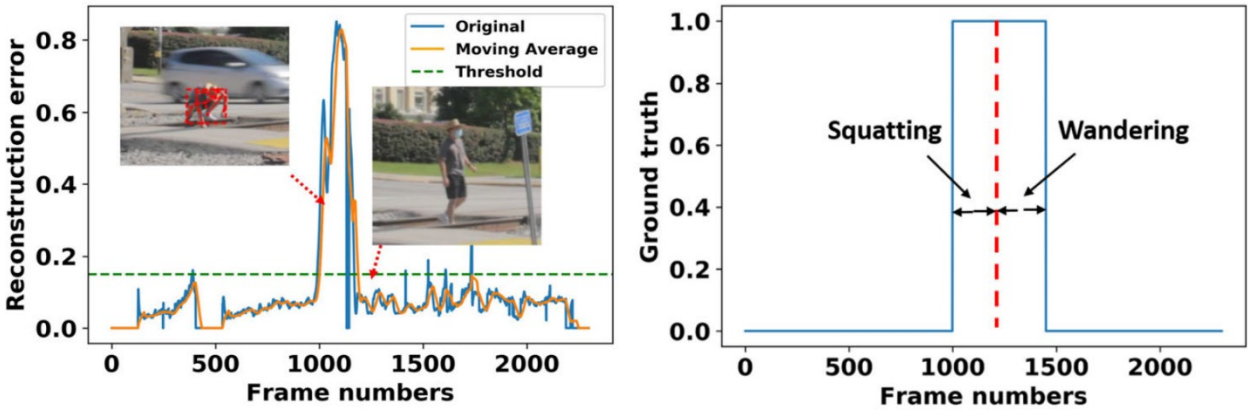
Figure 13. Anomaly localization results in Case Study 1

3.2.4.2 Case Study 2

A second case study was conducted at grade crossing spot 2, as illustrated in Figure 6b. Unlike the first case study, the team employed an opposite camera filming detection (located in the left bottom corner of Figure 6b) to collect the test data used to assess the robustness of the trained model. This consideration is crucial in real-world applications as the camera positioning may not always be consistent. This added variable made Case Study 2 more challenging than Case Study 1. Figure 14 presents an example of an anomalous event in this second case study, which similarly included abnormal behaviors such as lingering and squatting on the railway tracks.

Figure 14. An example of a frame with anomaly in Case Study 2

Figure 15 presents the results of anomaly detection and localization in Case Study 2. The same threshold value (0.15) was employed to differentiate between normal and abnormal events. Figure 15a shows that the abnormal behavior of a pedestrian was detected and localized within the video frames. Specifically, the squatting behavior was detected in frames approximately ranging from No. 1000 to No. 1150, where it produced a significantly larger reconstruction error compared to other frames. However, aside from this anomalous event, no others were detected. This contrasts with the ground truth depicted in Figure 15b, where lingering behavior was observed in frames No. 1150 to 1450. The inability to detect lingering behavior in this case study could be attributed to several factors. First, the lingering motion generally resembles regular walking, making the reconstruction error of skeleton trajectories less distinguishable between the two. Second, different camera filming directions were employed for collecting training and testing data at grade crossing spot 2. This discrepancy in the field of view between the training and testing data presents additional challenges for anomaly detection, as it becomes difficult to ascertain whether the lingering behavior is occurring on the rail or on the road. These issues can potentially be mitigated by leveraging a more comprehensive model or training on a richer dataset that includes pedestrian skeleton trajectories from numerous grade crossings. Such strategies will be considered in future work.



(a) (b)

Figure 15. Results of anomaly detection and localization in Case Study 2

These test results corroborate the robust performance of the proposed model in detecting and localizing anomalous behaviors such as squatting and lingering. This was demonstrated in the test data sourced from the same location with near-identical filming conditions. However, when applied to a new location or under significantly varied filming conditions, the model detection capabilities could be limited in identifying squatting behavior.

3.2.5 Comparison to Other Models

The proposed model was further evaluated in the other five cases using common assessment metrics. The receiver operating characteristics (ROC) curve was employed, which quantifies both the true positive rate (TPR) and false positive rate (FPR) with varying threshold values:

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{14}$$

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{15}$$

where TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative. The proposed method was quantitatively evaluated using the area under the curve (AUC); a higher AUC value indicates better anomaly detection performance.

Figure 16 shows the ROC curve for the proposed method at the frame level.

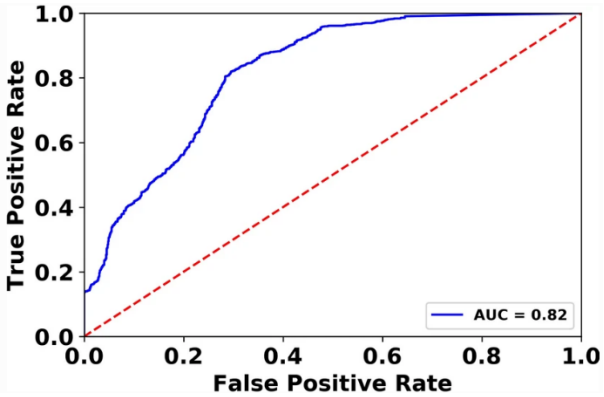


Figure 16. The ROC curve of the proposed method

The average AUC value across all seven test cases was 0.82. This indicates the method's strong ability to detect frame-level anomalies and associate them with individual pedestrians within a given frame. As discussed in the preceding section, erroneous detections predominantly occur due to the similarities in motion patterns between lingering and regular walking.

To further assess the efficacy of the proposed method for detecting anomalous pedestrian behaviors at grade crossings, it was compared against two established anomaly detection methods using the collected dataset. These included a baseline spatial-temporal autoencoder network (ST-AE) as per Chong and Tay (2017), and the state-of-the-art memory-guided normality for anomaly detection (MNAD) approach by Park et al. (2020). Table 1 illustrates a comparative analysis of these methods, highlighting the average frame-level AUC achieved using the study dataset.

Table 1. Comparison of the present method with other video-based models

Methods	AUC
ST-AE	0.73
MNAD w/o Mem	0.75
MNAD Mem	0.78
Proposed method	0.82

Both the ST-AE and MNAD methods registered lower AUC scores compared to the proposed model, despite achieving high values on public datasets such as UCSD and ShanghaiTech. Several factors contributed to this disparity in performance. In public datasets, pedestrian walking is considered the sole “normal” event, while other behaviors (e.g., running, skateboarding, bicycling, etc.) and the presence of non-pedestrian objects (e.g., vehicles, golf cars, trucks, etc.) are all classified as anomalies. However, within the context of the grade crossing scenes, such motion patterns exhibited by pedestrians, and even the presence of vehicles and trucks, were treated as normal. Consequently, this dataset, replete with a multitude of vehicles and trucks, introduced complexity that can confound traditional video-based anomaly detection models during the training phase.

In contrast, the proposed model extracts the skeletal trajectories of pedestrians from raw video data, effectively disregarding non-pedestrian entities. This significantly mitigates any interference during the training stage. Therefore, the current framework can analyze pedestrian behaviors with higher accuracy within a complex grade crossing environment containing various types of objects. This nuanced analysis can pose a significant challenge to existing end-to-end models that learn directly from videos.

3.3 Analysis of Abnormal Pedestrian Behaviors at Grade Crossings Based on Semi-Supervised Generative Adversarial Networks (GAN)

Figure 17 shows the proposed anomaly detection pipeline.

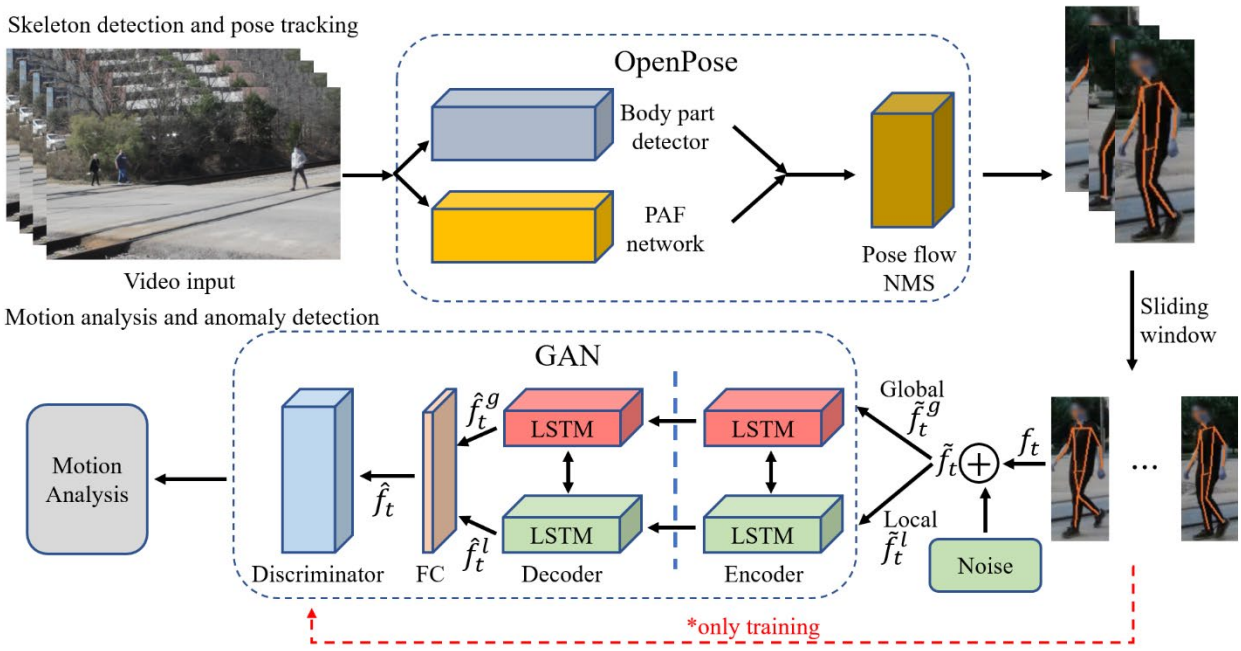


Figure 17. Overview of the anomaly detection pipeline

As shown in Figure 17, the proposed pipeline consists of three key modules:

1. Skeleton Detection and Pose Tracking: This module detects and tracks the skeletal points of each pedestrian in the video to generate trajectories that represent the corresponding pedestrian's movement. To facilitate swift analysis, the team used OpenPose (Cao et al. 2017) for frame-level skeleton extraction of each pedestrian. OpenPose employs a dual-branch convolutional neural network that inputs an image, detects individual parts of a human body, and predicts the associations between these parts within the part affinity fields. As a result, 2D skeleton points and human poses are generated by integrating the intermediate outputs from these two branches. The OpenPose output comprises a set of 17 key skeletal point trajectories, which serve as descriptors of pedestrian behavior in the video. In this way, more nuanced information about behaviors (e.g., localized body motion) can be distilled. Compared to other methods that use images as inputs, the use of key points enables the extraction of richer behavior and pose features. The data preprocessing employed in this study aligns with the approach described in Section 3.2.1.
2. Anomaly Detection: The trajectories collected from the previous stage are divided into two categories: normal and abnormal behavior. For behavior analysis and anomaly detection, only the normal trajectories are used to train a Generative Adversarial Network (GAN) model. To enrich the feature sets representative of gait and body motion (e.g., relative movement between limbs, torso, head, etc.), the trajectories are normalized in both global and local coordinates, and then input into the GAN model. Owing to its semi-supervised learning nature, this GAN model only learns the motion patterns of normal pedestrians. Specifically, the GAN model consists of a generator (G) and a discriminator (D). The generator primarily functions as a reconstruction network and consists of two interacting branches. Each branch comprises a LSTM-based recurrent neural network that reconstructs corresponding inputs in the temporal domain. The discriminator, on the other hand, is a fully connected neural network that generates an anomaly score indicative of

the likelihood of encountering previously unseen patterns. The use of the discriminator varies in the training and testing stages. During training, it processes both the original input samples and the samples reconstructed by the generator, assigning different scalar values of the anomaly score to both sets. In the testing stage, the discriminator only inputs reconstructed samples and outputs the anomaly score for each pedestrian. This score is then used as a criterion for anomaly detection.

3. Motion Analysis: If the anomaly score for a particular pedestrian exceeds a predefined threshold, a bounding box is generated. This box is formed using the coordinates of the individual's skeletal points at the current frame. The bounding box then tracks the motion of this pedestrian in the video until the anomaly score dips below the threshold.

For testing, the three modules above run sequentially.

3.3.1 GAN-based Anomaly Detection Model

Given the high level of unpredictability inherent in human behaviors, collecting sufficient data on all abnormal pedestrian movements presents a considerable challenge. Moreover, manually labeling ground truths for all trajectories can be an arduous process, given the large volume of the dataset. Therefore, in this study, the team adopted a semi-supervised learning method to train the GAN model. This approach allowed researchers to detect abnormal pedestrian movements at grade crossings in videos by identifying them as outliers, relative to the features learned exclusively from normal pedestrians. In essence, the model first learns the features of normal motions by training with trajectory sequences of normal pedestrians. These features are then used to reconstruct skeletal trajectories from either normal or abnormal pedestrians. The model finally detects anomalies in behaviors by evaluating the reconstruction error.

As mentioned previously, each pedestrian's global features are represented through bounding box information formed by all their skeleton points, while local features capture the relative motion of these points, normalized by the global features. Both global and local features share a strong correlation. For instance, a pedestrian positioned far from the camera on a global scale will also exhibit low magnitude local relative motion and vice versa. Hence, to enhance model accuracy and generalizability, the network should allow early exchange of global and local features through a message-passing scheme, rather than processing them separately. Given that the skeletal trajectories represent time-series data, the team used an encoder-decoder network based on LSTM cells as the generator to learn normal pedestrian trajectory patterns while accounting for behavioral correlations between time instants. Figure 18 illustrates the architecture of the generator, which comprises two recurrent encoder-decoder branches. Each block along a branch signifies an LSTM cell designed to process time-series data (Greff et al., 2016; Chen et al., 2022). The left branch (colored red) processes the global features, while the right branch (colored green) handles local features. These two branches exchange their information in the temporal direction, achieved by treating the cell state of one node in each branch as additional input to the node at the next time instant in the other branch. Consequently, every cell, except for the first layers of both the encoder and decoder, receives information from both branches from the preceding time instant.

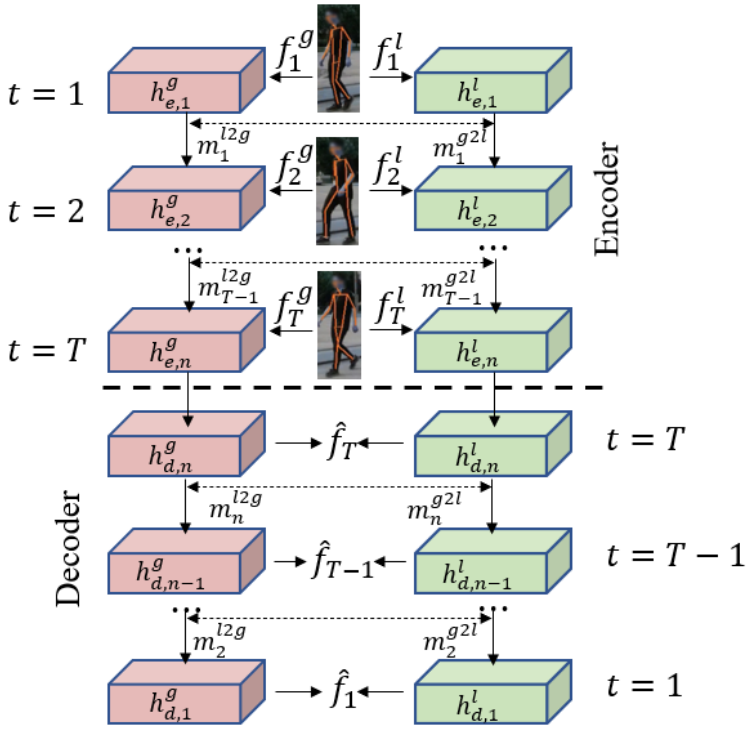


Figure 18. Model architecture of the generator

The hidden state h of all LSTM cells in the encoder are initialized to null. f_t^l and f_t^g of the skeleton points at each time step are obtained following the motion decomposition procedure presented in Section 3.2.1. The message exchanged between the local and global branches are given in Equation 16 and Equation 17:

$$m_t^{l2g} = \sigma\left(W^{l2g} h_{t-1}^l + b^{l2g}\right) \tag{16}$$

$$m_t^{g2l} = \sigma\left(W^{g2l} h_{t-1}^g + b^{g2l}\right) \tag{17}$$

where the superscript $l2g$ denotes the information from the local branch to the global branch, and $g2l$ for the information from the global branch to the local branch. W and b are the weight and bias, respectively, and will be learned during training, while σ is the activation function which in this model is the Rectified Linear Unit (ReLU). Then, for $t = 1, 2, \cdots n$, the LSTM encoder operation for both global branch and local branch is described as:

$$h_{e,t}^l = \text{LSTM}\left(\left[f_t^l, m_t^{g2l}\right], h_{e,t-1}^l\right) \tag{18}$$

$$h_{e,t}^g = \text{LSTM}\left(\left[f_t^g, m_t^{l2g}\right], h_{e,t-1}^g\right) \tag{19}$$

where subscript e denotes quantities associated with the encoder. After encoding the input, the decoders for both the global and local features attempt to reconstruct the input following the same procedure but in a temporally reversed direction. The initialization of the hidden states in each cell is defined as $h_{d,t}^{l/g} = h_{e,t}^{l/g}$. Therefore, the state in the decoder cell is computed by:

$$h_{d,t-1}^l = \text{LSTM}\left(m_t^{g2l}, h_{d,t}^l\right) \tag{20}$$

where subscript d denotes the quantities for the decoder. Then the reconstructed local features are calculated through a fully connected layer using the hidden state of each cell as $\hat{f}_t^l = \mathrm{FC}(h_{d,t}^l)$. The decoder in the global branch uses the same operation to compute $\hat{f}_t^g$. The reconstructed trajectory is generated through a fully connected layer by incorporating outputs from both global and local branches.

$$\hat{f}_t = FC(\hat{f}_t^l, \hat{f}_t^g) \tag{21}$$

The architecture of the discriminator, as shown in Figure 19, is a fully connected neural network of two layers with the Sigmoid activation function. The number of neurons in the first fully connected layer is equivalent to the size of input trajectories sequence (i.e., T). The number of neurons in the second fully connected layer is reduced to only $T/3$.

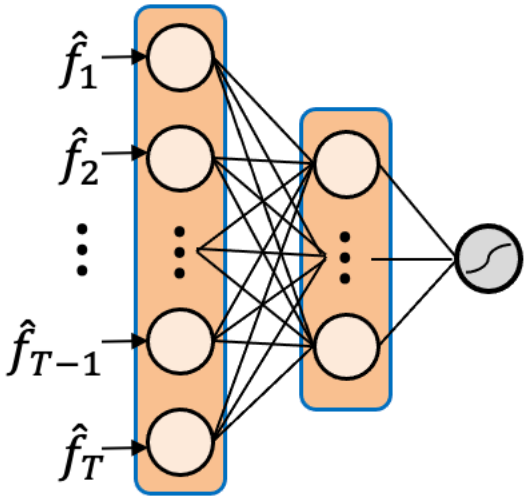


Figure 19. Model architecture of the discriminator

The generator and the discriminator of the GAN model are trained and updated adversarially in the semi-supervised manner. Only trajectories of normal behaviors are used for training the model. During training, the generator takes skeleton trajectories of normal behaviors as inputs and learns to reconstruct them to minimize the 2-norm loss function defined below:

$$l_{rec} = \|f_t - \hat{f}_t\|_2 \tag{22}$$

where f_t represents input samples of the trajectory sequence and $\hat{f}_t$ describes the reconstructed samples. The properly trained generator is expected to reconstruct inputs with minimal reconstruction error, meaning the reconstructed samples should closely resemble the original inputs. During each training epoch, the discriminator is applied twice, once with the reconstructed samples as input and then again with the original samples. A scalar value, or anomaly score, is assigned to each sample. The discriminator network is trained to differentiate between these two categories of samples, assigning reconstructed samples a value close to one, while original samples are assigned a value close to zero. Upon completion of training, the discriminator is anticipated to assign a low value to samples reconstructed from normal patterns, given that their distributions should now mirror those of the normal input samples. Consequently, the loss function used for training the discriminator is defined in Equation 23:

$$l_D = \log\left(1 - D\left(G\left(\tilde{f}_t\right)\right)\right) + \log\left(D(f_t)\right) \tag{23}$$

where $\tilde{f}_t$ is the input of skeleton key point trajectories with added noises, and its details are given below. $G(\cdot)$ represents the output from the generator with this noisy input $\tilde{f}_t$. Similarly, $D(\cdot)$ is the output result of the discriminator.

During the testing phase, the trajectory sequences extracted from images, which may contain abnormal behaviors, are input into the anomaly detection model. The reconstruction of abnormal behaviors will vary significantly from normal ones because the generator, having never learned the features in abnormal trajectories, fails to accurately reconstruct them based solely on the distribution of normal behaviors. Unlike the training phase, the discriminator in the testing phase operates in a single pass for each run, taking the reconstructed sample from the generator as input. Sequences of abnormal trajectories will exhibit a significant reconstruction error after processing by the generator, resulting in a high anomaly score (close to one) from the discriminator, marking it as an outlier. The higher the score, the greater the probability of anomaly in the motion patterns.

It is important to note that the essence of this anomaly detection model is to exploit the overfitting of the GAN model to generate large errors for unseen motion patterns. However, the accompanying issue with this overfitted model is that it can also reject unseen normal behaviors, such as those from different scenes (represented by various pedestrians, grade crossings, or camera view angles). To mitigate this issue and enhance model generalization, noise is added to the original training samples of trajectory sequences before they are input into the generator. Consequently, the generator acts as a denoising network, revealing key trajectory patterns even in variable environments. The noise must follow a specific distribution with appropriate amplitude to ensure optimal detection performance without severely contaminating the dataset. The dataset with added noise used for training is presented in Equation 24:

$$\tilde{f}_t = f_t + \left(\eta \sim N(0, \sigma^2)\right) \tag{24}$$

where η is the added noises in the normal distribution with the mean equal to zero and the standard deviation equal to σ. In this case, σ is empirically set to 10 percent of the maximum input amplitude, which was found to retain important motion patterns in the original dataset and yield the best performance in generalization and accuracy.

As previously mentioned, the discriminator assigns scalar values based on the likelihood of its input conforming to the distribution of normal patterns, thereby enhancing the reconstruction performance of the generator. Therefore, thresholding can be exclusively applied to the discriminator network. Frames exhibiting anomalous motions can receive high anomaly scores from the discriminator multiple times, as they may be included in several overlapping segments. Let $\mathbb{S}_{il}$ denote the set of all segments containing a specific frame l of specific pedestrian i, then the anomaly score associated with this pedestrian at frame l, D_{il} can be calculated as the average score within the set $\mathbb{S}_{il}$:

$$D_{il} = \frac{\sum_{j=1}^{\left|\mathbb{S}_{il}\right|} D_j}{\left|\mathbb{S}_{il}\right|} \tag{25}$$

3.3.2 Results and Discussion

This section presents the experimental details and evaluation results using the collected dataset. The training dataset comprises over 2,000 trajectories of normal pedestrians, 80 percent of which are used for training and the remaining 20 percent for validation. The testing dataset includes trajectories of both normal and abnormal behaviors from both locations, but different camera view angles are used at the second location, as depicted in Figure 6b. The use of various angles aims to verify the robustness and generalizability of the proposed framework. The model, which is built on PyTorch v1.11.0, was trained and runs on an NVIDIA GeForce RTX 2080Ti GPU, equipped with 4,352 CUDA cores and 11GB of memory.

3.3.2.1 Case Study 1

Figure 20a depicts the frame-level anomaly score predicted by the discriminator for a pedestrian exhibiting abnormal behavior alongside the ground truth. To filter out high-frequency local variations and noise to achieve a smoother prediction, a moving average operation with a window size of 15 was applied to the original output curve. Two fixed thresholds were established to detect the two abnormal behaviors, lingering and squatting: a lower threshold (illustrated in green) set at a value of 0.5, and a higher one (depicted in purple) set at 0.9. Frames with anomaly scores below the lower threshold were categorized as normal, while those above the higher threshold correspond to squatting motion. Frames that capture lingering action yielded scores that fell between these two thresholds. The detection results align closely with the ground truth in the temporal domain. The ground truth indicates that the pedestrian squats between frames No. 900 and No. 1250, and lingers from frame No. 340 to No. 900 and from No. 1250 to No. 1450. It's noteworthy that squatting behavior is significantly easier to detect because its anomaly score reaches a plateau, exhibiting unmistakably high values.

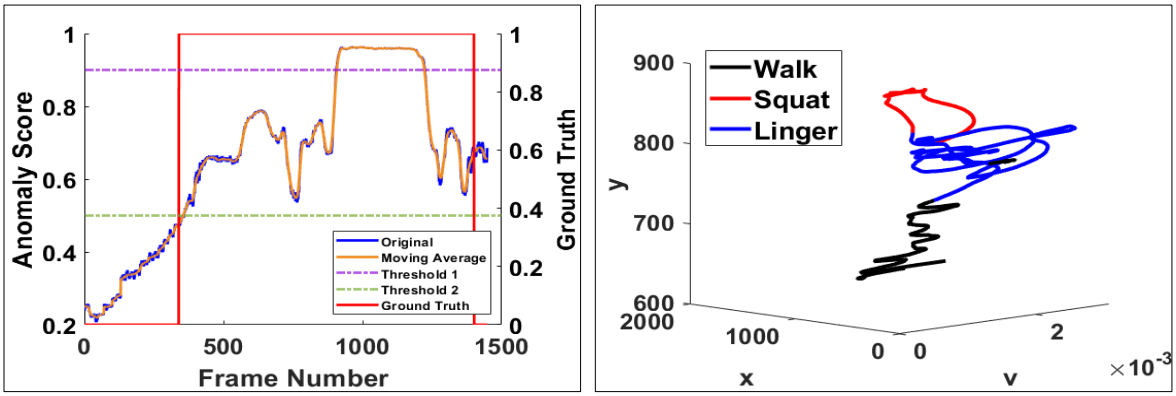


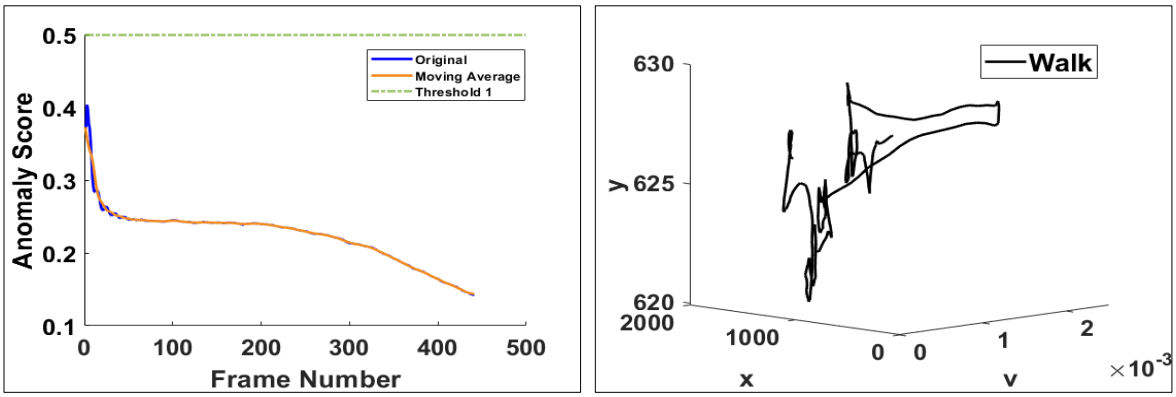
Figure 20. Anomaly detection results for the abnormal behavior in Case Study 1

The video coordinates used in all case studies were identical, with the origin located at the top left corner of the frame and the horizontal and vertical directions denoted by the x- and y- axes, respectively. The trajectory of the pedestrian is visualized in Figure 20b to assist in understanding the behavior, where the x- and y-coordinate values stand for the center position of the bounding box (c_x and c_y) as pixel indices along the horizontal and the vertical directions in the video. The v- axis is the speed of the pedestrian, all indicative of the dynamic characteristic of different pedestrian motions. The speed of a pedestrian at frame t was calculated using the information of the bounding box (c_x, c_y, w, h) as follows:

$$v_t = \sqrt{\left(\frac{c_{x,t+1} + c_{x,t-1} - 2c_{x,t}}{w}\right)^2 + \left(\frac{c_{y,t+1} + c_{y,t-1} - 2c_{y,t}}{h}\right)^2} \quad (26)$$

The central differencing scheme was employed for normalized velocity calculation. Different colors are used to represent various segments of the curve, reflecting the ground truth provided in Figure 20a. Specifically, segments representing normal walking, squatting, and lingering behaviors are colored in black, red, and blue, respectively. During normal walking, the *x*- and *y*-coordinate values, along with velocity *v*, undergo smooth changes within a specific period. In contrast, during lingering, the trajectory fluctuates randomly in both direction and magnitude at a high frequency, a clear deviation from normal walking behavior. For the squatting motion, a peak in the *y*-coordinate is the primary indicator, accompanied by a minor velocity change range. In essence, the distinction between lingering and squatting activities lies in the peak *y*-coordinate value and velocity range presence, and primarily stems from the *x*- and *y*- coordinate value fluctuations, which mirror changes in motion direction and frequency.

Figure 21 showcases the detection result of a pedestrian walking at a standard pace at the same grade crossing, with videos captured using identical camera settings. As demonstrated in Figure 21a, the discriminator's outputs for this pedestrian consistently remained below the lower threshold. Figure 21b delineates the *x*- and *y*-positions and the pedestrian's speed, clearly indicating a continuous trajectory along the *x*-direction. This pattern, coupled with a significant increase in pixel indices and speed oscillation between 0.001 and 0.003, aligns with normal walking behavior. The results from this first case study affirm the proposed pipeline's effectiveness.



(a) (b)

Figure 21. Anomaly detection results for the normal behavior in Case Study 1

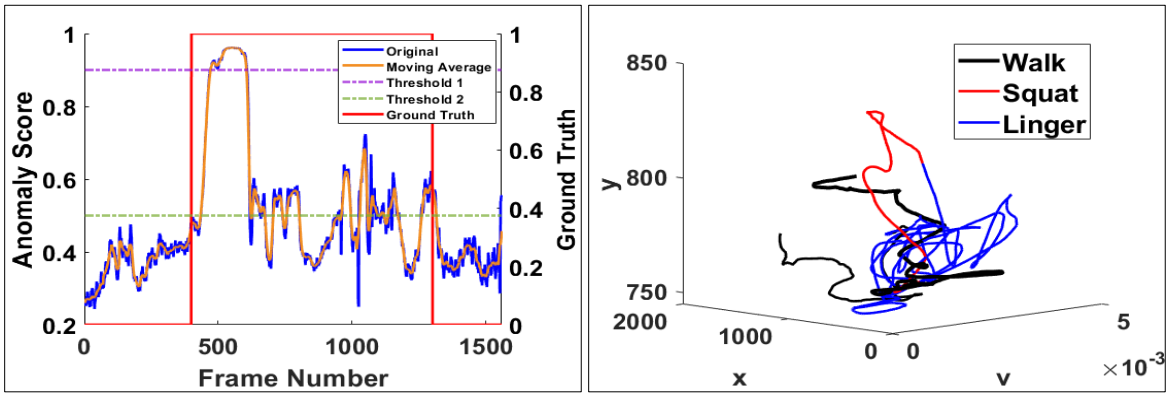
3.3.2.2 Case Study 2

The second case study used video collected from a different location, as depicted in Figure 6b. By including tests at an alternative grade crossing, the robustness and applicability of the proposed framework and model were evaluated, both crucial for real-world usage. The two anomalous activities, lingering and squatting on the rail track at this location, are showcased in Figure 22. Unlike the previous study, the lingering behavior in this case is bookended by normal walking and squatting. Furthermore, the pedestrian in this case study has a different appearance compared to that in Figure 10. Also, the videos capturing the abnormal behavior were primarily shot from the pedestrian's front, as opposed to the rear view in the previous case.

(a) (b)

Figure 22. Video sequences of two examples of abnormal behaviors in Case Study 2

Figure 23a depicts the frame-level output of the anomaly score from the discriminator, compared with the ground truth. Like the previous case, the moving average operation was used on the original output from the discriminator to smooth the results. The two constant thresholds from the first case study are still applicable here: frames with an anomaly score above 0.9 signal squatting behavior, while those with an anomaly score between 0.5 and 0.9 indicate lingering behavior. Frames with anomaly scores beneath the bottom threshold denote normal walking. In this case study, the pedestrian begins with normal walking, subsequently squats at the railroad track, and finally lingers at reduced speeds. Consequently, the highest plateau of the anomaly score corresponding to squatting was observed before lingering, mirroring the profiles observed in the first case study. Figure 23b presents the pedestrian's trajectory (x- and y-coordinates within the video frame) and their moving speed (represented on the v -axis). The red trajectory, occurring between frames No. 400 to No. 620, signifies the squatting motion, which serves as the ground truth for performance evaluation. The lingering trajectory, colored in blue, spans from frames No. 620 to No. 1300.

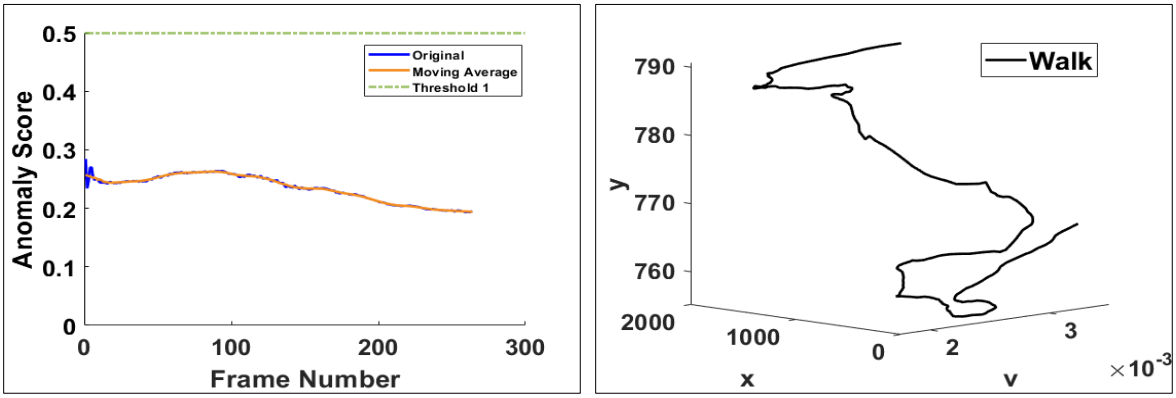


(a) (b)

Figure 23. Anomaly detection results for the abnormal behavior in Case Study 2

The variations in the profiles of x- and y-values, along with the v-speed, can effectively distinguish between the three motion patterns. For instance, the abrupt increase in the y-coordinate value (around 100 pixels) denotes squatting behavior. A trajectory with a speed slower than normal walking (specifically, $v < 0.001$), accompanied by randomly fluctuating positions, indicates lingering behavior.

Another test video showcasing only pedestrians demonstrating normal behavior at the same location was analyzed, as well. The results of the anomaly score and the trajectory are shown in Figure 24. The anomaly score consistently remained below 0.5, indicative of normal behavior. A substantial and continuous shift in the *x*-coordinate value was distinctly observed, whereas the *y*-coordinate remained relatively constant (fluctuating within a 50-pixel range). This mirrors the pattern of normal pedestrian motion as depicted in Figure 21b.



(a) (b)

Figure 24. Anomaly detection results for the normal behavior in Case Study 2

The comparison between Case Study 1 and Case Study 2 reinforces the model's generalizability. The model can be extrapolated to other grade crossings, and the need to retrain the model can be significantly reduced, if not completely eradicated. Furthermore, the two predefined thresholds for anomaly detection can still be effectively employed.

3.3.2.3 Case Study 3

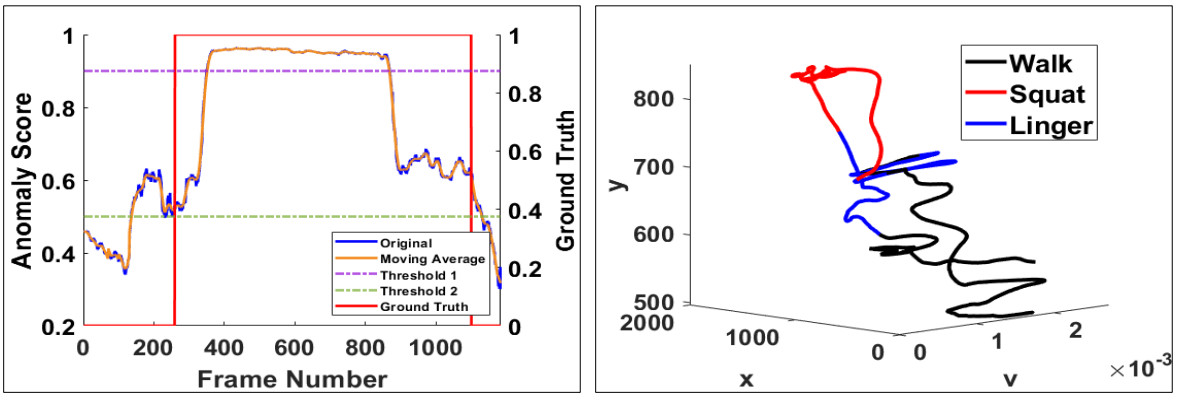
The third test video was captured at the same grade crossing as Case Study 1, but from a different camera view angle. This case study was designed to assess the impact of different camera angles on the robustness of the trained model. Videos displaying the same abnormal behaviors on the track were recorded and are displayed in Figure 25. In comparison to those shown in Figure 10 and Figure 22, the pedestrian is positioned closer to the camera, further testing the generalizability of the model.

(a) (b)

Figure 25. Video sequences of two examples of abnormal behaviors in Case Study 3

Figure 26a displays the frame-level anomaly score as predicted by the discriminator, with both original and moving average values compared against the ground truth. The squatting behavior, observed between frames No. 340 and No. 890, was successfully detected as their anomaly scores exceed the upper threshold. By contrasting the anomaly score with the upper and lower thresholds, frames from No. 260 to No. 340 and No. 890 to No. 1100 were identified as exhibiting lingering behavior.

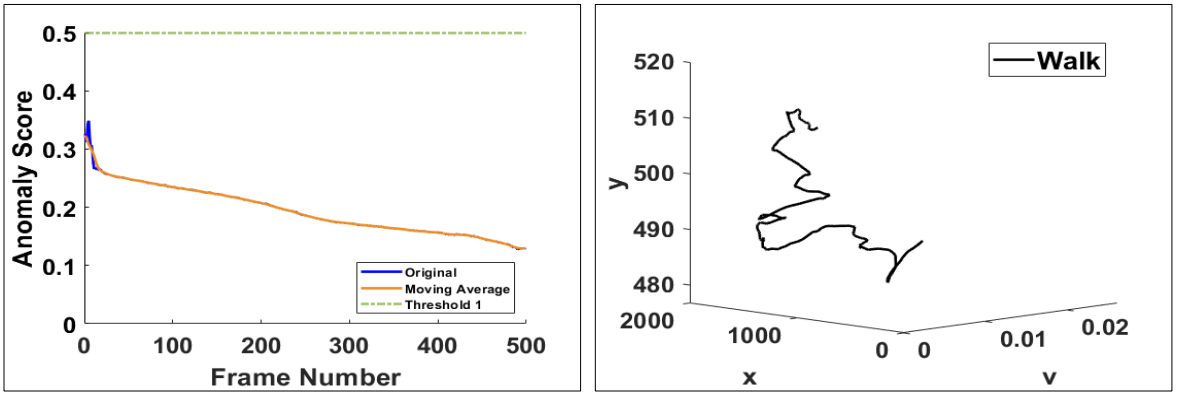
Figure 26b depicts the trajectory of the bounding box center in each frame, represented by the *x*- and *y*-coordinates. The value along the *v*-axis indicates the speed of the bounding box. The trajectory successfully detected and segmented normal walking, lingering, and squatting sections. This pedestrian exhibited unique gaits and patterns; the squatting motion manifested as a distinct, single peak in the *y*-coordinate, while the walking curve varied more smoothly, with fewer oscillations and a smaller *y*-coordinate range compared to the lingering motion. Lingering behavior (including short periods of stationary motion) is represented by more chaotic, slow-changing trajectories in one direction.



(a) (b)

Figure 26. Anomaly detection results for the abnormal behavior in Case Study 3

Figure 27 presents the results from analyzing a video containing only pedestrians walking normally at the same location. Findings consistent with previous studies were observed. The consistent results in both Case Study 2 and 3 affirm that the model performance is largely unaffected by the camera view angle, and that the threshold values of 0.5 and 0.9 remain valid.



(a) (b)

Figure 27. Anomaly detection results for the normal behavior in Case Study 3

3.3.2.4 Comparison to other benchmark models

The present model was evaluated in comparison with seven contemporary methods and models for anomaly detection in human behavior. These include MNAD (Park, et al., 2020), ASTNet (Le and Kim, 2023), OCGAN (Perera, et al., 2019), Dual Discriminator (Dong, et al., 2020), Spatiotemporal Dissociation Autoencoder (Chang, et al., 2022), Skeleton-transformer (Pang, et al., 2022), and the previous work discussed in Section 3.2. MNAD and the previous work used a deep auto-encoder reconstruction network, where abnormal behaviors are determined by reconstruction error. MNAD employs an additional memory module to record normal patterns and improve detection performance. The previous work used a model with two interacting branches to fully identify the normal behavior's skeletal trajectory patterns. ASTNet combines an attention mechanism with a residual autoencoder to automatically select features for learning. OCGAN and Dual Discriminator are based on a generative adversarial network. OCGAN employs two discriminators, one to restrict latent space and another to enhance image reconstruction performance. The Dual Discriminator model features two discriminators that assess whether input images and motion trajectories are authentic or simulated. The Spatiotemporal Dissociation Autoencoder model proposed by Chang et al. uses convolutional neural networks to disentangle spatiotemporal representation and learn their features independently. Specifically, the spatial autoencoder is designed to reconstruct the input of the initial individual frame, while the temporal part simulates the motion of the optical flow using the RGB difference of the first four consecutive frames as inputs. Pang et al.'s Skeleton-transformer employs a multi-head self-attention module and temporal convolutional layers as the core architecture for predicting future poses in video frames. Errors between predictions and reality serve as anomaly scores for distinction.

Table 2 shows a comparison in terms of average frame-level AUC when these six methods were tested on a custom dataset at grade crossings. The semi-supervised GAN model-based method surpassed the other five models in performance. Despite the impressive performance of MNAD, ASTNet, OCGAN, and Dual Discriminator on several public datasets (e.g., UCSD and ShanghaiTech), they seemed to underperform in this task. This is likely due to differences in dataset composition and application intent. While these public datasets treat non-pedestrian objects (e.g., vehicles) as anomalies, the model in this research, aided by OpenPose and skeleton detection, excludes such objects from consideration. This model is, therefore, more focused on analyzing trajectories and motion patterns, which results in a more precise representation of individual pedestrian motion patterns and the detection of abnormal behaviors at grade crossings without vehicle interference. Compared to the prior work, the neural network-based discriminator in the current model delivered a more comprehensive understanding of skeleton trajectory reconstruction. In the training phase, the generator network learns and updates by optimizing both reconstruction error and discriminator output, thereby extracting features of normal behaviors more effectively. During testing, detections based on the discriminator output are more resilient than those derived from a single static threshold on reconstruction error. This adaptability makes the proposed model more transferable and extendible to other grade crossings without the need for retraining. The use of overfitting enhances the distinction between abnormal and normal behaviors, and the introduction of added noise further augments the model's generalization capabilities and applicability across diverse environments.

Table 2. Comparison of methods on custom dataset

Methods	AUC
OCGAN	0.833
Dual Discriminator	0.807
ASTNet	0.846
MNAD	0.783
Spatiotemporal dissociation autoencoder	0.879
Skeleton-transformer	0.877
Previous work	0.822
The proposed method	0.89

3.4 Implementation in Real-time Testing

The team developed an online testing platform featuring a real-time monitoring video capture module (shown in Figure 28). This module uses a Logitech HD Pro Webcam C920, mounted on a tripod, capable of capturing 1080p resolution videos that meet the necessary image quality for effective skeleton detection and tracking. Computation and image processing were performed using an NVIDIA Jetson AGX Orin Developer Kit in real-time. To expedite the computational processes, all PyTorch models trained offline were converted to the TensorRT framework. In a summary of the testing process, video footage from the webcam was transferred to the Orin edge computer via USB-B port. Both the skeleton detection and tracking model, as well as the GAN model inference and post-processing, operate on the Orin system, ensuring that the platform is completely field-ready.

Figure 28. Real-time monitoring video capture module hardware implementation

Additional real-time field testing was conducted at the grade crossing located at 305 Country Club Dr, Titusville, FL, as shown in Figure 29.

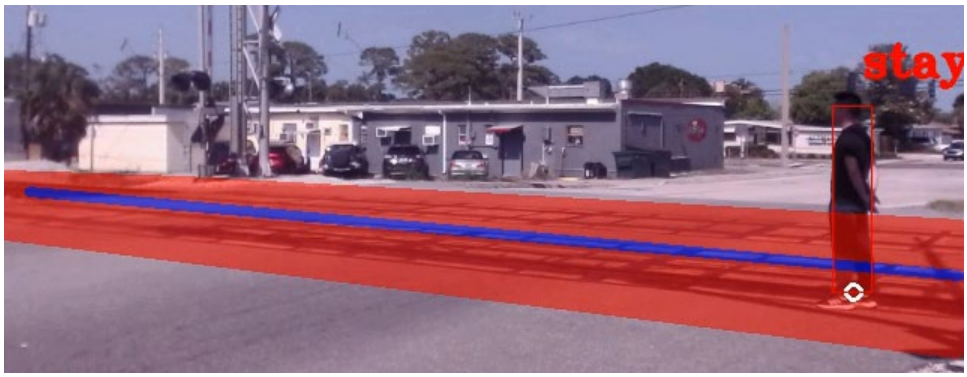
(a) **(b)**

Figure 29. The grade crossing at 305 Country Club Dr, Titusville, FL

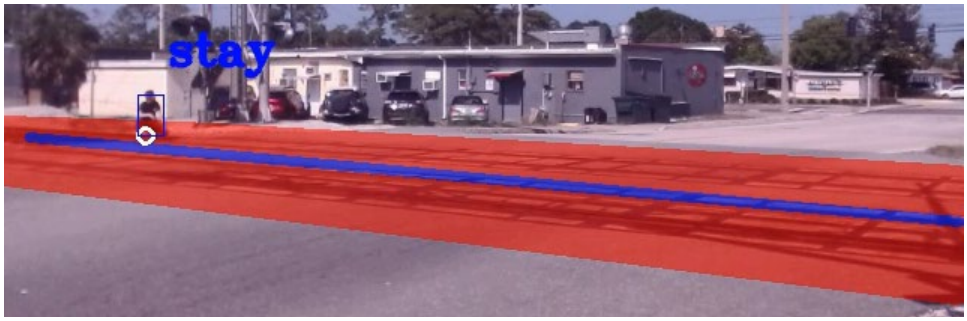
To concentrate on the grade crossing area, both the region of interest and rail track location were manually delineated on the first video frame captured following model initialization. A pedestrian's associated trajectories were sent to the GAN model for motion analysis only if the center of their feet appeared within the region of interest. Moreover, the coordinates of the center of the feet was used to detect an additional anomalous behavior – walking along the rail track. If the normalized variation in the distance between the center of the feet and the rail track remains at a low magnitude (i.e., below a predefined threshold) for several consecutive frames, the pedestrian in question was considered to be walking along the rail track. The numerical equation for calculating distance variation Δd_t is given by:

$$\Delta d_t = \frac{|d_t - d_{t-1}|}{d_t} \tag{27}$$

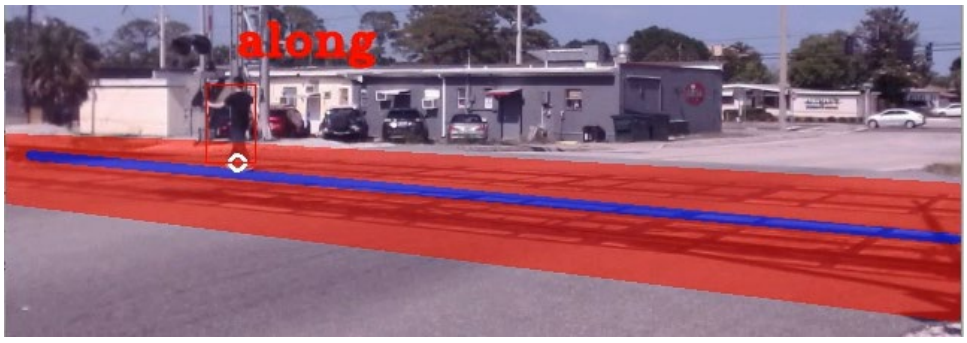
Figure 30 and Figure 31 illustrate the visual results with a single pedestrian and two pedestrians, respectively. In Figure 30, the region of interest is emphasized in red, while the blue straight line represents the approximate location of the rail tracks. The center of the pedestrian's feet is marked with a white circle. Anomalous behaviors such as lingering in the near view, squatting in the far view, and walking along the rail track were successfully detected. A bounding box around the corresponding pedestrian, accompanied by behavior annotation, serves as the visual alarm in the video.



(a) Lingering observed in near view



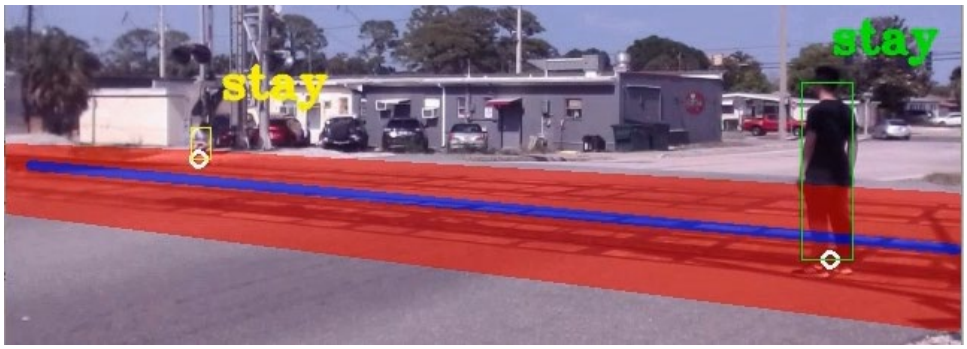
(b) Squatting observed in far view



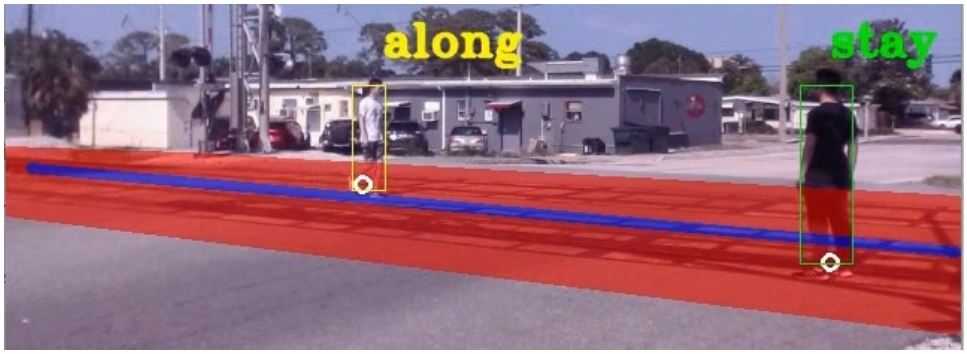
(c) Walking along the rail track

Figure 30. The visual detection result in the case of single pedestrian

In a scenario with two pedestrians, the GAN model successfully identified abnormal behaviors in both individuals. Consequently, the anomaly detection system generated visual alarms surrounding the associated pedestrians. Additionally, the proposed framework can process images and detect and analyze abnormal behaviors at a frequency of approximately 12 frames per second (fps) in scenarios involving multiple pedestrians. This frame rate is satisfactory for real-time testing requirements considering the typical movement rate of the human body.



(a) One pedestrian lingers in the near view and one pedestrian squats in the far view



(b) One pedestrian lingers in the near view and one pedestrian walks along the rail track

Figure 31. The visual detection result in the case of two pedestrians

4. Outliers Monitoring based on Foreground Segmentation

Beyond pedestrians, vehicles and other unexpected objects within a grade crossing can present significant safety risks. To address this, the research team created an auxiliary module, dubbed the Railroad Crossing Surveillance and Foreground Extraction Network (RC-SAFE), to augment the i-ASAP system. This enhancement enables the accurate and efficient detection and tracking of both stationary and mobile objects in the railroad crossing area.

4.1 Advantage of Foreground Detection

Foreground segmentation, often referred to as change detection, is a vital process in computer vision applications. Its main objective is to differentiate the moving outliers (i.e., foreground) from the static or stationary components of a scene (i.e., background). This critical separation of dynamic elements from an inert backdrop can be used in a multitude of applications including video surveillance, autonomous vehicles, anomaly detection, and many more.

This differs significantly from the more frequently used approach of CNN-based object detectors. These detectors classify objects into specific pre-established categories, while foreground detection employs a more universal approach, capturing all anomalies or outliers in each scenario as the Region of Interest (ROI), regardless of their classifications. Figure 32 highlights the distinctions between these two methodologies.

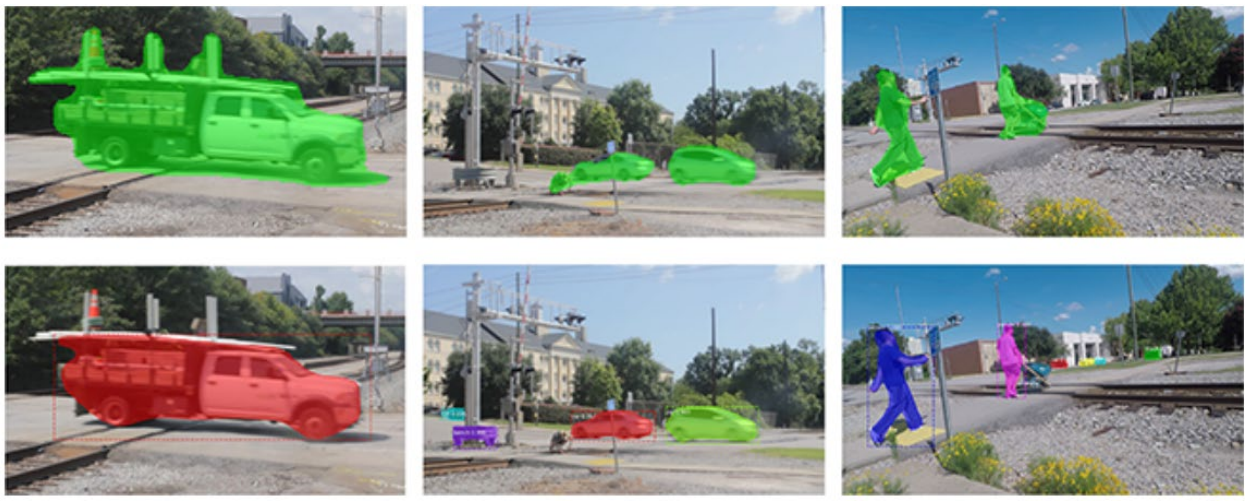


Figure 32. Differences between foreground segmentation and object detection (from top to bottom: foreground detection, object detection)

Object detection, powered by sophisticated CNN algorithms, excels at recognizing and categorizing familiar objects within a scene such as pedestrians or cars, thereby providing a nuanced understanding of the scene's contents; however, this capability comes with certain limitations. It can falter when faced with unusual or atypical objects like pipes, a squatting person, or a wheelbarrow. These errors may manifest as false negatives (i.e., where an object of interest goes unnoticed or undetected by the system) or false positives (i.e., where irrelevant objects or background elements are erroneously identified as objects of interest).

Foreground detection, on the other hand, adopts a considerably more inclusive approach. It proficiently identifies all moving elements, anomalies, or outliers within a scene as foreground

objects, without any inherent preference for the object's type or category. Given this distinction, foreground detection is ideally suited for monitoring applications at railway crossings. Its ability to detect and identify all stationary and moving outliers within the railway crossing area, regardless of their nature, provides a thorough situational overview. Consequently, this ensures a higher degree of safety by detecting potential hazards in real time.

4.2 Methodology

As shown in Figure 33, the proposed RC-SAFE model inputs the backgrounds along with a video frame and generates the corresponding foreground detection and tracking results. This method comprises three key parts: 1) a model for generating backgrounds, 2) a CNN designed for foreground segmentation, and 3) a weakly supervised training technique. These elements are explained in the following sections.

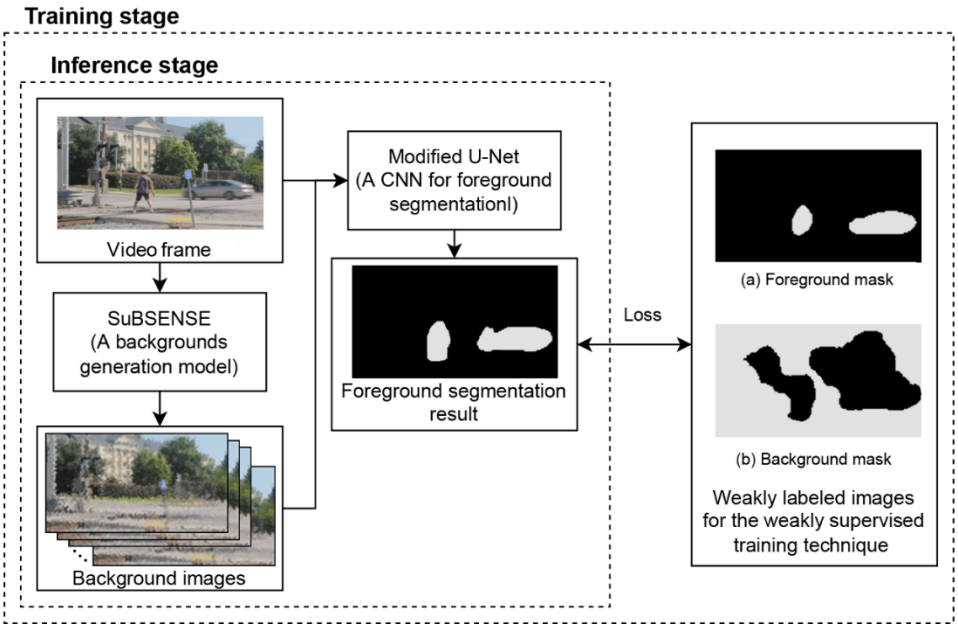


Figure 33. Data pipeline of the proposed RC-SAFE network

4.2.1 Background Generation by SuBSENSE Algorithm

The background model serves as a point of reference for video frames, aiding the CNN in discerning the discrepancies between the background images and the video frames. This model employs SuBSENSE for background generation, a renowned non-deep learning model for foreground segmentation and background modeling, which can deliver 50 background images (St. Charles, et al. 2014). The application of a multi-background scheme allows SuBSENSE to better handle background changes such as shifting light conditions, camera shake, and flowing water.

When a video frame is input into SuBSENSE, it iterates over each pixel to ascertain whether it belongs to the foreground or the background. SuBSENSE employs both the pixel's RGB value and Local Binary Similarity Pattern (LBSP) features to compute the discrepancies between the current pixel and its background library, which holds 50 background images. If at least two

background images identify a pixel as a background element, the pixel is classified as such; otherwise, it is deemed a foreground pixel.

SuBSENSE refreshes the background library through a conservative, stochastic, two-step approach. If a pixel is classified as a background pixel, a randomly selected background image has a 1/T chance of having its corresponding pixel replaced by the current frame pixel. Subsequently, one of the neighboring pixels also has a 1/T probability of being substituted by the same pixel, where T is a hyperparameter.

Figure 34 shows the first and last background images generated by SuBSENSE. Because of its stochastic background update scheme, the background images appear blurred. Since it is challenging to discern the differences between these background images with the naked eye, the team subtracted these two images from Figure 34d to accentuate their differences.

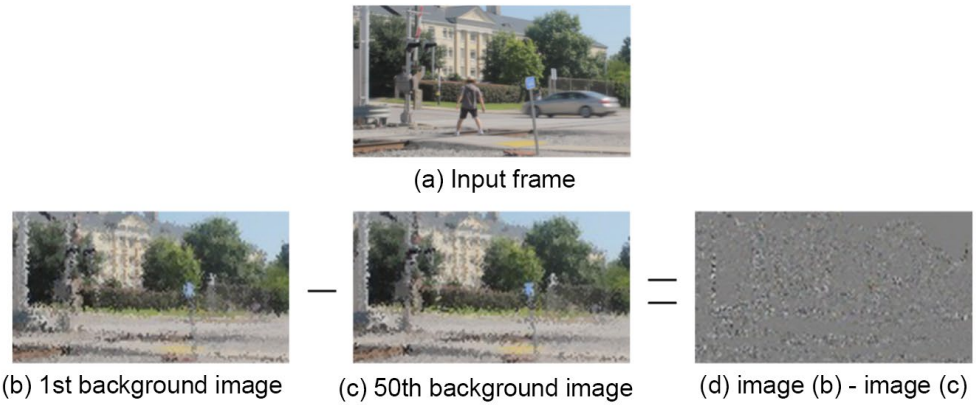


Figure 34. Example of the first and the last background images generated with SuBSENSE

4.2.2 Foreground Segmentation Network

Rather than focusing on object detection and segmentation, the RC-SAFE model centers on foreground segmentation. The team built RC-SAFE on the U-Net architecture (Ronneberger et al., 2015), which is a renowned, fundamental, and effective model used for image segmentation. The success of U-Net can largely be attributed to its large skip connections between the encoder and decoder. These connections help restore information lost during downsampling while upsampling, thereby enhancing segmentation accuracy.

As shown in Figure 35, the only deviation between RC-SAFE and the original U-Net is an additional input head. This element is tasked with maintaining a balance between the weights of the video frame and the background images. The video frame is an RGB image containing three channels, whereas the background consists of 50 RGB images, adding up to 150 channels. If the video frame and background images are merged as the network input, U-Net will place a disproportionately larger emphasis on the background images, making it challenging to identify any differences. The input head rectifies this by converting the video frame and background images into feature maps of identical size. The input head comprises two branches, each containing a single convolutional layer. The first branch is designated for the input frame, and the second for the 50 background images. Following the input head's processing, both the input frame and background images are transformed into a set of feature maps with 32 channels each. By interlinking these two sets, a 64-channel feature map is produced, which is then input into the U-Net structure.

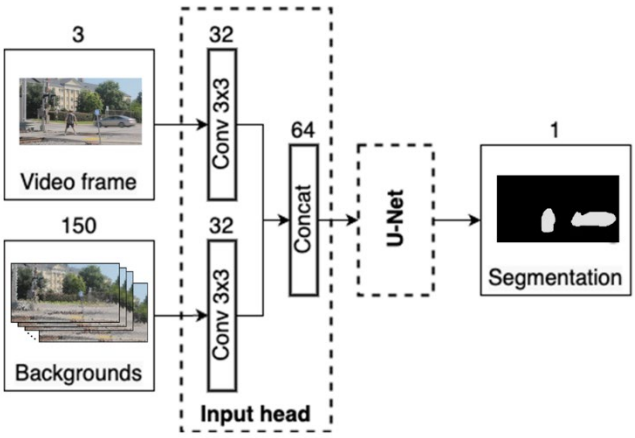


Figure 35. The U-Net-based network architecture of RC-SAFE

Figure 36 outlines the intricate network architecture of the proposed RC-SAFE model.

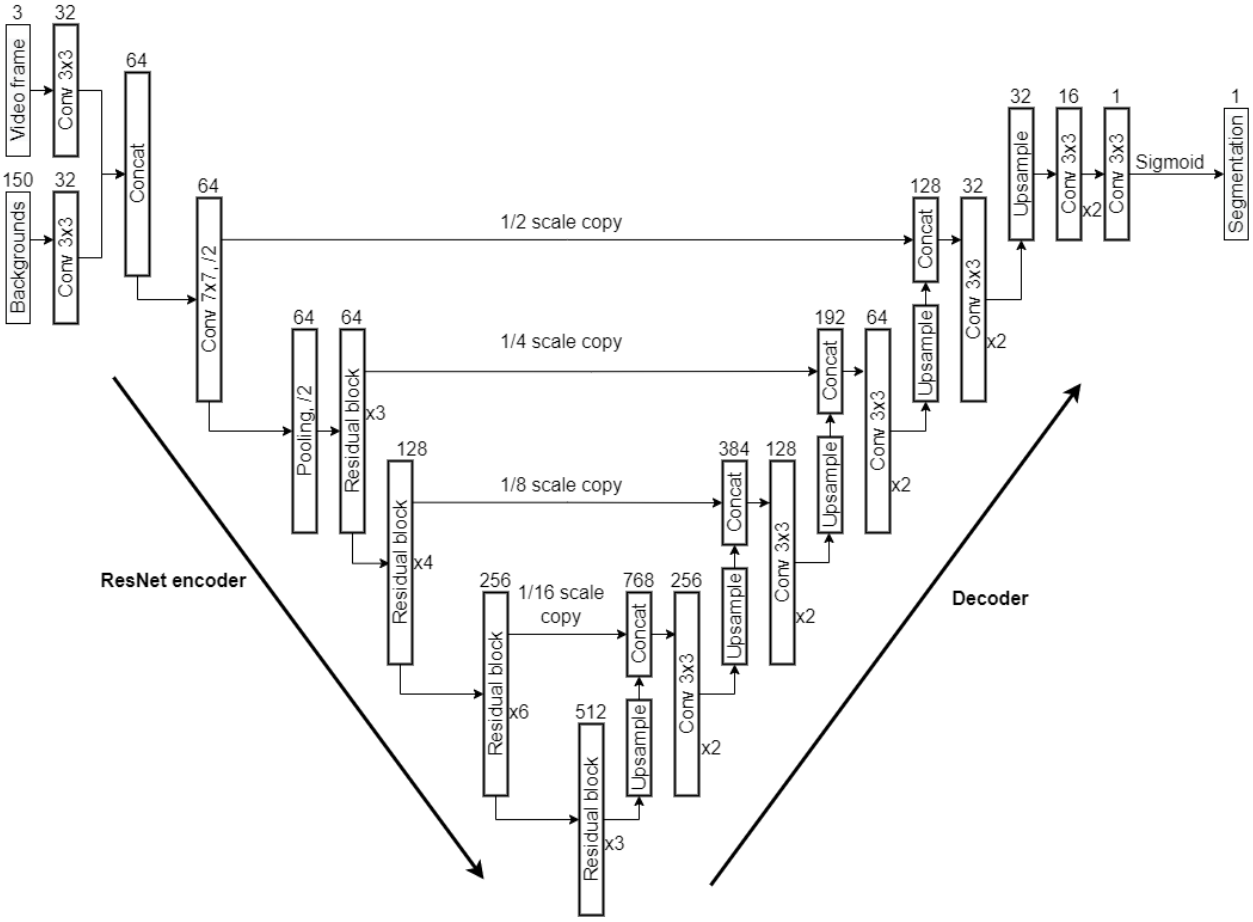


Figure 36. The detailed network architecture of RC-SAFE

ResNet-34 is employed as the encoder for the U-Net. ResNet (He et al. 2016) is a commonly used backbone in several CNN architectures such as U-Net, Feature Pyramid Network (FPN), and RetinaNet. In comparison to U-Net's original encoder (VGG-16), ResNet-34 is deeper yet comprises fewer parameters, surpassing VGG-16 in both accuracy and processing speed. A

critical component of ResNet is its double-layer skip connection, which addresses the vanishing gradient problem inherent in deep networks. During backpropagation, network parameters can readily obtain their gradients through the shortcut provided by the skip connections. Furthermore, as ResNet has smaller feature maps, it transmits fewer feature maps to the decoder at each scale, thereby enhancing the decoding speed of the U-Net.

ResNet initially employs a convolution layer with 7×7 kernels and a stride of two to downsample the input image by half. Subsequently, feature maps are downsized to a quarter of the original scale via a max-pooling layer. Following this, 19 residual blocks are deployed to amplify the network depth and nonlinearity. Specifically, at 1/4, 1/8, 1/16, and 1/32 scales, there are 3, 4, 6, and 3 residual blocks, respectively. Figure 37 provides a detailed view of the residual blocks at each scale.

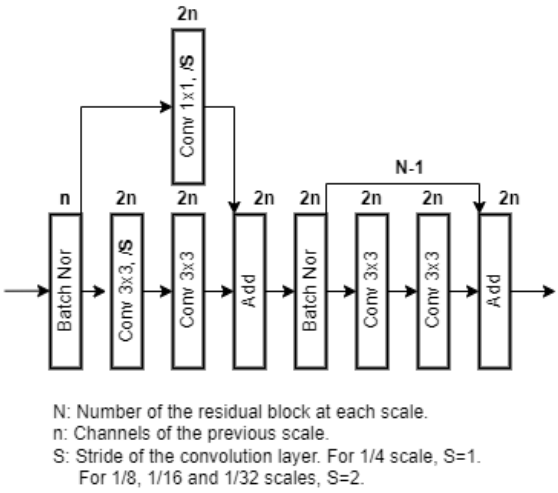


Figure 37. Details of the residual blocks at each scale

The U-Net decoder comprises five upsampling modules, with each one doubling the scale of the feature maps. The feature maps, downscaled to 1/32 by the ResNet, are eventually upscaled back to their original size. The first four upsampling modules follow the same procedure: they initially upsample the feature maps using the nearest interpolation, and then merge these upscaled feature maps with corresponding feature maps from the encoder. Subsequently, two convolution layers are deployed to extract information from the interlinked feature maps. The final upsampling module does not perform a linking operation.

The decoder ultimately produces a 1-channel segmentation result, which is activated by the sigmoid function. Within the segmentation results, foreground pixels are represented as 1, while background pixels are denoted as 0.

4.2.3 Weakly Supervised Training Technique

Currently, most CNN-based foreground segmentation methods rely on supervised learning. However, supervised learning implies that the entire training process must be aided by human attention, requiring a vast collection of accurately labeled images, which is labor-intensive. This research introduces a weakly supervised training technique for network training. With this proposed approach, the training of CNN no longer requires precisely and manually labeled images, but only weakly labeled ones. These weakly labeled images require significantly less effort for labeling and verification.

As illustrated in Figure 38, the weakly labeled images of a video frame consist of two images: the foreground mask and the background mask. The foreground mask includes the areas where the foreground pixels must appear (marked green), while the background mask outlines areas designated for the background pixels (marked red). Beyond the foreground and background masked areas lie the unlabeled pixels (marked yellow). CNN can classify these pixels by learning from the foreground and background masks.

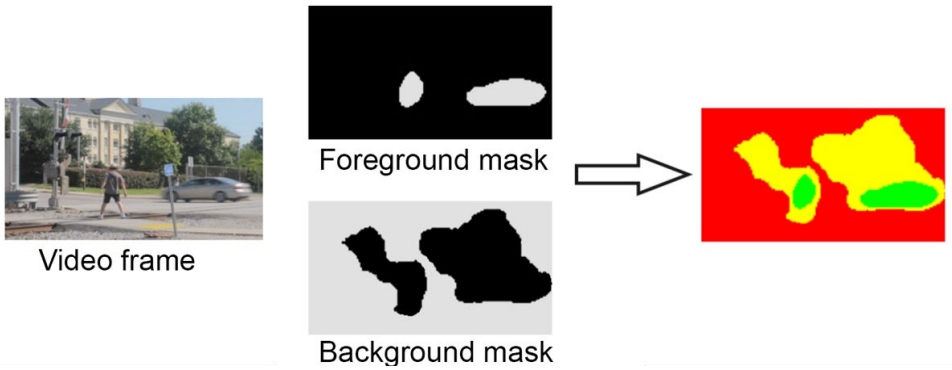


Figure 38. Foreground and background masks of the weakly supervised learning

The team used the SuBSENSE algorithm for weakly labeled image generation in this research. The sensitivity of the SuBSENSE was modified by adjusting the threshold of the pixel's RGB value R^0_{lbsp} to fit the specific segmentation task. Different thresholds were used for different training videos. The higher threshold was used for foreground mask generation, letting SuBSENSE detect the foreground pixels roughly and reduce the noisy pixels. In addition, some noticeable noises in the foreground mask were eliminated manually, if necessary. The lower threshold for background mask generation enabled the SuBSENSE algorithm to exclude the foreground pixels adequately and roughly detect background pixels.

The overall cost value has two items:

$$cost = cost_{fg} + cost_{bg} \tag{28}$$

where $cost_{fg}$ and $cost_{bg}$ are the cost values of the foreground and background masks, respectively. These two cost functions are modified from the Binary Cross Entropy (BCE):

$$BCE = -\frac{1}{N}\sum_{I=1}^{N}(y_t \log(y_i) + (1 - y_t)\log(1 - y_i)) * \beta \tag{29}$$

where y_i is the output probability ranging from [0, 1], Boolean y_t is the target value, and β is a network gradient magnification factor. Regularly, $\beta = 1$ and the network gradient will not be modified. In this study, the team used $\beta = 0$ and $\beta = 1$ to disable and enable network gradient in pixelwise to restrict the observation of the network.

With the weakly labeled images, the network costs of the foreground and background masks can be computed by Equations 30 and 31:

$$cost_{fg} = -\frac{1}{N}\sum_{I=1}^{N}\log(y_i) * fm_i \tag{30}$$

$$cost_{bg} = -\frac{1}{N}\sum_{I=1}^{N}\log(1 - y_i) * bm_i \tag{31}$$

The target value of foreground pixels is 1, whereas the target value of background pixels is 0. Booleans fm_i and bm_i are the pixels in the foreground and background masks, and they behave as the β value of Equation 29. The fm_i and bm_i will hide the network gradient if they are equal to 0 and keep the network gradient if they are equal to 1. Both fm_i and bm_i will hide the network gradient for the yellow region in Figure 38, and the CNN can classify these unlabeled pixels by learning from the foreground and background masks.

4.3 Results and Discussion

To conduct model testing and validation, the team used the publicly accessible CDnet 2014 dataset and a railroad crossing dataset curated specifically for this research (Wang et al., 2014). The CDnet 2014 dataset is currently the largest foreground segmentation dataset available, while the railroad crossing dataset contains three surveillance videos collected from different crossings in Columbia, South Carolina. Due to the limited number of videos in the railroad crossing dataset, it is unsuitable for deep network training. Therefore, the proposed RC-SAFE model was trained and validated using the CDnet 2014 dataset, while the railroad crossing dataset serves as a practical case study for performance testing and demonstration.

4.3.1 Evaluation Metrics

To assess the performance of RC-SAFE, the team used common model evaluation parameters: precision, recall, and F-Measure. Precision measures the percentage of detected foreground pixels that have been accurately classified, and it is used to evaluate the network's resistance to noise, as demonstrated in Equation 31. Recall represents the percentage of true foreground pixels that are correctly classified, serving as an estimate of the network's foreground recognition rate, as illustrated in Equation 32. The F-Measure is the harmonic mean of precision and recall; a high F-Measure indicates both high precision and recall, as depicted in Equation 33:

$$Precision = \frac{TP}{TP+FP} \tag{31}$$

$$Recall = \frac{TP}{TP+FN} \tag{32}$$

$$\mathrm{F-Measure} = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{33}$$

where TP, FP, and FN are the numbers of true-positive, false-positive, and false-negative pixels, respectively.

4.3.2 Experiments on the CDNet 2014 Dataset

The CDnet 2014 dataset includes 53 videos across 11 categories, representing various challenging situations such as adverse weather, dynamic backgrounds, and nighttime scenarios (Wang, et al., 2014), making it an ideal dataset for algorithm validation. CDNet 2014 comprises 159,278 frames, with 91,435 frames labeled. As the dataset is not specifically designed for deep learning studies, it does not differentiate between training and validation datasets. Some previous studies have divided the labeled frames into training and validation datasets, overlooking performance evaluation on the training frames. To thoroughly evaluate RC-SAFE across all scenarios, the team employed a cross-validation approach. In this method, one video is chosen

for validation, while the remaining videos serve for network training. The network is trained for 100 epochs, with validation occurring every five epochs. The network weight is saved if a higher F-measure is achieved for the current validation video. However, the videos in the Pan-Tilt-Zoom (PTZ) category may be excluded from each cross-validation as the camera's spinning feature in PTZ is irrelevant to railroad crossing monitoring tasks.

The CDnet 2014 also provides pixel-level ground-truth labels, allowing for the proposed RC-SAFE network to be trained via supervised learning for comparison purposes, which allows a fair comparison between weakly supervised learning and supervised learning. The primary difference between the two learning methods lies in the training data: supervised learning uses manually labeled images, while weakly supervised learning relies on weakly labeled images.

Table 3 presents the precision, recall, and F-measure of both weakly supervised and supervised networks, derived from different videos in the CDnet 2014 dataset. The supervised network only surpassed the weakly supervised network in categories of adverse weather and intermittent object motion videos, as it yielded an F-measure that is 2.52 and 0.97 percent higher, respectively. Both networks exhibited identical performance for baseline videos, achieving an F-measure of 95.67 percent. In 8 out of 10 categories, the weakly supervised network attained a higher F-measure. Overall, the weakly supervised network achieved an overall F-measure of 82.03 percent (excluding PTZ) and 76.01 percent (including PTZ), which are 1.57 and 1.48 percent higher than the supervised network, respectively. The proposed weakly supervised learning technique significantly reduces training costs while improving performance.

Table 3. The comparison of supervised learning and weakly supervised learning among CDnet2014 dataset

Video categories	Supervised learning			**Weakly supervised learning** (RC-SAFE)		
	Precision (%)	Recall (%)	F-measure (%)	Precision (%)	Recall (%)	F-measure (%)
Baseline	93.39	98.06	**95.67**	94.01	97.38	**95.67**
Bad weather	82.45	81.08	**81.76**	85.02	74.19	79.24
Dynamic background	68.02	94.82	79.21	75.64	89.99	**82.19**
Camera jitter	68.05	94.32	79.06	73.70	88.57	**80.45**
Intermittent object motion	75.24	93.36	**83.32**	74.45	92.14	82.35
Low framerate	64.60	83.79	72.96	77.42	87.48	**82.14**
Night videos	52.75	80.45	63.72	55.60	76.86	**64.53**
shadow	88.59	96.72	92.48	91.78	95.89	**93.79**
thermal	81.56	81.45	81.50	80.48	84.57	**82.47**
turbulence	66.31	86.01	74.89	75.08	80.03	**77.47**
Pan-Tilt-Zoom (PTZ)	19.29	86.39	31.54	19.34	94.01	**32.08**
Overall (exclude PTZ)	74.10	89.01	80.46	78.32	86.71	**82.03**
Overall (include PTZ)	69.11	88.77	76.01	72.96	87.37	**77.49**

Figure 39 shows some example detection results from the weakly supervised RC-SAFE network, the supervised network, and SuBSENSE. Both the weakly supervised and supervised networks exhibited a higher recognition rate for foreground pixels when compared to SuBSENSE. However, the weakly supervised network generated less noise than the supervised network.

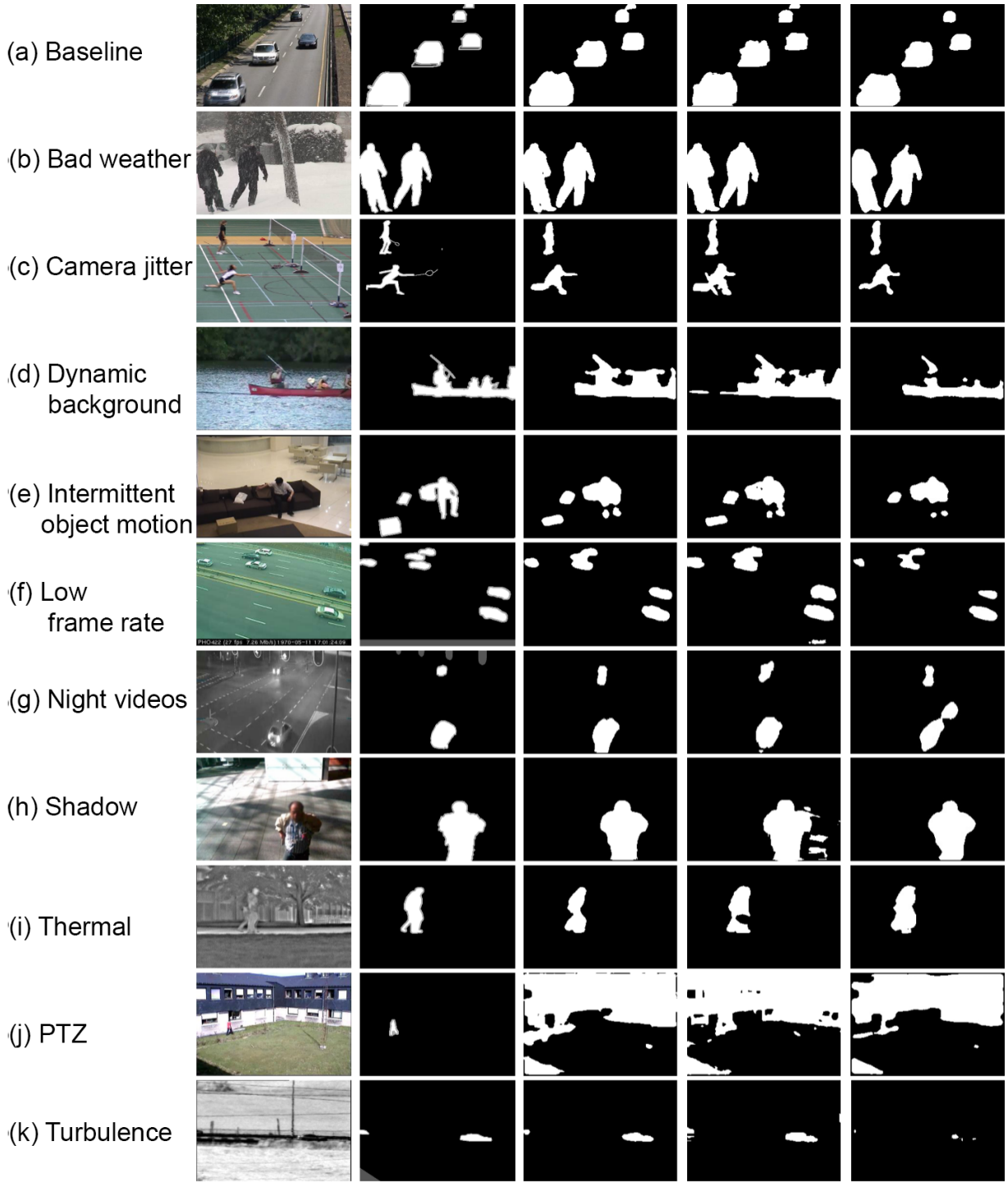


Figure 39. Example detection results on CDnet 2014 dataset (from left to right: input frame, ground-truth image, weakly supervised network (RC-SAFE), supervised network, SuBSENSE)

RC-SAFE was subsequently compared with several state-of-the-art algorithms. Table 4 shows the comparison of F-measure values derived from these methods. It's important to note that both RC-SAFE and DeepBS (10) are background reference-based models, but DeepBS employs supervised learning and employs a different scheme for generating background reference.

SimpleBSC (Minematsu et al. 2019) is another model using weakly supervised learning. SuBSENSE (St. Charles et al. 2014), PAWCS (St. Charles et al. 2015), and PBAS (Hofmann et al. 2012) are traditional image processing methods. All these methods failed in the PTZ category, with their F-measure values dropping below 50 percent. In comparison to these methods, RC-SAFE exhibited more robust performance. It scored the lowest F-measure, 64.53 percent, in the category of nighttime videos, which, nonetheless, was still the highest F-measure for this category. Ultimately, RC-SAFE achieved the highest overall F-measure, scoring 82.03 percent (excluding PTZ) and 76.01 percent (including PTZ). This surpasses the performance of SimpleBSC, DeepBS, SuBSENSE, PAWCS, and PBAS.

Table 4. F-measure comparison of different foreground algorithms among CDnet2014 dataset

Video categories	F-measure					
	RC-SAFE	SimpleBSC	DeepBS	SuBSENSE	PAWCS	PBAS
Baseline	95.67	**96.90**	95.80	95.03	93.97	92.42
Bad weather	79.24	72.20	83.01	**86.19**	81.52	76.73
Dynamic background	82.19	82.30	87.61	81.77	**89.38**	68.29
Camera jitter	80.45	60.50	**89.90**	81.52	81.37	72.20
Intermittent object motion	82.35		60.98	65.69	77.64	57.45
Low framerate	**82.14**		60.02	64.45	65.88	59.14
Night videos	**64.53**	37.70	58.35	55.99	41.52	43.87
shadow	**93.79**	56.00	93.04	89.86	89.13	81.43
thermal	82.47	66.10	75.83	81.71	**83.24**	75.56
turbulence	77.47	57.80	**84.55**	77.92	64.50	63.49
Pan-Tilt-Zoom (PTZ)	32.08		31.33	34.76	**46.15**	10.63
Overall (exclude PTZ)	**82.03**	66.20	78.91	77.01	76.82	69.06
Overall (include PTZ)	**77.49**		74.58	74.08	74.03	63.75

4.3.3 Experiments on the Railroad Crossing Dataset

Following the training and validation of the weakly supervised network using the CDnet 2014 dataset, RC-SAFE demonstrated promising results in general foreground segmentation. More importantly, RC-SAFE is expected to exhibit robust performance in the context of railroad crossing monitoring. Therefore, a railroad crossing dataset was established to assess the network. To establish the superiority of the proposed RC-SAFE, which is based on foreground segmentation, over object detection-based networks of railroad crossing monitoring, the renowned Mask-RCNN (He et al. 2017) was selected for comparison due to its robust performance. The Mask-RCNN was trained using the COCO dataset, encompassing 80 classes.

Table 5 outlines a performance comparison between RC-SAFE and Mask-RCNN across the railroad crossing dataset. Figure 40 shows sample detection results from both RC-SAFE and Mask-RCNN. Mask-RCNN only managed to detect commonplace objects like pedestrians and cars. Unfortunately, it failed to detect objects like pipes, a squatting person, and a wheelbarrow, leading to false-negative errors. Furthermore, Mask-RCNN tended to detect many background objects, causing substantial false-positive errors. Conversely, RC-SAFE successfully identified all objects not part of the railroad crossing as foreground objects, thus achieving high precision

and recall. Therefore, RC-SAFE not only outperformed Mask-RCNN in the railroad crossing monitoring task but also significantly reduced the effort required for model training.

Table 5 The comparison of RC-SAFE and Mask-RCNN railroad crossing dataset

Video	Mask-RCNN			RC-SAFE		
	Precision (%)	Recall (%)	F-measure (%)	Precision (%)	Recall (%)	F-measure (%)
Video-1	93.82	84.71	89.03	**99.44**	**97.85**	**98.64**
Video-2	57.84	**96.77**	72.40	**99.50**	91.94	**95.57**
Video-3	77.19	76.02	76.60	**99.97**	**95.48**	**97.68**
Overall	76.28	85.83	79.34	**99.64**	**95.09**	**97.30**

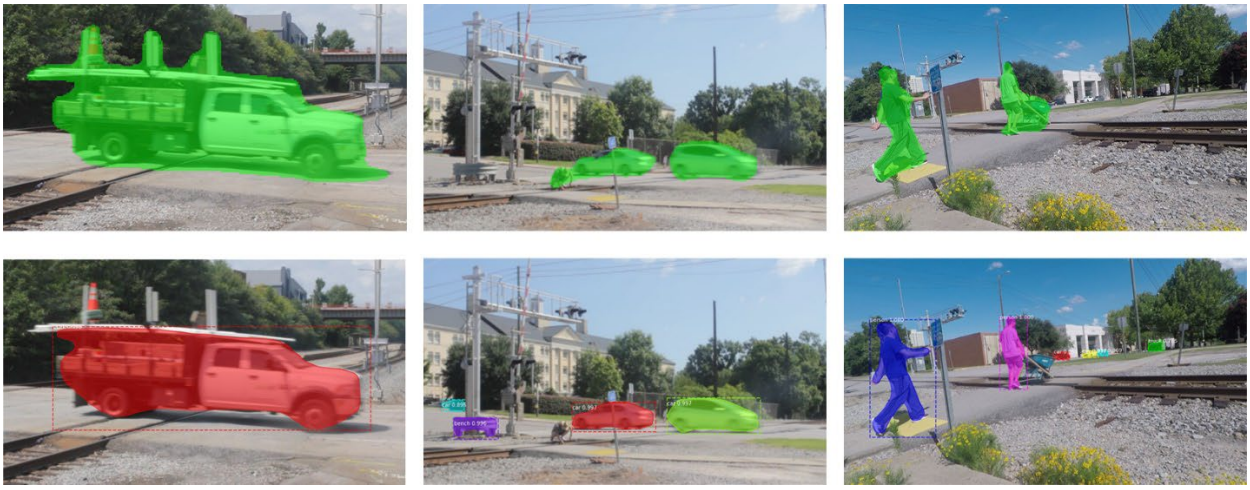


Figure 40. Example detection results on the railroad crossing dataset (from top to bottom: RC-SAFE, Mask-RCNN)

4.3.4 Field Testing

The online testing platform employed for these experiments aligns with the specifications detailed in Section 3. The setup included a high-performance computing NVIDIA Jetson AGX Orin and a Logitech HD Pro Webcam C920 mounted on a tripod for stability, as shown in Figure 28.

For computational efficiency, the team used the TensorRT framework for inferring the modified U-Net. To further enhance the system's performance and optimize hardware resource usage, the team designed a multi-processor pipeline to operate the RC-SAFE system. This arrangement enabled concurrent running of SuBSENSE, a background subtraction algorithm, and the modified U-Net on separate processors. Consequently, the RC-SAFE system's inference speed depends on the slowest component, adhering to the principle of concurrency in a multi-processor environment.

The team found that SuBSENSE and the modified U-Net operated at 25.04 FPS and 44.35 FPS, respectively. Nonetheless, given SuBSENSE's lower speed at 25.04 FPS, this value becomes the overall speed for the RC-SAFE system. Thus, the system's performance is essentially governed by its slowest component.

Field testing was conducted at a grade crossing situated at 300 Main Street, Columbia, SC, as illustrated in Figure 41. The implementation of the RC-SAFE system at this location yielded promising results. As evidenced in Figure 42, the RC-SAFE system effectively detected the outlier within the rail crossing. This successful detection underscores the practical effectiveness of the system in real-world scenarios.

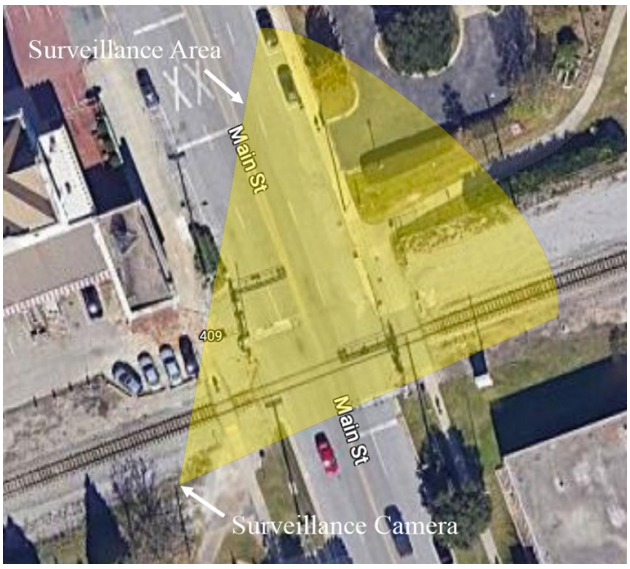


Figure 41. The grade crossing at 300 Main Street, Columbia, SC

Figure 42. Field testing illustration of RC-SAFE

5. Conclusion

Capitalizing on the progression of computer vision and AI, i-ASAP offers a comprehensive and advanced solution to bolster safety measures at railroad crossings. This system delivers real-time, detailed information concerning trespassing pedestrians, vehicles, and other unexpected obstructions - predominant contributors to accidents and fatalities at these junctions. The central results of this research are as follows:

1. Researchers developed an intelligent anomaly detection system that specifically focuses on pedestrian behaviors at grade crossing scenarios. This system leverages an open-source pose estimation model to numerically represent behaviors as skeleton trajectories and employs a message-passing recurrent neural network to understand these trajectory patterns. Experimental evaluation conducted on a custom dataset substantiated the feasibility and effectiveness of the proposed system in the detection of abnormal behaviors.
2. The team proposed a novel anomaly detection system, which integrates a GAN-enabled feature learning model to augment detection performance, replacing the previous recurrent neural network. The GAN module employs a secondary neural network (i.e., the discriminator) to enhance the performance of the feature learning module (i.e., the generator). This innovative anomaly detection system was trained in a semi-supervised manner, with training exclusively involving patterns of normal behaviors. As a result, anomalies can be pinpointed as outliers of the model, by observing anomaly scores produced by the discriminator. Experiments conducted on the custom dataset and real-time testing scenes substantiated the effectiveness and deployability of the proposed system to edge computing devices.
3. Researchers proposed a weakly supervised foreground segmentation network, RC-SAFE, for broader detection purposes. This network can accurately and effectively detect and track both stationary and moving objects within the railroad crossing area, outperforming object detection-based models such as Mask-RCNN.
4. Significantly, in this research the team employed a weakly supervised learning technique to train RC-SAFE, leading to substantial savings in annotation costs. Experimental results further endorse this methodology, demonstrating that RC-SAFE, when trained using weakly supervised learning, surpasses supervised networks and several other forefront foreground segmentation methodologies.

The development and implementation of the i-ASAP system highlights the promising prospects of integrating AI and computer vision technologies to improve railway crossing safety. The system's ability to process data in real-time and identify potential threats serves as an invaluable tool for railway operators and regulatory bodies. By delivering essential information for timely decision-making, i-ASAP holds significant potential to mitigate accidents, injuries, and fatalities associated with railway crossings.

6. References

Babaee, M., Dinh, D. T., and Rigoll, G. (2018). A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition, 76*, 635-649.

Bera, A., & Manocha, D. (2018). Interactive surveillance technologies for dense crowds. *arXiv preprint arXiv:1810.03965*.

Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).

Chang, Y., Tu, Z., Xie, W., Luo, B., Zhang, S., Sui, H., & Yuan, J. (2022). Video anomaly detection with spatio-temporal dissociation. *Pattern Recognition, 122*, 108213.

Chen, Y., Rao, M., Feng, K., & Zuo, M. J. (2022). Physics-informed LSTM hyperparameters selection for gearbox fault detection. *Mechanical Systems and Signal Processing, 171*, 108907.

Chong, Y. S., & Tay, Y. H. (2017). Abnormal event detection in videos using spatiotemporal autoencoder. *Advances in Neural Networks-ISNN 2017: 14th International Symposium* (pp. 189-196).

Dong, F., Zhang, Y., & Nie, X. (2020). Dual discriminator generative adversarial network for video anomaly detection. *IEEE Access, 8*, 88170-88176.

Du, Y., Fu, Y., & Wang, L. (2016). Representation learning of temporal dynamics for skeleton-based action recognition. *IEEE Transactions on Image Processing, 25*(7), 3010-3022.

Fang, H.-S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., . . . Lu, C. (2022). Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

FRA. (2018). National strategy to prevent trespassing on railroad property.

FRA. (2019). Highway-rail grade crossings overview.

FRA. (2021). Highway/rail grade crossing incidents.

Gao, Y., & Glowacka, D. (2016). Deep gate recurrent neural network. *Asian conference on machine learning* (pp. 350-365).

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems, 28*(10), 2222-2232.

Guan, L., Jia, L., Xie, Z., & Yin, C. (2022). A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved yolo-tiny network. *IEEE Transactions on Instrumentation and Measurement, 71*, 1-16.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2255-2264).

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hofmann, M., Tiefenbacher, P., & Rigoll, G. (2012). Background segmentation with feedback: The pixel-based adaptive segmenter. *2012 IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 38-43).

Hou, B., Liu, Y., Ling, N., Liu, L., & Ren, Y. (2021). A fast lightweight 3D separable convolutional neural network with multi-input multi-output for moving object detection. *IEEE Access, 9*, 148433-148448.

Le, V.-T., & Kim, Y.-G. (2023). Attention-based residual autoencoder for video anomaly detection. *Applied Intelligence, 53*(3), 3240-3254.

Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.-S., & Lu, C. (2019). Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10863-10872).

Lifesaver, O. (2023, 6). *Collisions and Casualties by Year*. Retrieved from Operation Lifesaver:

Lim, L. A., & Keles, H. Y. (2020). Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications, 23*, 1369-1380.

Lin, C., Yan, B., & Tan, W. (2018). Foreground detection in surveillance video with fully convolutional semantic network. *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 4118-4122).

Liu, W., Luo, W., Lian, D., & Gao, S. (2018). Future frame prediction for anomaly detection – a new baseline. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6536-6545).

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

Minematsu, T., Shimada, A., & Taniguchi, R.-i. (2019). Simple background subtraction constraint for weakly supervised background subtraction network. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-8).

Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., & Venkatesh, S. (2019). Learning regularity in skeleton trajectories for anomaly detection in videos. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11996-12004).

Nguyen, T.-N., & Meunier, J. (2019). Anomaly detection in video sequence with appearance-motion correspondence. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1273-1283).

Ogden, B. D., & Cooper, C. (2019). *Highway-rail crossing handbook.* United States, Federal Highway Administration.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. (2021). Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR), 54*(2), 1-38.

Pang, G., Yan, C., Shen, C., Hengel, A. v., & Bai, X. (2020). Self-trained deep ordinal regression for end-to-end video anomaly detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12173-12182).

Pang, W., He, Q., & Li, Y. (2022). Predicting skeleton trajectories using a Skeleton-Transformer for video anomaly detection. *Multimedia Systems, 28*(4), 1481-1494.

Park, H., Noh, J., & Ham, B. (2020). Learning memory-guided normality for anomaly detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14372-14381).

Perera, P., Nallapati, R., & Xiang, B. (2019). Ocgan: One-class novelty detection using GANS with constrained latent representations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2898-2906).

Piergiovanni, A., & Ryoo, M. S. (2019). Representation flow for action recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9945-9953).

Rahmon, G., Bunyak, F., Seetharaman, G., & Palaniappan, K. (2021). Motion U-Net: multi-cue encoder-decoder network for motion segmentation. *2020 IEEE 25th International Conference on Pattern Recognition (ICPR)* (pp. 8125-8132).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention--MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer.

ShanghaiTech campus dataset (Anomaly detection) (n.d.). Retrieved from SVIPLAB.

Siam, M., Valipour, S., Jagersand, M., Ray, N., & Yogamani, S. (2017). Convolutional gated recurrent networks for video semantic segmentation in automated driving. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1--7).

Sikora, P., Malina, L., Kiac, M., Martinasek, Z., Riha, K., Prinosil, J., . . . Srivastava, G. (2020). Artificial intelligence-based surveillance system for railway crossing traffic. *IEEE Sensors Journal, 21*, 15515-15526.

St-Charles, P.-L., Bilodeau, G.-A., & Bergevin, R. (2014). SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing, 24*, 359--373.

St-Charles, P.-L., Bilodeau, G.-A., & Bergevin, R. (2015). A self-adjusting approach to change detection based on background word consensus. *2015 IEEE Winter Conference on Applications of Computer Vision* (pp. 990-997).

UCSD Anomaly detection dataset (n.d.). Retrieved from UCSD.

Vijayan, M., Raguraman, P., & Mohan, R. (2021). A fully residual convolutional neural network for background subtraction. *Pattern Recognition Letters, 146*, 63-69.

Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., & Lee, H. (2017). Learning to generate long-term future via hierarchical prediction. *International conference on machine learning* (pp. 3560-3569).

Wang, Y., & Yu, P. (2021). A fast intrusion detection method for high-speed railway clearance based on low-cost embedded GPUs. *Sensors, 21*, 7279.

Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014). CDnet 2014: An expanded change detection benchmark dataset. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 387-394).

Xiu, Y., Li, J., Wang, H., Fang, Y., & Lu, C. (2018). Pose Flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*.

Xu, C., Liu, H., Li, T., Zhang, Y., Li, T., & Li, G. (2022). Cascaded feature-mask fusion for foreground segmentation. *IEEE Open Journal of Intelligent Transportation Systems, 3*, 340-350.

Yang, L., Li, J., Luo, Y., Zhao, Y., Cheng, H., & Li, J. (2017). Deep background modeling using fully convolutional network. *IEEE Transactions on Intelligent Transportation Systems, 19*, 254-262.

Zaman, A., Ren, B., & Liu, X. (2019). Artificial intelligence-aided automated detection of railroad trespassing. *Transportation research record, 2673*, 25--37.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., . . . Chawla, N. V. (2019). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *Proceedings of the AAAI conference on artificial intelligence, 33*(01), 1409-1416.

Zhang, Z., Zaman, A., Xu, J., & Liu, X. (2022). Artificial intelligence-aided railroad trespassing detection and data analytics: Methodology and a case study. *Accident Analysis and Prevention, 168*, 106594.

Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., & Hua, X.-S. (2017). Spatio-temporal autoencoder for video anomaly detection. *Proceedings of the 25th ACM international conference on Multimedia* (pp. 1933-1941).

Abbreviations and Acronyms

ACRONYM	DEFINITION
AE	Autoencoder
AI	Artificial Intelligence
AUC	Area Under the Curve
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
FRA	Federal Railroad Administration
FPR	False Positive Rate
FPS	Frames Per Second
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MA	Moving Average
MNAD	Memory-Guided Normality for Anomaly Detection
MPED-RNN	Message-Passing Encoder-Decoder Recurrent Neural Network
NMS	Non-Maximum Suppression
PAF	Part Affinity Field
PTZ	Pan-Tilt-Zoom
RC-SAFE	Railroad Crossing Surveillance and Foreground Extraction Network
ROC	Receiver Operating Characteristics
ROI	Region of Interest
ROW	Right of Way
SSD	Single Shot MultiBox Detector
SSTN	Symmetric Spatial Transformer Network
ST-AE	Spatial-Temporal Autoencoder Network
TPR	True Positive Rate