

C2 SMART

CONNECTED CITIES WITH
SMART TRANSPORTATION 

A USDOT University Transportation Center

New York University

Rutgers University

University of Washington

The University of Texas at El Paso

City College of New York

EXPLORING COST-EFFECTIVE COMPUTER VISION SOLUTIONS FOR SMART TRANSPORTATION SYSTEMS

May 2023



TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Exploring Cost-effective Computer Vision Solutions for Smart Transportation Systems		5. Report Date May 2023	
		6. Performing Organization Code:	
7. Author(s) Jingqin Gao, Chuan Xu, Daniel Zhang, Fan Zuo, Liu Yang, Omar Hammami		8. Performing Organization Report No.	
9. Performing Organization Name and Address Connected Cities for Smart Mobility towards Accessible and Resilient Transportation Center (C2SMART), 6 Metrotech Center, 4th Floor, NYU Tandon School of Engineering, Brooklyn, NY, 11201, United States		10. Work Unit No.	
		11. Contract or Grant No. 69A3551747119	
12. Sponsoring Agency Name and Address Office of Research, Development, and Technology Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296		13. Type of Report and Period Final report, 3/1/22-5/31/23	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract The key accomplishments of the project are summarized as follows. The team conducted a literature review on computer vision for smart cities with a focus on transportation and summarized a resource list that lists publicly available traffic camera systems in the U.S. The team also established an automatic pipeline for data acquisition and developed a web-based tool that integrated the computer vision algorithms for the two selected applications. The urban work zone application (WorkZoneX) leverages 900+ traffic cameras in NYC and provides real-time urban work zone identification, active work zone with workers detection, work zone size estimation, and traffic condition around the work zones. WorkZoneX achieved an average mAP of 74.1% across all work zone classes, an accuracy of 98.4% for scene identification, and an accuracy of up to 89.52% for size estimation. The safety risk index map application (SAFEExMAP) provides a risk indicator scoring system with a map interface that leverages near-miss data gathered from in-vehicle cameras via computer vision. A positive spatial correlation was found between near misses and crashes for the study area. Both applications were optimized for web-based access and prototyped for real-world deployment. A cost estimation of deploying the two applications was provided. This project stands to facilitate the adoption of computer vision in smart cities, potentially positively impacting transportation planning and operations by providing cost-effective solutions to the industry.			
17. Key Words		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161. http://www.ntis.gov	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 104	22. Price

Exploring Cost-effective Computer Vision Solutions for Smart Transportation Systems

PI: Jingqin Gao
New York University
0000-0002-1718-2432

Co-PI: Kaan Ozbay
New York University
0000-0001-7909-6532

Fan Zuo
New York University
0000-0002-6761-2808

Chuan Xu
New York University
0000-0001-5727-4561

Liu Yang
New York University
0009-0009-7929-3582

Daniel Zhang
New York University
0009-0007-3372-4565

Omar Hammami
New York University
0000-0002-4727-408X

C2SMART Center is a USDOT Tier 1 University Transportation Center taking on some of today's most pressing urban mobility challenges. Some of the areas C2SMART focuses on include:



Urban Mobility and Connected Citizens



Urban Analytics for Smart Cities



Resilient, Smart, & Secure Infrastructure

Disruptive Technologies and their impacts on transportation systems. Our aim is to develop innovative solutions to accelerate technology transfer from the research phase to the real world.

Unconventional Big Data Applications from field tests and non-traditional sensing technologies for decision-makers to address a wide range of urban mobility problems with the best information available.

Impactful Engagement overcoming institutional barriers to innovation to hear and meet the needs of city and state stakeholders, including government agencies, policy makers, the private sector, non-profit organizations, and entrepreneurs.

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation’s University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

The authors would like to thank the great support and advice provided by the advisory board, including Asheque Rahman (NYC DDC), Terri Matthews (NYC DDC), Howard Jiang (NYC DDC), Seth Berman (NYC DOT), Justin Romeo (NYC DOT), Rob Viola (NYC DOT), Maddalena Romano (NYC DOT), Zamir Alam (NYC DOT), Dan Wan (NYC DOT), Paul Rothman (NYC OTI), Ruoran Lin (NYC DCP), and Mark Davis (Mobileye/Vexcel). The authors also would like to thank Daniel Zhang, Angela Zhang, Lukelo Luoga for annotating and cleaning the training data. In addition to the funding support from C2SMART, some of the researchers were supported by NYU Tandon School of Engineering’s Summer Undergraduate Research Programs. Traffic camera data used in this project was extracted from <https://webcams.nycmc.org/> and was originally compiled by the New York City Department of Transportation (NYCDOT).

Executive Summary

Computer vision can turn existing infrastructure into smart sensors in a myriad of ways, and new applications are being continuously developed. While state-of-the-art computer vision techniques are well-documented in literature, this technology has not widely applied to existing transportation infrastructures for routine operations yet. This project aims to develop a deep learning-based data acquisition and analytics approach using vision-based sensors (i.e., cameras) to understand cities with machine eyes. The goal is to demonstrate the cost-effectiveness of computer vision technology to generate a new stream of mobility and safety data, thereby supporting planning and operational strategies. This approach will leverage both existing transportation infrastructure and emerging probe data from connected and automated vehicles (CAVs).

The research team first assessed the maturity of various smart city applications using computer vision and object detection (e.g., pedestrian detection, work zone identification, curb lane usage, connected and automated vehicles [CAVs]). Furthermore, needs assessment from multiple local agencies, including New York City (NYC) Department of Transportation (DOT), NYC Department of Construction and Design (DDC), NYC Department of City Planning (DCP), NYC Office of Technology and Innovation (OTI), was obtained through a voluntary advisory board. Based on the feedback received, two applications: 1) urban work zone detection, and 2) safety risk index view map, were developed.

The key accomplishments of the project are summarized as follows. The team conducted a literature review on computer vision for smart cities with a focus on transportation and summarized a resource list that lists publicly available traffic camera systems in the U.S. The team also established an automatic pipeline for data acquisition and developed a web-based tool that integrated the computer vision algorithms for the two selected applications. The urban work zone application (WorkZoneX) leverages 900+ traffic cameras in NYC and provides real-time urban work zone identification, active work zone with workers detection, work zone size estimation, and traffic condition around the work zones. WorkZoneX achieved an average mAP of 74.1% across all work zone classes, an accuracy of 98.4% for scene identification, and an accuracy of up to 89.52% for size estimation. The safety risk index map application (SAFEExMAP) provides a risk indicator scoring system with a map interface that leverages near-miss data gathered from in-vehicle cameras via computer vision. A positive spatial correlation was found between near misses and crashes for the study area. Both applications were optimized for web-based access and prototyped for real-world deployment. A cost estimation of deploying the two applications was provided. This project stands to facilitate the adoption of computer vision in smart cities, potentially positively impacting transportation planning and operations by providing cost-effective solutions to the industry.

Table of Contents

Executive Summary	iv
Section 1 Introduction	1
1.1 Background	1
1.2 Study Objectives	4
1.3 Collaborative Approach	5
Section 2 Literature Review	6
2.1 Publicly Available Traffic Camera Systems in the U.S.	6
2.2 Smart Transportation Applications Using Computer Vision	9
Section 3 Urban Work Zone Detection	21
3.1 Framework for Urban Work Zone Detection and Sizing	22
3.2 Experiment and Results Analysis	32
3.3 Limitations	37
3.4 Conclusion	38
Section 4 Safety Risk Index Map	39
4.1 Research and Development Framework	39
4.2 Data Preparation	41
4.3 Correlation between crash data and near misses	49
4.4 Safety Risk Index Development	55
4.5 Safety Analysis Results and Discussion	59
4.6 Injury and Fatal Crash related SSRI	62
4.7 VRU SSRI	65
4.8 Limitation and Conclusion	65
Section 5 Web-based Applications	66
5.1 Web Application Architecture	66
5.2 Urban Work Zone Web-Based Application	71
5.3 Safety View Map Application	78
5.4 Security	84
5.5 Cost Estimation	85
5.6 Summary and Lessons Learned	86
Section 6 Advisory Board	87
Section 7 Outreach and Technical Transfer	91
Section 8 Conclusion and Discussion	92
8.1 Research Conclusion	92
8.2 Discussion and Future Work	93
References	95
Appendix	100
A.1 Correlation between VRU crash data and VRU near misses	100
A.2 VRU SSRI	102

List of Figures

Figure 1. Examples of Smart City Applications Using Computer Vision	1
Figure 2. Computer Vision Applications Using Traffic Cameras (Source: C2SMART)	2
Figure 3. Computer Vision Applications Using Probe Vehicle/CAV-based Cameras (Source: Mobileye)	3
Figure 4. Collaborative approach.....	5
Figure 5. Spatial distribution and scale of the selected camera systems in the U.S.	7
Figure 6. Timeline of selected pedestrian detection technology deployments (Source: C2SMART & ITSJPO)	13
Figure 7. Examples of work zone detection research.....	14
Figure 8. Flowchart for urban work zone detection and sizing.	23
Figure 9. Training data sources.....	24
Figure 10. Data-Centric training pipeline.....	26
Figure 11. YOLOv8 (R=Reduce layer, T=Top-down layer, B=Bottom-up layer, C=Concat, U=Upsample, D=Downsample, O=Out layer)	26
Figure 12. Label distribution by object type.....	28
Figure 13. False positive example: Incorrect steam vent detection.....	28
Figure 14. Proposed work zone size estimation method.	31
Figure 15. PR curve of model validation from baseline model and data-centric (DC) trained models using different training data sources combinations.	33
Figure 16. Example of detection output (daytime)	35
Figure 17. Example of detection output (nighttime).....	36

Figure 18. Output of work zone detection, inference, and area estimation.	37
Figure 19. The crash fatalities and injuries (2009-2022) of the New York City	39
Figure 20. Research and development framework for the safety view map application	41
Figure 21. Grid Generation in the Study Area	43
Figure 22. The concepts of FCW, BCW and PCW (Source: Mobileye)	44
Figure 23. ME8 and OEM collision warning locations and distributions	45
Figure 24. Crash records in New York City during 7/5/2022-12/31/2022	46
Figure 25. Demonstration of computing VMT for a grid cell (Xie et al., 2017a)	47
Figure 26. The land use group spatial distribution in the study area	48
Figure 27. The correlation matrix of the safety related variables	50
Figure 28. Bivariate Moran's I for Near Misses and Crash Counts	52
Figure 29. Bivariate LISA Cluster Map using K-nearest neighbor (k=8)	54
Figure 30. Injury and Fatal Crash related Bivariate LISA cluster map using K-nearest neighbor (k=8)	55
Figure 31. The VIF of each safety risk related variable	56
Figure 32. The spatial distribution of grid-based scaled safety risk index	60
Figure 33. The spatial distribution of grid-based crash frequency	61
Figure 34. Safety Risk Index Spatial Distribution Using Only Near Miss Data (without Crash Counts)	62
Figure 35. The spatial distribution of grid-based injury and fatal crash related SSRI	63
Figure 36. The spatial distribution of grid-based injury and fatal crash frequency	64
Figure 37. Injury and fatal crash related scaled safety risk Index spatial distribution using only near miss data (without crash counts)	64
Figure 38. Microservice Architecture	66

Figure 39. Automated pipeline	68
Figure 40. List virtualization with a view window	70
Figure 41. Multi-threading algorithm performance evaluation	71
Figure 42. WorkZoneX interface	72
Figure 43. WorkZoneX dashboard interface.....	73
Figure 44. WorkZoneX: Map components and filters	74
Figure 45. WorkZoneX: Labels of non-work zone and detected work zone	75
Figure 46. WorkZoneX: Locate and Learn More functions.....	76
Figure 47. WorkZoneX: Number of work zones in each borough	77
Figure 48. WorkZoneX: Work zone duration chart for individual work zones	77
Figure 49. WorkZoneX: Historical work zone statistics and download function.....	78
Figure 50. SAFExMAP interface	79
Figure 51. SAFExMAP: Displaying SSRI, SSRI rank, and component variables.....	80
Figure 52. SAFExMAP: Pairwise Comparative Analysis of SSRI.....	81
Figure 53. SAFExMAP: Filter options and data layers.....	81
Figure 54. SAFExMAP: Multi-source data visualization	82
Figure 55. SAFExMAP: User-defined polygon for spatiotemporal analysis.....	83
Figure 56. SAFExMAP: Temporal pattern bar charts	84
Figure 57. The correlation plots of VRU crash data and VRU near misses	100
Figure 58. Bivariate LISA cluster map for VRU crash data and near misses (k-nearest neighbor, k=8)	102
Figure 59. The distribution of VRU SSRI.....	104

List of Tables

Table 1 List of the selected camera systems in the U.S.....	7
Table 2 Summary of recent literature on smart transportation apps using computer vision.	16
Table 3 Work zone objects.	27
Table 4 Example of ground truth information.....	29
Table 5 Training Performance Comparison.	34
Table 6 Work Zone Scene Identification Confusion Matrix.....	34
Table 7 Input Variable Summary	49
Table 8 Global Moran’s I statistics (k-nearest neighbors, k=8)	51
Table 9 The estimate results of best NB model.....	56
Table 10 The feature importance of XGBoost model	57
Table 11 Injury and Fatal Crash related SSRI Variables and Their Parameters	62
Table 12 Inter-Process Communication Methods.	67
Table 13 Query Speed Comparison with and without Indexing.....	69
Table 14 Capital cost (using individual computer).	85
Table 15 Capital cost (using server services).	85
Table 16 Cost summary	86
Table 17 Bivariate Global Moran’s I using K-nearest neighbor (k=8) for VRU crash data/near misses	101
Table 18 Pedestrian SSRI Variables and Their Parameters.....	102
Table 19 Cyclist SSRI Variables and Their Parameters.....	103

Section 1 Introduction

1.1 Background

The rapid development of the internet of things (IoT), sensing technologies, Artificial Intelligence (AI), machine learning and deep learning techniques, have yielded new perspectives on how novel technologies can be applied to smart cities. The New York City IoT Strategy report highlighted that the city has been home to a major expansion in IoT and AI use in the last decade, with impacts on many areas, including its transportation system. As a subfield of AI, Computer Vision is showing promising potential in understanding the realistic dynamics of cities. While cities are complex by nature, especially cities like NYC, the applications of computer vision show progress in tackling a variety of complex physical and non-physical visual tasks. In addition, computer vision can turn existing infrastructure into smart sensors in a myriad of ways, and new applications are being continuously developed. Agencies including NYC Department of Transportation (NYC DOT), NYC Department of Design and Construction (NYC DDC), and NYC Mayor's Office of the Chief Technology Officer (CTO) have identified a “Wishlist” of area of interests in using computer vision technologies for the City. The Wishlist include using computer vision tools to validate the accuracy of collected mobility data (e.g., turning movement counts, vehicle classifications, vehicle and pedestrian speed, etc.), parking utilization, work zone assessment, incident detection, mobility aids detection, vehicle-pedestrian conflicts as well as assessing how the computer vision algorithm can be trained for NYC conditions where there may be difficult sightlines and blockages and restrictions on camera placement. This “Wishlist” provides the foundation of understanding what can the technology offer and what are the needs from the agencies (Figure 1).

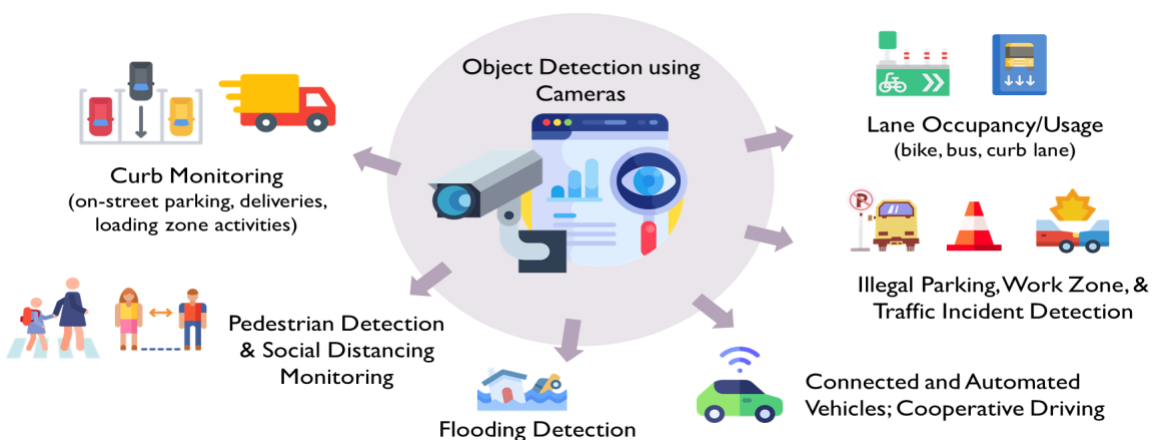


Figure 1. Examples of Smart City Applications Using Computer Vision

On the one hand, while many of the innovative computer vision technologies require agencies to replace their existing intelligent transportation systems (ITS) infrastructure with new camera devices or sensors, others can leverage existing resources making them more cost-effective. Utilizing existing ITS infrastructure not only reduces the deployment cost, but also reduces the chance of its obsolescence. The Closed-circuit television (CCTV) system, which is available for many transportation systems, is a valuable source of traffic condition information and can be used as vision sensors. Traffic video data can provide rich information, such as traffic volume, travel speed, curb activities, and incident information (Figure 2), to facilitate traffic operations and management.



Figure 2. Computer Vision Applications Using Traffic Cameras (Source: C2SMART)

On the other hand, emerging transportation data sources, such as camera-based imagery recorded from probe vehicles and CAVs can be used to augment existing transportation databases and enable real-time mobility and safety monitoring for our transportation systems. By sensing the driving scenes, vast amount of information can be extracted – free space, vehicle and pedestrian detection, traffic sign recognition, and lane markings (Figure 3). By connecting this image-based data with vehicle dynamics such as acceleration and braking, crucial information on dangerous driving and safety based on the environment can be learned.

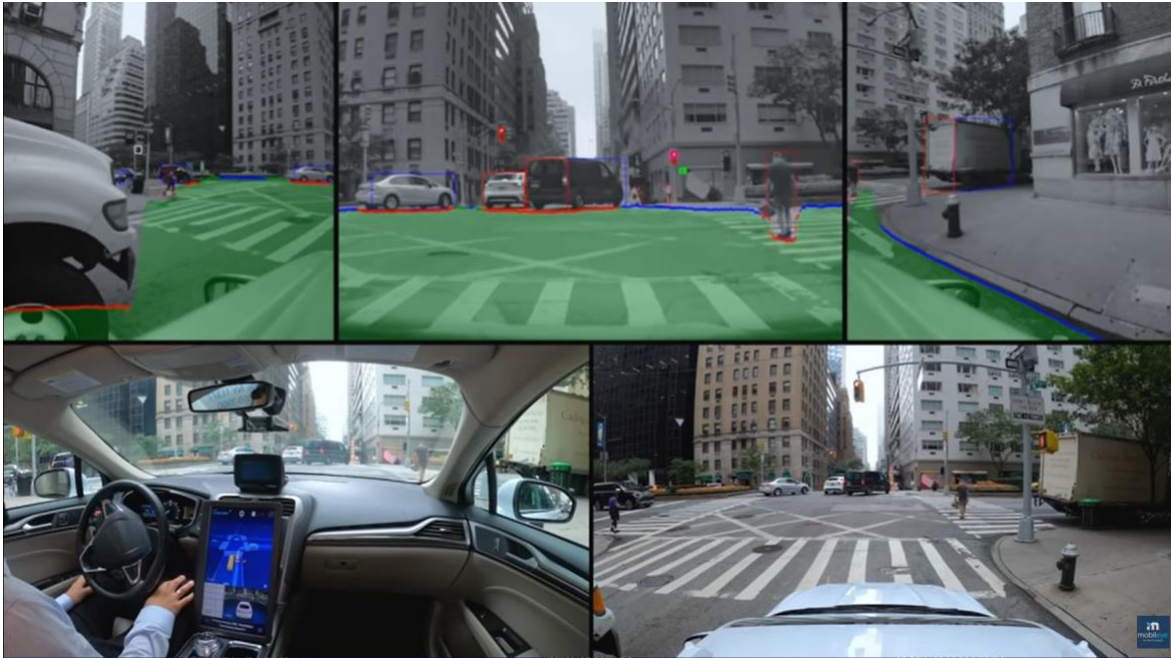


Figure 3. Computer Vision Applications Using Probe Vehicle/CAV-based Cameras (Source: Mobileye)

These new streams of transportation and mobility data collected and analyzed can be used in various ways. For example, pedestrian and bike detection output can provide input to the city's non-motorized count program; vehicle count and classification can be used to fill data gaps for traffic impact studies such as NYC congestion pricing before-and-after analysis; curb activity detection can provide support for new curb strategies like evaluating potential smart loading zones; pedestrian intention detection can be applied to connected and automated vehicle pedestrian safety applications. This fully remote approach also allows for data collection without deploying humans in the field providing not only labor cost savings, but also a safe alternative during public health crises in which disease spread is a concern.

However, there is a need to assess the maturity of these applications and the feasibility of connecting various different data sources concurrently. For example, fixed traffic camera imagery has limitations in terms of blind spots and visual conditions that vehicle-based imagery can overcome. When applying them to cities like NYC, the unique existing ITS infrastructure being used as the vision sensors must be investigated. Further research is needed to better understand the process and realize computer vision's true potential for large scale complex urban systems. Specifically, there is a lack of linking the

technology to the needs of agency users and validate and quantify the perceived benefits of computer vision technology to the existing data collection processes and planning/operation strategies.

1.2 Study Objectives

The primary objective of this research is to develop a cost-effective deep learning based data acquisition and analytics tool using vision-based sensors to facilitate smart transportation systems. We mainly used NYC as a living lab as it has a complex urban transportation system that provides opportunities for applying computer vision for different use cases. Given the limited duration of the project, we developed computer vision applications for two selected use cases that are already identified as high priority by the stakeholders and prepare a summary of existing and potential smart transportation applications that can benefit from the use of computer vision. The two selected use cases are: 1) real-time urban work zone detection, and 2) safety risk map for dangerous driving and vehicle-pedestrian/cyclist near misses. Specifically, the research aims to achieve the following goals:

- Identify smart city applications with a focus on transportation using computer vision and object detection from the existing literature and assess their maturity and state of adoption.
- Understand the needs from stakeholders and build a list of available CCTV systems in the U.S.
- Establish an automatic pipeline using computer vision algorithms for two selected use cases and validate their feasibility and applicability using existing traffic cameras and crowd sourced CAV-based traffic and camera data.
- For the safety risk map use case, develop a road risk scoring map application for all road users as well as vulnerable road user (VRU) near misses, including pedestrians and cyclists. For the work zone use case, develop an application to identify real-time work zones, traffic condition around work zone and validate temp work zones.
- Evaluate the cost-effectiveness of the proposed solutions.

This goal of this project also aligns with ongoing efforts with NYC DDC and NYC DOT through the NYC Town + Gown efforts as part of the City's broader Vision Zero initiative to improve the safety of city streets and previous experience from the COVID-19 social distancing study¹.

¹ University Transportation Centers (UTC) Programs, UTC Spotlight, "[Using Video Feeds from Public Traffic Cameras and Computer Vision to Analyze Social Distancing and Travel Patterns during the COVID-19 Pandemic](#)," Accessed November 11, 2021.

As a part of the safety risk map use case, the research team collaborated with its industry partner, Mobileye, a global leader in the development of vision technology for Advanced Driver Assistance Systems (ADAS) and autonomous driving. Mobileye has one of the world’s largest crowdsourcing fleet, which harvesting 450M miles globally. Figure 4 shows its equipped fleet vehicle coverage across North America. Mobileye has rich mobility intelligent information updated at a high refresh rate and near-real time information and imagery about traffic flow and traffic obstructions. For the proposed research project, Mobileye will share its cloud sourced connected vehicle data for NYC from camera-based technology that advance safety and urban mobility.

1.3 Collaborative Approach

This research applies a collaborative approach that facilitates partnerships between academic institutions, industry and government agencies (Figure 4).

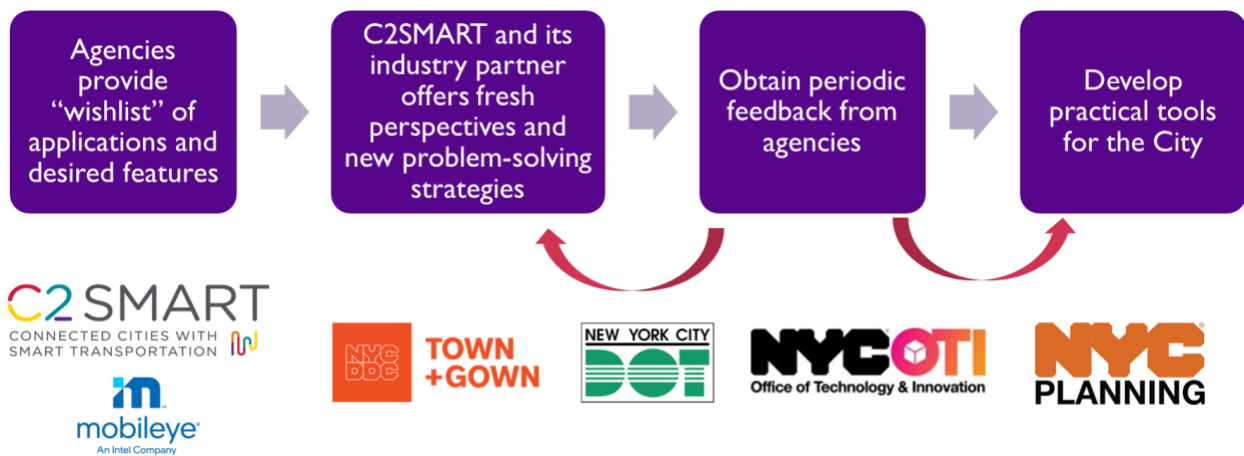


Figure 4. Collaborative approach

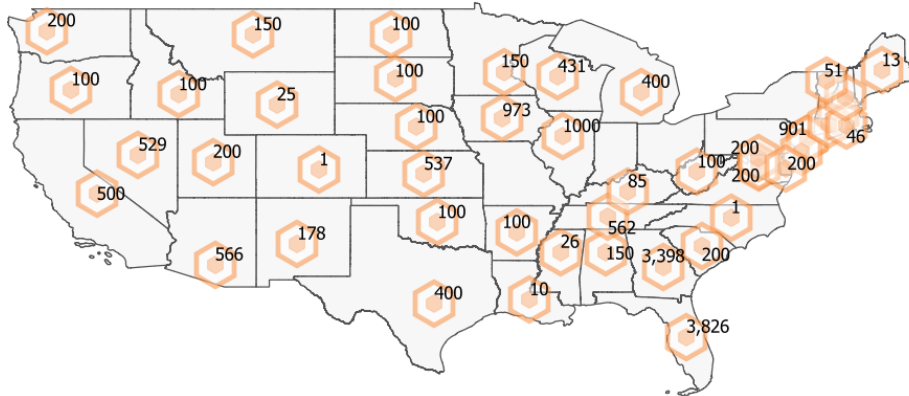
Section 2 Literature Review

Computer vision is reshaping the transportation industry and bringing its unique capabilities to the table to enable next generation smart transportation systems in many different ways. The state-of-the-art of IoT strategies and computer vision techniques is well-studied in the literature, some has already been tested and used for certain use cases, but this technology has not widely applied in the day-to-day operation to existing transportation infrastructures yet. A comprehensive review of both state-of-art and state-of-practice as well as gaps in terms of use cases and applications is needed. In addition, several major challenges that hinder further advances in computer vision-based smart transportation application development remain. This includes how to develop the transportation-specific computer vision techniques through advanced artificial intelligence (AI) and machine learning (ML) techniques; how to make use the outputs of the computer vision-based systems to enhance traffic safety and situational awareness; how to customize the solutions based on different objectives from the agencies and road users; how to improve the accuracy of these systems under conditions such as adverse weather; and finally, how to maintain the cost-effectiveness of these computer vision-based transportation solutions.

We conducted a multi-facet literature review that first examined the current use cases and transportation related applications that utilize the computer vision methodologies with a focus in urban areas, especially work zone and safety applications, and evaluated their applicability to various tasks of urban analytics, state of adoption, and limitations. The literature review then assessed if and how transportation equity is considered in the current state of adoption of computer vision/AI technology, for example, whether state-of-the-art object detection systems have equitable predictive performance on pedestrians with different skin tones.

2.1 Publicly Available Traffic Camera Systems in the U.S.

Using a screening analysis, we identified various existing publicly available CCTV camera systems across 45 states in the US. Figure 5 shows their spatial distribution and system scale. Table 1 lists the name of these systems and their streaming features, camera resolution, system scale and weather these cameras are facing urban streets or highways.



#CAMERAS

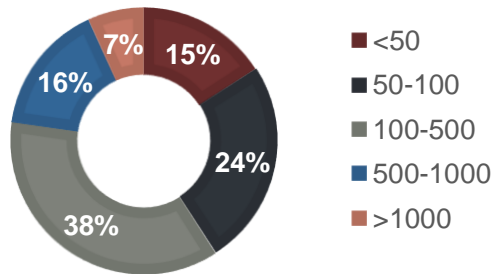


Figure 5. Spatial distribution and scale of the selected camera systems in the U.S.

Table 1 List of the selected camera systems in the U.S.

City/State	#Cameras	Continuous stream?	Resolution	Highway/Urban Street
NYC, NY	901	Every 1-5 seconds	Moderate	Both (some are at intersections)
NY	2,414	Yes	Moderate	Most highways and intersections
Nevada	529	Yes	Low	Only highways and big roads
DC, District of Columbia	200+	Every 3-7 seconds	Low	Mostly at crosswalks
Seattle, Washington	200+	Yes	High	Variety of different places
Florida	3,586	Yes	Moderate	Only highways and big roads
Austin, Texas	400+	Image updates every hour or so	Low	Good coverage for highways

Arlington, Virginia	200+	Yes	Moderate	Most in downtown and airport areas
Maryland	200+	Yes	Low	Covers big cities of DC and Baltimore and between cities, only highways
Illinois	around 1,000	Image updates every few minutes	Low	Only highways and expressways in the state
Minneapolis, Minnesota	around 100-200	Yes (videos can be played)	Low	Only highways
Oklahoma City, Oklahoma	around 100	Yes	High	Concentrated in the city, only highways
Birmingham, Alabama	around 100-200	Yes	Moderate	Only highways
Apex, North Carolina	1	Yes	720p	One urban street
Massachusetts	around 300-400	Image updates every 5-15 seconds	Low	Only highways
California	around 500	Yes	Low	Only highways
Georgia	3,398	Image updates every few minutes to hours	Low	Only highways
Arizona	566	Image updates every few minutes to hours	High	Only highways
Michigan	around 400	Image updates every 10-30 seconds	Low	Only highways
New Jersey	around 200	Yes	Low	Only highways
Colorado	1	Yes	Low	One highway
Oregon	100+	Image updates every few hours	Moderate	Only highways
Utah	around 200	Image updates every few hours	Moderate	Only highways
Maine	13	Image updates every few minutes	Moderate	Only highways
Tennessee	562	Yes	Low	Only highways
Louisiana	10	Yes	Low	Only highways
Connecticut	318	Image updates every 5-15 seconds	Moderate	Only highways
South Carolina	around 200	Yes	Moderate	Only highways
Wisconsin	431	Yes	High	Only highways
Mississippi	26	Image updates every few seconds	Moderate	Only highways
Arkansas	around 100	Yes	Moderate	Only highways
Iowa	973	Image updates every few minutes	Moderate	Only highways

Kansas	537	Image updates every few minutes	Moderate	Only highways
Nebraska	around 100	Image updates every hour	Moderate	Only highways
Wyoming	25	Image updates every hour	Moderate	Only highways
Idaho	around 100	Image updates every few minutes	Low	Only highways
Rhode Island	46	Image updates every few minutes	Moderate	Major highways and a few local streets
New Hampshire	100-200	Image updates every few minutes	Low	Only highways
Delaware	around 100	Yes	Moderate	Only highways
West Virginia	around 100	Yes	Moderate	Only highways
Vermont	51	Image updates every few minutes	Low	Only highways
South Dakota	less than 100	Image updates every few minutes to hours	Low	Only highways
North Dakota	around 100	Image updates every few minutes	Moderate	Major highways and a few local streets
New Mexico	178	Image updates every 1 minute	Low (320x240)	Major highways and a few local streets
Kentucky	85	Image updates every few minutes	Low (300x220)	Major highways and a few local streets
Montana	100-200	Image updates every few hours	Low (213x120)	Only highways

2.2 Smart Transportation Applications Using Computer Vision

Many computer vision approaches have been introduced for vehicle detection. Based on these approaches, numeric research has been focusing on traffic counting and traffic monitoring, including density and speed estimation, congestion detection and so on. For example, Muhammad (Fachrie, 2020) created a simple vehicle counting system to help human in classify and counting the vehicles that cross the street. YOLOv3 was used for object detection and pre-trained model was applied using Common Objects in Context (COCO) dataset, a large-scale object detection, segmentation, and captioning dataset that has annotations for 80 different objects. The system achieved a detection accuracy as high as 96.96% with 'motorcycle' and 'car' being the most accurate and 'truck' and 'bus' being the worst accurate vehicle category. Most of the studies used a centralized detection system with a few utilizing edge computing. Liu et al. (Liu et al., 2021) proposed a two-tier edge computing based

model for congestion and speed detection. They build their own video dataset using an IP-based camera. They also compared the edge and cloud schemes with the hybrid scheme (edge + cloud) and found that under good weather condition, the performance of the edge scheme is better than that of the cloud scheme while under bad weather condition (i.e., snowy), the performance of the cloud scheme is better than that of edge scheme.

Considerable development efforts have been made into autonomous driving using sensing technology and computer vision to find road obstacles and analyze the current traffic flow and surrounding conditions. Many review papers have been developed, for instance, (Chen et al., 2020) evaluated the technologies used to advance autonomous driving, including CNN, SSD, R-SNN, R-FCN and so on. The review paper identified that recurrent neural network (RNN) could be replaced by long-short term memory (LSTM) in terms of autonomous driving scenes because it could bring more efficiency. The authors tackled the existing works of these methods and concluded selected approaches to point their strengths and gaps. This study highlighted that since autonomous driving is fairly new to society, it is important to improve the weaknesses of scientific methods to help them become a safer option. Some studies focus on enhancing 3D object detection. Peng et al. (Peng et al., 2020) introduced a lightweight Instance-Depth-Aware (IDA) 3D Detection to approaching object detection in autonomous driving which accurately predicted the depth of the 3D bounding box's center by instance-depth awareness. Their method focused on objects and directly performs the instance depth regression and paid more attention on far-away objects by disparity adaptation and matching cost reweighting.

One of the vital application areas in smart transportation is accident detection. Ijjina et al. (Ijjina et al., 2019) developed a neoteric framework for detection of road accidents using road-traffic CCTV surveillance footage. This work was evaluated on vehicular collision footage from different geographical regions under various ambient conditions such as harsh sunlight, low visibility, daylight hours, snow and night hours. The dataset includes accidents in various ambient conditions such as harsh sunlight, daylight hours, snow and night hours. All videos were compiled from YouTube and were around 20 seconds. Their proposed framework was able to detect accidents correctly at a 71% detection rate with 0.53% false alarm rate on the accident videos. Another interesting research (Ghosh et al., 2017) used eye blink detection system based on object tracking and machine learning to alert drivers with high efficiency. Authors used real life dataset of drivers when they are commuting to a certain destination. This system had an efficiency of 80%, which means it could detect about 8 eye blinks in 10 actual blinks.

Various studies have also been conducted on parking occupancy detection using computer vision. Traditional approaches for parking occupancy detection include background subtraction and hand-

crafted feature (e.g., edges, color, texture) extraction (Acharya et al., 2018). Single shot detector (SSD) (Liu et al., 2016), You Look Only Once (YOLO) (Redmon and Farhadi, 2018) and its subsequent versions, and CNN-based frameworks (Krizhevsky et al., 2017) have achieved state-of-the-art accuracies in image classification and object detection. For instance, Acharya and Yan (Acharya et al., 2018) used deep Convolutional Neural Networks (CNNs) trained from public datasets (PKLot) and a binary Support Vector Machine (SVM) classifier to achieve outdoor parking occupancy detection. The detection accuracies of the model are reported to be 99.7% and 96.7% for a public dataset and for a new dataset generated by the authors. Amato et al. (Amato et al., 2016) developed a solution for visual parking space occupancy detection using a deep CNN model robust to light condition changes, presence of shadows, and partial occlusions. The authors tested two CNN architectures, mAlexNet and mLeNet, based on (Krizhevsky et al., 2017) and (LeCun et al., 1998) and reported an overall accuracy 82.9% on CNRPark, and 90.4% on PKLot dataset using mAlexNet. Bulan et al. (Bulan et al., 2013) presented a video-based real-time on-street parking occupancy detection system using background subtraction, motion detection, and occlusion detection. To eliminate unreliable frames and regions for vehicle detection, they applied occlusion detection based on the position of a foreground blob with respect to a parking region. The parking occupancy detection method performs in real time with a 91% average detection accuracy for each camera. The authors stated that the video-based approach could replace the in-ground sensors since the former has a higher detection accuracy than that of in-ground sensors in San Francisco.

A natural value-added option to on-street parking occupancy detection is to perform illegal parking detection simultaneously. For example, Bulan et al. (Bulan et al., 2013) integrated parking angle violation detection, parking boundary violation detection, and exclusion zone violation detection, into their parking occupancy detection model. Other than fixed traffic or surveillance cameras, Gkolia and Vlahogianni (Gkolia and Vlahogianni, 2018a) developed data science models to detect empty on-street parking spaces in urban networks based on in-vehicle cameras. Ranjan et al. (Ranjan et al., 2019) introduced StreetHAWK that leverages the rear camera of a dashboard mounted smartphone to identify potential parking violations. Other value-added features can be considered for adoption, such as bus or bike lane occupancy and violation detection. In the literature, most studies focus on parking lot usage detection (Acharya et al., 2018; Amato et al., 2016; Lee et al., 2005; Marmol and Sevillano, 2016; Wu et al., 2007; Yin et al., 2019) but illegal parking detection is mostly needed on-street. Only a few studies (Bulan et al., 2013; Gkolia and Vlahogianni, 2018a) have tested for on-street parking occupancy of curb lanes. Previous studies have often relied on moderate to high resolution videos (over 480p) and consecutive video frames (>1 frame per second (fps)) (Chen and Yeo, 2019; Lee et al., 2009; Marmol and Sevillano, 2016; Ranjan et al., 2019; Xie et al., 2017c) and many models use vehicle tracking (Marmol and Sevillano, 2016) (Chen and Yeo, 2019; Tang et al., 2020; Xie et al., 2017c) for event detection. Since

public traffic surveillance cameras suffer from low image resolution and frame rate, an effective solution that accounts for this feature is needed.

Besides vehicle detection, enhancing the safety of vulnerable road users (VRUs) is also of critical importance to achieving the objectives of USDOT's National Roadway Safety Strategy (NRSS), and vision zero goals. According to data from the National Highway Traffic Safety Administration (NHTSA), in 2020 there were 10,626 traffic fatalities in the United States at roadway intersections, including 1,674 pedestrian and 355 bicyclist fatalities. These fatalities at intersections represent 27% of the total of 38,824 road traffic deaths recorded in 2020. Previous detection methods for VRUs, especially pedestrians, mainly using infrared sensors, radar sensors, thermal imaging, microwave sensors and so on. Figure 6 shows the evolution of pedestrian detection technologies and vision-based detection system showed an increasing trend in recent years. More details about pedestrian detection deployment can be found in the interactive timeline and map visualizations developed by the C2SMART research team for USDOT Intelligent Transportation Systems Joint Program Office (ITSJPO) at <https://www.itskrs.its.dot.gov/decision-support>. Besides the general VRU detection application, some studies also extended the use case to social distancing measuring or pedestrian intention predictions (Wang et al., 2022; Zuo et al., 2021). Zuo et al. (Zuo et al., 2021) developed a reference-free video-to-real distance approximation-based urban social distancing analytics. Their method measured pedestrian distancing and density at crosswalks and sidewalks in complex urban environments to quantify social distancing to better understand the new norm of urban mobility amid COVID-19 pandemic. Wang et al. (Wang et al., 2022) added a Temporal Attention (TA) to the encoding and decoding layers of the Generative Adversarial Network (GAN) to improve pedestrian intention prediction. Such predictions can be further incorporated into various applications such as jaywalker detection and cooperative perception. Review of the literature also revealed that only a few studies centered on detecting people with mobility aids. Kollmitz et al. (Kollmitz et al., 2019) collected of over 17,000 annotated images from a hospital in Frankfurt, Germany and developed a model to detect people with mobility aids to benefit robots operating in hospitals. Their dataset contained five classes, including pedestrian, person in wheelchair, pedestrian pushing a person in a wheelchair, person using crutches, and person using a walking frame. The study only focused on indoor environment and its performance on outdoor environments such as crosswalks is unknown.

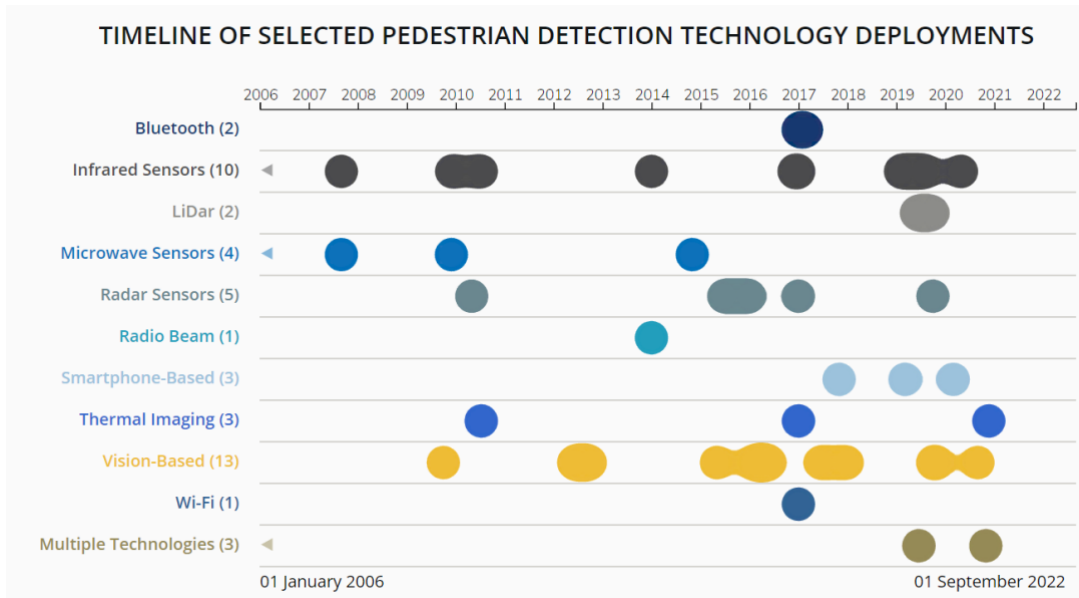


Figure 6. Timeline of selected pedestrian detection technology deployments (Source: C2SMART & ITSJPO)

2.2.1 Work Zone Detection Using Computer Vision

The literature review also revealed that progress has been made in computer vision, but mainly on pedestrians and vehicles. Computer vision for other use cases, such as work zone detection, is still very limited. Most existing studies focus on off-street work zones or a single type of work zone object (e.g., traffic cones). In addition, almost all of the existing literature emphasized that the main challenge for work zone detection is the scarcity of publicly available, large-scale, domain-specific, annotated dataset of work zone imagery.

For example, Nath and Behzadan (Nath and Behzadan, 2020) used a CNN model that laid out a framework for detecting the most common types of off-street construction objects, namely, buildings, equipment, and workers (Figure 7 (a)). They recognized the lack of publicly available annotated work zone imagery dataset and introduced a systematic approach to visual data collection through crowdsourcing and web-mining and annotating the dataset for AI model training to overcome the limitation. The results showed that models perform best when trained on combined (crowdsourced and web-mined) data. They collected 3,500 images with 11,500 work zone objects and tested both YOLO-v2 and -v3. The study found the best-performing model is YOLO-v3, which had a 78.2% mAP.

Duan et al. (Duan et al., 2022) also stated that the lack of large-scale, open-source dataset for the construction industry limited the development of computer vision algorithms as they are often data-hungry. This study developed a new large-scale work zone image dataset, Site Object Detection dAtaset (SODA), which was collected from the real construction site and contained 15 types of object classes categorized by workers, materials, machines, and layout. A total of 19,846 images including 286,201 objects were mined and annotated. Their model achieved a maximum mAP of 81.47%. They also suggested field data acquisition could adopt methods such as using drones, handheld monocular camera shooting, and construction site monitoring video. The limitation of this dataset is it is mainly for off-street work zones and may not be suitable for detecting work zones that occur on the roadways.

A recent study conducted by Katsamenis et al. (Katsamenis et al., 2023) used Yolov5 for traffic cone detection using a training dataset of 500 traffic cones images (Figure 7 (b)). The data used in this paper was collected and manually annotated under the framework of the H2020 HERON project. The results showed that the proposed computer vision model could achieve a 91% accuracy in detecting traffic cones. However, work zones, especially urban work zones often composed by multiple types of construction objects and have no standard work zone set up, single object type detection may not be as effective as expected in such cases. This demands the needs of building and sharing a comprehensive publicly available, domain-specific, annotated dataset of urban work zone (on urban streets and sidewalks) imagery.



Figure 7. Examples of work zone detection research

Additionally, we found most of the existing studies rely on specific cameras while a few of them utilized existing intelligent transportation systems (ITS) infrastructures such as closed-circuit television (CCTV) cameras (Ijjina et al., 2019; Zinchenko et al., 2020; Zuo et al., 2021).

Table 2 synthesizes some of the most recent literature on smart transportation applications using computer vision. While not exhaustive, it provides a representative sample of recent research effort.

Table 2 Summary of recent literature on smart transportation apps using computer vision.

Study	Year	Application(s)	Goal	Method	Training data publicly available?
(Sajib and Bhuiyan, 2019)	2019	Traffic monitoring, Traffic count	Propose a vision-based traffic monitoring system detect the number of vehicles that monitors the density of the roads.	Haar feature based Adaboost classifier and virtual detection lines (VDL)	No
(Liu et al., 2021)	2021	Traffic monitoring (speed estimation & congestion detection)	Propose a two-tier edge computing based model for congestion and speed detection	Gaussian Mixture Model and Global Foreground Detection	No
(Fachrie, 2020)	2020	Traffic counting	Aim to create a simple vehicle counting system to help human in classify and counting the vehicles that cross the street.	YOLOv3 & counting using coordinates or location of the vehicles	Yes
(Kousar Nikhath et al., 2021)	2020	Traffic counting	Develop a video-based system that can be used to count the road traffic, and it does not disturb traffic flow	Background extraction	Yes
(Leroux et al., 2022)	2022	Traffic Counting	Develop small, location-specific object detection models for traffic counting without needing manual data labeling	location specific models	Yes
(Zinchenko et al., 2020)	2020	Traffic counting	Incorporate an intelligent traffic light controlling system using an algorithm that consumes real data from closed-circuit television (CCTV) cameras	Neural network-based models	Yes
(Khan et al., 2020)	2020	Pedestrian detection	Develop an accurate computer vision-based system to track and count passengers for both indoor and outdoor scenarios.	SVM classifier and histograms of orientated gradient descriptor	Yes

(Zuo et al., 2021)	2021	Pedestrian detection/Social distancing	Measure pedestrian distancing and density to quantify social distancing to better understand the new norm of urban mobility amid the pandemic	Reference-free distance measure algorithm & YOLOv3	No
(Wang et al., 2022)	2022	Pedestrian intension estimation	Add a Temporal Attention to the encoding and decoding layers of the Generative Adversarial Network to improve pedestrian intention prediction	Generative Adversarial Network based on Temporal Attention	Yes
(Kollmitz et al., 2019)	2019	People with disabilities	Detect people with mobility aids to benefit robots operating in indoor environment such as hospitals.	Deep convolutional neural network (CNN)	Yes
(Gao et al., 2022)	2022	Parking management/Illegal parking	Develop a computer vision–based data acquisition and analytics approach for curb lane monitoring and illegal parking impact assessment	YOLOv3 & Mask R-CNN	Yes
(Gkolias and Vlahogianni, 2018b)	2018	Parking management	Develop data science models for the detection of empty on-street parking spaces in urban road networks based on data provided by in vehicle cameras	CNN	Yes
(Nath and Behzadan, 2020)	2020	Work Zone Detection (off-street)	Detect construction objects at off-street construction sites	YOLOv2/v3	Yes
(Duan et al., 2022)	2022	Work Zone Detection (off-street)	Develop a large-scale off-street construction site image dataset	YOLOv3/v4	Yes
(Katsamenis et al., 2023)	2022	Work Zone Detection (Traffic cone only)	Detect construction buildings, equipment, and workers	YOLOv5	Yes
(Kanchana et al., 2021)	2021	Autonomous driving	Evaluate the technologies used to advance autonomous driving, including CNN, SSD, R-SNN, and R-FCN	CNN	Yes
(Peng et al., 2020)	2020	Autonomous driving	Introduces Instance-Depth-Aware (IDA) 3D Detection as to approaching object detection in autonomous driving which accurately predicts the depth of the 3D	IDA 3D Detection	Yes

			bounding box's center by instance-depth awareness		
(Chen et al., 2020)	2020	Autonomous driving	Review and develop advanced technologies for the visual sensing system of autonomous vehicles from standard computer vision to event-based neuromorphic vision	CNN & Neuromorphic-vision algorithms	Yes
(Ijjina et al., 2019)	2019	Accident Detection	Develop a neoteric framework for detection of road accidents using road-traffic CCTV surveillance footage	Mask RCNN	No
(Yan et al., 2020)	2020	Accident detection	Detection of workers and heavy vehicles, three-dimensional (3D) bounding box reconstruction, depth and range estimation in the monocular 2D vision, and 3D spatial relationship recognition	CNN	Yes (for COCO) No (for KITTI)
(Ghosh et al., 2017)	2017	Accident detection	Develop an eye blink detection based alert system	Eye blink detection	No
(Arshad et al., 2019)	2019	Flood management/monitoring	Present a systematic review of the literature regarding computer vision applications in flood monitoring and mapping	Artificial Neural Networks and so on	Yes

2.2.2 Safety Analysis

In this project, we aim to generate a safety risk index utilizing the data extracted using computer vision techniques from in-vehicle cameras in CAVs. Safety risk of urban traffic is complex that has many different ways to estimate. There are two key elements of the estimation: variable selection and data fusion method.

Variable selection

Regarding to variable selection, crashes, near misses, traffic facilities, traffic characteristics, demographic information, land use and other variables were found in the previous literature. Jiang et al. (2020) proposed safe route mapping model that integrates crash and conflict risk to score the safety of roadways. He et al. (2021); Santhanavanich et al. (2020) used crash-based kernel density estimation method to find the high safety risk locations. Abdelrahman et al. (2019) developed per segment risk indexing using crashes and near misses.

Vehicle Miles Traveled (VMT) serves as an exposure variable, reflecting the volume of motor vehicle traffic. Xie et al. (2017a) identified VMT as key traffic exposure indicator contributing to pedestrian-vehicle collisions in Manhattan. Further supporting this, a positive correlation was found between VMT and the number of accidents Yang et al. (2019). In addition to the exposure indicator, characteristics of the road network, such as intersection density, number of intersections, and road length, also significantly influence road safety risk. Huang et al. (2018) utilized a geographically weighted model to uncover that greater four-way intersection density within block groups typically correspond with higher collision rates. Moreover, Ding and Sze (2022) found a positive correlation between intersection density and bicycle-vehicle collisions, suggesting that high intersection density might adversely affect road safety. Aligning with these findings, Silva et al. (2020) argued that road length stands out as a critical factor when modeling traffic collision frequency.

As it pertains to land use, Yang et al. (2019) discovered that a higher proportion of commercial areas positively correlates with the total number of collisions occurring throughout the day. Conversely, a greater proportion of park areas tends to diminish the frequency of collisions. This indicates that land use patterns significantly influence traffic safety. Expanding on this, Lym and Chen (2020) found that commercial land use tend to elevate the risk of Property Damage Only (PDO) and Injury-related accidents. Further reinforcing these findings, studies focusing on risks associated with cycling (Xie et al., 2021) and pedestrian activities (Xie et al., 2017a) demonstrated that commercial areas exert a larger

positive influence on accident rates compared to other land use patterns. The implication is that the characteristics of land use significantly contribute to the variation in accident rates.

Data fusion

Data fusion technology is a critical component in effectively integrating and weighting various data sources, thereby providing a comprehensive understanding of traffic safety risks. Several techniques, such as the entropy weight method, divergence method, and fuzzy comprehensive evaluation method, have been proposed in past research to amalgamate different road risk-related variables into a singular index that accurately represents traffic safety.

The entropy weight method (Huang et al., 2021) is an approach that offers a systematic way of assigning weights based on the inherent variability of data. The method gets its name from the concept of entropy in information theory, where entropy is a measure of the uncertainty or randomness of information. If a variable has higher information entropy, its weight is larger. The primary advantage of this method is its objectivity, thus avoiding the pitfalls of subjectivity involved in manually defining weights. However, a significant drawback is its inability to consider the correlation between indicators, making it more suitable for scenarios where variables exhibit weak interrelationships.

On the other hand, the scatter degree method (Huang et al., 2021) assigns weight coefficients that reflect the maximum overall difference between the objects evaluated. When applying in safety risk evaluation, this approach is advantageous as it optimizes the variance between data points, enhancing the identification of distinct road safety risk factors. However, a potential limitation is that it might overlook similarities or patterns that could be meaningful in the risk analysis.

Jiang et al. (2020) introduced a fuzzy logic-based method to unify single indicators. The strength of this method lies in its tolerance for imprecise data and its ability to model complex non-linear relationships through flexible fuzzy rules. This flexibility makes it particularly useful for handling intricate and vague real-world data scenarios often encountered in traffic safety research. However, a disadvantage of this method is the potential subjectivity introduced in designing fuzzy rules and weights, which may result in deviations in the comprehensive evaluation results.

Section 3 Urban Work Zone Detection

The occurrence of work zone activities within highway networks and urban roadways can induce substantial traffic interruptions. These operational disruptions often result in roadway capacity reduction (i.e., by closing down one or more lanes) that might lead to severe congestion and roadway crashes. On the one hand, work zones have contributed to approximately ten percent of highway congestion in the U.S., resulting in an estimated annual loss of \$700 million in fuel alone (Edara et al., 2013), while exacerbating the negative environmental effects of vehicle emissions and increasing safety risks (Edara et al., 2017). For example, one study found that the crash rate increased by 24.4% under work zone conditions compared to non-work zone conditions (Ozturk et al., 2014). On the other hand, while offline data sources such as work zone permit data may be available, the majority of U.S. cities have yet to establish a real-time approach to monitor actual activities during operational periods of work zones throughout the road network.

Given these concerns, real-time work zone detection becomes crucial. Knowing the location, duration, size of the work zones in real-time can provide vital insights into their impact on traffic flow and safety and help decision makers strategically allocate resources through the Transportation Management System (TMS).

One viable approach to detecting work zones in real-time is using vision-based detection by means of existing traffic cameras. While these traffic cameras are extensively used for pedestrian and vehicle detection, their application to road work zone detection, especially in complex urban settings, remains limited. Urban work zones are uniquely challenging due to pedestrian and vehicular activities, complex surroundings, and lack of standardized setups.

As mentioned in Section 2, the majority of existing studies using computer vision for work zones focus on off-street sites that might employ different types of equipment than those seen in roadside work zones, or they concentrate solely on detecting a single type of work zone component (e.g., traffic cones) (Katsamenis et al., 2022). Moreover, the mere recognition of work zone equipment does not necessarily confirm the presence of a work zone, as such equipment can sometimes serve other purposes, including regulating traffic (e.g., using barrels to separate traffic lanes). Consequently, there is a need for a new methodology that identifies work zone scenes in their entirety rather than simply detecting individual pieces of some work zone equipment.

Additionally, most of the existing Artificial Intelligence (AI) applications adopt model-centric approaches wherein data collection is perceived as a one-time event to improve the model architecture to enhance

its performance (Motamedi et al., 2021). However, given the inherent scarcity of open-source training samples for work zone detection, the current limitation in this domain is the lack of data rather than shortcomings in the model. This constrains the development of computer vision algorithms for this specific problem, which typically require substantial amounts of data.

In this project, we introduce a deep learning-based framework to effectively recognize urban work zone scenes and their sizes. The main contributions of this research are summarized as follows:

- We propose a data-centric training approach designed to iteratively improve the performance of work zone object detection by augmenting a customized training dataset fused from multiple data sources to overcome the sparsity of annotated real-world work zone images. These sources include 2,600 images with 15,000 work zone object labels from traffic cameras, web-mined images, and synthetic work zone images generated through a 3D simulator.
- We implemented a topology-based inference method using XGBoost to automatically identify work zone scenes. This innovative approach is designed to deal with the complexities of work zone scene detection caused by the fact that recognizing individual or certain combination of work zone components alone (e.g., a traffic cone behind a car) may not necessarily represent a true work zone.
- We developed a reference-free work zone size estimation method, which utilizes the standard heights of common work zone equipment, to provide a generalized real-pixel distance rate method.

3.1 Framework for Urban Work Zone Detection and Sizing

The emerging field of Data-Centric AI is anticipated to introduce techniques for dataset optimization, thus enabling detection algorithms to be effectively trained even with relatively small datasets (Motamedi et al., 2021). In this project, we propose a data-centric framework for urban work zone detection and sizing which contains three modules: 1) a data-centric training that systematically augments training datasets with the goal of enhancing the accuracy of the work zone detection model, 2) a topology-based work zone scene inference that can identify work zones by understanding the positional relationships and connections among detected work zone objects (e.g., cones placed adjacent to a line of fences), and 3) a reference-free estimation for work zone size. We describe each part of the proposed detection methodology in the following sections. Figure 8 shows the working flow of the proposed method.

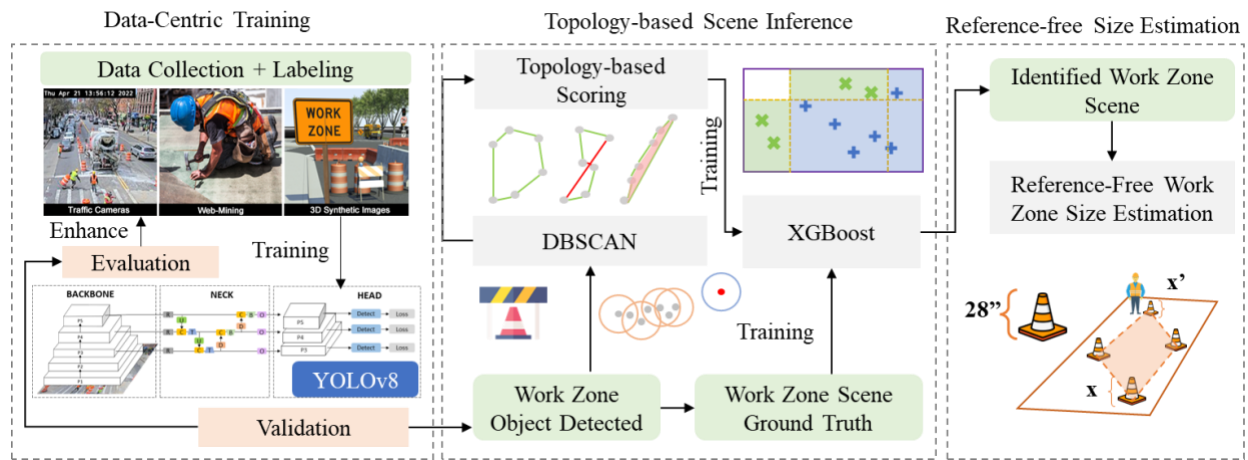


Figure 8. Flowchart for urban work zone detection and sizing.

Data-Centric Training for Work Zone Object Detection: We begin with collecting a customized dataset of 2,600 work zone images with about 15,000 labels from diverse sources including CCTVs, web-mined images, and a 3D simulator, offering a wide array of work zone scenarios (Figure 9). This model is designed to identify several key work zone objects, such as traffic cones, construction workers and vehicles. Web-mined and 3D synthetic images primarily serve to fill the gaps in certain subcategories of the training data, which may be sparse in the CCTV images, yet their detection accuracy is vital for the model. For instance, web-mining is a good supplemental source for augmenting training images for construction vehicles. The quality of the data was incrementally improved by correcting label errors and pruning noisy labels based on data quality objectives.

Topology-Based Work Zone Scene Inference: Once work zone objects are identified, their topological arrangement is analyzed. We calculate a topology complexity score based on the positional relationships and connections among these objects. The score serves as an indicator of whether the scene represents an organized work zone or a random accumulation of work zone objects for non-work zone purposes. The score is then fed into an XGBoost classifier, which is trained on ground truth work zone scene data.

Reference-Free Work Zone Size Estimation: After the presence of a work zone is confirmed, an estimation algorithm is performed to approximate the size of the work zone. This is accomplished using a reference-free method, which utilizes the standard heights of common work zone equipment to establish a scale. With this scale, the distances and sizes of the work zone scene can be inferred, allowing for the estimation of the work zone's size.



Figure 9. Training data sources.

3.1.1 Data-Centric Training Pipeline

The key to work zone detection and sizing lies in accurately recognizing work zone-related objects. For the purposes of this study, we focus on seven key objects, including traffic cones, barricades, barrels, chain fences, construction vehicles, signs, and workers. The YOLOv8 model that integrates cutting-edge backbone and neck architectures with the mosaic augmentation method is used as it enhances both feature extraction and object detection, compared to previous YOLO versions (Jocher et al., 2023). Given the absence of a pre-trained YOLOv8 model tailored to our needs due to the lack of publicly available labeled roadway work zone data, we curated and manually labeled our own training set featuring work zone objects from the sources described in Figure 9.

Data-centric model training recognizes that quality data is key to achieving better model performance, especially when dealing with real-world scenarios that are diverse and often unpredictable. Consider the objective of data-centric training in a simplified manner. Assume that we have a model, defined by its parameters θ , and a dataset $D = (x_i, y_i)$, where x_i is an instance (image in our case) and y_i is the corresponding label. In the typical model-centric training, we want to find optimal parameters θ^* that minimize a loss function L , averaged over all instances in the dataset:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum L(y_i, f(x_i; \theta)), \text{ for all } i \text{ in } D \quad (1)$$

Where $f(x_i; \theta)$ is the output of the model given instance x_i and model parameters θ .

In contrast, for the data-centric training, we recognize that our dataset D itself might be sub-optimal, due to label errors or lack of image diversity. We hence introduce a notion of "dataset quality" $q(D)$, which outlines how good our data is. The goal then becomes to optimize not just the model parameters θ but also the dataset D itself:

$$(\theta^*, D^*) = \underset{\theta, D}{\operatorname{argmin}} \frac{1}{N} \sum L(y_i, f(x_i; \theta)) - \lambda^* q(D), \text{ for all } i \text{ in } D \quad (2)$$

Here, λ is a regularization parameter balancing model loss and data quality. Enhancing dataset D might include correcting labeling errors, ensuring data representation, and introducing edge cases for model generalization.

In this study, we refined the dataset based on model performance, adding data and correcting/pruning label errors specifically for subclasses falling short of desired accuracy. The data-centric training pipeline is illustrated in Figure 10.

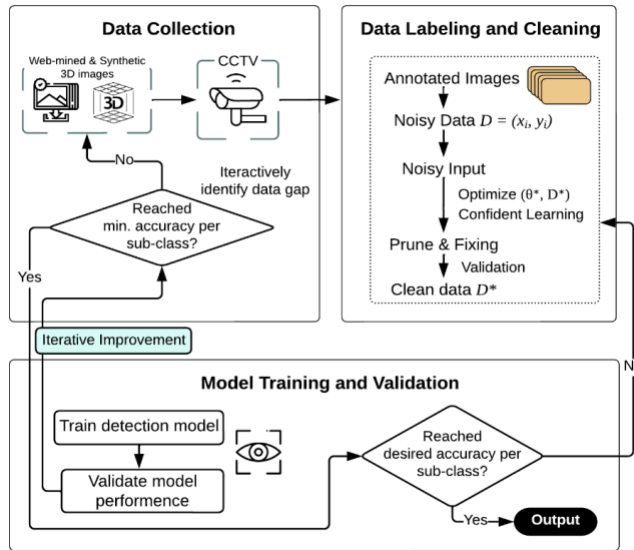


Figure 10. Data-Centric training pipeline.

A customized YOLOv8 model utilized the training dataset generated is used as the object detection model. YOLOv8 (Jocher et al., 2023) is the latest in the YOLO series of real-time object detectors, provides outstanding performance in terms of speed and accuracy, making it ideal for a variety of object detection tasks. It employs advanced backbone and neck architectures for improved feature extraction and object detection and features an anchor-free split detection head for enhanced accuracy. Furthermore, YOLOv8 offers an optimal balance between accuracy and speed, and provides a range of pre-trained models for various tasks such as object detection, instance segmentation, pose/keypoints detection, and classification. When compared to YOLOv5, YOLOv8 is faster, more accurate, and provides a more comprehensive framework for developers.

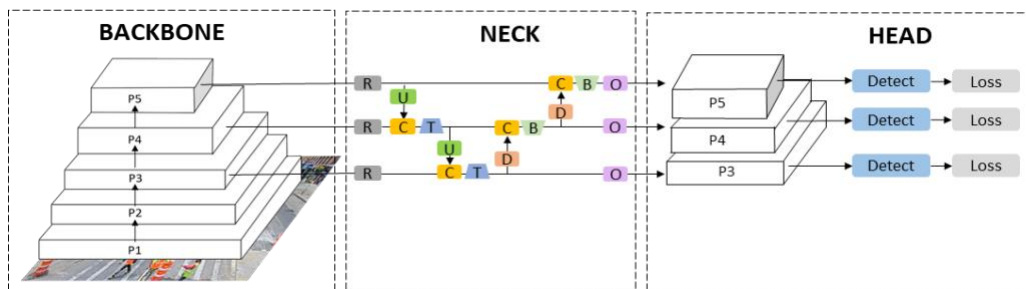













Figure 11. YOLOv8 (R=Reduce layer, T=Top-down layer, B=Bottom-up layer, C=Concat, U=Upsample, D=Downsample, O=Out layer)

Initially, 11 work zone-related objects were considered for training and testing (refer to Table 3). However, following an evaluation of the label distribution based on object type (as shown in Figure 12) and examples of false positives (Figure 13), changes were made to the work zone object classes. Specifically, 1) the ' Manhole Guard Rail' class was removed due to insufficient training data, 2) 'Steam Vent', 'Fence', 'Trench Cover', and 'Delineator' classes were removed due to high false positive rates, and 3) a 'Chain Fence' class was added, as it was identified as a common work zone object in NYC.

Table 3 Work zone objects.

ID	Object Type	Example	ID	Object Type	Example
1	Traffic cones		7	Construction Workers	
2	Barrels/Drums		8	Construction Vehicles	
3	Barrier & Barricades		9	Work Zone Signs	
4	Fence		10	Trench Covers	
5	Delineators and Channelizer		11	Manhole Guard Rail	
6	Stream Vent				

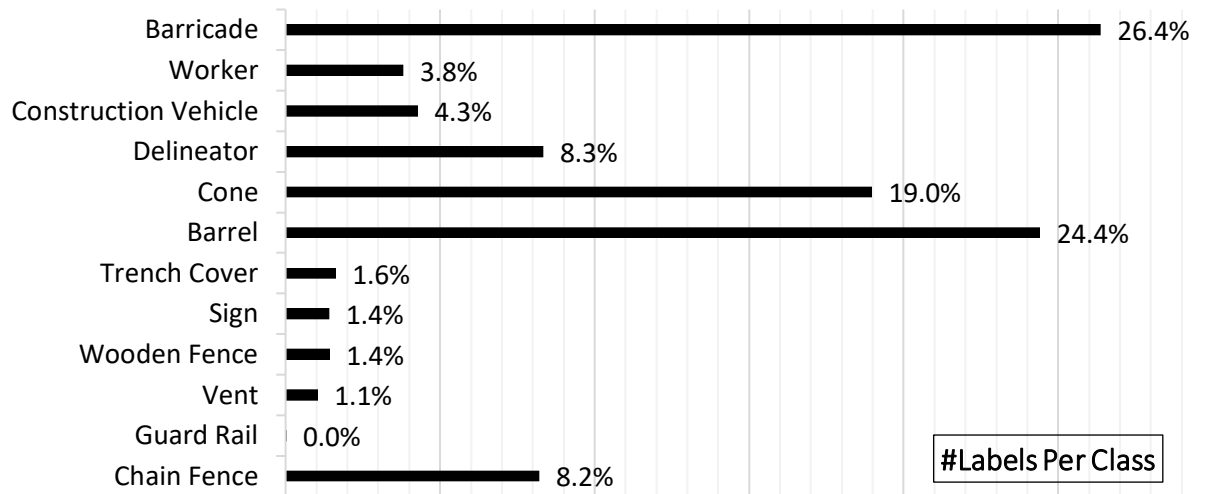


Figure 12. Label distribution by object type



Figure 13. False positive example: Incorrect steam vent detection

3.1.2 Topology-based Work Zone Inference using XGBoost

Not all scenes containing work zone objects are actual work zones, especially in urban areas. Therefore, a work zone scene identification model is needed. In this project, we introduced an innovative topology-based work zone inference using a supervised learning classifier, XGBoost, to reliably identify work zone scenes under real-world conditions.

Firstly, we manually collected ground truth data based on CCTV cameras from real-world work zone and non-work zone scenes that contains work zone objects. Table 4 provides an example of the information collected.

Table 4 Example of ground truth information collected from real-world work zone and non-work zone scenes that contains work zone objects.

ID	Cones	Barrels	Barriers	Fences	Workers	Const. Vehs.	Signs	Work Zone?	In travel lane?	In Parking Lane?
1		4						No	No	No
2			22	6				Yes	No	Yes
3			10	27	2	1		Yes	No	Yes
4	30				4	2	1	Yes	Yes	No
5	1							Yes	Yes	No
6	1							No	No	No
7	2	1						Yes	No	Yes
8			4					No	No	Yes
9	1	2						No	No	No
10		30					4	Yes	Yes	No

Second, instead of relying solely on the presence of work zone objects, this methodology also considers their arrangement and inter-connectedness within the work zone. A topology complexity score is derived from these detected objects, illustrating their layout complexity. Work zone objects are clustered using a density-based clustering algorithm, DBSCAN (Ester et al., 1996). Any cluster with fewer than three items is considered as noise (i.e., non-work zone). The topology complexity score views the detected work zone as a graph. From any random vertex, the algorithm searches for the nearest vertex with a degree less than two, then adds an edge between them. This process is repeated until every vertex in the graph is connected, and no vertex has a degree greater than two. The complexity is measured by the features of the generated graph, including the number of edge cross points, the ratio of cycles to chordless cycles, and the length distribution of edges.

These measurements serve as input variables for a supervised learning classifier, which identifies whether a particular scene represents a work zone or not. For the selection of the classifier, we prioritized inference speed, training speed, and accuracy. Since Tree-based models continue to outperform deep learning on medium tabular data (Grinsztajn et al., 2022) and our data is tabular with a limited sample size, we decided to apply XGBoost, which is a gradient-boosting algorithm that is well-known for its efficiency and performance. The detailed algorithm is provided in Algorithm 1.

This classifier achieved a model accuracy of 91-94% on test data. In addition, feature importance test showed barrier, cones, barrels and construction vehicles are the most important features for identifying a true work zone.

Algorithm 1 Topology Complexity Score

Require:

- 1: Initialize a list of qualified clusters C_q from the results of DBSCAN consists with C .
 - 2: Initialize an empty Graph $G(V, E)$.
 - 3: Initialize a list of available vertices V_a consists with all vertices v_a in G having a degree less than or equal to 1.
 - 4: N is the total number of vertices in C , n is the total number of vertices in G .
 - 5: Original Vertex: v_0 ; Destination Vertex: v_d ; Number of cycles: N_c ; Number of chordless cycles: N_{ch} ; Number of crossing edges: N_{cc}
 - 6:
 - 7: **for** each cluster C in C_q **do**
 - 8: **for** each vertex v in C **do**
 - 9: **while** $n < N$ **do**
 - 10: **if** $G = \emptyset$ **then**
 - 11: $v_0 \leftarrow v$
 - 12: **else if** Degree of $v_d \geq 2$ **then**
 - 13: $v_0 \leftarrow v_d$
 - 14: **else**
 - 15: $v_0 \leftarrow \forall v \in V_a$
 - 16: **end if**
 - 17: **if** V_a has more than two vertices **then**
 - 18: Find the nearest v_a and set it as v_d
 - 19: Add an edge between v_0 and v_d
 - 20: Update G and V_a
 - 21: **else if** V_a has only two vertices **then**
 - 22: Set the last v_a other than v_0 as v_d
 - 23: Add an edge between v_0 and v_d
 - 24: Update G and V_a
 - 25: **end if**
 - 26: **end while**
 - 27: Calculate N_c, N_{ch}, N_{cc} and the length of edges in G , store these attributes.
 - 28: **end for**
 - 29: Find graph $G_f \leftarrow \text{argmin}_{N_c}(G)$
 - 30: Obtain topology scores from G_f
 - 31: **end for**
-

3.1.3 Reference-free Work Zone Size Estimation

To overcome variance in camera positioning and perspective, we suggest a reference-free methodology for work zone size estimation, based on standard heights of common work zone equipment such as traffic cones. This method, initially proposed from our previous work (Zuo et al., 2021) for pedestrian detection, eliminates the need for a physical scale reference in the scene which usually requires on-site human investigators.

The reference-free method divides the image into several hyper-planes, oriented perpendicularly to the horizontal plane and vanishing lines. Due to perspective effects, each hyper-plane exhibits a unique real-to-pixel distance ratio (RP-rate) as the number of pixels corresponding to a given real-world length varies between hyper-planes. Next, we propose that each specific type of work zone equipment in the image stands perpendicular to the horizontal plane and maintains a uniform actual height h_r . Figure 14 provides an example illustrating the proposed area estimation method.

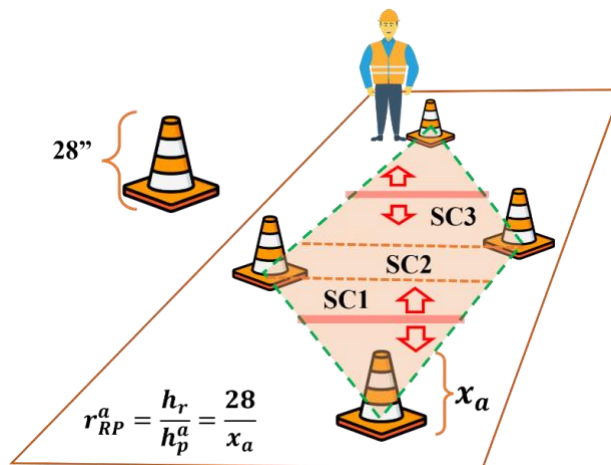


Figure 14. Proposed work zone size estimation method.

Subsequently, the detected object (vertex) is ranked according to the RP rate, and one edge is then added to the adjunct vertices. Combining the final graph G_f generated from Algorithm 1 and removing the duplicated edges, a set of sub-cycles are generated. The calculation of the real distance is considered as the integration of the product between RP-rate and pixel distance Δp (Zuo et al., 2021). Similarly, the real area size of each chordless cycle can be formulated as the integration of the product between RP rate and the pixel area size Δs . Given that each cycle comprises at least three vertices, it's

straightforward to divide a cycle into sub-cycles between two neighboring vertices using a supplementary vertical line.

A horizontal line is added from each vertex inside the generated work zone area, and then the area is separated into sub-region (SC1 to SC3 in Figure 14). Each sub-region is considered as many small rectangles, and the length of the rectangle is the real distance between the region's bounds, l , which can be calculated using the pixel distance multiplied by the RP rate. The rectangle width is calculated using the method introduced in (Zuo et al., 2021) due to the perspective effect. Thus, each sub-region can be calculated as a double integral that is written as:

$$A_{sc} = \int_a^b \int_{r_a}^{r_b} l_r dldr \quad (3)$$

where the a and b are the two vertices, l_a and l_b are the horizontal distance between the work zone bounds at vertex a and b ; r_a and r_b are the RP rates of vertex a and b , which can be calculated as follow:

$$r_{RP}^i = \frac{h_r}{h_p^i} \quad (4)$$

where h_p^i is the pixel height of the detected equipment, and h_r is the real height of the equipment type. In this study, the height of a barrel, a cone and a barricade is assumed to be 37, 28, and 42 inches, respectively.

3.2 Experiment and Results Analysis

To prove the effectiveness of our proposed framework in urban work zone detection, we evaluated the detection model performance using precision-recall (PR) curve and Mean Average Precision (mAP) over IoU 0.5 (mAP@0.5) based on different training datasets. Then, we assessed the performance of the work zone scene identification and size estimation based on confusion matrix, accuracy and F1 score.

3.2.1 Work zone object detection model performance

Our training data primarily derives from a subset of the 900+ fixed CCTV traffic cameras in New York City (nyctmc.org), providing a variety of urban work zone images under different lighting, weather, and traffic conditions, although with relatively low resolution (i.e., 240p). Free stock images sourced from the web and synthetic images from a 3D simulator are also incorporated as supplement data sources in addition to the CCTV images. The free stock images provide high-resolution depictions of specific classes, such as construction workers or vehicles, while the 3D simulator generates synthetic work zone

images from various angles and work zone setups under controlled conditions. This approach bridges gaps in areas where real-world data may be scarce. For instance, if there are few examples of night-time work zones in the actual data, these scenarios can be simulated in 3D. This ensures the data-centric training dataset comprehensively covers all possible scenarios, a crucial aspect for training a robust model.

Among the 2,600 images collected and annotated, 890 CCTV, 850 stock, and 280 synthetic 3D images were used as the training data, and 580 CCTV images were used as the test/validation data. We used the YOLOv8 as our model, trained on four customized training models for 300 epochs, and evaluated on the test set. The four training models are: 1) **Baseline Model** uses original CCTV data without data-centric processing, 2) **DC-CCTV** uses CCTV data with data-centric processing, 3) **DC-CCTV+Stock** uses CCTV and free stock data with data-centric processing, and 4) **DC-CCTV+Stock+3D** uses CCTV, free stock and synthetic 3D data with data-centric processing.

Figure 15 displays the PR curve resulting from different training model combinations. The performance, compared to the baseline, saw substantial improvement after implementing data-centric training in DC-CCTV, which included data cleaning and re-labeling. Furthermore, when additional data sources were added to the training data pool, overall model performance increased due to data enhancement processing (Figure 15(c) and (d)), especially for subclass barricade, construction vehicle, work zone sign, and chain fence. Table 5 presents the mAP@0.5 scores for each work zone object class. Interestingly, a marginal decline is observed in performance for certain specific types upon data augmentation. A possible explanation is that the inclusion of stock and synthetic 3D images broadened the model's generalization thus reduced the risk of overfitting.

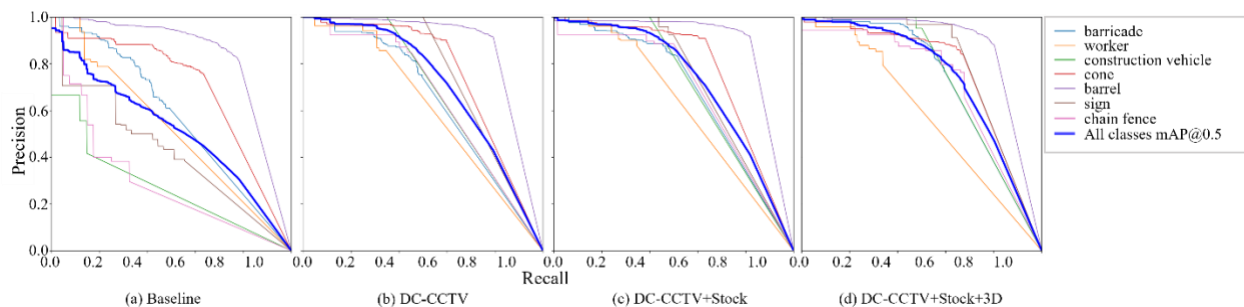


Figure 15. PR curve of model validation from baseline model and data-centric (DC) trained models using different training data sources combinations.

Table 5 Training Performance Comparison.

Detection Type	Base	Data-Centric Training		
	CCTV	CCTV	CCTV+Stock	CCTV+Stock+3D
Barricade	0.572	0.644	0.680	0.725
Worker	0.516	0.610	0.620	0.579
Const. veh	0.272	0.675	0.700	0.737
Cone	0.692	0.755	0.773	0.764
Barrel	0.830	0.871	0.882	0.872
Sign	0.424	0.750	0.727	0.799
Chain fence	0.291	0.655	0.680	0.713
ALL	0.514	0.708	0.723	0.741

All values are Map@0.5

3.2.2 Work zone scene identification model performance

For XGBoost training, we selected 684 CCTV images, 399 containing unique work zones and 285 containing work zone objects that do not constitute work zones (e.g., cones for lane control). Features like work zone shape, equipment count, number of crossing edges, the ratio of the cycle over the chordless cycle, and the number of edge outliers were manually labeled for model training. We tested the trained model on 853 CCTV images, with the confusion matrix presented in Table 6.

Table 6 Work Zone Scene Identification Confusion Matrix.

	True WZ	True Non-WZ
Predicted WZ	15	5
Predicted Non-WZ	7	826

Upon analyzing the confusion matrix, it is noted that the accuracy of the model is 98.4%, whereas the F1 score stands at 0.713. The discrepancy between these metrics can largely be attributed to the skewed distribution of the dataset, where the majority of data points are not associated with a work zone. Despite the inherent bias indicated in the F1 score, the results showcase practical applicability and offer satisfactory performance in real-world contexts. As demonstrated in Figure 16 and Figure 17, the work zone scene identification model functioned effectively under both daytime and nighttime conditions. It also performed well even under rainy conditions or when the camera views were blurred, as shown in the top two pictures of Figure 16.



Figure 16. Example of detection output (daytime)

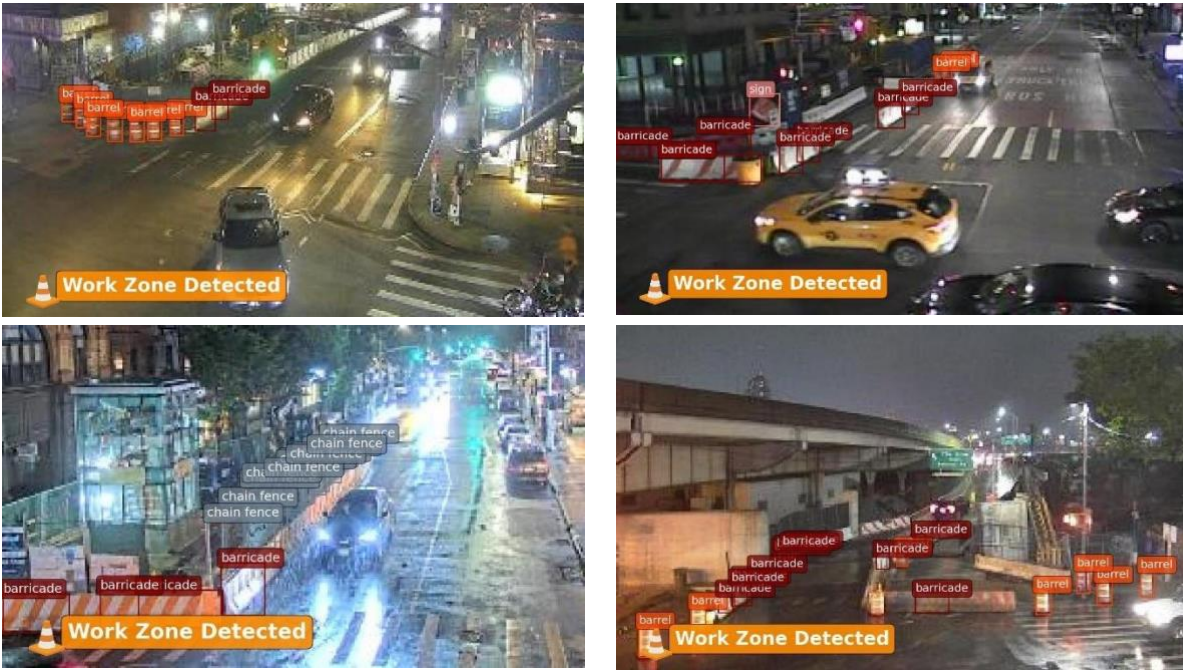


Figure 17. Example of detection output (nighttime)

3.2.3 Work zone size estimation model performance

Figure 18 presents illustrative examples of work zone inference and area estimation. The first two images accurately identify work zones and their sizes, while the last two images avoid false positives despite the presence of work zone equipment. This demonstrates our model's accuracy even in complex environments. We evaluated our method using ten real-world images, manually estimating work zone sizes via satellite images. The method achieved accuracy within a range of 67.71% to 89.52%.



Figure 18. Output of work zone detection, inference, and area estimation.

The top two images highlight detected work zones in orange, with estimated sizes at the bottom right. The bottom images show correct detection of non-work zones containing work zone equipment.

3.3 Limitations

One limitation we found in the proposed detection model concerns distinguishing between a construction worker and a traffic enforcement agent. Although human judgement was applied when annotating the training data, since both of them often wear similar yellow or orange safety vests, the model sometimes recognizes traffic enforcement agents as construction workers. Given that construction workers have a relatively high importance score in the work zone scene identification model (i.e., a scene with a detected worker is highly likely to be a true work zone), this can potentially lead to the misclassification of work and non-work zones. Future work should explore solutions to this issue.

The area estimation methodology also carries certain limitations that necessitate further improvement. First, it functions optimally when the identified boundary equipment forms a closed enclosure. Further improvement is needed for work zones that utilize natural boundaries such as road curbs or walls in addition to typical enclosures like barrels or cones. Also, the reference-free estimation approach assumes uniform heights for each object type, which could introduce bias if actual sizes deviate.

3.4 Conclusion

The urban work zone application aims to address the challenges of automatically recognizing and sizing work zones in complex urban environments. The research team introduced a multi-facet framework that combines data-centric AI training, topological analysis, gradient boosting classification, and reference-free size estimation, forming a comprehensive work zone detection toolkit. We developed a deep-learning based work zone object detection model with a data-centric approach to iteratively enhance the model's performance by augmenting a custom training dataset collected from multiple sources, thereby overcoming the sparsity of annotated real-world work zone images. The training data is acquired from traffic cameras, mined from the web, and 3D-simulated work zone images. An innovative topology-based inference method is introduced, using XGBoost, for distinguishing true work zones from non-work operational zones with some work zone features. We also developed a reference-free work area size estimation method, which utilizes the standard heights of common construction equipment to provide a generalized real-pixel distance approximation.

Our model's efficacy is demonstrated with an average mAP of 74.1% across all work zone classes, an accuracy of 98.4% for scene identification, and an accuracy of up to 89.52% for size estimation. This holistic approach enables real-time identification and size estimation of work zones with reasonable accuracy under complex urban settings. This can potentially facilitate the provision of more informed active work zone management, thereby improving safety and mobility in the presence of work zones. As the proposed approach was empirically validated using existing traffic camera infrastructure in NYC, it also shows its potential as a new mechanism for generating real-time work zone data remotely in a cost-effective manner. We aim to combine the proposed approach with traffic and incident detection, as well as work zone permit database, to facilitate a more effective TMS in the future.

Section 4 Safety Risk Index Map

4.1 Research and Development Framework

According to the World Health Organization, approximately 1.35 million people die in road traffic crashes annually (WHO, 2022), making it the eighth leading cause of death worldwide. Furthermore, an additional 20 to 50 million individuals suffer non-fatal injuries (ASIRT, 2022), often resulting in long-term disabilities. In densely populated urban areas like New York City, traffic safety concerns are exacerbated by the sheer volume of vehicles, pedestrians, and cyclists that share the streets (see Figure 19). With over 259 fatalities and 50,733 injuries reported in 2022 (CHEKPEDS, 2023), the city faces numerous challenges, including the high cost of accidents, the vulnerability of certain road users, and the impact of crashes on traffic congestion. Addressing these issues is of utmost importance for the well-being of the city's residents, commuters, and visitors.

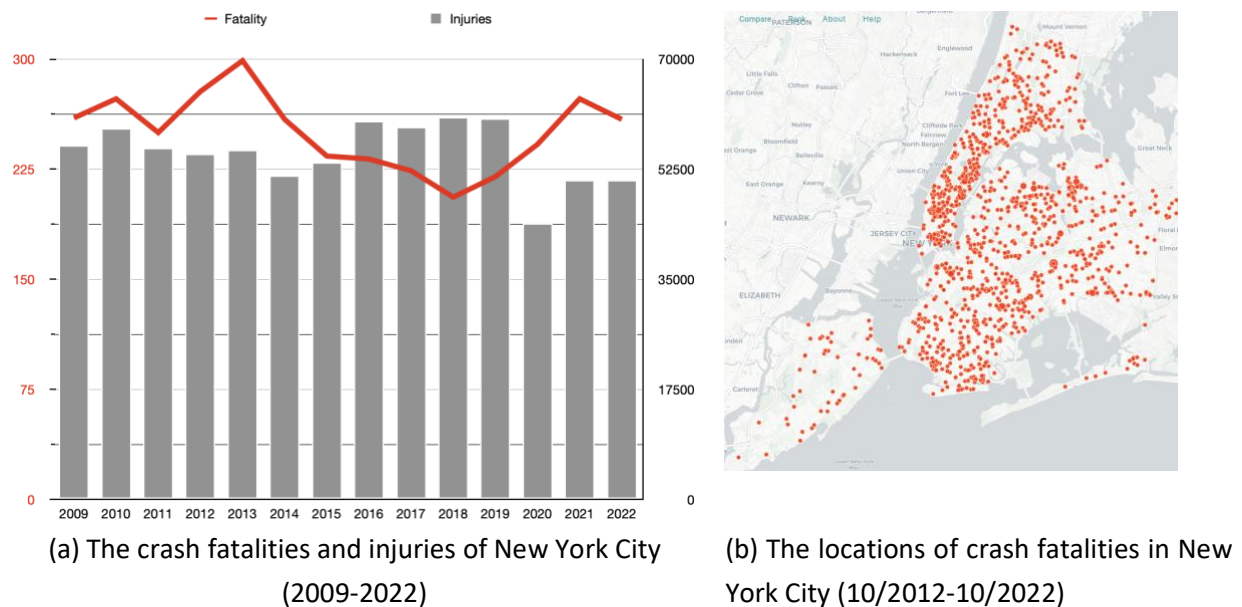


Figure 19. The crash fatalities and injuries of the New York City

Fortunately, a unified commitment to promoting safety for all is demonstrated at both federal and city levels. The Federal Highway Administration's (FHWA) Strategic Plan for 2022-2026 (FHWA, 2022) places traffic safety at the forefront of its goals, emphasizing five key components: Safety Design, Safety System, Safe Public, Safe Workers, and Critical Infrastructure Cybersecurity. This comprehensive approach highlights the need for innovative infrastructure solutions, data-driven methodologies, public

awareness, worker safety, and cybersecurity measures to address the multifaceted challenges of traffic safety. New York City's Vision Zero Plan (NYC Vision Zero, 2023), launched in 2014, is another key initiative that shares the objective of improving traffic safety. The plan aims to eliminate traffic-related fatalities and serious injuries by implementing a combination of enforcement, education, and engineering measures. Despite a nationwide rise in traffic fatalities, New York City defied the trend in 2022 with a 6.6% reduction in overall traffic fatalities, and a 6.3% reduction in pedestrian fatalities, with Vision Zero in effect. These federal and city-level initiatives underscore the collective responsibility to prioritize traffic safety.

Historical crash data plays a critical role in traffic safety analysis, as it provides a foundation for understanding the underlying factors contributing to accidents, identifying high-risk areas, and informing targeted interventions. However, despite its value, the rarity of crash, underreport issue, low location accuracy limits its usage in practice. With the advancement of computer vision technology, either via fixed infrastructures (e.g., CCTV cameras) or on-board devices (e.g., in-vehicle cameras), have emerged as a valuable tool in detecting near misses. These devices can capture real-time data on near-miss incidents, providing a more comprehensive understanding of potential hazards and close calls that might not result in reported accidents. Analyzing near miss incidents effectively can offer insights that extend beyond individual incidents and can be vital to improving road safety and preventing serious, fatal, or expensive accidents. In addition to crash and near miss data, traffic volume, road attributes, road network features, transport facility distributions, land use, population and social demographics data are available in some cities and can provide supplemental information for traffic safety analysis.

This study introduces a safety risk index based on data fusion, which consolidates multisource data and presents crash-related data through a highly interactive map application. This would be particularly useful in urban areas, such as New York City. To effectively organize the data, we employed a grid-based analysis that enables the integration of various factors affecting traffic safety. We first examined if a correlation exists between the crash records and near miss data collected via in-vehicle camera through computer vision technologies for both vehicle crashes and vulnerable road user (VRU) crashes (i.e., pedestrian and cyclist). The near miss data is provided by the industry partner Mobileye. Then, we modeled the crash frequency by considering several variables, including near misses, traffic volume, the number of intersections, road length, land use percentage, and population density. Utilizing machine learning methods such as XGBoost, we calculated the importance of each variable in relation to crash frequency. Based on the calculated importance scores, we generated a scaled safety risk index that incorporates crash frequency-related variables weighted by their respective importance. This risk index aims to provide a comprehensive measure of traffic safety risk in the urban environment, considering

the complex interplay of various factors. To make this valuable information accessible, we also developed a web-based safety map that visualizes the spatial distribution of the safety risk index, as well as other crash related information, such as speeding tickets or aggregated link speed, across the city.

This data-driven, interactive map will allow stakeholders — including city planners, traffic engineers, and policymakers — to easily pinpoint high-risk areas requiring safety interventions and provides insights into key factors contributing to traffic safety issues. The research and development framework for the safety view map application can be seen in Figure 20.

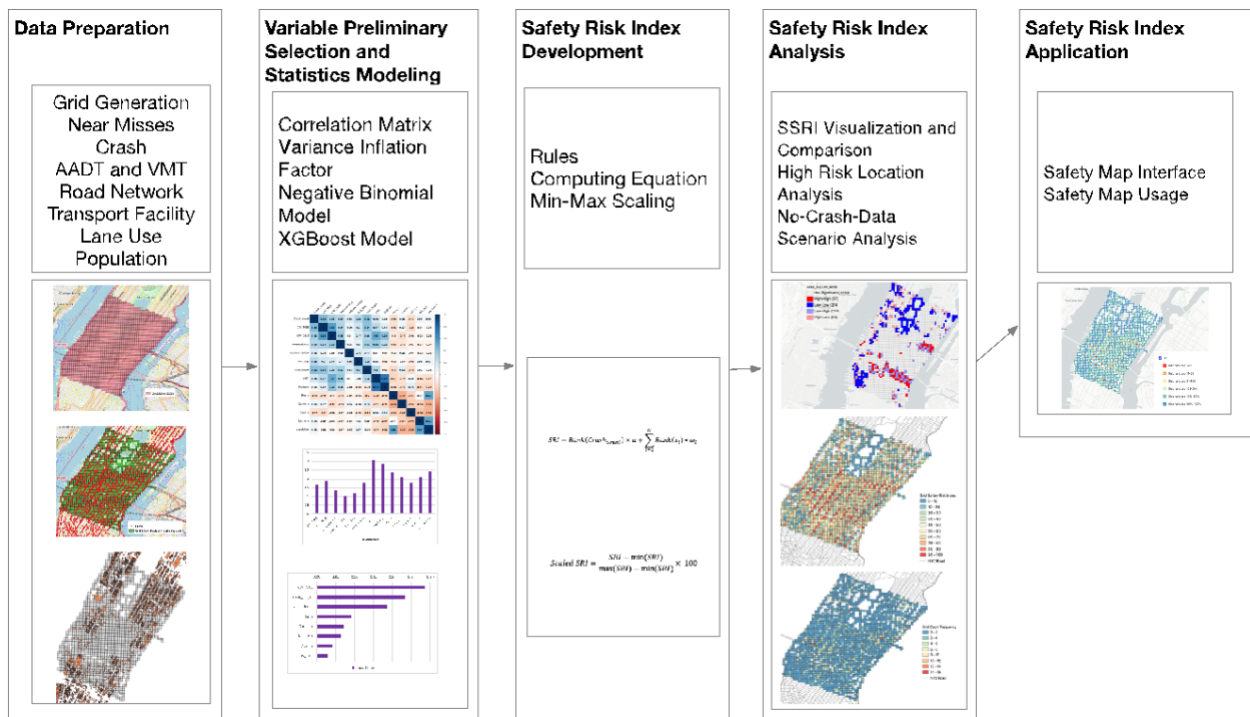


Figure 20. Research and development framework for the safety view map application

4.2 Data Preparation

The grid cell-based framework is a spatial analysis method that divides a geographical area into a grid of uniformly sized and shaped cells. Each cell represents an independent spatial unit used to collect, analyze, and display data. It has been used in various past studies (Gao et al., 2018; Xie et al., 2017a; Xie et al., 2017b) and is an effective method for addressing limitations in traditional spatial analysis

techniques, especially in traffic safety data analysis. Traditional methods often rely on Traffic Analysis Zones (TAZs) or census tracts, which can lead to issues when crashes occur at or near their boundaries. The grid cell-based framework offers several advantages, including improved resolution, as a finer grid allows for more precise identification of crash locations and helps identify problem areas or hotspots that might be overlooked in coarser spatial units. It also eliminates boundary issues by treating each cell as an independent unit that captures crash data within its boundaries. Grid cells are uniform in size and shape, allowing for more consistent comparisons and analysis, while also being easily scalable for analysts to adjust the resolution based on their needs and data availability. Furthermore, grid cells can be easily integrated with other spatial data sources, such as road networks, land use, and demographic information, providing additional context for safety analysis and helping identify potential contributing factors to crashes. Therefore, we chose grid cell-based framework to build the risk map.

4.2.1 Grid Generation

The selected study area for this analysis encompasses midtown Manhattan, specifically between 16th and 89th streets, where Mobileye near-miss data is covered. The study area was uniformly divided into a total of 2,343 cells, each measuring 300 ft by 300 ft (see Figure 21 (a)). This cell size was chosen because it closely aligns with the standard block width in Manhattan (264 ft) and the block length (900 ft) is divisible by 300 ft. By using cells of 300 ft in length, location-specific features can be captured more accurately, allowing for a detailed street-by-street resolution that enhances risk analysis. After cleaning the cells that do not contain any datapoints (e.g., around lakes), the grid map with complete data is shown in Figure 21 (b).



(a) Grid 300ft*300ft



(b) Grid 300ft*300ft with complete data

Figure 21. Grid Generation in the Study Area

4.2.2 Traffic Safety Related Data

Near Misses

The near-miss data is extracted from Mobileye collision warning events that were reported by fleet vehicles equipped with Mobileye after-market Advanced driver assistance systems (ADAS) solution. A Collision Warning (CW) event is an alert generated when a Mobileye-equipped vehicle is on a trajectory to collide with another vehicle, pedestrian, or bicyclist in its path. Mobileye's Collision Warning system is based on the estimated Time to Collision (TTC), a function of velocity and distance. There are three types of collision warnings (see Figure 22) reported: Forward Collision Warning (FCW), Pedestrian Collision Warning (PCW), and Bicyclist Collision Warning (BCW). FCW indicates a potential vehicle-to-vehicle collision, detected up to 80 meters (260 feet) ahead and active for speeds between 1 km/h and 200 km/h (124 mph). TTC threshold for FCW is triggered at 2.7 seconds. Both BCW and PCW involve potential collisions with bicyclists and pedestrians, respectively, detected up to 28 meters (90 feet) ahead and active for speeds between 1 km/h and 50 km/h (31 mph). The TTC threshold for these warnings is 2 seconds.

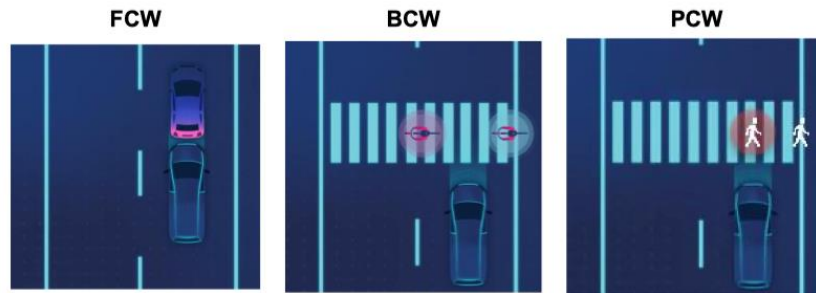


Figure 22. The concepts of FCW, BCW and PCW (Source: Mobileye)

There are two types of Mobileye-equipped vehicles:

- **ME8:** The first type is fleet vehicles with Mobileye 8 connect (ME8) technology. All collision warnings generated from the ME8 vehicles are collected.
- **OEM:** The second type is consumer vehicles with Original Equipment Manufacturer (OEM) Mobileye technology. A subset of Collision Warnings was observed from OEM vehicles, and the default collection rate is set to gathering information from 1 driver per road segment per hour.

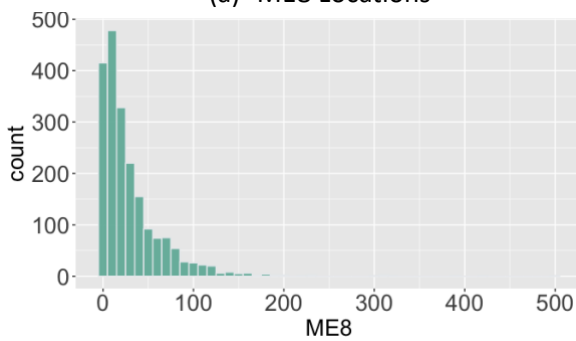
ME8 Near-miss data used in this project covers the period from July 5, 2022, to December 31, 2022. OEM Near-miss data covers the period from August 16, 2022, to December 31, 2022. During this timeframe, a total of 59,277 (ME8) and 2559 (OEM) warnings were generated respectively (Figure 23), offering invaluable insights into potential collision events. In ME8 near-miss data, the vast majority of these warnings were FCW (58,210 events), representing 98.2% of all warnings. PCW and BCW comprised a smaller portion of the dataset, with 711 (1.2%) and 356 (0.6%) warnings respectively. Similarly, OEM near-miss data are consist of 91.8% FCW (2,005 events), 4.0% PCW (88 events) and 4.2% BCW (91 events). For each grid cell, the total count of the three collision warning types (FCW, PCW, and BCW) was calculated to represent the number of near misses. This calculation was performed using the Spatial Join tool in ArcGIS.



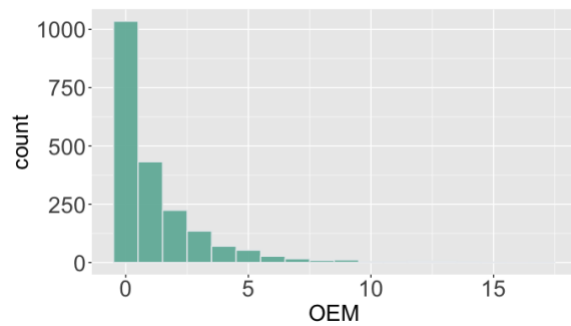
(a) ME8 Locations



(b) OEM Locations



(c) ME8 Histogram



(d) OEM Histogram

Figure 23. ME8 and OEM collision warning locations and distributions

Crash Data

The historical motor vehicle crash data for this analysis was obtained from Open NYC (<https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95>). To maintain consistency with the near-miss data, the crash data was filtered to include incidents occurring between July 5, 2022, and December 31, 2022. Furthermore, any crash data entries lacking coordinate information were removed from the dataset. The crash data is then aggregated to each grid cell.

Crashes can be categorized based on their severity into property damage only (PDO), injury crash, and fatal crash. However, feedback from the advisory board suggests that PDO crashes might be underreported, for instance due to limited resources during the COVID-19 pandemic. Such underreporting

could lead to biased parameter estimates (Donnell et al., 2020). One strategy to mitigate the impact of underreporting on safety risk is to focus solely on severe crashes (i.e., injury and fatal crashes), which are typically less susceptible to underreporting. Consequently, we have separately calculated total crash records and Injury and Fatal Crash (IFC) records.

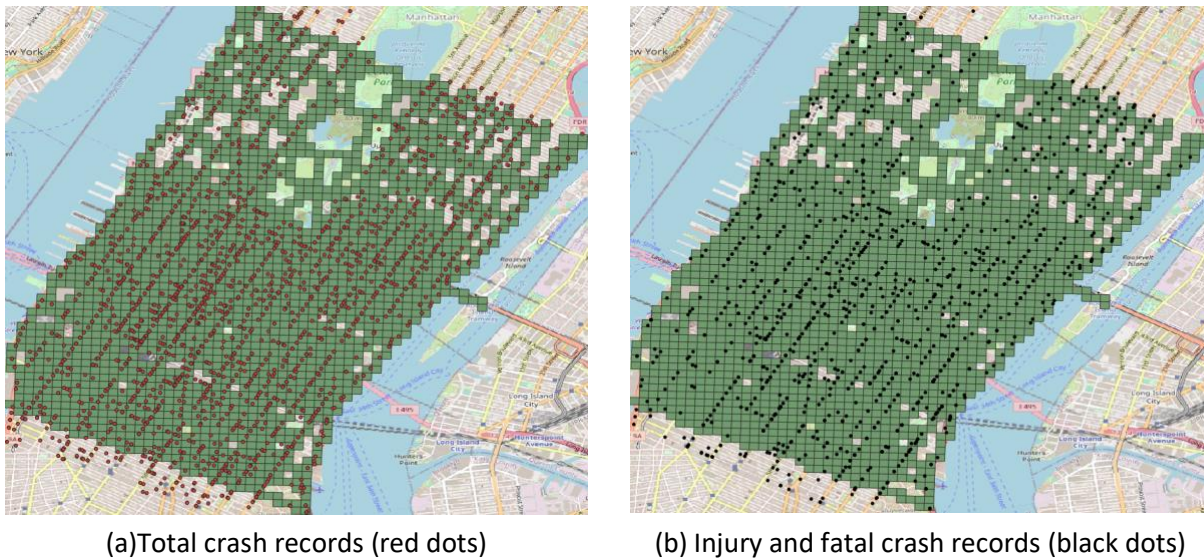
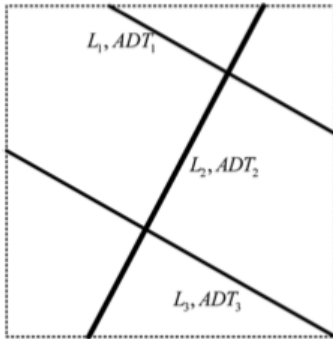


Figure 24. Crash records in New York City during 7/5/2022-12/31/2022

Traffic Exposure Data

In this study, Annual Average Daily Traffic (AADT) and Vehicle Miles Traveled (VMT) data are used as approximations for traffic exposure. The most recent AADT data (2021) was acquired from the NYSDOT Traffic Data View website (<https://www.dot.ny.gov/tdv>). As AADT data is a link-based feature of road networks, we calculated the VMT for each grid cell using this information. First, we identified long links that were not split at intersections and divided them, accordingly, assigning the same AADT value to the resulting segments. Next, we used the road length and AADT data to calculate the VMT. By employing spatial tools in ArcGIS, roadways were divided at the boundaries of each grid cell to obtain the length of individual road segments within each cell. Finally, VMT was determined by summing the products of road segment lengths and their corresponding average daily traffic values (Figure 25).



$$vmt = 365 \times \sum_{j=1}^3 L_j \times ADT_j$$

L_j : The length of the road segment j in the grid cell

ADT_j : The average daily traffic of the road segment j

Figure 25. Demonstration of computing VMT for a grid cell (Xie et al., 2017a)

Road Network

The road network data for New York City was obtained from [Data.gov](https://data.gov). Subsequently, the total road length and the total number of road network nodes within each grid cell were calculated. To compute the total road length, the roads were divided at the grid cell boundaries. We also aggregated the total counts of road network nodes (e.g., intersections) as input. Moreover, a binary highway index was established for each grid cell based on the presence (=1) or absence (=0) of urban highways.

Transport Facility

The total number of bus stops and subway stations per cell are also used as features of safety risk. The two latest shapefiles, showing the NYC bus stops and subway stations on November 2020, are acquired from the website of the Newman Library of Baruch College, CUNY (Newman Library of Baruch College, 2020).

Land Use

Land use data was acquired from NYC Department of City Planning (NYC DCP) Map PLUTO version 2022, v3.1 (<https://www.nyc.gov/site/planning/data-maps/open-data/dwn-pluto-mappluto.page>). Each building class is assigned to the most appropriate land use category. We derived four land use category groups for this study, including residential area (R), commercial area (C), mixed residential and commercial area (Mixed R & C), and open space area (Figure 26). Land use data, represented as polygon features, were utilized to calculate the area percentage of each of the four land use category groups within the grid cells. To accomplish this, the polygons were split at the grid cell boundaries, and the

Spatial Join tool in ArcGIS was employed to determine the percentages of residential area, commercial area, mixed residential and commercial area, and open area within each grid cell.

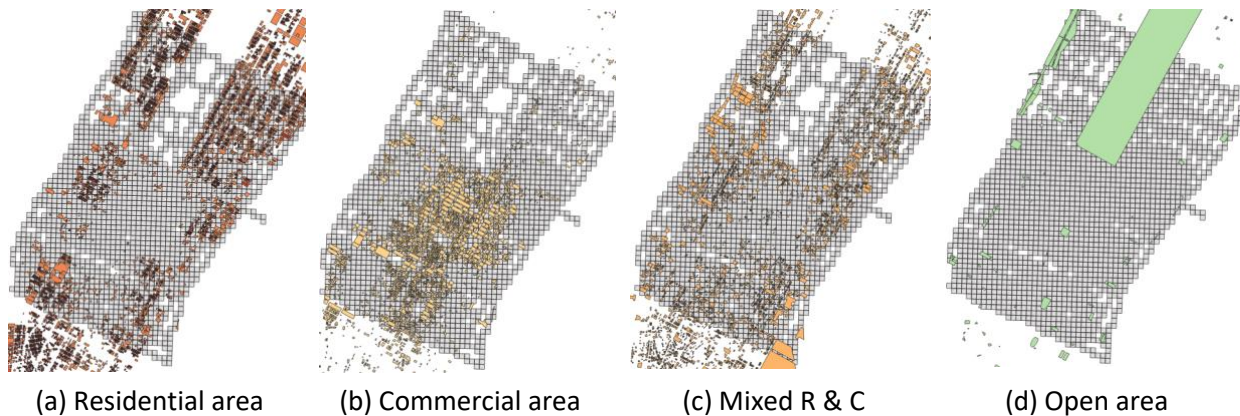


Figure 26. The land use group spatial distribution in the study area

Population Density

Population data was obtained from the U.S. Census Bureau (<https://data.census.gov/>), which offers various aggregation levels, such as census tracts, census tract block groups, and census tract blocks. The most recent data (2020 P1 Race table) aggregated at the census tract block level was used for calculating the population within each grid cell. The calculation method for determining the population in each grid cell assumes a uniform distribution of population within the region. The population within a grid cell originating from a specific census block was then calculated by multiplying the total population by the area ratio of the grid cell.

4.2.3 Variable Summary

The summary of the input variables used for safety analysis, including their mean, standard deviation (S.D.), median, minimum (Min), and maximum (Max) values, is provided in Table 7. The development of the safety risk index will be based on these variables, ensuring a comprehensive understanding of the potential factors influencing safety risk within the study area. These variables are included to cover aspects not only direct results in crashes (e.g., crash records), but also indirectly increase or decrease crash risk, such as traffic exposure, road geometry, and land use.

Table 7 Input Variable Summary

Variable	Mean	S.D.	Median	Min	Max
Crash_tot	1.43	2.02	1	0	15
Crash_fi	0.58	1.06	0	0	8
CW_ME8	30.87	38.30	19	0	504
CW_OEM	1.24	2.01	0	0	17
Intersection	0.93	1.40	1	0	18
Subway_station	0.026	0.17	0	0	2
Bus_stop	0.34	0.65	0	0	4
Road_length (ft)	407.97	176.82	355.22	3.64	1032.05
VMT (mi*veh)	1259	1869	723	0	15777
Highway	0.08	0.27	0	0	1
Res_r	15%	20%	5%	0%	98%
Comm_r	16%	23%	2%	0%	87%
Open_r	9%	26%	0%	0%	100%
Mix_rc_r	14%	16%	8%	0%	91%
Population	259	226	227	0	1367

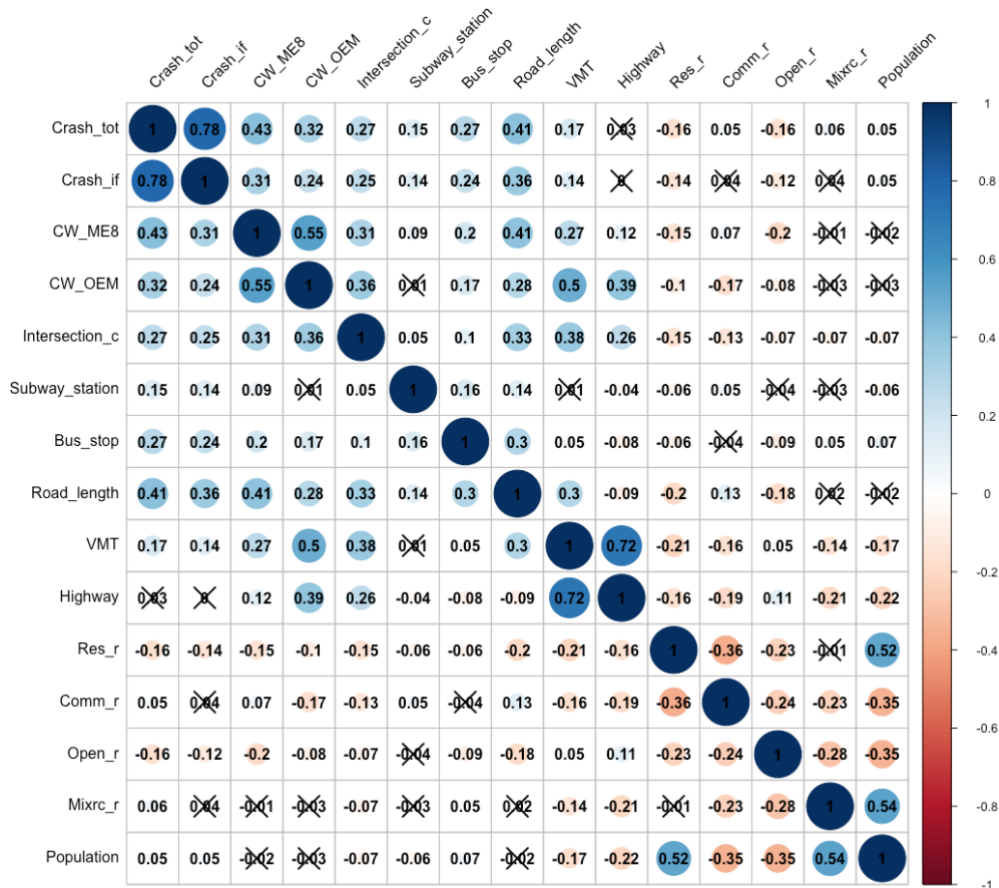
Note: Crash_tot is the total crash count in a grid, Crash_fi is the injury and fatal crash count in a grid, Res_r is the residential area rate in a grid, Comm_r is the commercial area rate in a grid, Open_r is the open area rate in a grid, Mix_rc_r is the mixed residential area and commercial area in a grid.

4.3 Correlation between crash data and near misses

4.3.1 Correlation Coefficient

We first examine the correlation between the variables using correlation matrix, which is a statistical technique used to evaluate the relationship between a pair of variables in a data set. As shown in Figure 27, the correlation coefficients of the total crash counts and other independent variables range from -0.16 to 0.43. Specifically, the correlation coefficients of crash count and near misses (CW_ME8) is 0.43, which is the highest among those correlation coefficients for crash count. The correlation coefficients of

injury and fatal crash counts and other independent variables range from -0.14 to 0.36. Specifically, the correlation coefficients of crash count and near misses (CW_ME8) is 0.31, which is the second highest among those correlation coefficients for crash count. This indicates that the crash count (both total crashes and injury and fatal crashes) and near misses have a moderate positive linear relationship.



Note: X on a number means the correlation coefficient is insignificant. Crash_tot is the total crash count in a grid, Crash_if is the injury and fatal crash count in a grid.

Figure 27. The correlation matrix of the safety related variables

4.3.2 Spatial Correlation

4.3.2.1 Global Bivariate Moran's I

Moreover, the bivariate Moran's I statistics (Moran, 1950) is used to measure spatial correlation between a pair of variables. It assesses whether the value of one variable at one location is associated

with the value of another variable at a neighboring location. The z-score of Moran's I (Z_I) and pseudo p-value (Anselin et al., 2006) obtained from permutation test is used to assess the significance of Moran's I. Z_I can be computed as:

$$Z_I = (I - E[I])/SD[I] \quad (5)$$

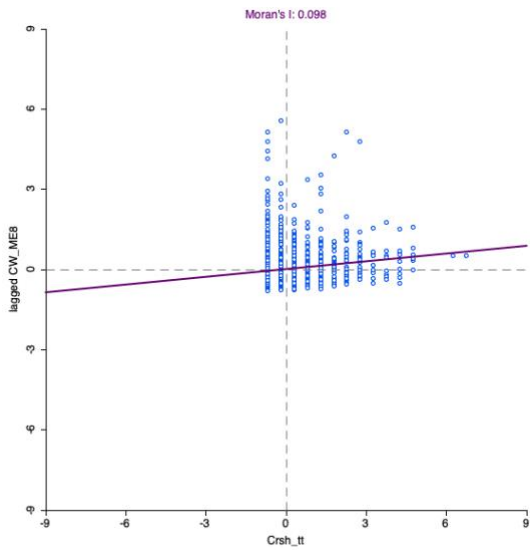
where $E[I]$ is the expectation of I and $SD[I]$ is the standard deviation of I . A positive Z_I indicates the observation distribution is spatially clustered (Xie et al., 2015) and a pseudo p-value less than 0.05 confirms that I is statistically significant at the confidence level of 95% (Goodchild, 1986). More details about Global Moran's I practice can be found in previous studies (Gao et al., 2018; Xie et al., 2015; Xie et al., 2014).

We assume the null hypothesis is complete spatial randomness. The Moran's I test was conducted using several weight matrices, including threshold distances and k-nearest neighbors, on four pairs of variables: 1) crash count and near misses, 2) near misses and crash count, 3) IFC count and near misses, and 4) near misses and IFC count. The goal of testing these pairs is to understand whether a location with a high crash count also experiences a high number of near misses (Pair 1 & 3), and if a location with a high near miss count also has a high crash count in its vicinity (Pair 2 & 4). Near miss data is using the data extracted from ME8. The results of global Moran's I test are presented in Table 8 and Figure 28.

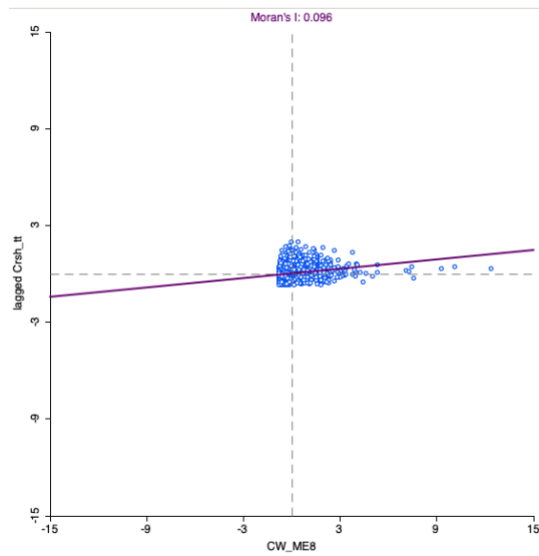
Table 8 Global Moran's I statistics (k-nearest neighbors, k=8).

8-nearest neighbors	I	E(I)	SD(I)	Zi	Pseudo p-value
Total Crash-Near Miss	0.0977	-0.0005	0.0095	10.2509	0.001
Near Miss-Total Crash	0.0956	-0.0005	0.0095	10.0311	0.001
Injury and Fatal Crash -Near Miss	0.0588	-0.0005	0.0090	6.4906	0.001
Near Miss-Injury and Fatal Crash	0.0587	-0.0005	0.0089	6.5677	0.001

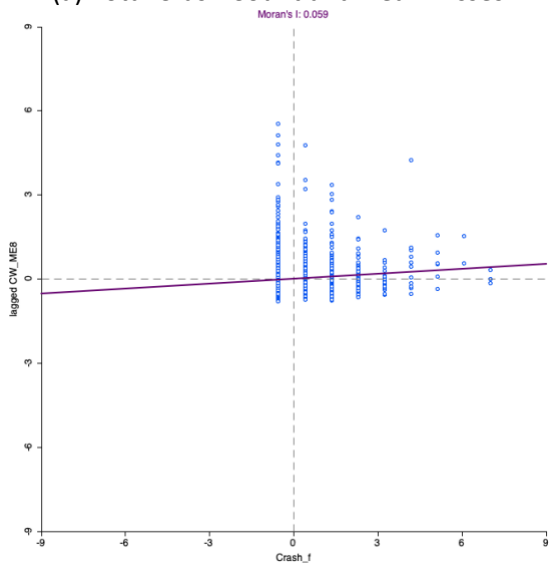
Since all the p-values are statistically significant, and the z-scores are positive using different weight matrix. We can reject the null hypothesis. The results indicate a statistically significant spatial correlation - that is to say, the spatial distribution of high and/or low values in all four pairs is more spatially clustered than would be expected if the underlying spatial processes were random.



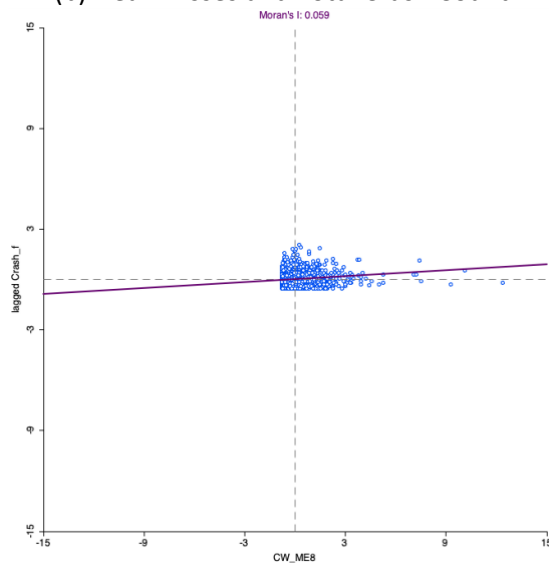
(a) Total Crash Count and Near Misses



(b) Near Misses and Total Crash Count



(c) Injury & Fatal Crash Count and Near Misses



(d) Near Misses and Injury & Fatal Crash Count

Figure 28. Bivariate Moran's I for Near Misses and Crash Counts

4.3.2.2 Bivariate Local Indicators of Spatial Association (BiLISA)

While bivariate Moran's I provides a global measure of spatial correlation, we also investigated bivariate Local Indicators of Spatial Association (BiLISA). This local measure evaluates the spatial correlation between two different variables and can identify specific locations as either spatial clusters (hotspots) or

spatial outliers (cold spots). Local Moran's I for the observation z_i, z_j in cell i, j with weight matrix w_{ij} and N observations (Anselin, 1995) can be computed as:

$$I_i = \frac{z_i}{\left(\frac{\sum_i z_i^2}{N}\right)} \sum_j w_{ij} z_j \quad (6)$$

The bivariate LISA can identify four types of spatial association: (1) High-High, (2) Low-Low, (3) High-Low, and (4) Low-High. High-High and Low-Low are known as spatial clusters. They have a high value for one variable and are surrounded by areas with high values for the other variable or have a low value for one variable and are surrounded by areas with low values for the other variable, respectively. This indicates positive local spatial correlation. High-Low and Low-High are known as spatial outliers. They have a high value for one variable and are surrounded by areas with low values for the other variable or have a low value for one variable and are surrounded by areas with high values for the other variable, respectively. This indicates negative local spatial correlation.

Bivariate LISA (Total Crashes)

Figure 29 presents the total crash related bivariate LISA results for the two pairs “Crash-Near Miss” and “Near Miss-Crash”. In our case, both Bivariate LISA have more clusters with positive local spatial correlation than that with negative local spatial correlation. For positive local spatial correlation locations, in both Bivariate LISAs, the low-low clusters are around the central park and Chelsea area, this can be viewed as low-risk locations. The high-high locations in Crash-Near Miss Bivariate LISA (Figure 29(a)) are mostly near the Queensboro bridge linkage road area and Midtown East between 34th Street and 42nd Street while the high-high locations in Near Miss-Crash Bivariate LISA (Figure 29(b)) are around the Queensboro bridge linkage road area and Midtown East between 34th Street and 42nd Street.

For negative local spatial correlation locations, High-Low clusters in Crash-Near Miss Bivariate LISA (high crash count surrounded by low near miss count) and Low-High clusters in Crash-Near Miss Bivariate LISA (low near miss count surrounded by high crash count) are most concerned. Based on the Bivariate LISA results, near-miss data may not be a good approximation for crashes in the upper west side (west side of Manhattan, near Central Park). Therefore, the use of near-miss data in this area should be approached with caution. The low-high locations in the Near Miss-Crash Bivariate LISA may be of less concern, as most of them are close to high-high clusters, indicating a potential spillover effect.

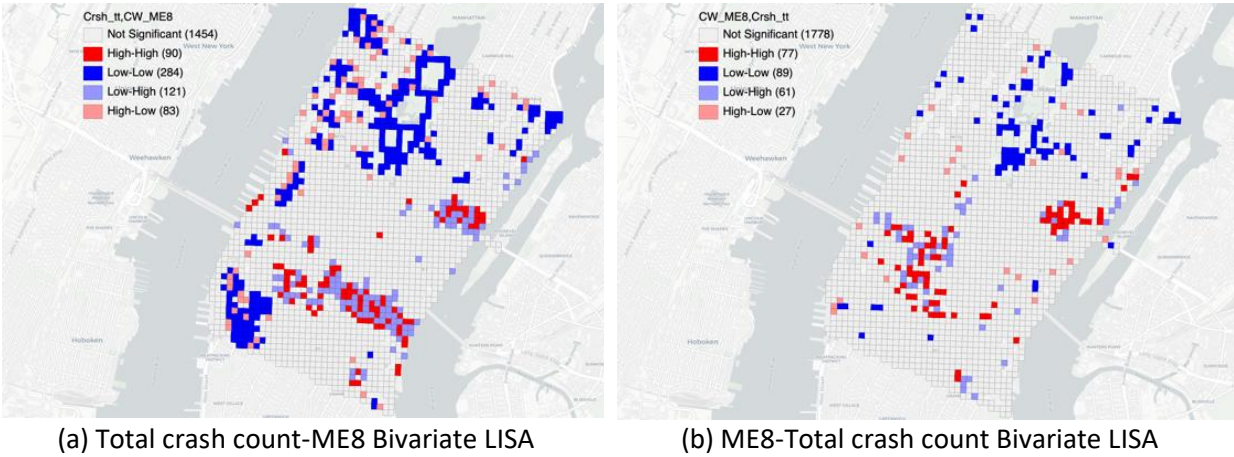


Figure 29. Bivariate LISA Cluster Map using K-nearest neighbor (k=8)

In summary, the High-High clusters (in red) and Low-Low clusters (in blue) in Near Miss-Crash Bivariate LISA Map (Figure 29(b)) are the areas where near-miss data can be reliably used to signify high and low crash risk. Similarly, the High-High and Low-Low clusters on the Crash-Near Miss Bivariate LISA Map (Figure 29(a)) are regions where near-miss data may likewise offer a relatively valuable indication of either high or low crash risk. Nonetheless, the utilization of near-miss data should be carefully considered when applied to the Upper West Side.

Bivariate LISA (IFC)

Figure 30 illustrates the bivariate LISA results for the two pairs “IFC crashes-Near Miss” and “Near Miss-IFC crashes”. In Figure 30 (b), clusters with negative local spatial correlation outnumber those with positive local spatial correlation. The high-high clusters primarily reside in areas such as Queensboro Bridge, 34th Street, and the Penn Station vicinity. Compared with the ME8-Crash Bivariate LISA (Figure 29(b)), the ME8-IFC Bivariate LISA map doesn’t exhibit as many low-low clusters. High-low clusters are spread throughout central Manhattan, with some nestled between FDR and 9A. Low-high clusters are mainly found in the Broadway area between 50th and 55th streets, and in the Kips Bay area.

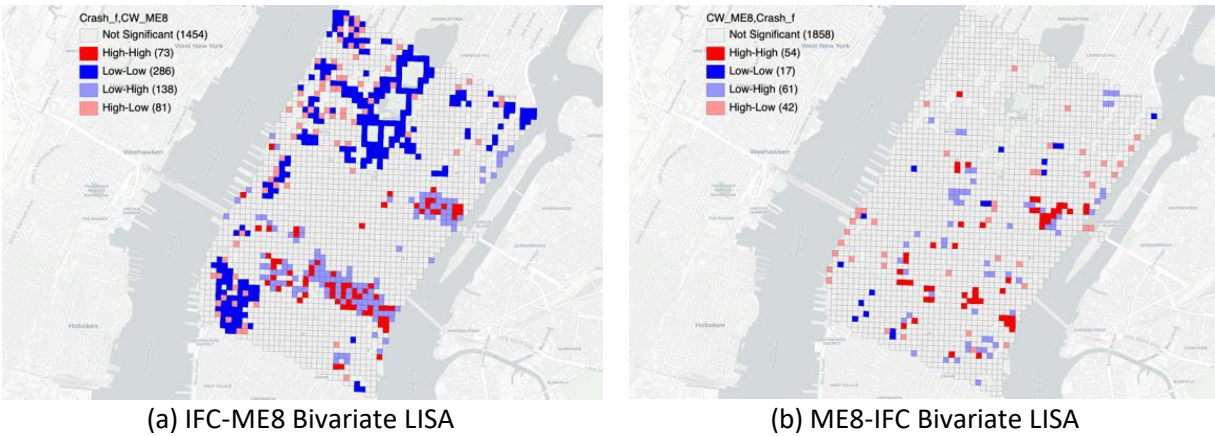


Figure 30. Injury and Fatal Crash related Bivariate LISA cluster map using K-nearest neighbor (k=8)

Generally, when we shift from considering total crashes to IFC crashes, the crash-ME8 bivariate LISA map remains largely unaffected. However, changes are found in the ME8-crash map. These changes might be a result of the decrease in the number of crashes, which could in turn influence the statistical significance of the tests.

4.4 Safety Risk Index Development

In this section, we introduce the method to develop safety risk index using total crash frequency case as an example. The safety risk index using other crash frequency, such as injury and fatal crash frequency, VRU involved crash frequency, can be found in section 4.6 and 4.7.

4.4.1 Variable Preliminary Selection

We applied the correlation matrix (Figure 27) and Variance Inflation Factor (VIF) to identify highly correlated variables. The VIF is a measure that directly quantifies the ratio of the variance in a model that includes all features, compared to the variance in a model that includes only the feature under consideration. Upon examining the correlations between variables, we observed that CW_OEM is highly correlated with both CW_ME8 and VMT. Given that CW_ME8 has a larger sample size and higher correlation coefficients with Crash_count, we have decided to exclude CW_OEM from the safety risk index development. Another variable exhibiting high correlation coefficients is population, which is strongly correlated with Res_r and Mix_rc_r. Therefore, we opted to remove the population from

subsequent steps. Lastly, VMT and Highway demonstrate high correlation with one another. In consideration of the VIF results (refer to Figure 31), where only the VIFs of VMT and Highway exceed 2.5, we have decided to exclude both variables to avoid multicollinearity issues.

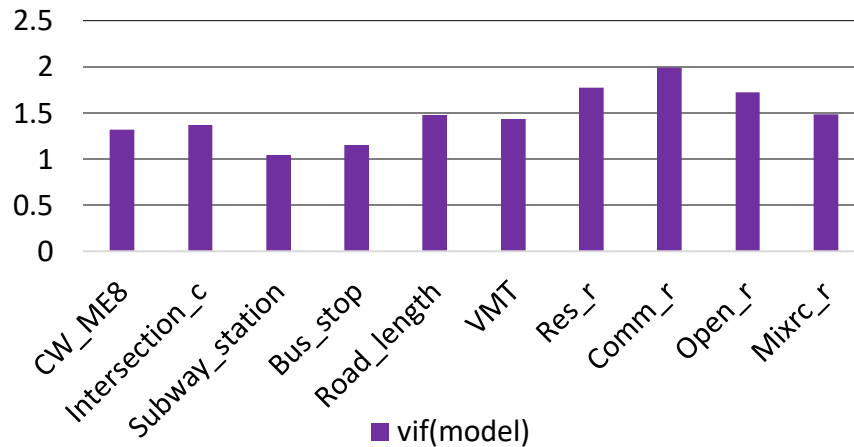


Figure 31. The VIF of each safety risk related variable

4.4.2 Statistics Modeling

To provide a more interpretable analysis and to understand the positive or negative effects of variables on crash frequency, a Negative Binomial (NB) model is employed. The NB model is a widely used statistical method for crash frequency modeling and is an extension of the Poisson model. The Poisson model assumes that the mean and variance of crash count data are equal, but real-world crash data often exhibit overdispersion, with the variance exceeding the mean. To account for this overdispersion, the NB model introduces an additional parameter to capture the unobserved heterogeneity in the data.

Based on the variable selection process, we utilized the preliminarily selected variables to construct an NB model. The StepAIC function in R was employed to identify the optimal variables for input. The resulting best NB model is presented in Table 9.

Table 9 The estimate results of best NB model.

	Estimate	Std. Error	z value	Pr(> z)
Intercept	-0.88	0.09	-9.389	0.00**

CW_ME8	0.01	0.00	10.694	0.00**
Intersection_c	0.12	0.02	6.893	0.00**
Subway_station	0.20	0.14	1.506	0.132
Bus_stop	0.19	0.04	4.878	0.00**
Road_length	0.01	0.00	10.152	0.00**
Res_r	-0.73	0.17	-4.419	0.00**
Open_r	-1.04	0.16	-6.358	0.00**
Mix_rc_r	0.28	0.17	1.587	0.112

Note: ** denotes significance at the 5% level, * denotes significance at the 10% level.

The StepAIC function removed the Comm_r variable, and apart from the insignificant Subway_station variable, all other variables were found to be significant in the NB model. CW_ME8, Intersection_c, Bus_stop, Road_length, and Mix_rc_r have a positive impact on crash frequency, while Res_r and Open_r exhibit a negative influence on crash frequency.

4.4.3 Variable Importance by Machine Learning Method

XGBoost (Chen and Guestrin, 2016), short for eXtreme Gradient Boosting, is an advanced machine learning algorithm based on scalable tree boosting system that has gained significant popularity due to its remarkable performance in various applications. It is an ensemble learning technique that employs the gradient boosting framework to construct a series of decision trees, improving the predictive performance by combining the results of multiple weak learners. The XGBoost model is known for its capability to handle a wide range of data types, scalability, and efficiency in training large datasets (Chen and Guestrin, 2016). In this project, we utilized the XGBoost model to assess the relative importance of the variables in predicting crash frequency. The feature importance results of the variables are shown in Table 10.

Table 10 The feature importance of XGBoost model.

Feature	Gain	Cover	Frequency	Importance
CW_ME8	0.35	0.27	0.26	0.35
Intersection_c	0.27	0.14	0.08	0.27
Road_length	0.23	0.31	0.32	0.23

Res_r	0.09	0.15	0.20	0.09
Open_r	0.04	0.09	0.09	0.04
Bus_stop	0.02	0.04	0.05	0.02
CW_ME8	0.35	0.27	0.26	0.35
Intersection_c	0.27	0.14	0.08	0.27

4.4.4 Safety Risk Index (SRI)

We propose the following six steps to construct the safety risk index.

- *Step 1 - Variable Selection:* Apply variable selection based on correlation matrix and Variance inflation factor (VIF) analysis. Build an optimum Negative Binomial model to link crash frequency to the remaining variables. The significant variables in the optimum Negative Binomial model are the variables used for subsequent SRI development.
- *Step 2 - Utilize rank values instead of raw values:* Leveraging the advantages of robustness and independence from distribution assumptions, we determine to use rank values rather than raw values for each variable.
- *Step 3 - Employ feature importance from XGBoost as weights:* When combining different variables, the weight assigned to each variable is determined by its feature importance score in the XGBoost model, with linear combination as the method of aggregation.
- *Step 4 - Derive the sign of the weight from the estimates of the Negative Binomial model:* As some variables have a positive impact on safety risk while others have a negative impact, we adopt the sign of the weight from the Negative Binomial model when developing the index.
- *Step 5 - Incorporate crash frequency into the index, assigning it a weight equal to α :* Inspired by a previous study (Jiang et al., 2020), we include crash frequency in the safety risk index as historical crash count is still one of the most important indicators for potential crash risk at a certain location.
- *Step 6 – Multiply each variable's rank score by its weight to compute SRI.*

By adhering to these steps, we developed a robust and comprehensive safety risk index that can effectively incorporate the relevant factors in assessing safety risk within the study area. The safety risk index (SRI) can be computed by:

$$SRI = \alpha * Rank(Crash_{count}) + \omega_i \sum_{i=1}^N Rank(x_i) \quad (7)$$

Where Rank(.) is a function that assigns an order value to a variable, arranging them from smallest to largest value, $Crash_{count}$ is the crash frequency in a grid, α is the weight of crash frequency, x_i is the selected variables related to safety risk, ω_i is the weight of each variable, it uses the feature importance score in XGBoost with the sign of the estimate from the NB model. We used $\alpha = 1$ in this study.

4.4.5 Min-Max scaling and Scaled Safety Risk Index (SSRI)

Ultimately, to facilitate comparison and interpretation of the results, we employ a linear scaling technique known as Min-Max scaling to standardize the safety risk index within a 0-100 range. The Min-Max scaled Safety Risk Index (SSRI) is computed using the following formula:

$$Scaled\ SRI\ (SSRI) = \frac{SRI - \min(SRI)}{\max(SRI) - \min(SRI)} \times 100 \quad (8)$$

4.5 Safety Analysis Results and Discussion

We calculated the SSRI based on the proposed methodology by fusing the multiple datasets described in the data preparation section. For better visualization, the SSRI values were divided into ten intervals, with each interval assigned a distinct color. The SSRI effectively combines a variety of information related to traffic safety (crashes, near misses, land use, roadway features, traffic exposure etc.), providing a comprehensive view of the spatial distribution of safety risks, as illustrated in Figure 32.

For comparison purposes, we also develop a visualization using only crash count data (Figure 33). Figure 33 reveals that crash frequency values per cell only span from 0 to 15, with 15 being the maximum. This indicates that crash frequency is constrained to integer values within this range. When trying to understand and interpret cells with identical crash counts, and particularly in identifying underlying contributing factors that might necessitate safety intervention, a data fusion approach becomes critical. Furthermore, the data reveals that the majority of grid cells have a crash frequency within the narrow 0-2 range. Sole reliance on crash frequency data thus limits the granularity of safety risk representation, as the range is insufficiently detailed.

Two patterns can be observed from Figure 32. Firstly, the grids containing linkage roads of bridges and tunnels have higher SSRI values, which aligns with the findings of previous research (Xie et al., 2021). This observation may be attributed to the complexity of the road network at these locations. The complex road network (more merge and diverge at those locations) necessitates more frequent lane

changes by drivers at specific locations, which can lead to an increased likelihood of human errors by drivers. These errors, in turn, may contribute to higher safety risks observed in such areas. In addition, the large disruption of traffic may also increase the likelihood of conflicts among motor vehicles, pedestrians, and bicyclists (Xie et al., 2021). Second, the high-risk index grids are predominantly distributed along the avenues from 6th Ave to 8th Ave. These avenues are characterized by a high density of pedestrian activity, likely due to the presence of commercial establishments, public transit access points, and other urban amenities. The increased pedestrian presence in these areas, combined with vehicular traffic, can lead to a higher potential for conflicts and safety risks.



Figure 32. The spatial distribution of grid-based scaled safety risk index



Figure 33. The spatial distribution of grid-based crash frequency

A significant challenge lies in the fact that crashes are infrequent occurrences, leading to scenarios where crash data may be limited. Therefore, we also constructed a hypothetical scenario where crash data is unavailable, a situation that might occur if the observation period is too short to yield enough crash data, but the near-miss count has sufficient samples. In this scenario, the near miss count becomes the highest weighted variable, and the combination of near miss count, and other information serves the need to indicate the safety risk within a city. Figure 34 illustrates the SSRI spatial distribution using only near miss data (excluding crash counts). It exhibits relatively similar patterns compared to Figure 32, as both capture the grids surrounding the linkage roads of bridges and tunnels, grids along 6th-8th Avenues in midtown, and grids near Penn Station. This suggests that even in the absence of crash data, the safety risk map can still be effective in capturing the primary patterns of safety risk within a city.

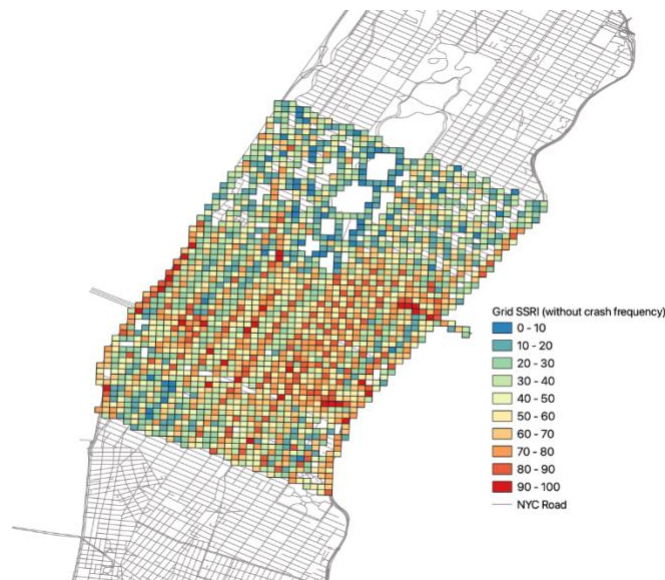


Figure 34. Safety Risk Index Spatial Distribution Using Only Near Miss Data (without Crash Counts)

4.6 Injury and Fatal Crash related SSRI

Given the underreporting issue mentioned in 4.2.2, we have also constructed an Injury and Fatal Crash Related Safety Situation Risk Index (IFCR-SSRI) followed the steps listed in Section 4.4. Table 11 presents the variables that were used to calculate the IFCR-SSRI and their parameters.

Table 11 Injury and Fatal Crash related SSRI Variables and Their Parameters.

Variable	Injury and Fatal Crash	ME8 Near Misses	Road Length	#Intersections	Population
Weight	1	0.24	0.22	0.21	0.13
Sign	+	+	+	+	+
Variable	Residential Land Used Rate	Mixed Residential and Commercial Land Used Rate	#Bus Stops	Open Area Land Used Rate	#Subway Station
Weight	0.08	0.06	0.03	0.03	0.01
Sign	-	+	+	-	+

Similar to the analysis conducted in Section 4.5, Figure 35, Figure 36, and Figure 37 display the distribution of IFCR-SSRI, Injury and Fatal Crash Frequency (IFCF), and the IFCR-SSRI exclusive of IFCF, respectively. Similar implications can be drawn from these three figures: Figure 35 indicates that the grids comprising the linkage roads of bridges and tunnels had higher SSRI values, likely due to the complex nature of the road networks at these locations. In Figure 35, the high-risk index grids are predominantly distributed along the avenues from 6th Ave to 8th Ave, which may be resulted from a high density of pedestrian activity. Figure 36 reveals that IFCR values per cell only range from 0 to 8, with 8 being the maximum. Thus, exclusive reliance on IFCR crashes may limit the detailed representation of safety risk, as the range is insufficiently detailed. Figure 37 illustrates the IFCF-SSRI spatial distribution using only near miss data (excluding crash counts) and shows that with the absence of crash data, the IFCF-SSRI can still be effective in capturing the primary patterns of safety risks.

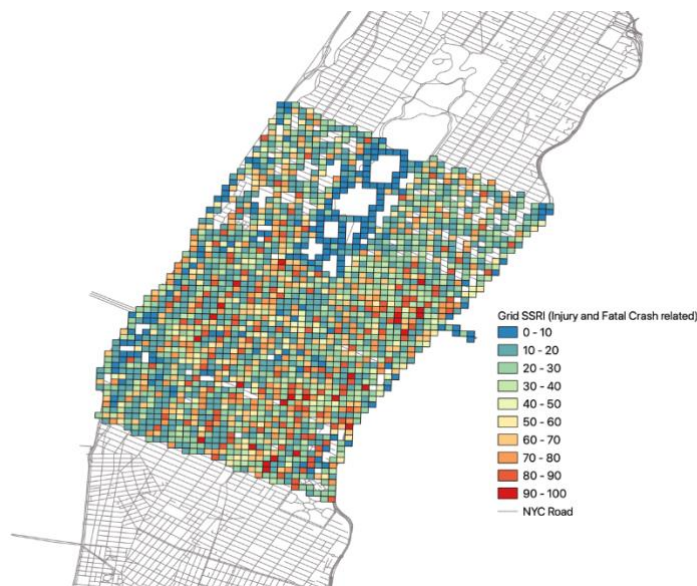


Figure 35. The spatial distribution of grid-based injury and fatal crash related SSRI

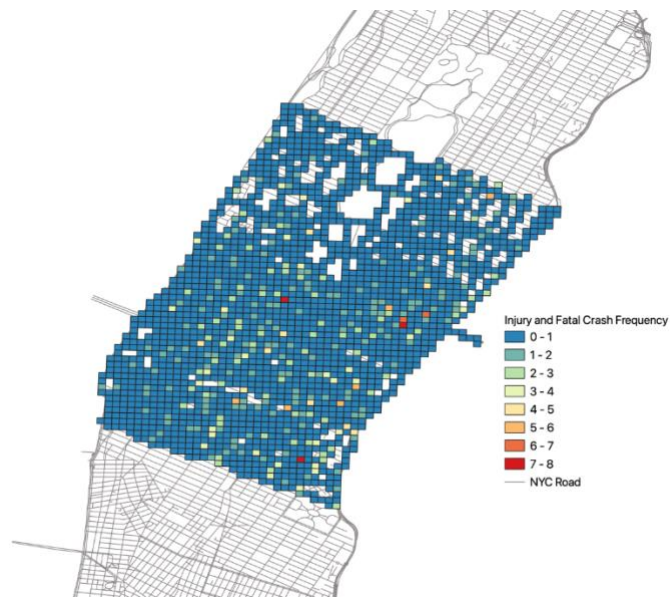


Figure 36. The spatial distribution of grid-based injury and fatal crash frequency

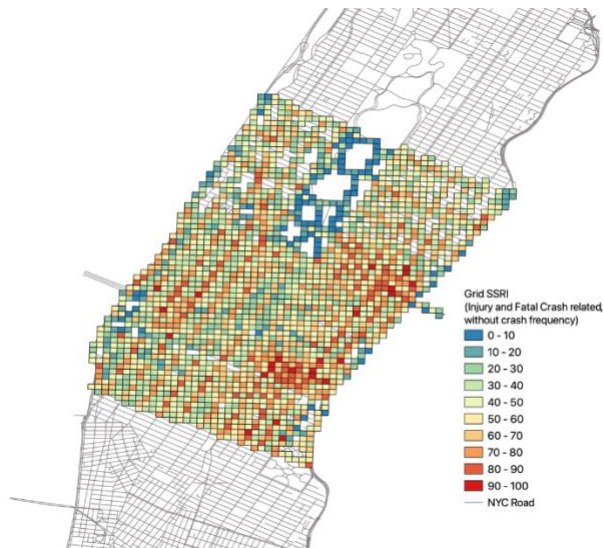


Figure 37. Injury and fatal crash related scaled safety risk Index spatial distribution using only near miss data (without crash counts)

4.7 VRU SSRI

We also constructed VRU SSRI using pedestrian and cyclist data. However, due to less VRU near miss events identified by the crowd sourced CAVs, the usage of VRU SSRI may be limited. Details of the VRU SSRI can be found in the Appendix.

4.8 Limitation and Conclusion

By leveraging the power of multisource data, statistical modeling, and machine learning techniques, the proposed SSRI can be used as a safety risk index for traffic planning. It provides a more comprehensive and accurate representation of the confounding factors influencing traffic safety in a complex urban network. This index can potentially be used by decision-makers to prioritize strategies, allocate resources efficiently, and assess the impact of their implemented measures on enhancing traffic safety.

The proposed approach has certain limitations. While the initial intent was to include both time variant and static variables to construct the SSRI, we ultimately only incorporated crash frequency and CW_ME8 as dynamic variables, which can be updated at a relatively high frequency. Other directly related safety variables, such as speeding tickets, are not sufficiently prevalent in the area we tested. Other variables like land use, road network, and traffic facility locations are considered static variables, which are typically updated less frequently. Currently, the SSRI is employed as a one-time index representing the safety risks associated with the study period. In the future, we plan to implement it as a time-dependent algorithm that will adjust based on the user-selected study period. We also aim to expand its application to encompass the entire city, so that other time variant variables such as speeding tickets can be incorporated.

Section 5 Web-based Applications

5.1 Web Application Architecture

As a part of this project, two applications were developed: 1) WorkZoneX, a web-based application using computer vision and publicly available traffic camera for urban work zone detection in NYC; and 2) SAFExMAP, a web-based application that provides data visualization and spatiotemporal analysis of various safety-related data in NYC. This includes crash records, near-misses, and other road hazards detected by computer vision techniques via in-vehicle cameras, as well as speeding tickets.

Both applications are embedded in an all-in-one web-based platform called “Urban Intellivision” hosted on a C2SMART server. The work zone application works by capturing live CCTV camera images, detecting work zones within the image data, performing analysis on the detection results, and presenting live statistics on the web page. The safety view map filters traffic data stored in our on-premise databases and utilizes deck.gl to render the data as different map layers for comparison and aggregation. As different modules may be implemented using different frameworks and programming languages (such as TypeScript for the front-end, Java for the back-end, and Python for the detection algorithm), their interaction can be complicated, leading to high coupling when using a monolithic system. To address this issue, we have adopted a microservices architecture (Figure 38), which separates different functions and makes the system easier to develop, test, and maintain.

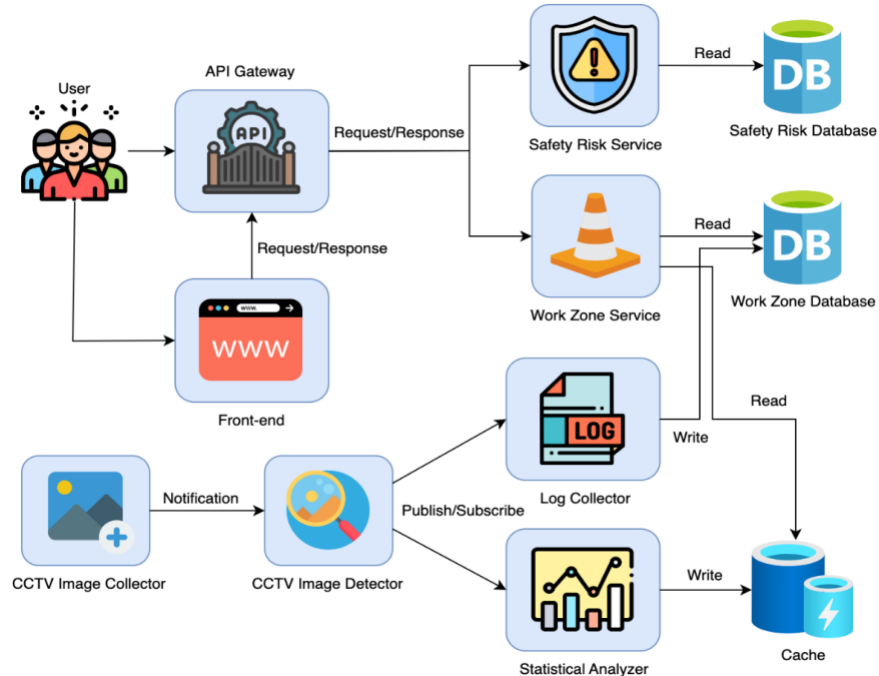


Figure 38. Microservice Architecture

The microservice architecture includes several distinct components, such as a collector that regularly retrieves the most recent CCTV images (every fifteen minutes), a detector that applies computer vision algorithms to identify work zone objects in the images and determine the existence of work zones, a log collector and a statistical analyzer that gather and evaluate the detection results. Additionally, we have a work zone service, a safety risk service, a front-end, and an API gateway, which offer web services.

5.1.1 Inter-process communication

In the architecture described above, each service is implemented as an individual process. As a result, the services must communicate with each other using an inter-process communication (IPC) mechanism. When selecting an IPC mechanism for a service, it's critical to consider how the services will interact. There are various interaction styles between clients and servers that we are utilizing.

Table 12 Inter-Process Communication Methods.

	One-to-one	One-to-many
Synchronous	Request/Response (REST API)	N/A
Asynchronous	Notification (RabbitMQ)	Publish/Subscribe (RabbitMQ)

Synchronous one-to-one communication refers to direct and immediate information exchange between two parties, in a request-response pattern. REST APIs are commonly used (Polák and Holubová, 2015) to implement this kind of communication, where a REST API request is sent from the front-end to the back-end to retrieve the corresponding traffic data and show it on the page. On the other hand, asynchronous one-to-one communication involves a delay in exchanging information between two parties, where both the sender and the receiver can continue with their processing without blocking each other. For example, when the image collector acquires a certain number of CCTV images and intends to transfer them to the image detector for detection, it doesn't want to wait for the detection results to halt image collection. At the same time, the image detector may also not be available to receive the request. To overcome this issue, the image collector delivers the metadata of the images that need to be detected as notifications in a message queue called RabbitMQ. The image detector then retrieves this metadata from the queue to perform detection when it is ready, ensuring real-time detection to the fullest extent. Asynchronous one-to-many communication involves a sender transmitting a message to multiple recipients, and the sender does not hold up its own processing to wait for responses from any recipients. This pattern is useful when multiple services need the detection results to collect logs or calculate corresponding statistics after image detection is finished. To accomplish this feature, a

publish/subscribe architecture is used, where the publisher sends messages to RabbitMQ once and all subscribed recipients receive and process them independently.

5.1.2 Automated pipeline

In addition to automating data flow, our application deployment process is also fully automated through a pipeline (Figure 39). In a manual deployment process, we would have to manually upload the code to the server and then log in to execute the appropriate build and deployment operations. This manual process is both time-consuming and error-prone. However, with our automated pipeline, the code is uploaded to a private Git repository, and Jenkins automatically recognizes this action and pulls the code to the server. From there, it runs the necessary build and test scripts, uploads the resulting Docker image to our self-hosted Docker Registry, and finally retrieves the latest version from the Registry for deployment. This automated deployment process not only improves the application's reliability and deployment efficiency but also provides backups for each version, which can be used in case of version rollback situations.

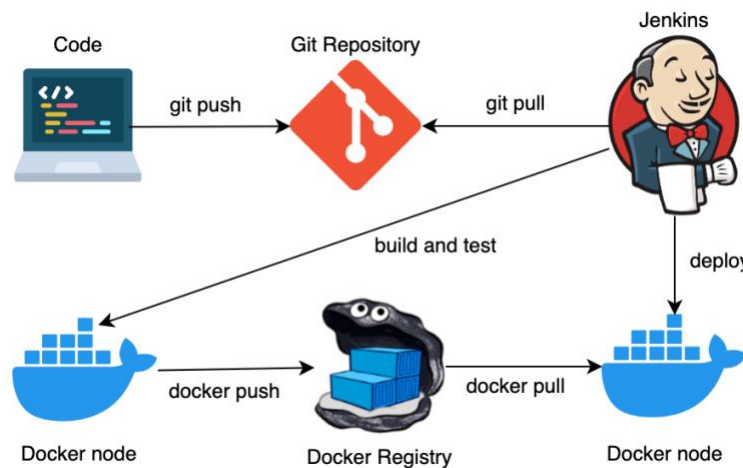


Figure 39. Automated pipeline

5.1.3 Indexing and caching

We are utilizing indexing and caching to boost query performance in our web application. Indexing refers to a database technique that pre-sorts the values of one or more attributes using a data structure. By leveraging an index, the database system can quickly locate the relevant records, avoiding the need to scan the entire table and significantly improving query speed. On the other hand, caching is a data

storage technique that involves storing data in a high-speed medium, such as memory or a fast disk, to enable faster retrieval and updating, thereby enhancing the application's overall performance.

SAFExMAP uses several static data sets, including traffic crashes and speeding tickets, which consist of millions of records. Performing full table scans on these data sets would lead to significant delays and negatively impact the user experience. To address this issue, we created indexes on primary query keywords, such as timestamps, to enhance search performance. The table below showcases the time required to query one month of data in each table, with and without indexes:

Table 13 Query Speed Comparison with and without Indexing.

Table	Query speed (without index)	Query speed (with index)
ME8	895 ms	88 ms
OEM	812 ms	89 ms
Traffic crash	2.4 s	275 ms
Speeding ticket	4.8 s	510 ms

The results demonstrate that indexes yield a performance boost of almost ten times. Additionally, as the data tables are typically read-only, we can avoid the additional cost of index maintenance.

The WorkZoneX Dashboard requires real-time statistical analysis data, which includes obtaining the latest count of work zones in each borough, the current duration of work zones, the total number of work zones in the past four hours, and the duration of work zones with the end time at the previous hour. However, retrieving this analysis data through database queries can be a cumbersome process, as it may involve large volumes of data or multiple queries. Considering that the calculation of statistics is based on the results of image detection, which is cyclical in nature. Therefore, as an alternative, it is more efficient to calculate the results of the statistics in the backend as triggered by the completion of image detection and put them in the cache for backend service calls.

5.1.4 List virtualization

The data source we are using includes data from close to 1,000 CCTV cameras in New York City, and we need to display an equal number of detected images in WorkZoneX during each time period in the form

of an image gallery. However, the traditional front-end rendering approach loads all the images simultaneously, which is inefficient due to the corresponding loading of DOM elements and network requests for each image component. Furthermore, this rendering approach is unnecessary, as the user cannot possibly view so many images at once. To enhance front-end rendering efficiency, we utilized a list virtualization optimization pattern for efficiently displaying large data lists (Figure 40). This method involves virtualizing a list of items by maintaining a window and moving it around the list to display only items visible to the user. As a result, the page only renders a few image components initially and loads more data in real-time as the user slides through the list. This method significantly optimized the page load speed from 8 seconds to 1 second, an 87.5% improvement in speed.

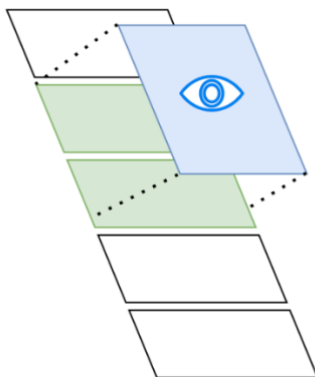


Figure 40. List virtualization with a view window

5.1.5 Multi-threading detection

Detecting work zones can be a slow task, as it typically requires significant system resources. To speed up the computation of work zone detection algorithms, there are various methods that can be applied, including upgrading hardware, compressing models, and utilizing distributed computing. One effective way to make the most of computational resources is to use multiple CPU cores for multi-threading parallel computation. To balance the trade-off between image detection throughput and average latency, we have found it effective to assign the task of detecting 100 images to a single thread for execution. However, it's important to note that using more threads doesn't necessarily lead to better computational efficiency, as the overhead of executing the thread lifecycle and methods gradually increases with the number of threads used, even if CPU utilization may also increase. Therefore, it's crucial to carefully decide the number of threads used in order to achieve optimal performance. We conducted a performance evaluation for one timestamp of all the CCTV image data by recording the

running time of the work zone detection algorithm with varying numbers of threads, which generated the following curve. The graph indicates that the optimal performance is achieved when the number of threads is set to 5, after which it stabilizes. Hence, we have determined 5 to be the ideal number of threads to be employed.

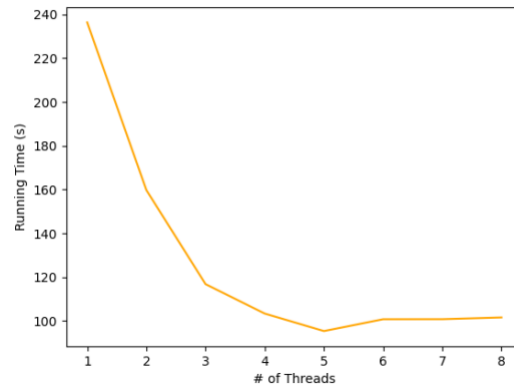


Figure 41. Multi-threading algorithm performance evaluation

5.2 Urban Work Zone Web-Based Application

5.2.1 WorkZoneX Interface

The urban work zone web-based application includes two sub-boards. The first one (Figure 42) features an interactive map displaying real-time detected work zones alongside the original CCTV camera images and images with detection output (e.g., bounding boxes of detected work zone-related objects). The detected work zones are also presented in a list format for ease of navigation.

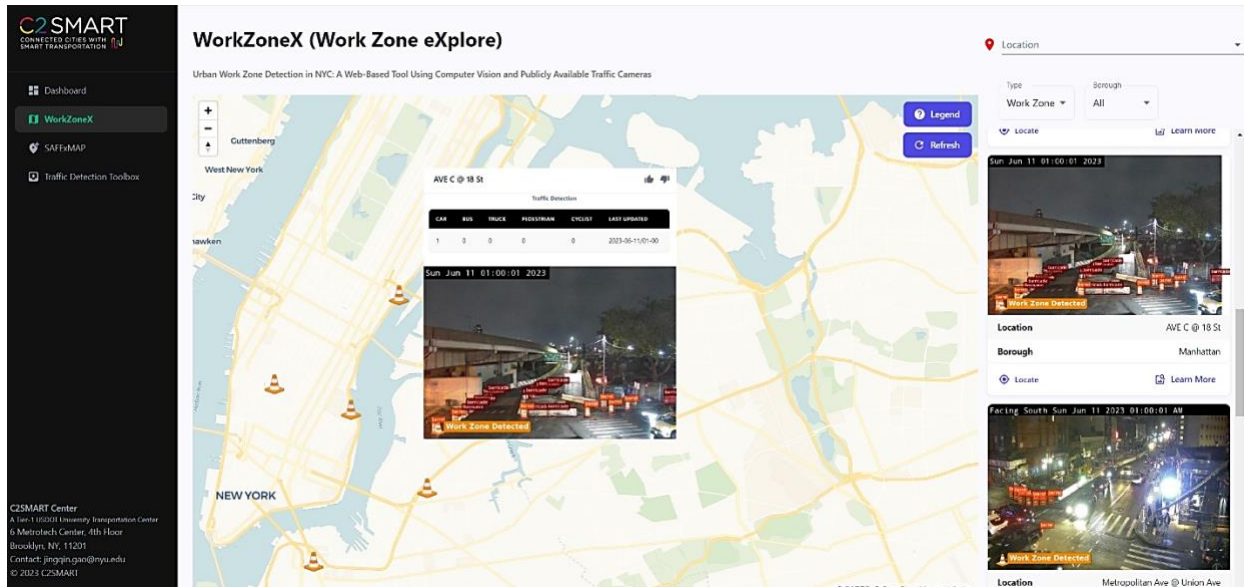


Figure 42. WorkZoneX interface

The second sub-board is a data dashboard that provides both disaggregated and aggregated statistics of detected work zones in real time (Figure 43). Historical data dating back to February 2023 is also available for download.

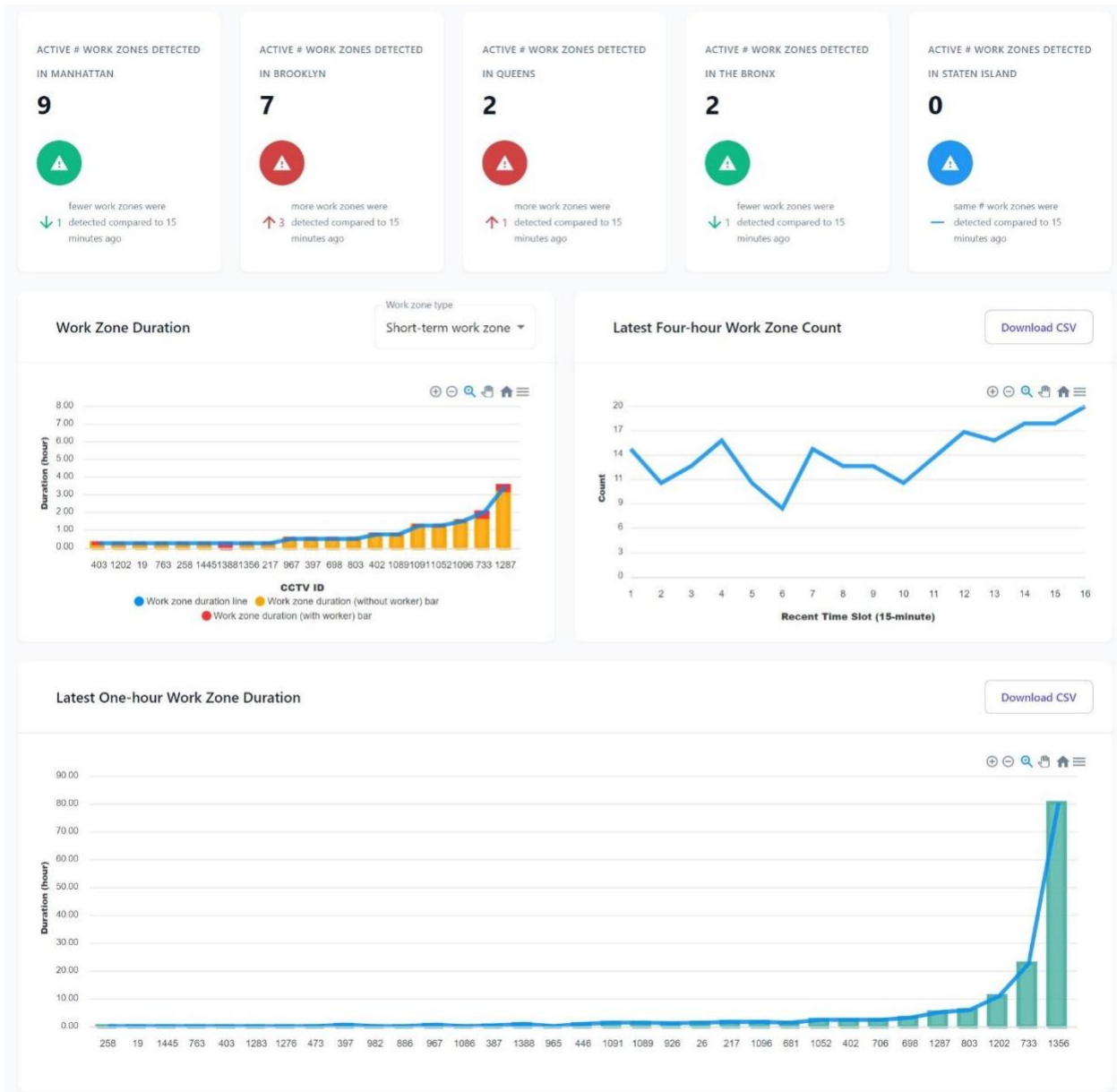


Figure 43. WorkZoneX dashboard interface

5.2.2 WorkZoneX Functionalities

The main features of the WorkZoneX app include the following:

- Highlighting locations where work zones have been detected

- Highlighting locations with active work zones, including the presence of construction workers
- Displaying bounding boxes of detected work zone objects
- Displaying traffic volume by type around detected work zones
- Displaying interactive charts that showcase disaggregated or network-level work zone statistics, including the number of work zones per borough over time, work zone durations, and work zone durations with the presence of construction workers
- Providing download functions for historical work zone statistics

Highlighting locations where work zones or construction workers have been detected

The interactive map parses images from publicly available CCTV cameras operated by NYCDOT every 15 minutes. Clusters of all the cameras are shown in the default setting and individual cameras can be viewed when zooming in (Figure 44). The developed detection algorithms are embedded and run in the backend in real time. A filter is included in the interactive map that can filter cameras with a work zone detected, and cameras with an active work zone with the presence of construction workers.

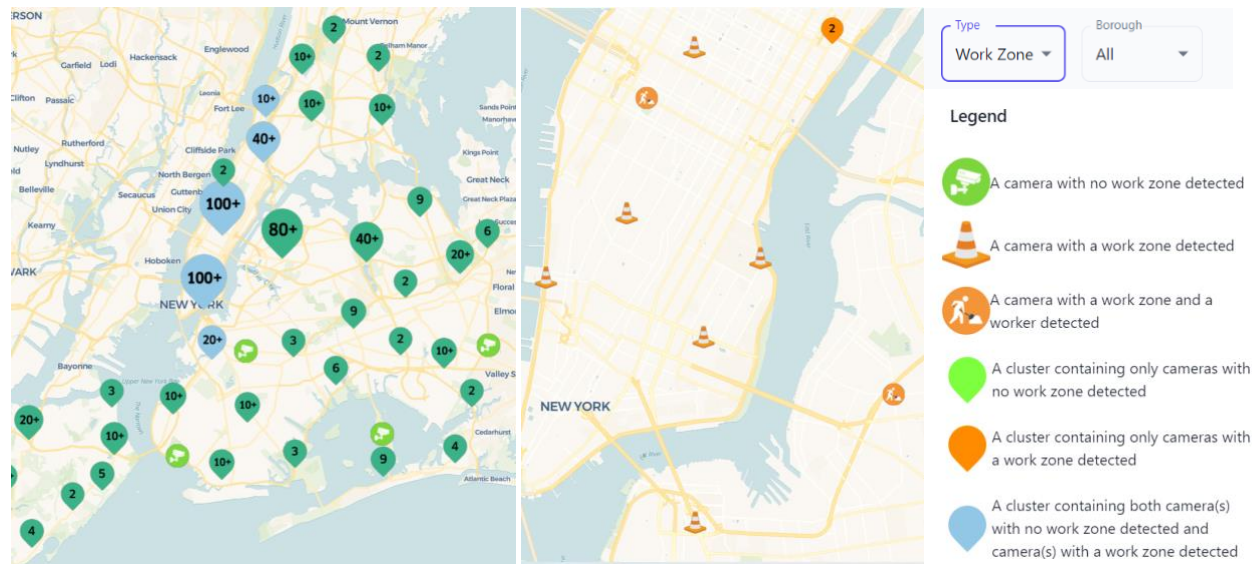
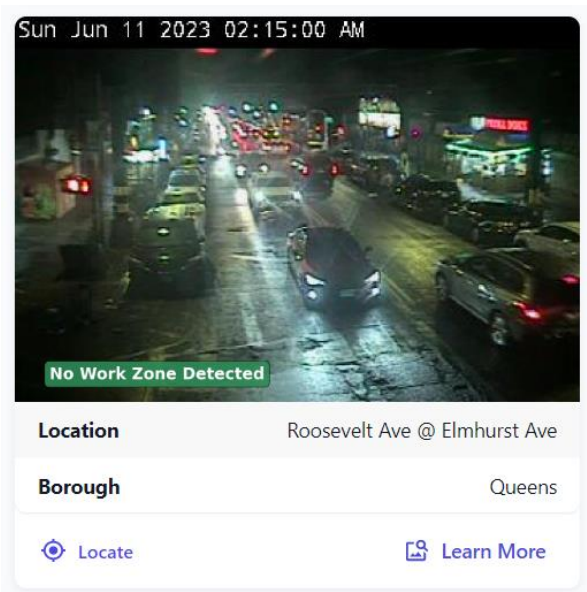
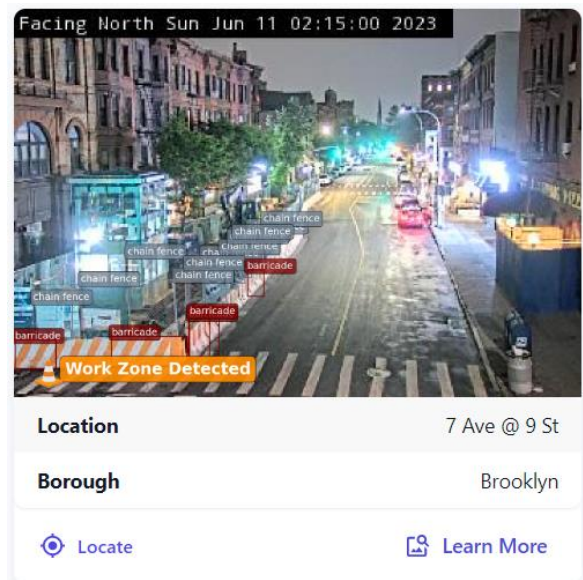


Figure 44. WorkZoneX: Map components and filters

If no work zone is detected, a green label stating "No Work Zone Detected" will appear at the bottom left of the camera image (Figure 45). Conversely, if a work zone is detected, an orange label stating "Work Zone Detected," accompanied by a small traffic cone icon, will be displayed at the bottom left of the camera image.



(a) A non-work zone (green label)



(b) A detected work zone (orange label)

Figure 45. WorkZoneX: Labels of non-work zone and detected work zone

Displaying bounding box of work zone objects & traffic volume around the detected work zones

Within each detected work zone, we provide a "Locate" function and a "Learn More" function. The "Locate" function automatically centers the map on the detected work zone that the user is viewing and displays the original CCTV image without detection bounding boxes (Figure 46). The "Learn More" function offers an enlarged image featuring the bounding boxes of detected work zone objects, along with a snapshot of traffic detection by type (car, bus, truck, pedestrian, cyclist) for the detected work zone at the timestamp indicated in the table (a timestamp within the most recent 15 minutes). A voting feature, with thumbs up and down options, is also included as shown in Figure 46. This allows users to vote, and the results of these votes can be utilized for future improvement.

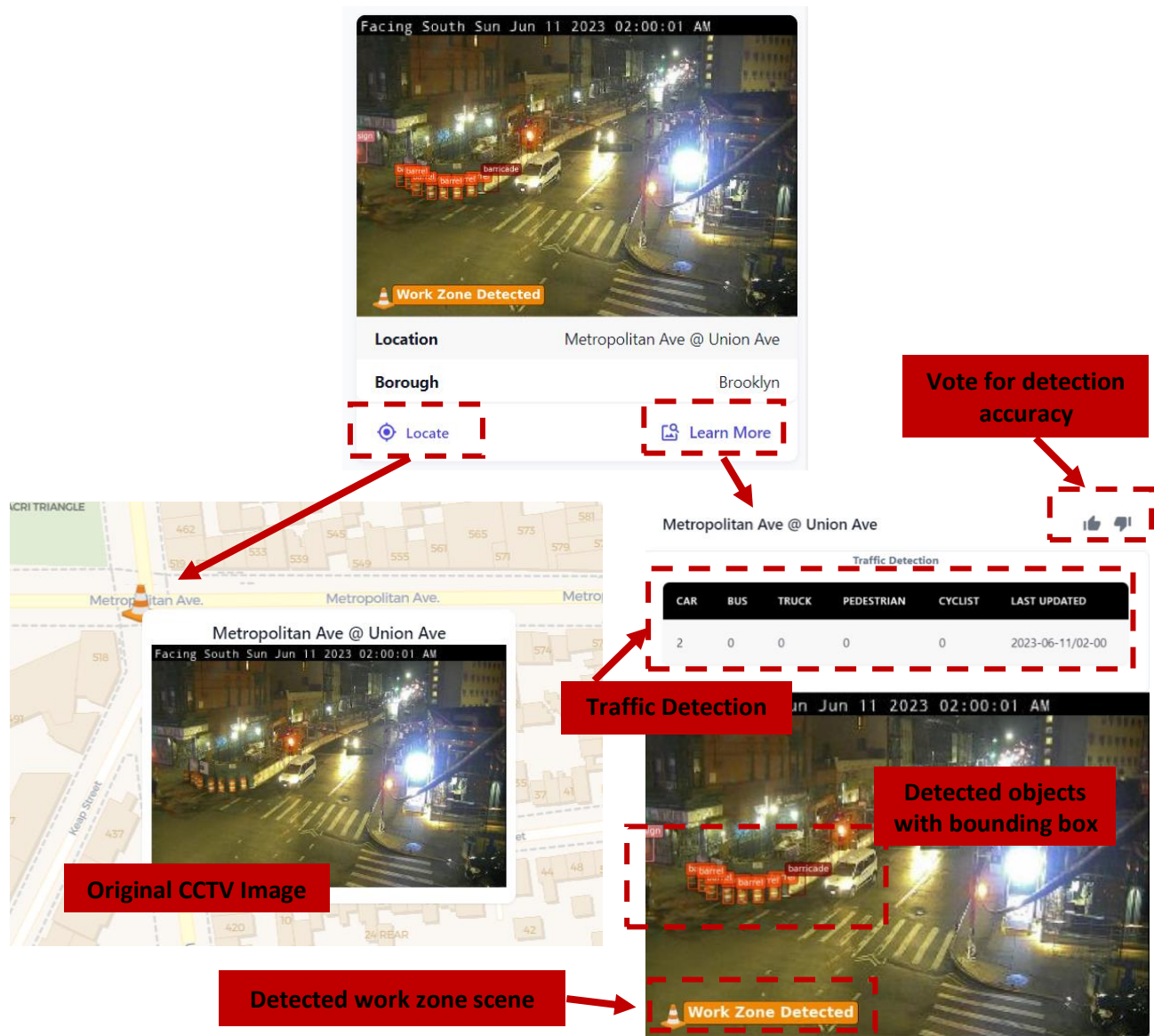


Figure 46. WorkZoneX: Locate and Learn More functions

Displaying interactive charts of work zone statistics & Provide download functions for historical data

Several interactive charts of work zone statistics are provided in the Dashboard sub-board. The top of the Dashboard displays button-like card that shows the number of detected work zones in each borough in real time and provides a comparison of the number with the number of work zones in the previous 15 minutes (Figure 47).

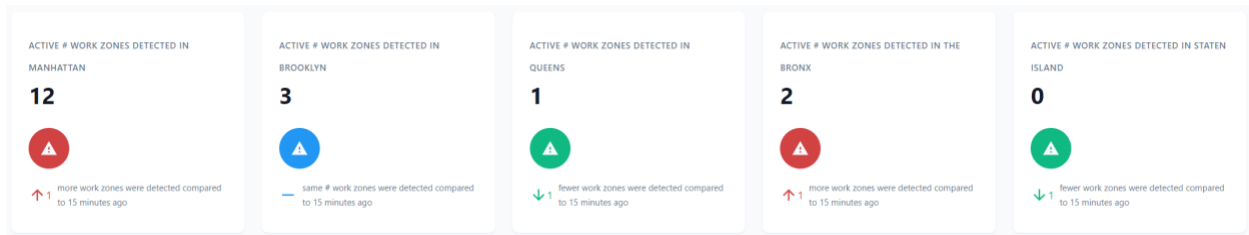


Figure 47. WorkZoneX: Number of work zones in each borough

The second interactive chart presents the duration of individual work zones. As shown in Figure 48, we provide separate sub-charts for two types of work zones: 1) short-term work zones (those with a duration less than 24 hours), and 2) intermediate/long-term work zones (those with a duration longer than 24 hours). For each work zone displayed in the chart, its location, CCTV ID, and work zone duration both with and without the presence of construction workers are shown.



Figure 48. WorkZoneX: Work zone duration chart for individual work zones

Figure 49 presents the two additional charts that provide temporal distributions on both the latest four-hour work zone count and the latest one-hour work zone duration. These two charts also allow for the download of all historical data since the launch of the web-based platform's first prototype in February 2023.



Figure 49. WorkZoneX: Historical work zone statistics and download function

5.3 Safety View Map Application

5.3.1 SAFExMAP Interface

The SAFExMAP provides data visualization and spatiotemporal analysis of various safety-related data in NYC. This includes crash records, near-misses, and other road hazards detected by computer vision techniques via in-vehicle cameras (provided by our industry partner, Mobileye), as well as speeding tickets. The interface of SAFExMAP is shown in Figure 50. The platform is divided into two modules: 1) Data Visualization (Data Viz), and 2) Spatiotemporal Analysis. The Data Visualization module facilitates the display of different types of data in a layered format and provides a variety of visualization options, including circle tile grid maps, scatter plots, polyline visualization, and 2D and 3D heatmaps. The Spatiotemporal Analysis module allows users to create customized polygon selections for multi-data analysis, considering both temporal and spatial contexts.

SAFExMAP

DATA VIZ

SPATIOTEMPORAL ANALYSIS

The **SAFExMAP** provides data visualization and spatiotemporal analysis of various safety-related data in NYC. This includes crash records, near-misses, and other road hazards detected by computer vision techniques via in-vehicle cameras (provided by our industry partner, Mobileye), as well as speeding tickets.

The **Scaled Safety Risk Index (SSRI)** is calculated using a method that combines sociodemographic data, crash records, and near-misses identified by computer vision. A higher rank (e.g., Rank 1) indicates a higher safety risk in the region.

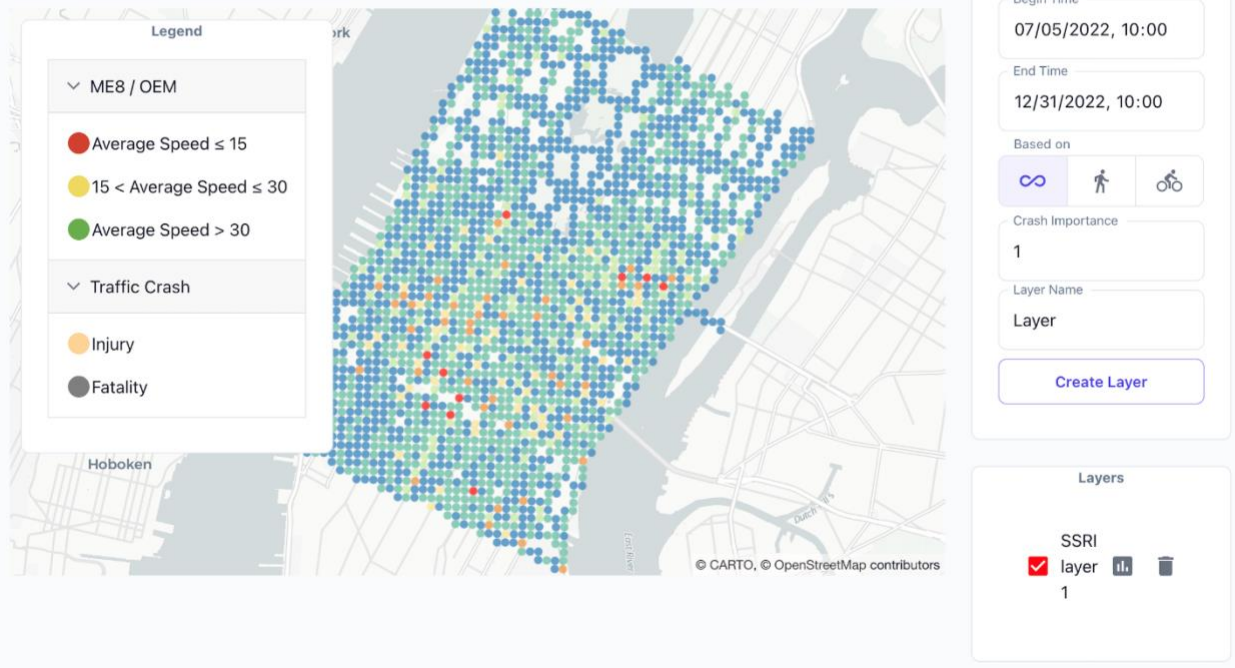


Figure 50. SAFExMAP interface

5.3.2 SAFExMAP functionalities

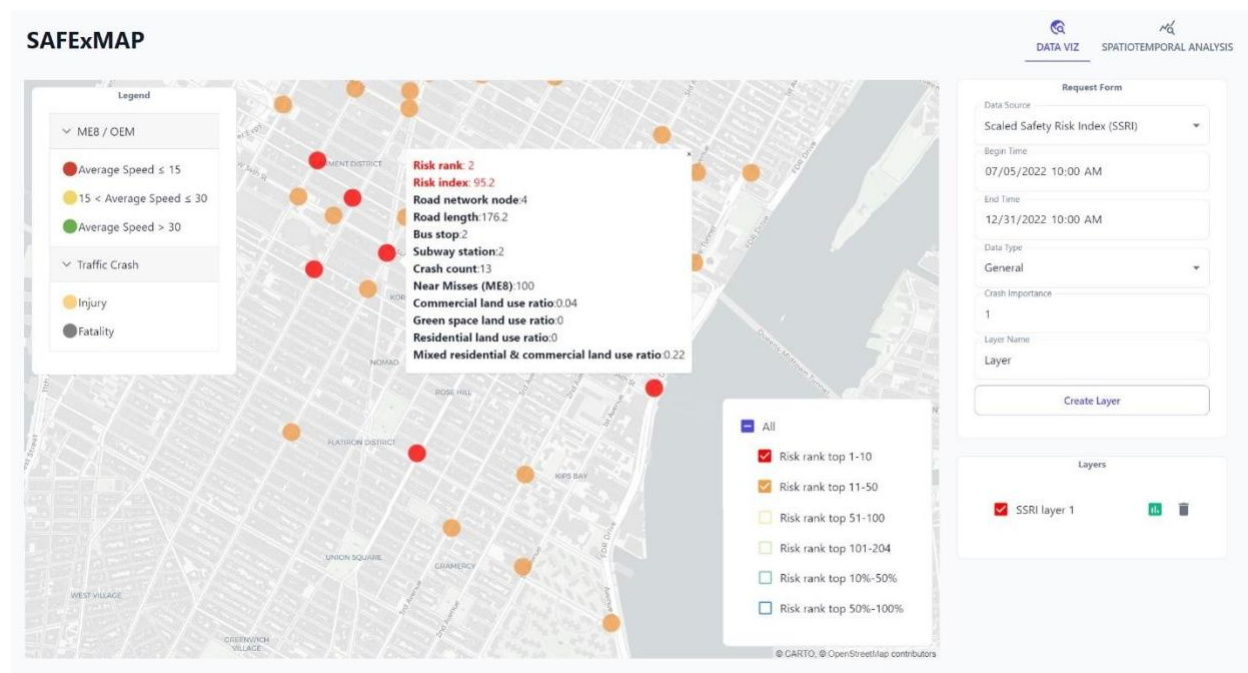
The main features of the SAFExMAP app include the following:

- Displaying SSRI, SSRI rank, and component variables in a circle tile grid map
- Providing a pairwise comparison table of user-selected locations on SSRI and its data-fusion based inputs
- Displaying multi-source data in a layered format, facilitating a spatial understanding of traffic safety-related data
- Offering various visualization options, such as scatter plots, bubble maps, 2D and 3D heatmaps, and bar charts
- Allowing for customized user input regarding the study time period

- Conducting spatiotemporal analysis that permits user-defined polygon selections for multi-data analysis

Displaying SSRI, SSRI rank, and component variables in a circle tile grid map

As demonstrated in Figure 50, a circle tile grid map is used to show the grid-based SSRI results. Given that SSRI results are generated for all crashes/near misses, pedestrian-only crashes/near misses, and cyclist-only crashes/near misses, these filter options are available in the right panel. The interactive legend tied to the SSRI results allows users to filter results based on risk rank. Figure 51 illustrate an example of displaying only the top 10 (in red) and top 50 (in orange) high SSRI locations. When a data point is clicked, additional information about the risk index and its input is shown in a pop-up dialog box.



**Figure 51. SAFExMAP: Displaying SSRI, SSRI rank, and component variables
Providing a pairwise comparison table of user-selected locations on SSRI**

SAFExMAP allows users to select a pair of locations and compare them in a table format. Figure 52 illustrates an example of a table comparing two locations with SSRI ranks 1 and 7.

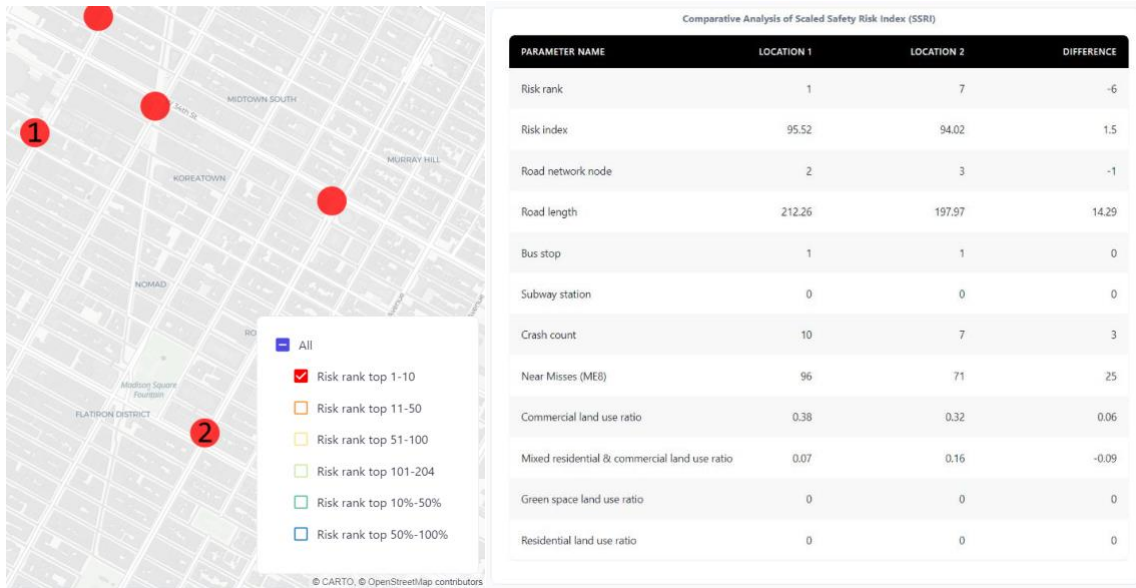


Figure 52. SAFExMAP: Pairwise Comparative Analysis of SSRI
 Displaying multi-source data in a layered format and offering various visualization options

SAFExMAP integrated various safety related data and offers different visualization options to display them. Figure 53 shows the filter option and how the data layers look like in the right panel of the interface. Figure 54 demonstrates some examples using different visualization options.

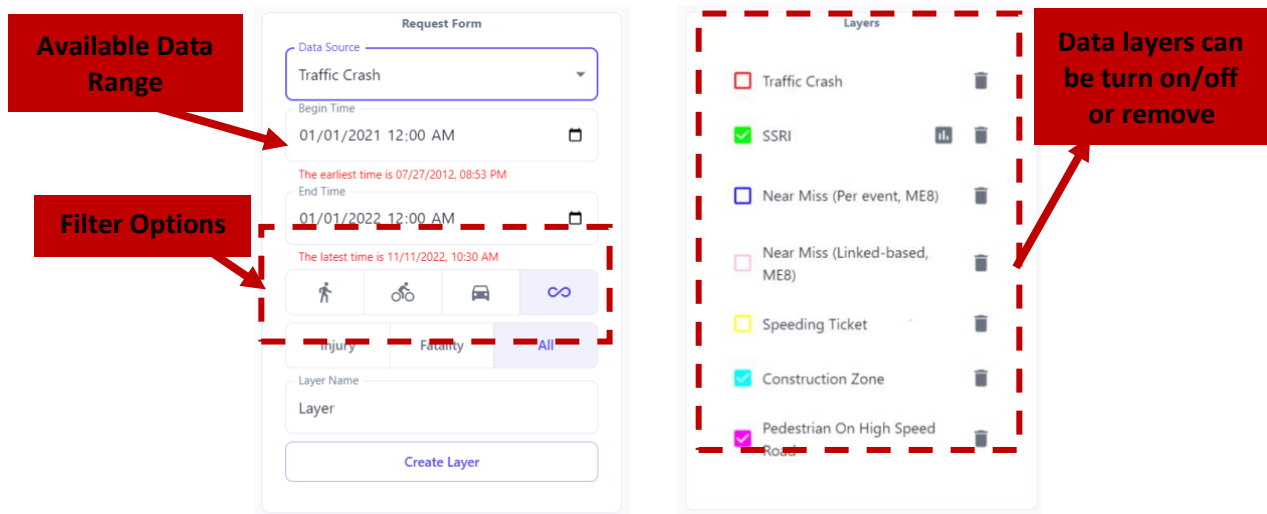


Figure 53. SAFExMAP: Filter options and data layers



Figure 54. SAFExMAP: Multi-source data visualization

Spatiotemporal analysis

The second module in SAFExMAP allows for spatiotemporal analysis that enables user-defined polygon selections for multi-data analysis (Figure 55). Once the polygon is defined, it generates bar charts illustrating temporal patterns from four data sources: Near Miss (ME8), Near Miss (OEM), Traffic Crash, and Speeding Ticket (Figure 56). The range for the temporal analysis is also customizable based on user input. Furthermore, this module provides corresponding JSON files for all the data of the selected polygon. These files can be downloaded, enabling users to conduct further offline analyses.



Figure 55. SAFExMAP: User-defined polygon for spatiotemporal analysis

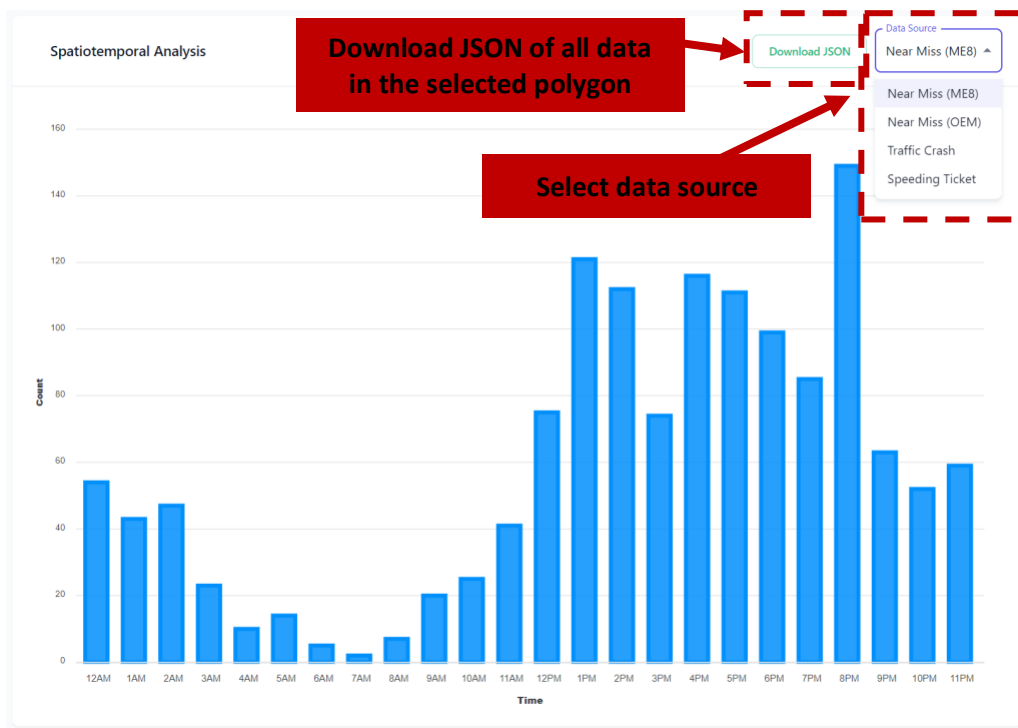


Figure 56. SAFExMAP: Temporal pattern bar charts

5.4 Security

The web-based platform has several security measures implemented, including:

1. **Secure communications:** To ensure that sensitive information exchanged between users and the application remained private and secure, we utilized NGINX as a reverse proxy to implement HTTPS communications in the web application.
2. **Input validation:** To prevent injection attacks like SQL injection, we validated all input received from users on both the front-end and back-end.
3. **Data cryptographic hash:** We utilized SHA-256 to create a secure hash of sensitive data such as passwords. This made it virtually impossible to reverse-engineer the original password from the hash value.
4. **Access controls:** We implemented access controls to restrict access to specific features such as historical detected images based on user roles and permissions.

5.5 Cost Estimation

The web-applications can be implemented on individual computers or servers. The capital cost of the application depends on various factors like architecture complexity, traffic patterns, and data processing and storage.

5.5.1 Capital cost (if individual computer is used)

If we consider a scenario with 300-500 of concurrent users, the following estimates outline the capital costs if individual computer is used.

Table 14 Capital cost (using individual computer).

	Processor	Memory	Storage	Network	Total cost
Minimum	Intel Core i5 or AMD Ryzen 5 with 4 cores and a clock speed of 2.5GHz or higher	8GB RAM	256GB SSD	10 Mbps to 100 Mbps	\$250-\$450
Desired	Intel Core i7 or AMD Ryzen 7 with 6 or more cores and a clock speed of 3.0GHz or higher	16GB RAM	512GB SSD	100 Mbps to 1 Gbps	\$500-\$800

5.5.2 Capital cost (if a local or cloud-based server is used)

The following are the costs for deployment on local servers and cloud services.

Table 15 Capital cost (using server services).

	Local server	Total cost
Minimum	Dell PowerEdge T150 Tower Server	\$850
Desired	Dell PowerEdge T350 Tower Server	\$1,650

	Cloud service	Annual cost
Minimum	AWS EC2 a1.xlarge	\$900
Desired	AWS EC2 a1.2xlarge	\$1,750

5.5.3 Application development and operation costs

The operating cost of the application includes the cost of server maintenance, which amounts to approximately \$300 per year. The labor costs are categorized into development and maintenance expenses. The application development took approximately ten weeks, with a commitment of five hours per week. On the other hand, maintenance activities required two hours per week. Considering a developer salary of \$27.50 per hour, the total cost for development amounts to \$1,375, while the yearly maintenance cost is estimated at \$2,860.

5.5.4 Cost summary

The total capital and annualized operation cost estimates are summarized as follows.

Table 16 Cost summary.

	Individual Computer	Local Server	Cloud Server
Computer/Server Acquisition (*desired requirement is used for estimation)	\$800 (one-time)	\$1,650 (one-time)	\$1,750 (per year)
App development	\$1,375 (one-time)	\$1,375 (one-time)	\$1,375 (one-time)
Server maintenance	-	\$300 (per year)	\$300 (per year)
App maintenance	\$2,860 (per year)	\$2,860 (per year)	\$2,860 (per year)
Total	\$2,175 (one-time) + \$2,860 (per year)	\$3,025 (one-time) + \$3,160 (per year)	\$1,375 (one-time) + \$4,885 (per year)

Note: Cost estimates based on 300-500 of concurrent users.

5.6 Summary and Lessons Learned

In a nutshell, the Urban Intellivision web application implements a microservice-based architecture that facilitates inter-process communications and automated pipelines for real-time analysis and visualization of urban traffic data and work zones in New York City. Our team has applied various optimizations, such as list virtualization, indexing caching, and multi-threading detection, which significantly lower rendering, query, and detection latency by 87.5%, 89.1%, and 59.6%, respectively. Lessons learned from building this web-based tool are also summarized as follows:

- **Utilize parallel computing to speed up computer vision detections.** The complexity of our computer vision models, and the substantial amount of input data necessitate a considerable amount of resources, resulting in a performance burden. Nevertheless, the implementation of multi-threading enhances the effectiveness of CPU cores, ultimately increasing the overall throughput and decreasing the processing time of the program.
- **Apply list virtualization to accelerate page rendering performance.** When rendering a large list, React has to traverse and diff the entire list, even if only a few items are visible on the screen, which can be a time-consuming process. However, we used list virtualization to allow only a small subset of the full list that is visible to the user to be rendered, resulting in a significant optimization of the rendering performance.

Section 6 Advisory Board

The collaborative approach in this project, involving academia, industry, and agencies, was primarily facilitated through regular communication with the advisory board. The board was assembled to include both technical and policy experts from various agencies and industries. Their role was to provide insights into the applicability of available camera and other transportation data, comprehend the operational and planning needs of agencies, and suggest desired functionalities for computer vision use cases. We found the 'wishlist' approach highly useful for identifying high-priority use cases of interest to local agencies.

The research team organized three advisory board meetings, including a kick-off meeting and a follow-up meeting after the first prototype of each app was developed. These meetings provided a platform for live demonstrations of the applications and also offered opportunities to gather regular feedback during the development of the two computer vision applications. This “Aigle Management” type of approach (an iterative approach to managing app development that focuses on continuous releases and customer feedback) helped minimize the risk of pursuing misguided or practically unfeasible research outcomes.

For the work zone application, our focus was on obtaining stakeholder feedback about how the computer vision application could enhance the real-time work zone information database, manage work zone impacts dynamically, and improve street construction permit enforcement. For the safety risk map application, we concentrated on obtaining stakeholder feedback about their needs for using near-miss data for traffic safety analysis.

Throughout the project, the flexible format of the volunteer advisory board allowed for wider awareness about the project, attracting more agency personnel to join the board. The final advisory board included the following:

- **Paul Rothman**, Director of Smart Cities + IoT, Strategic Initiatives, NYC Office of Technology and Innovation
- **Asheque Rahman**, Senior Program Traffic Engineer, The Port Authority of New York & New Jersey
- **Howard Jiang**, Engineer in Charge, Construction Management Unit within Infrastructure Division, NYC Department of Design and Construction
- **Justin Romeo**, Director of Special Project, Regional & Strategic Planning, New York City Department of Transportation
- **Rob Viola**, Director of Safety Policy & Research, New York City Department of Transportation
- **Maddalena Romano**, Director of the Data/Asset Management Unit, New York City Department of Transportation
- **Seth Berman**, Senior Transportation Planner, Transportation Planning and Management Division, New York City Department of Transportation
- **Zamir Alam**, Deputy Director, Modeling and Data Analysis, New York City Department of Transportation
- **Dan Wan**, Data Scientist, Modeling and Data Analysis, New York City Department of Transportation
- **Ruoran Lin**, Transportation Planner, Transportation Policy and Analytics Division, New York City Department of Planning
- **Terri Matthews**, Director of Town+Gown:NYC, a citywide university-community built environment research program at NYC Department of Design and Construction
- **Mark Davis**, General Manager, Infrastructure & Government Sectors, Vexcel Imaging (Formally Head of Data Services & Business Development (North America), Mobileye).

From the three meetings, we derived several key takeaways, which were used to refine our research approaches and outcomes or incorporated into our future app development plans:

- The advisory board emphasized the importance of understanding how pedestrians and vehicles navigate around work zones. They suggested that if computer vision could estimate the size of the work zone or the number of lanes blocked by the work zones, it would be beneficial. Responding to this feedback, our research team developed an algorithm to estimate work zone

size and integrated traffic detection functions, enabling users to understand the number of vehicles and vulnerable road users around the work zones.

- The advisory board expressed interest in understanding whether the near-miss data collected by the project team could predict injuries, as this is a question/concern for the NYCDOT. They also mentioned a desire for a usable matrix to indicate areas with better or worse traffic safety. However, they noted that an extremely detailed spatial aggregation level might not be necessary due to the potential for false precision. These comments guided the research team in refining their approach to examining near-miss data, its spatial correlation with crashes and surrounding environments, and the suitable level of spatial aggregation.
- The advisory board suggested that instead of using general crash data, injury and fatality records should be used as they are more pertinent to the safety indicator. This is because Property Damage Only (PDO) crashes, although more common, may merely indicate friction and not all of them may be responded to by the police (e.g., due to limited resources during COVID), rendering them unreliable data points. While crashes involving injuries or fatalities might constitute a smaller dataset, they are more relevant when considering a safety indicator. Based on this comment, the research team has included an additional safety analysis that uses only injury and fatality crash data and near-miss events.
- Privacy concerns in the use of video for analytics were raised by the advisory board. They suggested adopting a "privacy by design" approach, which in this case involves de-identifying any information from the CCTV feeds, such as blurring pedestrians, license plates, and other identifiable information. This requirement would be crucial for any project scaling beyond a research initiative, emphasizing the importance of planning for it during development. The research team agreed that privacy is always a paramount consideration. For CCTV cameras, the current public feeds are low resolution (<240p) and discontinuous (one image per 1-7 seconds), making privacy less of a concern as pedestrian faces or license plates cannot be seen, nor can pedestrians or vehicles be tracked. However, in keeping with the "privacy by design" principles, the research team plans to use a previously developed pedestrian mask algorithm to protect any identifiable information, which can be integrated into the current framework if privacy becomes a concern.
- The advisory board recommended the collection of Maintenance and Protection of Traffic Plan (MPT) samples. MPTs could help define a work zone and its start time. They also suggested comparing detected "active" work zones with the permitted construction time periods. Although not yet implemented in the app's current version, the research team is working on these items, with plans to incorporate validation and comparison features in the future.

- The advisory board suggested for the Urban Work Zone Detection application, one sub-application would be the ability to identify whether a construction site is adhering to the City's rule regarding temporary biking infrastructure. The rule states that if a construction project needs to take over a bike lane, it must make temporary accommodations for the lane to exist adjacent to the construction. Many times, the lane is just blocked without accommodation. Being able to identify this condition and request remediation of the situation from the construction team would help bolster safety for cyclists. The research team acknowledged observing bike lanes often blocked for various reasons. While this sub-use case wasn't tested due to the limitations of CCTV images capturing these situations, the research team is interested in developing a sub-use case if suitable data becomes available.

Section 7 Outreach and Technical Transfer

This research project facilitates the adoption of computer vision for smart cities that will bring positive impacts on transportation planning and operations and provide cost-effective solutions to the transportation industry. Particularly, this project developed two prototype web-based applications for urban work zone detection and traffic safety risk analysis.

The research outcomes are disseminated via the following ways:

- The urban work zone detection algorithms were compiled into a technical paper for the 26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023).
- The prototype system has been tested using NYC data. Demonstrations of the applications have been disseminated to transportation agencies including the NYC Department of Transportation, NYC Department of Design and Construction, NYC Department of City Planning, NYC Office of Technology Innovations, and the Port Authority of New York & New Jersey via virtual meetings.
- The proposed computer vision approach was introduced at two peer-exchange events in NYC: “Vision Zero Research on the Road” and “Vision Zero Research Collaboration Forum”. These events were hosted by Town+Gown NYC, an initiative organized by the NYC Department of Design and Construction and the NYC Department of Health & Mental Hygiene.
- The results and findings of this research were showcased at the ITS-NY 30th Annual Meeting and Technology Exhibition via a student poster presentation and was awarded the 2nd Prize.
- Both the proposed computer vision approach and the web-applications were introduced during the four-day International KN-C³ Workshop. Faculty and students from the Korea Advanced Institute of Science & Technology (KAIST) were in attendance.
- The results and findings of this research were presented at an international academic workshop “Application of Deep Learning & Data Science in Transportation and Smart City”, co-hosted by NYU Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning and Tongji University.

Section 8 Conclusion and Discussion

8.1 Research Conclusion

This research project facilitates the integration of computer vision technologies within complex urban environments. This approach holds significant potential to positively impact transportation planning and operations and offers economically efficient real-world solutions for the transportation industry. Specifically, this research project has led to the development of two highly interactive web-based applications - one dedicated to real-time urban work zone detection, and the other for conducting comprehensive traffic safety risk analysis. These prototypes not only present practical solutions to existing challenges but also underscore the substantial potential of utilizing deep-learning and vision techniques based on both fixed camera facilities and crowd sourced CAV-based traffic and camera data for traffic planning and operations. The key accomplishment of the real-time urban work zone detection and the safety risk index view map can be summarized as follows:

- Our team introduced a comprehensive framework that merges various elements, namely, data-centric AI training, topological analysis, gradient boosting classification, and reference-free size estimation. This forms a robust toolkit for work zone detection. The deep-learning based model for work zone object detection was developed using a data-centric approach. This approach aims to iteratively enhance the model's performance by augmenting a custom training dataset gathered from multiple sources, including traffic cameras, web-mined images, and 3D-simulated work zone images. This methodology helps overcome the scarcity of annotated real-world work zone images.
- Our approach includes the introduction of an innovative topology-based inference method, utilizing XGBoost, to differentiate true work zones from other operational zones that feature some elements of work zones. Alongside this, we developed a reference-free work area size estimation method. This method uses the standard heights of common construction equipment to provide a generalized real-pixel distance approximation.
- Evidence of the effectiveness of our model is seen in its performance - an average mAP of 74.1% across all work zone classes, an accuracy of 98.4% for scene identification, and an accuracy of up to 89.52% for size estimation. The system is capable of real-time identification and size estimation of work zones in complex urban settings, which can aid in the provision of more informed active work zone management, ultimately improving safety and mobility.
- We embedded the proposed algorithms into an urban work zone detection web-base application, WorkZoneX, which allows real-time detection of the work zones in New York City via

900+ existing CCTV cameras. Users are able to view the detected work zones with and without the presence of construction workers (i.e., active work zones) as well as traffic conditions around the work zone. Individual and network-level work zone statistics including counts and durations are provided in a dashboard format.

- This research also developed a Scaled Safety Risk Index (SSRI), using multisource data, statistical modeling, and machine learning techniques, as a comprehensive safety risk index for traffic planning. A unique data source used in this approach is the crowd sourced CAV-based near miss data extracted using computer vision technology.
- A positive correlation was found between crashes and near misses, indicating the potential of using near miss data as supplemental safety data when crash information is not adequate.
- Based on the work zone detection and SSRI algorithms, two highly interactive and user-friendly web-applications were built using microservice architecture. Automatic pipelines were developed for both applications to minimize manual efforts.
- Cost estimations were provided for operating the applications on an individual computer (\$2,175 (one-time cost) + \$2,860 (per year)), a local server (\$3,025 (one-time cost) + \$3,160 (per year)) or on a cloud-based server (\$1,375 (one-time cost) + \$4,885 (per year)).

Each accomplishment detailed above demonstrates a substantial step forward in leveraging AI for improving urban traffic management and safety. This research not only provides valuable real-world applications but also showcases the potential for further developments in these areas.

8.2 Discussion and Future Work

This research project, while advancing computer vision technologies, has identified some limitations and corresponding potential areas as the directions for future work:

- The current urban work zone detection model encounters difficulties distinguishing between construction workers and traffic enforcement agents, due to their similar attire such as yellow or orange safety vests. As the presence of construction workers is significant in identifying true work zones, any misclassification may lead to errors in work and non-work zone identification. Future work needs to address this challenge. Another area of concern is the methodology used for area estimation. While it works effectively when boundary equipment forms a closed enclosure, it struggles with work zones that utilize natural boundaries like road curbs or walls. Also, the estimation method assumes uniform heights for each object type, which can introduce bias. Future efforts should aim to enhance the model's versatility and accuracy.

- The SSRI model currently incorporates only crash records and near miss data as dynamic variables. While these can be updated frequently, other directly related safety variables, such as speeding tickets, were not prevalent in the test area. In addition, the SSRI currently serves as a one-time index for the study period as the sample near miss data is limited. In future iterations, we aim to develop the SSRI as a time-dependent algorithm, adjusting based on the user-selected study period. We also plan to expand its application to cover the entire city to incorporate other time variant variables, such as speeding tickets.

These limitations open avenues for future work to refine the models and increase their applicability and accuracy in a wider range of scenarios. Periodic feedback from the advisory board provided an excellent communication channel to hone our research approach and enhance application features. Furthermore, the methodology proposed in this project offers valuable insights for potential future expansion of using computer vision techniques for other prioritized use cases (e.g., bike and scooter safety) in urban areas.

References

1. Abdelrahman, A., Hassanein, H.S., Abu-Ali, N., 2019. IRouteSafe: Personalized cloud-based route planning based on risk profiles of drivers, *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, pp. 1-6.
2. Acharya, D., Yan, W., Khoshelham, K., 2018. Real-time image-based parking occupancy detection using deep learning, *Research@ Locate*, pp. 33-40.
3. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vairo, C., 2016. Car parking occupancy detection using smart camera networks and deep learning, *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, pp. 1212-1217.
4. Anselin, L., 1995. Local indicators of spatial association—LISA. *Geographical analysis* 27(2), 93-115.
5. Anselin, L., Syabri, I., Kho, Y., 2006. GeoDa: an introduction to spatial data analysis. *Geographical analysis* 38(1), 5-22.
6. Arshad, B., Ogie, R., Barthelemy, J., Pradhan, B., Verstaevel, N., Perez, P., 2019. Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review. *Sensors* 19(22), 5012.
7. ASIRT, 2022. ROAD SAFETY FACTS.
8. Bulan, O., Loce, R.P., Wu, W., Wang, Y.R., Bernal, E.A., Fan, Z., 2013. Video-based real-time on-street parking occupancy detection system. *Journal of Electronic Imaging* 22(4), 041109.
9. CHEKPEDS, 2023. NYC Crash Mapper.
10. Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., Knoll, A., 2020. Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine* 37(4), 34-49.
11. Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794.
12. Chen, W., Yeo, C.K., 2019. Unauthorized Parking Detection using Deep Networks at Real Time, *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, pp. 459-463.
13. Ding, H., Sze, N., 2022. Effects of road network characteristics on bicycle safety: A multivariate Poisson-lognormal model. *Multimodal transportation* 1(2), 100020.
14. Donnell, E.T., Hanks, E., Porter, R.J., Cook, L., Srinivasan, R., Li, F., Nguyen, M., Eccles, K.A., 2020. The Development of Crash Modification Factors: Highway Safety Statistical Paper Synthesis. United States. Federal Highway Administration. Office of Safety Research and
15. Duan, R., Deng, H., Tian, M., Deng, Y., Lin, J., 2022. SODA: Site Object Detection dAtaset for Deep Learning in Construction. *arXiv preprint arXiv:2202.09554*.

16. Edara, P., Rahmani, R., Brown, H., Sun, C., 2017. Traffic impact assessment of moving work zone operations. Smart Work Zone Deployment Initiative.
17. Edara, P.K., Sun, C., Robertson, A., 2013. Effectiveness of Work Zone Intelligent Transportation Systems. Iowa State University. Institute for Transportation.
18. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, *kdd*, pp. 226-231.
19. Fachrie, M., 2020. A simple vehicle counting system using deep learning with YOLOv3 model. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)* 4(3), 462-468.
20. FHWA, 2022. FHWA Strategic Plan 2022-2026.
21. Gao, J., Xie, K., Ozbay, K., 2018. Exploring the spatial dependence and selection bias of double parking citations data. *Transportation Research Record* 2672(42), 159-169.
22. Gao, J., Zuo, F., Ozbay, K., Hammami, O., Barlas, M.L., 2022. A new curb lane monitoring and illegal parking impact estimation approach based on queueing theory and computer vision for cameras with low resolution and low frame rate. *Transportation Research Part A: Policy and Practice* 162, 137-154.
23. Ghosh, A., Chatterjee, T., Samanta, S., Aich, J., Roy, S., 2017. Distracted driving: A novel approach towards accident prevention. *Adv. Comput. Sci. Technol* 10(8), 2693-2705.
24. Gkollias, K., Vlahogianni, E.I., 2018a. Convolutional neural networks for on-street parking space detection in urban networks. *IEEE Transactions on Intelligent Transportation Systems*.
25. Gkollias, K., Vlahogianni, E.I., 2018b. Convolutional neural networks for on-street parking space detection in urban networks. *IEEE Transactions on Intelligent Transportation Systems* 20(12), 4318-4327.
26. Goodchild, M.F., 1986. *Spatial autocorrelation*. Geo Books.
27. Grinsztajn, L., Oyallon, E., Varoquaux, G., 2022. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*.
28. He, S., Sadeghi, M.A., Chawla, S., Alizadeh, M., Balakrishnan, H., Madden, S., 2021. Inferring high-resolution traffic accident risk maps based on satellite imagery and gps trajectories, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11977-11985.
29. Huang, W., Zhang, Y., Yu, Y., Xu, Y., Xu, M., Zhang, R., De Dieu, G.J., Yin, D., Liu, Z., 2021. Historical data-driven risk assessment of railway dangerous goods transportation system: Comparisons between Entropy Weight Method and Scatter Degree Method. *Reliability Engineering & System Safety* 205, 107236.
30. Huang, Y., Wang, X., Patton, D., 2018. Examining spatial relationships between crashes and the built environment: A geographically weighted regression approach. *Journal of transport geography* 69, 221-233.

31. Ijjina, E.P., Chand, D., Gupta, S., Goutham, K., 2019. Computer vision-based accident detection in traffic surveillance, *2019 10th International conference on computing, communication and networking technologies (ICCCNT)*. IEEE, pp. 1-6.
32. Jiang, S., Jafari, M., Kharbeche, M., Jalayer, M., Al-Khalifa, K.N., 2020. Safe route mapping of roadways using multiple sourced data. *IEEE Transactions on Intelligent Transportation Systems* 23(4), 3169-3179.
33. Jocher, G., Chaurasia, A., Qiu, J., 2023. YOLO by Ultralytics. URL: <https://github.com/ultralytics/ultralytics>.
34. Kanchana, B., Peiris, R., Perera, D., Jayasinghe, D., Kasthurirathna, D., 2021. Computer vision for autonomous driving, *2021 3rd International Conference on Advancements in Computing (ICAC)*. IEEE, pp. 175-180.
35. Katsamenis, I., Karolou, E.E., Davradou, A., Protopapadakis, E., Doulamis, A., Doulamis, N., Kalogeras, D., 2022. TraCon: A novel dataset for real-time traffic cones detection using deep learning, *Novel & Intelligent Digital Systems: Proceedings of the 2nd International Conference (NiDS 2022)*. Springer, pp. 382-391.
36. Katsamenis, I., Karolou, E.E., Davradou, A., Protopapadakis, E., Doulamis, A., Doulamis, N., Kalogeras, D., 2023. TraCon: A novel dataset for real-time traffic cones detection using deep learning, *Novel & Intelligent Digital Systems Conferences*. Springer, pp. 382-391.
37. Khan, S.H., Yousaf, M.H., Murtaza, F., Velastin, S., 2020. PASSENGER DETECTION AND COUNTING FOR PUBLIC TRANSPORT SYSTEM. *NED University Journal of Research* 17(2).
38. Kollmitz, M., Eitel, A., Vasquez, A., Burgard, W., 2019. Deep 3D perception of people and their mobility aids. *Robotics and Autonomous Systems* 114, 29-40.
39. Kousar Nikhath, A., Sailaja, N.V., Vasavi, R., Vijaya Saraswathi, R., 2021. Road traffic counting and analysis using video processing, *Intelligent System Design*. Springer, pp. 645-651.
40. Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84-90.
41. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278-2324.
42. Lee, C.-H., Wen, M.-G., Han, C.-C., Kou, D.-C., 2005. An automatic monitoring approach for unsupervised parking lots in outdoors, *Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology*. IEEE, pp. 271-274.
43. Lee, J.T., Ryoo, M.S., Riley, M., Aggarwal, J., 2009. Real-time illegal parking detection in outdoor environments using 1-D transformation. *IEEE Transactions on Circuits and Systems for Video Technology* 19(7), 1014-1024.
44. Leroux, S., Li, B., Simoens, P., 2022. Automated training of location-specific edge models for traffic counting. *Computers and Electrical Engineering* 99, 107763.
45. Liu, G., Shi, H., Kiani, A., Khreishah, A., Lee, J., Ansari, N., Liu, C., Yousef, M.M., 2021. Smart Traffic Monitoring System using Computer Vision and Edge Computing. *IEEE Transactions on Intelligent Transportation Systems*.

46. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, *European conference on computer vision*. Springer, pp. 21-37.
47. Lym, Y., Chen, Z., 2020. Does space influence on the frequency and severity of the distraction-affected vehicle crashes? An empirical evidence from the Central Ohio. *Accident Analysis & Prevention* 144, 105606.
48. Màrmol, E., Sevillano, X., 2016. QuickSpot: a video analytics solution for on-street vacant parking spot detection. *Multimedia Tools and Applications* 75(24), 17711-17743.
49. Moran, P.A., 1950. Notes on continuous stochastic phenomena. *Biometrika* 37(1/2), 17-23.
50. Motamedi, M., Sakharnykh, N., Kaldewey, T., 2021. A data-centric approach for training deep neural networks with less data. *arXiv preprint arXiv:2110.03613*.
51. Nath, N.D., Behzadan, A.H., 2020. Deep convolutional networks for construction object detection under different visual conditions. *Frontiers in Built Environment* 6, 97.
52. Newman Library of Baruch College, C., 2020. NYC Mass Transit Spatial Layers.
53. NYC Vision Zero, 2023. What It Is.
54. Ozturk, O., Ozbay, K., Yang, H., 2014. Estimating the impact of work zones on highway safety.
55. Peng, W., Pan, H., Liu, H., Sun, Y., 2020. Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13015-13024.
56. Polák, M., Holubová, I., 2015. REST API management and evolution using MDA, *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*, pp. 102-109.
57. Ranjan, A., Misra, P., Vasan, A., Krishnakumar, S., Sivasubramaniam, A., 2019. City Scale Monitoring of On-Street Parking Violations with StreetHAWK, *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 31-40.
58. Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
59. Sajib, M.S.R., Bhuiyan, T.A.-U.-H., 2019. Computer vision based traffic monitoring and analyzing from on-road videos. *Global journal of computer science and technology* 19(2), 19-24.
60. Santhanavanich, T., Wuerstle, P., Silberer, J., Loidl, V., Rodrigues, P., Coors, V., 2020. 3D SAFE ROUTING NAVIGATION APPLICATION FOR PEDESTRIANS AND CYCLISTS BASED ON OPEN SOURCE TOOLS. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 6.
61. Silva, P.B., Andrade, M., Ferreira, S., 2020. Machine learning applied to road safety modeling: A systematic literature review. *Journal of traffic and transportation engineering (English edition)* 7(6), 775-790.

62. Tang, H., Peng, A., Zhang, D., Liu, T., Ouyang, J., 2020. SSD Real-Time Illegal Parking Detection Based on Contextual Information Transmission. *CMC-COMPUTERS MATERIALS & CONTINUA* 62(1), 293-307.
63. Wang, Y., Li, X., Jiao, Z., Zhang, L., 2022. Pedestrian trajectory prediction based on temporal attention, *International Conference on Algorithms, Microchips and Network Applications*. SPIE, pp. 509-514.
64. WHO, 2022. Road traffic injuries.
65. Wu, Q., Huang, C., Wang, S.-y., Chiu, W.-c., Chen, T., 2007. Robust parking space detection considering inter-space correlation, *2007 IEEE International Conference on Multimedia and Expo*. IEEE, pp. 659-662.
66. Xie, K., Ozbay, K., Kurkcu, A., Yang, H., 2017a. Analysis of traffic crashes involving pedestrians using big data: Investigation of contributing factors and identification of hotspots. *Risk analysis* 37(8), 1459-1476.
67. Xie, K., Ozbay, K., Yang, D., Xu, C., Yang, H., 2021. Modeling bicycle crash costs using big data: A grid-cell-based Tobit model with random parameters. *Journal of Transport Geography* 91, 102953.
68. Xie, K., Ozbay, K., Yang, H., 2015. Spatial analysis of highway incident durations in the context of Hurricane Sandy. *Accident Analysis & Prevention* 74, 77-86.
69. Xie, K., Ozbay, K., Zhu, Y., Yang, H., 2017b. Evacuation zone modeling under climate change: A data-driven method. *Journal of Infrastructure Systems* 23(4), 04017013.
70. Xie, K., Wang, X., Ozbay, K., Yang, H., 2014. Crash frequency modeling for signalized intersections in a high-density urban road network. *Analytic methods in accident research* 2, 39-51.
71. Xie, X., Wang, C., Chen, S., Shi, G., Zhao, Z., 2017c. Real-time illegal parking detection system based on deep learning, *Proceedings of the 2017 International Conference on Deep Learning Technologies*, pp. 23-27.
72. Yan, X., Zhang, H., Li, H., 2020. Computer vision-based recognition of 3D relationship between construction entities for monitoring struck-by accidents. *Computer-Aided Civil and Infrastructure Engineering* 35(9), 1023-1038.
73. Yang, D., Xie, K., Ozbay, K., Yang, H., Budnick, N., 2019. Modeling of time-dependent safety performance using anonymized and aggregated smartphone-based dangerous driving event data. *Accident Analysis & Prevention* 132, 105286.
74. Yin, Z., Xiong, H., Zhou, X., Goldberg, D.W., Bennett, D., Zhang, C., 2019. A deep learning based illegal parking detection platform, *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pp. 32-35.
75. Zinchenko, V., Kondratenko, G., Sidenko, I., Kondratenko, Y., 2020. Computer vision in control and optimization of road traffic, *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, pp. 249-254.

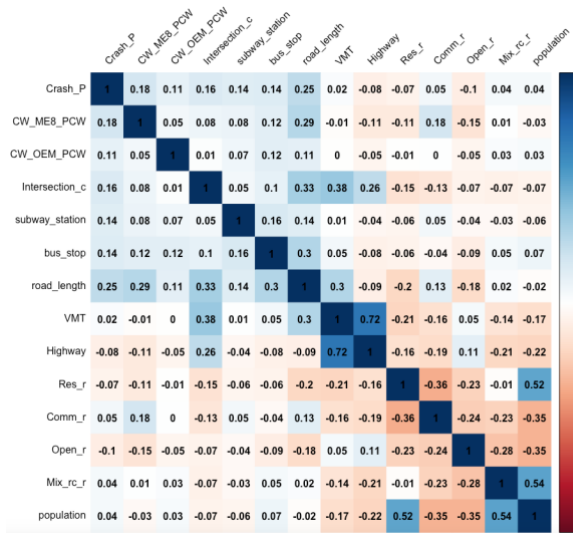
76. Zuo, F., Gao, J., Kurkcu, A., Yang, H., Ozbay, K., Ma, Q., 2021. Reference-free video-to-real distance approximation-based urban social distancing analytics amid COVID-19 pandemic. *Journal of Transport & Health* 21, 101032.

Appendix

A.1 Correlation between VRU crash data and VRU near misses

A.1.1 Correlation

Figure 57 shows the correlation between VRU crash data and VRU near misses. For pedestrians and cyclists, the values of correlation coefficient are 0.18 and 0.11 respectively. This indicates that the VRU crash count and near misses have a weak positive linear relationship.



(a) Pedestrian safety risk related variables



(b) Cyclist safety risk related variables

Figure 57. The correlation plots of VRU crash data and VRU near misses

A.1.2 Spatial Correlation

(1) Bivariate Global Moran's I

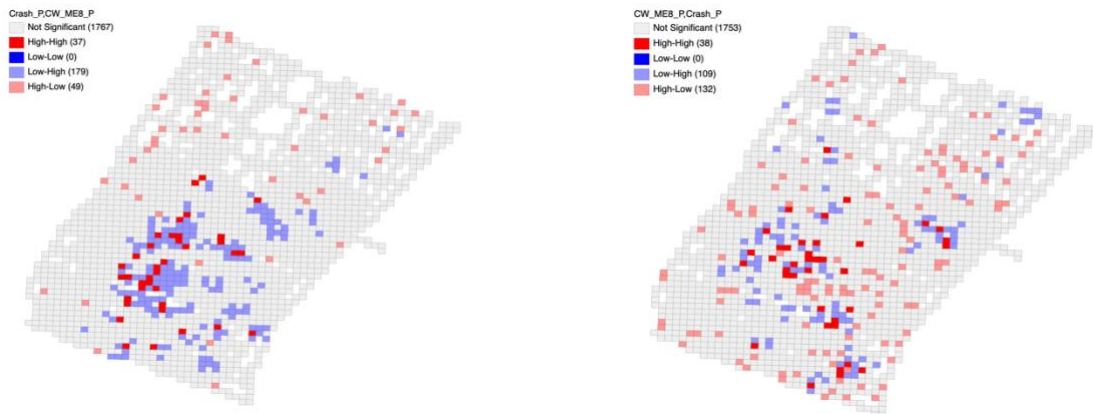
Table 17 presents the Bivariate Global Moran's I using k-nearest neighbor (k=8) for VRU crash data and near misses. All four bivariate global moran's I are significant in p-value (<0.05) .These results indicate a statistically significant spatial correlation.

Table 17 Bivariate Global Moran's I using K-nearest neighbor (k=8) for VRU crash data/near misses.

Bivariate Global Moran's I	I	E(I)	SD(I)	Zi	Pseudo p-value
Pedestrian Crash-Near Miss	0.046	-0.0005	0.0089	5.1746	0.001
Pedestrian Near Miss-Crash	0.046	-0.0005	0.0089	5.1402	0.001
Cyclist Crash-Near Miss	0.025	-0.0005	0.0087	2.867	0.004
Cyclist Near Miss-Crash	0.025	-0.0005	0.0085	2.942	0.002

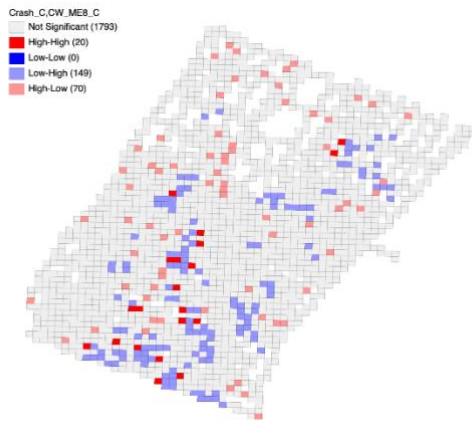
(2) Bivariate LISA Cluster Map

Figure 58 shows the bivariate LISA results for the two pairs “Crash-Near Miss” and “Near Miss-Crash” for pedestrian and cyclist respectively. Both Bivariate LISA have more clusters with negative local spatial correlation than that with positive local spatial correlation. Overall, due to less VRU near miss events identified by the crowd sourced CAVs, the usage of VRU SSRI may be limited.

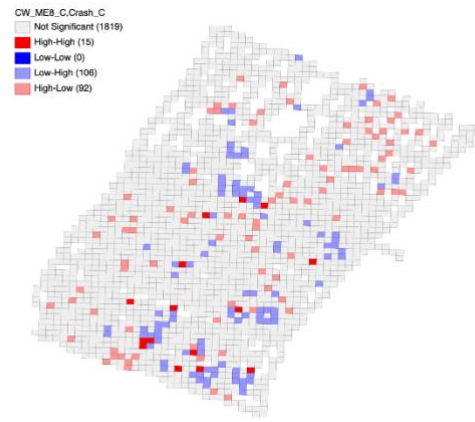


(a) Pedestrian crash count-ME8 pedestrian near misses bivariate LISA

(b) ME8 near misses - pedestrian crash count bivariate LISA



(c) Cyclist crash data-ME8 cyclist near misses bivariate LISA



(d) ME8 cyclist near misses - cyclist crash data bivariate LISA

Figure 58. Bivariate LISA cluster map for VRU crash data and near misses (k-nearest neighbor, k=8)

A.2 VRU SSRI

A.2.1 VRU SSRI Calculation

Based on the steps proposed in Section 4.4, two VRU SSRI, which are Pedestrian SSRI and Cyclist SSRI were calculated. The main difference between VRU SSRI and SSRI is that VRU SSRI is using the crash count and near misses for a specific VRU. Table 18 and Table 19 present the variables that were used to calculate the SSRI and their parameters.

Table 18 Pedestrian SSRI Variables and Their Parameters.

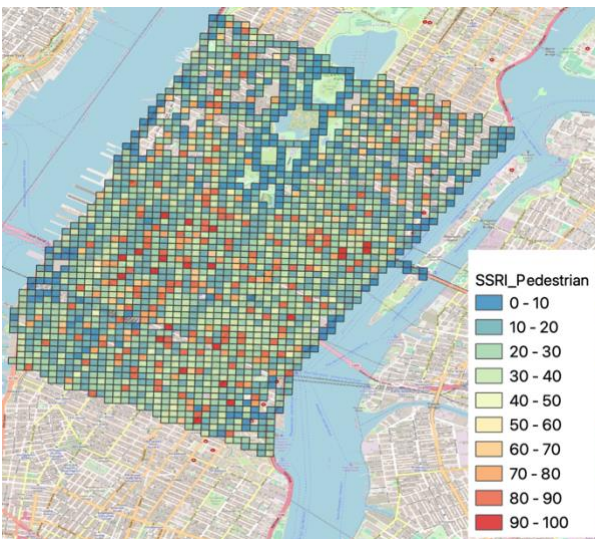
Variable	Pedestrian Crash	Road Length	#intersections	Population	Commercial Land Used Rate	ME8 Near Misses	#Bus Stops	Open Area Land Used Rate	#Subway Station
Weight	1	0.271	0.243	0.181	0.166	0.045	0.044	0.028	0.023
Sign	+	+	+	+	+	+	+	-	+

Table 19 Cyclist SSRI Variables and Their Parameters.

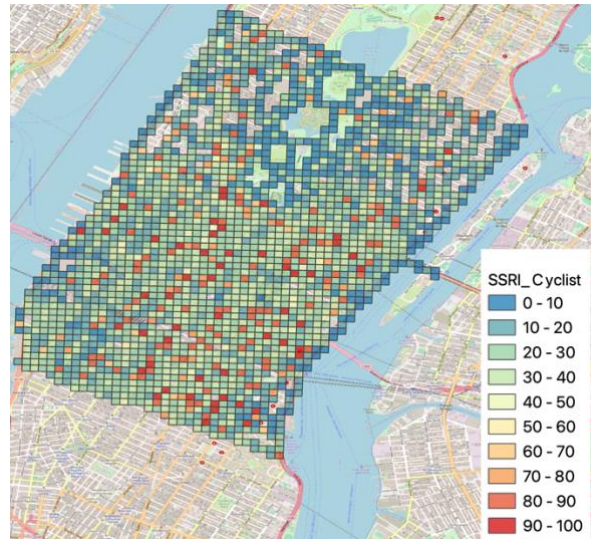
Variable	Cyclist Crash	Road Length	Population	Residential Land Used Rate	Commercial Land Used Rate	#Bus Stops	ME8 Near Misses
Weight	1	0.425	0.248	0.164	0.088	0.048	0.027
Sign	+	+	+	+	+	+	+

A.2.2 VRU SSRI Distribution

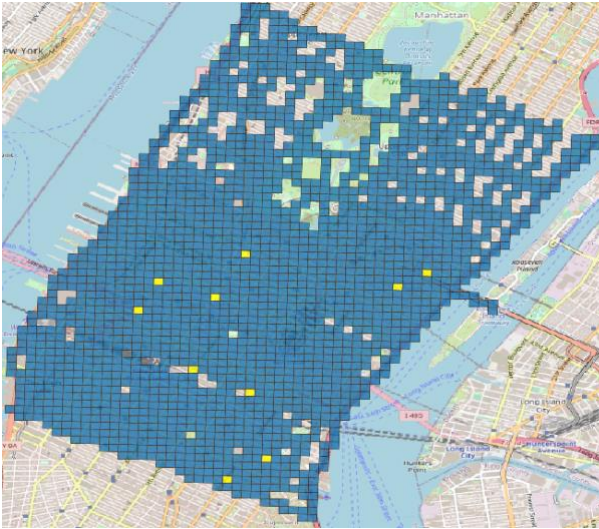
The distributions of VRU SSRI are shown in Figure 59. For pedestrians, 9th avenue, 7th avenue, 2nd avenue and 34th street are the roadways that have two locations that are high risk pedestrian SSRI locations (within top 10), respectively. For Cyclist, 2nd avenue and 51st street both have three locations with high cyclist SSRI (within top 10).



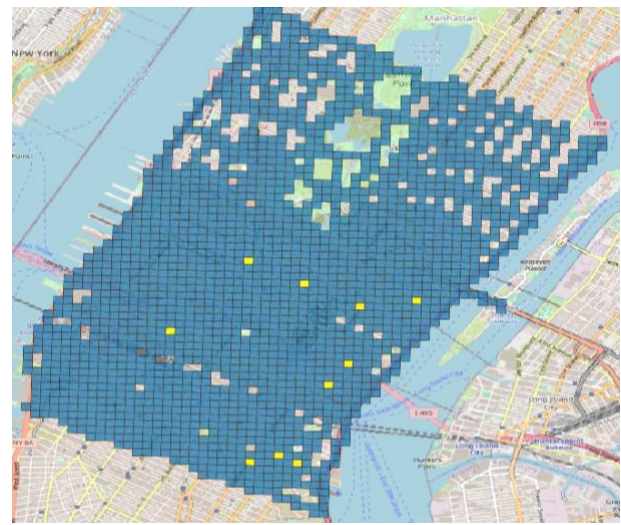
(a) Pedestrian SSRI



(b) Cyclist SSRI



(a) Top 10 Pedestrian SSRI locations
(highlighted in yellow)



(b) Top 10 Cyclist SSRI locations
(highlighted in yellow)

Figure 59. The distribution of VRU SSRI