Real-time Risk Prediction at Signalized Intersections Using a Graph Neural Network

December 2023 | Final Report

SAFE-D
SAFETY THROUGH DISRUPTION

VIRGINIA TECH TRANSPORTATION INSTITUTE
VIRGINIA TECH

Texas A&M Transportation Institute

San Diego State University
Leadership Starts Here

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.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. 06-012	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Real-time Risk Prediction at Signalized Intersections Using a Graph Neural Network	5. Report Date December 2023 6. Performing Organization Code:
7.Author(s) Akash Sonth Abhijit Sarkar Sparsh Jain Hirva Bhagat Zachary Doerzaph	8. Performing Organization Report No. 06-012
9. Performing Organization Name and Address: Safe-D National UTC Virginia Tech Transportation Institute 3500 Transportation Research Plaza Blacksburg, VA 24061	10. Work Unit No. 11. Contract or Grant No. 69A3551747115/Project 06-012
12. Sponsoring Agency Name and Address Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)	13. Type of Report and Period Final Research Report 05/2022-10/2023 14. Sponsoring Agency Code

15. Supplementary Notes

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

16. Abstract

Intersection-related traffic crashes and fatalities are major concerns for road safety. This project aimed to understand the major causes of conflicts at intersections by studying the intricate interplay between roadway agents. The approach involved using the current traffic camera systems to automatically process traffic video data. As manual annotation of video datasets is a very labor-intensive and costly process, this research leveraged modern computer vision algorithms to automatically process these videos and retrieve kinematic behavior of the traffic actors. Results demonstrated how traffic actors and road segments can be modeled independently via graphs and how they can be integrated into a framework that can model traffic systems. The team used a graph neural network to model (a) the interaction of all the roadway agents at any given instance and (b) their role in road safety, both individually and as a composite system. The model reports a near-real-time risk score for a traffic scene. The study concludes with a presentation of a new drone-based trajectory dataset to accelerate research in intersection safety.

17. Key Words Intersection safety, crash causation analysis, graph neural network, computer vision, traffic camera	18. Distribution Statement No restrictions. This document is available to the public through the Safe-D National UTC website, as well as the following repositories: VTechWorks, The National Transportation Library, The Transportation Library, Volpe National Transportation Systems Center, Federal Highway Administration Research Library, and the National Technical Reports Library.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 55	22. Price $0

Abstract

Intersection-related traffic crashes and fatalities are major concerns for road safety. This project aimed to understand the major causes of conflicts at intersections by studying the intricate interplay between roadway agents. The approach involved using the current traffic camera systems to automatically process traffic video data. As manual annotation of video datasets is a very labor-intensive and costly process, this research leveraged modern computer vision algorithms to automatically process these videos and retrieve kinematic behavior of the traffic actors. Results demonstrated how traffic actors and road segments can be modeled independently via graphs and how they can be integrated into a framework that can model traffic systems. The team used a graph neural network to model (a) the interaction of all the roadway agents at any given instance and (b) their role in road safety, both individually and as a composite system. The model reports a near-real-time risk score for a traffic scene. The study concludes with a presentation of a new drone-based trajectory dataset to accelerate research in intersection safety.

Acknowledgements

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program. We would also like to thank Jean Paul Talledo Vilela, Surendrabikram Thapa, and Sandesh Jain for their help in data collection. The team wants to thank Oracle for providing us with computational resources. Oracle gifted $100,000 as cloud computation credit that was utilized for this project. We specifically want to thank Dr. Bryan Baker from Oracle for making this possible. The project team would also like to thank MicroTraffic (currently, MioVision) for providing valuable input for traffic safety and processing. We want to thank Dr. Craig Milligan from MicroTraffic for all the valuable discussion sessions. We would additionally like to thank Mr. Mike Fontaine for providing valuable feedback during the project and providing technical review for the final report.

Table of Contents

List of Figures

List of Tables

Introduction

Motor vehicle crashes continue to claim a significant number of lives in the United States. From 2015 to 2021, there were over 35,000 deaths every year due to motor vehicle crashes [1]. For example, in 2019, there were approximately 6.76 million police-reported crashes in the U.S., resulting in 36,355 fatalities [2, 3]. In 2020 and 2021, there were approximately 38,824 and 42,915 fatalities, respectively [4, 5]. The financial impact of these incidents is substantial, with an estimated cost of $242 billion attributed to crashes, prompting the U.S. Department of Transportation to address road safety as a critical national public health concern.

According to studies, more than 25% of all traffic fatalities and more than 50% of traffic injuries occur at or near intersections. According to NHTSA, roughly 22% of intersection-related crashes occur when the vehicle is turning left, 12% while crossing over, and 22% while traveling off the road [6, 7]. These crashes result in significant damage in the form of human injury, property damage, and economic costs [4, 5]. Intersection-related crashes are common because multiple approaching vehicles from two or more intersecting roads create the potential for conflicts. Maneuvers such as turning left, turning right, crossing over, or yielding to other vehicles, pedestrians, and cyclists often increase the risk of crash occurrence. Intersections are a special roadway segment where different streams of traffic intersect as well as their speed and directions change.

Several factors can affect the likelihood of intersection crashes, including traffic control devices, weather and road conditions, critical pre-crash events, and driver-related information. Driver behavior in safety critical event also plays key role [8-10]. These factors may also include infrastructure-specific characteristics such as number of lanes, width of lanes, lighting conditions, traffic signal patterns and timing, and presence of visual obstacles [1, 6, 7]. Driver behavior also plays a major role in intersection-related crashes, such as failing to yield to right-of-way traffic, overspeeding, running a red light, distraction, speeding to avoid a red light, distracted driving, and unsolicited lane changes [11-16].

Understanding the factors contributing to these intersection crashes is crucial in developing effective strategies to prevent them. Analyzing traffic participants' behavior in various intersection scenarios provides valuable insights into the causes of crashes and enables the implementation of targeted preventive measures. Identification of crash-related factors typically rely on lagging indicators, such as police accident reports. Although these reports are informative, identification of driver behavior can be challenging due to reliance on eyewitness testimony and post hoc crash reconstruction. In addition, it can take several years to compile enough data to identify hazardous locations and the underlying issues. Thus, new approaches are required to evaluate the interplay between behavior of vehicles and infrastructure elements that will inform strategies to proactively reduce crashes and their associated injuries and fatalities.

Stakeholders need a system that can analyze traffic safety in real time. The system should be able to automatically identify potential crash-related conflicts, identify the factors associated with the conflict, and provide suggestions for countermeasures. The goal is to build a system that can provide real-time assessment of traffic safety at each intersection, record safety indicators, and track them over time. With the growing volume of traffic, it is important to identify potential hazardous locations even before the crash takes place. The continuous assessment of intersections is therefore a critical first step. By conducting a comprehensive analysis of traffic participants, we can identify patterns and trends that contribute to the occurrence of crashes. Analyzing the behavior of drivers, pedestrians, and cyclists in different scenarios will allow us to understand the prevalence of these risky behaviors and develop strategies to mitigate them. Thus, we can learn intricate patterns from intersections that have recorded an historically high volume of crashes and distinguish them from intersections that are not prone to crashes. Learning traffic patterns across adjacent intersections and the full roadway network as a system will also help alleviate traffic congestion.

Ultimately, understanding and preventing crashes requires a multidimensional approach that considers the complex interactions between various traffic participants and their environment. Analyzing the behavior of drivers including distraction, gaze fixation [17], and secondary behavior [18], behavior of pedestrians, and cyclists in different scenarios through a data-driven method provides a solid foundation for developing evidence-based interventions that can effectively reduce the number of crashes, injuries, and fatalities on our roads.

Project Scope and Objectives

In this project, we first analyzed crash statistics at Virginia intersections; however, the scope of this research goes beyond mere analysis and extends to developing an innovative approach for real-time detection, tracking, and estimation of potential crash situations at intersections. The utilization of the Virginia Department of Transportation's (VDOT) network of traffic video cameras enabled us to implement a graph neural network (GNN) machine learning (ML) model for this purpose. This study specifically focused on signalized intersections.

By leveraging video footage from these cameras, we first deployed modern computer vision methods to detect and track traffic participants (e.g., car, pedestrian, bicycle, truck) in the scene. We then used geometric transformation methods to convert those trajectories in GPS format. This helped us in studying the kinematics of the traffic participants. We next developed a risk analysis method using a GNN. The risk analysis algorithm (i) extracts behavioral features from the kinematics in an aggregated roadway scenario where multiple roadway agents interact with each other; and (ii) uses the behavioral features along with infrastructural features to design a semi-supervised ML model that defines road safety/risk in near real time. The ML model is expected to automatically identify a stable and safe traffic flow condition and differentiate it from safety-critical roadway scenarios. The proposed GNN-based method aims to observe and analyze traffic flow patterns at intersections. Its objective is to identify potential events that may be safety critical.

Through the analysis of various parameters, such as vehicular speed, acceleration, and time to collision (TTC), the system can make predictions regarding these crash factors [19]. It is important to note that these crash factors are already documented in the Virginia FR300 report for each reported crash. This project addressed the following research questions:

1. How do different roadway agents interact in high-conflict intersections?
2. How does the kinematic behavior at any intersection affect the overall safety of the intersection?
3. How can infrastructural elements and driver behavior in roadway safety be modeled?
4. Can a graph-based model be used to model a traffic scene and include all dynamic and static components?
5. How effective are traffic cameras in continuous monitoring of traffic and safety analysis in real time?

Graph-based Analysis of Traffic at Intersections

The inherent complexity of traffic scenes, where numerous elements dynamically interact within the confines of road networks, demands a modeling framework that can effectively capture these intricate relationships and aggregate over time. Graph-based systems offer an ideal solution due to their capacity to represent interconnected entities and their interactions. In traffic scenarios, vehicles, pedestrians, roadways, and signals can all be represented as nodes, while the edges between them depict the various relationships and dependencies. This representation enables quicker analysis, facilitates the utilization of graph theory algorithms, and enables the integration of contextual information. As a result, this holds the power to transform traffic management, enhance safety measures, and optimize transportation systems in the ever-growing congestion of urban environments. Figure 1 shows example of a traffic scene and its representation as a graph, where the graph captures the semantic interaction between the vehicles. The connection between the nodes (vehicles) is determined by their potential interactions. This interaction can be longitudinal (if the other vehicle is at the front or back of the ego vehicle), lateral (vehicle is at the left or right of the ego vehicle), or intersecting (there is a potential conflict inside the intersection while taking a turn). This modeling approach helps bring a physical construct of the traffic scene with its actors to a topological framework. This modeling facilitates the aggregation of information using advanced algorithms like a GNN.

Literature Review

Modeling intersection safety requires a proper amalgamation of multiple fields of research. Building on previous research related to intersection safety and modeling of roadway intersections, we have introduced a graph-based modeling technique to existing data sources and existing infrastructure systems. In this section, we summarize previous work in two major areas: 1) how to model a traffic scene with graph-based methods and GNNs; and 2) traffic safety analysis through advanced methods. To limit the scope of the project, we mainly focused on the most recent research on graph-based methods. This includes work on scene graphs and methods summarizing spatiotemporal systems.

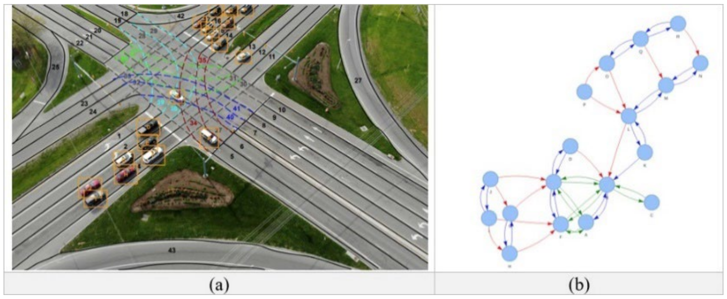


Figure 1. Photo and graph. Modeling traffic using graphs: (a) vehicles labeled for one of the frames from the intersection drone footage from Blacksburg, VA; (b) semantic scene graph for the drone image. Different colored edges indicate different types of edges: Longitudinal (red), lateral (blue), and intersecting (green).

Traffic Scene Representation as a Graph

A “graph” stands as a fundamental and versatile abstract data structure. Graphs provide a powerful framework for representing and analyzing relationships or connections between various entities, making them a cornerstone in solving complex problems. In essence, a graph consists of a collection of nodes (vertices) and the connections (edges) that link these nodes together. The choice of nodes and edges and their characteristics are design parameters to engineers. In recent years, researchers have used different combinations of objects, including vehicles, buildings, pedestrians, bicycles, road elements, and roadside elements, as nodes for modeling [20-24]. As a result, pedestrians and bicycles mainly exhibit topological relationships with roadside elements. On the other hand, vehicles establish multiple relationships with different object types. These relationships encompass topological relations, relative orientation, relative trajectories, relative speed, metric relations, and order relations. Additionally, vehicles also establish connections with structures (such as lane dividers, signboards, traffic signals, and guardrails), road segments (including crosswalks, edge lines, stop lines, center lines, etc.), roadsides, intersections, and road markings. By incorporating these diverse relationships, these models provide a comprehensive representation of the intricate interactions between vehicles and their surroundings. Most researchers have used these objects as nodes. While most studies have reported a 2D representation of the systems, Zhang et al. [25] used 3D modeling of the traffic scene. Edges are modeled to represent the relationship between the objects. These can be their relative positions, velocities, region connection calculus (RCC) , relative orientation, relative trajectories, relative speed, quantitative distance, and order relations [23, 24].

A road scene can be complex depending on many factors. Inclusion of all these elements may not be trivial. To effectively handle graphs composed of subgraphs that do not have significant influence on each other, two approaches are employed. The first approach involves dividing the road into smaller sections, allowing the creation of multiple graphs. These individual graphs can then be connected, forming subgraphs that collectively make up a larger graph. The second approach involves dividing the road segment into non-overlapping sectors. This division can be based on either the sector length or the road geometry itself. By implementing this division, the road segment is explicitly represented as distinct bidirectional carriageways, which may consist of single or multiple lanes. These graphs are used through different ontologies and classification methods. This differs by their scale. Typical classification is node-based classification (behavior of each object), edge classification (relation between two objects/elements), and graph classification (subset of the road segment with all its element).

Traffic Safety Analysis

Computer vision and advanced ML have been widely used for traffic safety analysis and prediction. Chen et al. [26] utilized high-resolution video from traffic cameras, transformed the data into a top-down perspective to facilitate object detection and tracking in small road segments, and computed safety metrics. Candela et al. [27] employed a GNN to enhance vehicle trajectory prediction, subsequently assessing potential collisions. Glasmacher et al. [28] defined three traffic scenario scores, incorporating trajectory data, metadata, and a semantic map. These scores were combined across multiple layers for a comprehensive evaluation. Diehl et al. [22] utilized vehicle information to create features for nodes in a graph. A graph attention network (GAT) was employed to capture relevant interactions among vehicles. Malawade et al. [21] used real-time video feed analysis to predict the likelihood of collisions. This process involved scene-graph creation, spatiotemporal embedding modeling, and an Long Short Term Memory model for crash prediction. Furthermore, Mo et al. proposed a comprehensive framework [20] that considered individual dynamics, interactions, and road structure in predicting trajectories by placing each agent in its own coordinate system to eliminate coordinate shifting discrepancies. Finally, innovative risk estimation approaches using Gaussian mixture models and scenario safety forecasting models were discussed in Jin et al. [29] and Gang and Zhuping [30], respectively. Huang et al. [33] introduced a novel mechanism, rank influence learning (RIL), to address limitations in existing models for spatiotemporal forecasting using deep learning techniques. In this project, we adopted a scene graph-based method that considers dynamics of the traffic actors.

Traffic Modeling Using Graphs

Significant progress has been made in modeling traffic scenarios by adopting a graph-based approach. Graphs provide a versatile framework for representing and analyzing complex relationships among different elements in a system. In the context of traffic scenes, a graph-based representation enables the depiction of various entities as nodes and their interactions as edges.

By constructing a graph representation of a traffic scene, it becomes easier to capture the spatial and temporal relationships between different participants. For instance, vehicles can be represented as nodes, and the edges can denote their proximity or interactions, such as overtaking or following. A graph-based system is object class agnostic; therefore, vulnerable road users including pedestrians and cyclists can also be incorporated into the graph as separate nodes, allowing a comprehensive representation of the entire traffic ecosystem.

The graph-based approach offers several advantages for analyzing traffic scenarios. First, it provides a more compact and structured representation compared to raw video frames. Instead of processing every pixel in each frame, the focus shifts to analyzing the relationships between nodes and edges in the graph, which reduces computational complexity. This enables faster analysis and extraction of relevant information.

Second, the graph-based representation allows the application of various algorithms and techniques from graph theory to gain insights into traffic dynamics. For example, centrality measures can identify influential vehicles or nodes with high traffic density. Graph-based clustering algorithms can group similar entities together, helping to identify traffic patterns or anomalies. Additionally, graph-based simulations can predict the future behavior of traffic participants based on their interactions and historical data. Furthermore, the graph-based approach facilitates the integration of additional contextual information, such as road infrastructure data, traffic regulations, and historical traffic patterns. This enriched representation enhances the understanding of traffic scenarios and enables more accurate analysis and decision-making processes.

Semantic Scene Graphs

A scene graph is a hierarchical data structure commonly employed in computer graphics to represent objects within a scene. Unlike typical graphs, scene graphs possess a specific structure with parent-child relationships, rather than arbitrary connections between nodes and edges. Each node in a scene graph typically contains information about its position, orientation, scale, and other attributes that define its appearance and behavior in the scene.

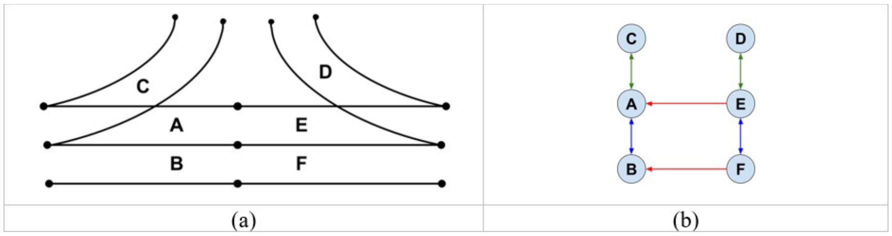


Figure 2. Graphs. (a) Example road intersection: intersections labeled with the different road segments, called "lanelets." (b) Road graph for the road layout shown in (a) with different colors indicating different types of edges: consecutive (red), adjacent (blue), and overlapping (green).

A semantic scene graph (SSG), on the other hand, is a specialized type of scene graph utilized in computer vision to represent a 3D scene in terms of its semantic meaning. In an SSG, every node corresponds to an object present in the scene and is labeled with a category or class of object, such as "car," "person," "bicycle," and so on.

Diverging from a regular scene graph, which solely encodes spatial relationships between objects, an SSG also captures functional and semantic relationships between objects. For instance, it might indicate that a "person" node is "driving" a "car" node or that a "cup" node is "on" a "table" node.

This supplementary semantic information can be leveraged to support an array of applications, including object recognition, scene comprehension, and task planning.

Lanelet Map and the Road Graph

A lanelet divides a road into smaller segments such that the physical road can be identified by a continuous set of indices. Each index represents each lanelet. This lets us use each small road segment, henceforward referred as lanelet, as a single node. These lanelets are homogeneous in nature and their physical attributes, which helps to create a set of roadway nodes for graph generation. Figure 2a presents the creation of a lanelet map by following the established protocol for generating standard bird's-eye view datasets. Using the lanelets, we created a directed graph, as shown in Figure 2b. To construct the directed road graph, we followed the methodology outlined in Zipfl and Zöllner [32]. Each lanelet has been assigned a unique number for easy identification within the road graph.

The graph consists of three types of edges, distinguished by color. Consecutive edges (red) indicate that two road segments/lanelets/nodes are positioned consecutively, with one following the other. These edges are unidirectional, as traffic can only flow in one direction through two consecutive road segments. Adjacent edges (blue) signify that two road segments are positioned next to each other. These edges are bidirectional, allowing traffic to move in both directions. Finally, overlapping edges (green) represent road segments that overlap or cross over each other, sharing a specific area of the road. These are also bidirectional in nature.

Spatial Abstraction

The position and pose of a vehicle in space can be influenced by various factors. In certain situations, a participant may have projections onto multiple road segments, particularly when the vehicle is changing lanes. This scenario is common at intersections where multiple road segments meet and overlap (Figure 3). To accurately represent these projections, each participant's projection is aligned with the corresponding road segment's Frenet coordinates [33].

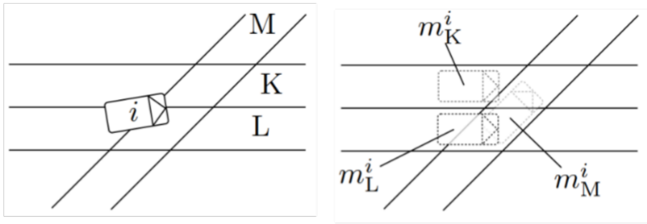


Figure 3. Diagram. Spatial abstraction of a vehicle having projection onto three different road segments. Image source: Zipfl and Zöllner [62].

SSG Creation

In the proposed system, despite participants having multiple projection identities, they are consolidated into a single object node. This ensures that all edges in the graph originate from and terminate at a single node. Let p_{ab} represent a path in the road graph G_{road} connecting two road segments v^{a}_{road} and v^{b}_{road} (represented as nodes in the graph) where two projection identities m^{a}_{road}

and m^b_{road} exist. If all edges in p_{ab} are labeled as "consecutive," it indicates a longitudinal relation (shown in red in Figure 4) between the two traffic participants i and j.

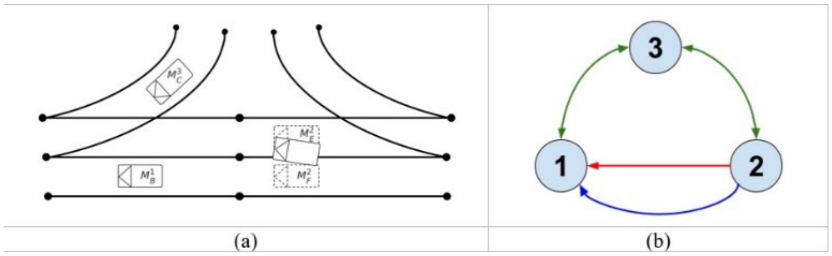


Figure 4. Graphs. (a) Projecting traffic participants onto their corresponding lanelets; Vehicle 2 has multiple projections, as it is partially present in multiple lanelets. (b) Semantic scene graph for the example scenario shown in (a); each node represents an individual traffic participant from the scene, and different colors indicate edges with various semantic relations: longitudinal (red), lateral (blue), and intersecting (green).

When vehicles travel in adjacent lanes, they are linked in the scene graph through a lateral relationship (shown in blue). This means that when a path p_{ab} exists between v^a_{road} and v^b_{road} in G_{road}, where each edge in p_{ab} has the "consecutive" attribute and exactly one edge has the "adjacent" attribute, the corresponding traffic participants in the scene graph are labeled with the attribute k_{rel} = lateral.

Intersecting (shown in green) traffic participants i and j (k_{rel} = intersecting) are those traveling in lanes that will either overlap or merge. This condition is met when there exists a path p_{ab} in G_{road} where each edge in p_{ab} is labeled either "consecutive" or "adjacent" and exactly one edge is labeled "overlapping." It is important to note that once an edge with the "overlapping" attribute is included in p_{ab}, all subsequent edges must be reversed.

To maintain a realistic representation, the length of the path $|p|$ is limited (usually set to 30 meters). This prevents the inclusion of edges between traffic participants that are significantly distant from each other.

Data Collection and Processing

This section includes the methodology employed to create a graph-based model solely from infrastructure cameras. First, we introduce the traffic camera dataset and selection of intersections. Next, we discuss the computational process that used computer vision and roadway structure information. The computer vision process included object detection, object tracking, and homography transformation. The detection and tracking methods provided trajectory information for each dynamic actor on the road in an image coordinate system. The homography transformation helped convert the image pixel information to GPS coordinates. Next, we used the roadway information, including from OpenStreetMap (OSM), to model each intersection into a lanelet-based structure. This allowed us to relate an object to the structure of the road.

Selection of High-risk Intersections

We utilized publicly available datasets, including the VDOT Crash Analysis Tool website and virginiaroads.org containing crash statistics, to analyze and identify (a) intersections with a history of high crash occurrence and (b) another set of intersections with a history of low crash occurrence but high vehicle volume. We shortlisted two sets of intersections with similar traffic volumes. The first set had a history of high crash occurrence, while the second set had low crash occurrence. Distribution of severe crashes was compared with respect to the multiple parameters obtained from the crash database, including time of day (by hour), day of week, and month of year,

The riskiest five intersections were obtained from the Virginia cities of Portsmouth (1), Hampton (3), and Newport News (1). The team located other high-risk intersections in Virginia Beach, Richmond, and Fairfax. We selected a total of 20 (comprised of 10 high-crash and 10 low-crash) intersections for further inspection. In addition to their rankings, additional parameters such as the number of lanes and average annual traffic volume were considered during the inspection process:

i. Signalized control (i.e., controlled by traffic lights)
ii. At least 20-25 lanes (incoming and outgoing combined)
iii. Moderate/heavy annual average daily traffic volume (estimated, >20,000)

We further used the GPS locations of the cameras and the crash locations to verify the selection of the intersection. In this report, we present our analysis of the VT-CAST (Traffic Cameras for Advanced Safety Technologies) 2020 dataset [34]. Traffic cameras spread throughout Virginia stream live video feeds on the VDOT server. These streams are recorded in segments of 1-hour videos to form the dataset. These cameras are positioned to capture a wide field of view and offer an oblique perspective that allows visibility of the surrounding road area. While the cameras do not provide a top-down view, they ensure comprehensive monitoring of traffic conditions. The cameras solely offer raw video feeds and do not provide specific information regarding the kinematics or movement patterns of the traffic participants in the observed scenario.

As this project used camera data, we further used the camera quality and movement as selection criteria. We manually reviewed and selected camera data to check whether the data was valid. Often cameras are rotated and do not capture the traffic intersection and traffic dynamics. We chose only those videos that captured relevant information. A more detailed analysis of the data collection and intersection selection can be found in Appendix C: Virginia Intersection Data.

Object Detection and Tracking

Object detection and tracking play a crucial role in determining the precise location of participants through the identification of bounding boxes [35] . However, in scenarios where the video quality is poor, the object detection process is prone to generating both false positives and false negatives. Research shows that transfer learning can help overcome this challenge [17, 18, 36 – 38], partially, hence shows promises. However, it becomes necessary to explore and evaluate different algorithms to obtain the most effective model. We tested a total of three algorithms: (a)

MMDetection + Graph Convolutional Neural Network Match (GCNNMatch) [39-41]; (b) Global Tracking Transformer (GTR) [42] ; and (c) YOLOv7 + BoT-SORT [43, 44] (Simple Online and Realtime Tracking; more details can be found in Appendix A: Object Detection and Tracking). The primary objective of object detection and tracking is to accurately identify and locate participants within a given video. This task involves detecting objects of interest, such as vehicles, pedestrians, or other relevant entities, and creating bounding boxes around them. Figure 5a shows an example of the object detector and tracker.

The participant's approximate position can be best estimated by identifying the point closest to the ground. When working with data from a monocular camera, it is not possible to construct a 3D bounding box with a ground plane. Therefore, we obtained an estimation of each participant's position by considering the midpoint of the lowermost edge of the bounding box (see Figure 5b). This approximation allows us to represent a participant with a single point, minimizing noise. This approach is superior to using the centroid of bounding boxes, as the centroid may be elevated from the ground, resulting in an inaccurate representation of the participants' positions.

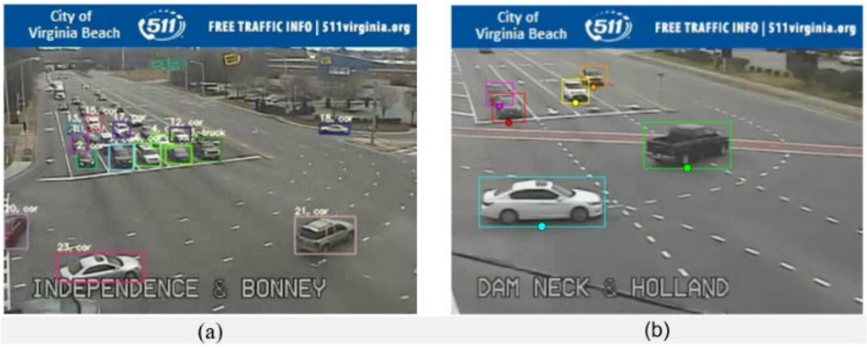


Figure 5. Video images. Examples of (a) object detection and tracking and (b) centroid detection to identify their ground location.

Pixel-to-GPS Transformation

Pixel-to-GPS conversion involves transforming an image captured by a camera into geographic coordinates using GPS data. This technique is widely used in the field of remote sensing and allows researchers to obtain precise geolocation information from satellite or aerial imagery. This method maps any pixel in the camera into a GPS coordinate. In this effort, we first made manual annotations between the Google Earth view of an intersection and an image of the same intersection from the camera. Then, we computed a homography matrix that transformed the image pixels to a Google Earth image. This helped to identify the GPS position of a traffic actor when it is identified in a specific pixel location. Figure 6 shows an example of the point-based matching between the two views.

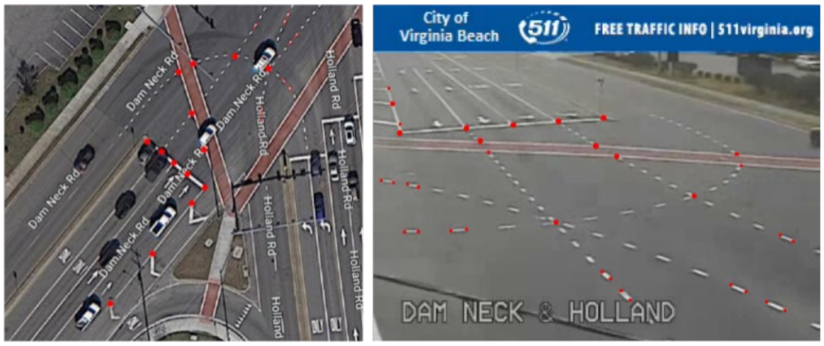


Figure 6. Photo and video image. Matching points from the Google Maps top-down view (left) and camera view (right) for homography computation.

Lanelet Design of an Intersection

Next, we used the Lanelet2 [45] library to segment the road structure into smaller segments (i.e., lanelets). This would allow us to model the roadway as a graph with node-edge formulation where each node is homogeneous in nature. We additionally used OSM [46] and JOSM (Java OpenStreetMap Editor) [47]. OSM is an inclusive and collaborative mapping initiative that strives to develop a comprehensive, freely accessible map of the entire world. JOSM helps to build an additional map from the information received from OSM, including GPS. We used these tools and libraries to manually annotate intersections into smaller segments.

The core data structure in the Lanelet2 library is the lanelet, which represents an individual drivable lane on the road. Each lanelet can be visualized as a segment of the road that includes its left and right boundaries, determining the flow of traffic. Moreover, each boundary is assigned a property of "virtual" or "road boundary." The virtual property means that the boundary is between two sets of lanes moving in the same or opposite direction. The road boundary property means that the boundary is between a road and a non-road surface. These boundaries are not mere lines but are additionally annotated with relevant associated traffic rules. This comprehensive annotation approach enables the Lanelet2 model to provide a nuanced and thorough understanding of the road environment.

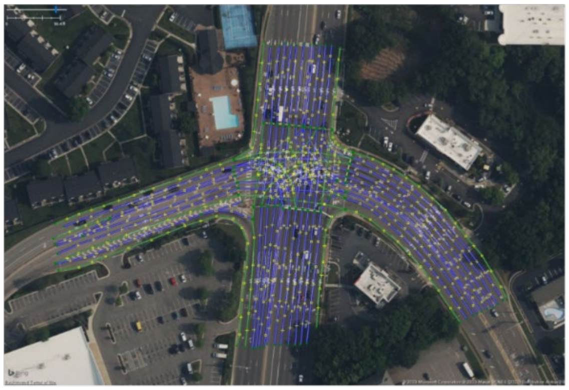


Figure 7. Aerial photo. Lanelets created for one of the risky intersections in Virginia Beach.

Figure 7 shows the lanelet maps created for one of the risky intersections in Virginia Beach. The map has been overlayed on Bing Maps aerial imagery to better visualize the lanelets according to the actual road structure. Next, we utilized Frenet coordinates (See Appendix D: Technical Background), which offer a more intuitive representation of a position on a road compared to the traditional Cartesian coordinates. By utilizing these concepts, we can accurately represent the position of vehicles on the road using Frenet coordinates.

Safety Analysis of Intersection Using a GNN

In this section, we introduce a modeling technique for traffic safety using a GNN. We used the graph structure and kinematic feature extraction methods introduced in the previous sections and integrated them to develop a safety score. Finally, using real-world intersection data, we demonstrated the effectiveness of the safety modeling. Figure 8 shows the full schematic depicting the components and features that we used to derive the risk score model.

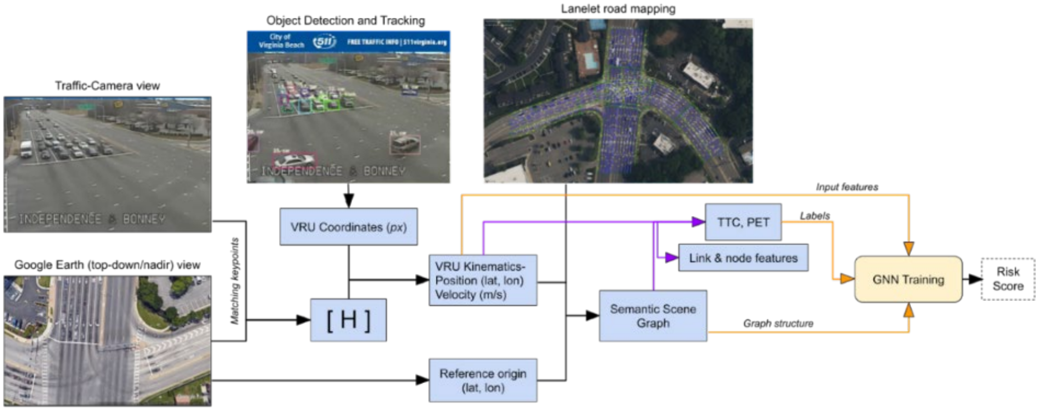


Figure 8. Diagram. Comprehensive overview of the entire pipeline for our proposed approach.

GNNs

GNNs have gained widespread use across various domains due to their ability to learn from data modeled as a graph or network data with nodes and edges. Over the past few years, there has been a significant surge in the growth and application of GNN models. A GNN is a dedicated neural network model that is created only for graphs. GNNs have gained prominence owing to their ability to effectively capture intricate relationships and dependencies within graph-structured data, which is prevalent in numerous real-world scenarios such as social networks, recommendation systems, and biological networks. A GNN operates by aggregating information from a node's neighbors and iteratively updating node representations, allowing it to learn complex patterns and features inherent in the graph. This dynamic, recursive nature of GNNs enables them to perform tasks such as node classification, link prediction, and graph classification. One major advantage of GNNs is that the training is transferable across structures. Once the GNN is trained on a specific set of the training graphs or domain, the learnings are transferable to similar graphs and domains. This characteristic makes it useful for applications like traffic scenes, where we can train on a set of intersections, and then transfer the knowledge onto other intersections. Also, a GNN does not need enormous amounts of data like other deep neural network models do.

Model Selection

Selecting the appropriate GNN capable of integrating node and edge embedding information is of utmost significance. In our research, we relied upon the GNN cheat sheet offered by PyG [48] as a valuable reference. For this work, we selected two state-of-the-art models: TransformerConv and GINEConv (Graph Isomorphism Network with Edges). TransformerConv is a novel attention-based model that was proposed in Shi et al. [49]. It is designed to be more efficient and effective than traditional attention mechanisms for graph-structured data. GINEConv [50] is a graph convolutional neural network (GCNN) that uses a novel message passing mechanism to aggregate node features. The message passing mechanism is based on the Graph Isomorphism Network (GIN) [51], but it also incorporates edge features. This allows GINEConv to learn more complex representations of graphs than GNNs that do not use edge features.

Results and Discussion

Node and Link Features in Elementary Graph

To assess the traffic density of a specific traffic scene, we employed a simplified graph representation. We adopted this approach due to the presence of multiple edges with distinct properties between two nodes in the SSG, where network theory concepts fail to differentiate between these various edge types. As a result, we employed a preliminary method that established connections between a node and other nodes within its vicinity. In this graph, each node represents a traffic participant, and edges are established between any two distinct traffic participants (u, v) if their mutual Euclidean distance is less than a predefined margin value (see Figure 9).

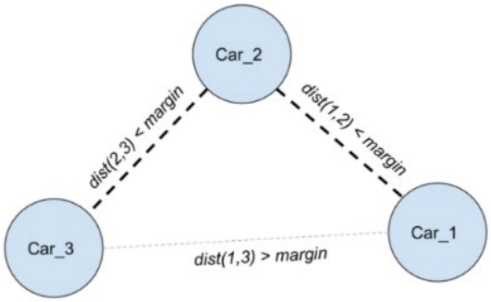


Figure 9. Diagram. Creation of a graph structure using roadway participants.

Table 1 and Table 2 present the node and link features, respectively, for one of the intersections in Virginia. This particular intersection has a documented history of a high frequency of crashes. The node and edge features provided in the tables were obtained from a 50-minute video feed captured during the afternoon. The distance margin was chosen as 10 meters for creating the graph, as 10 meters approximates 1 second of headway for a 25-mph road. Based on the obtained graph, these features were calculated. The details of all measures in the tables can be found in Appendix E: Network Theory.

Table 1. Average Values for Various Node-level Features such as Degree, Centrality, and Clustering Coefficient Computed for One of the Videos from the VDOT Database

Node	Count	Degree	Centrality			Clustering coefficient
			Eigenvector	Betweenness	Closeness	
Car	8177	2.48659	0.18796	0.03496	0.24433	0.32658
Truck/Bus	3110	2.57098	0.21295	0.02449	0.25161	0.3795
Pedestrian	120	1.46054	0.10737	0.016178	0.18812	0.25529
Bicycle	23	4.81395	0.19966	0.05233	0.41798	0.42497
Motorcycle	23	2.6279	0.24813	0.05588	0.3292	0.40337

Table 1 presents some interesting characteristics about the traffic scene; it is evident that cars, trucks, and motorcyclists typically have an average degree of approximately 2.5. This suggests that, on average, they are surrounded by two to three other vehicles or traffic participants. On the other hand, bicyclists have a higher degree because they share edges with both pedestrians and vehicles. Consequently, cyclists are generally at a higher risk, as they have a greater number of traffic participants in proximity.

Moreover, among all the road users, bicyclists experience the least amount of safety. This is primarily due to their higher exposure to nearby traffic participants and the inherent risks associated with sharing the road with vehicles. In contrast, pedestrians typically walk on sidewalks and thus have the fewest number of connections or edges with other participants. Additionally, it appears that there are only a limited number of pedestrians captured in the video data.

Eigenvector centrality measures a traffic participant's significance within a network. Examining Table 1, it becomes evident that pedestrians hold the lowest influence on the network, while other participants exhibit relatively comparable levels of influence. Notably, motorcyclists emerge as the group with the most substantial impact.

Closeness centrality is a measure that quantifies how close a traffic participant is to other participants within a network. It is calculated based on the distances between the participant and all other participants. Here, distance is the length of the shortest path between two nodes in the graph. The distance is based on an adjacency matrix where two nodes connected directly have a distance of 1, and so on. The values of closeness centrality follow a similar trend as the average degree for different traffic participants in the network.

The clustering coefficient is a measure that indicates the tendency of nodes in a network to form clusters or groups. It is observed that bicycles exhibit the highest tendency to form clusters, followed by motorcycles. In contrast, pedestrians have the least inclination to form such clusters.

Table 2. Average Values for Various Link-level Features such as Jaccard's Coefficient, Adamic-Adar Index, and Katz Index Computed for One of the Videos from the VDOT Database

Node 1	Node 2	Jaccard	Adamic-Adar
Bicycle	Car	0.20755	0.85407
Bicycle	Pedestrian	0.05555	0.2103
Bicycle	Truck/Bus	0.19010	0.89363
Car	Car	0.2341	0.82552
Car	Motorcycle	0.21041	0.72338
Car	Pedestrian	0.191915	0.5831
Car	Truck/Bus	0.25405	0.89782
Motorcycle	Pedestrian	0.03809	0.16156
Motorcycle	Truck/Bus	0.2	0.63092
Pedestrian	Truck/Bus	0.12771	0.43341
Truck/Bus	Truck/Bus	0.29414	0.99186

Based on the local-overlap data (Jaccard's coefficient and Adamic-Adar index) presented in Table 2, motorcycles and pedestrians, and bicycles and pedestrians have the least number of edges with common traffic participants. This could be because most of the edges of traffic participants are with cars, given the high number of cars in the video. The values for the various other node types are very similar. It is additionally observed that in the case when two trucks/buses are present in the frame, there are more common traffic participants than usual.

The Katz index is typically used in network analysis to measure similarity or overlap between nodes based on their connectivity patterns within a network. It is commonly applied to social networks, where nodes represent individuals and edges represent relationships between them. In such networks, the Katz index can capture the degree of similarity or overlap in terms of shared connections. However, in a traffic network, where nodes represent traffic participants and edges represent proximity, the concept of similarity or overlap between nodes may not be meaningful in the same way. In this context, the focus is on proximity and interaction between participants rather than shared connections. In traffic networks, other measures such as traffic flow, congestion, shortest paths, or centrality measures like betweenness centrality or closeness centrality may be more relevant for understanding the dynamics and efficiency of the network.

Risk Score Design Using Traffic Dynamics

From the VT-CAST 2020 dataset, we carefully chose two intersections for our study. One of these intersections had a significant history of crashes, making it statistically notable. Conversely, the other intersection had a minimal record of past crashes. To ensure a comprehensive analysis, we selected three videos from each intersection. For the training subset, we utilized two videos from each intersection, while one video from each intersection was allocated to the test subset. To maintain consistency, all videos were recorded at a frame rate of 15 frames per second.

Given our specific focus on intersections, we concentrated on consolidating data related to overspeeding, rapid acceleration, rapid deceleration, and the number of traffic rule violations concerning TTC and post-encroachment time (PET). More details on the safety metrics can be found in Appendix D: Technical Background. These various factors were combined to assign a binary label, either "risky" or "non-risky," to each frame in the dataset. Equation 1 shows the combination of the different safety parameters used to define the risky or non-risky label. The variable c_f denotes the class label for a frame *f* from the video. A value of 1 denotes that the situation is risky, and 0 denotes that it is non-risky.

$$\boldsymbol{c_f = \min\left(1,\ floor(0.5 \times n_{PET} + 0.25 \times n_{TTC} + 0.5 \times n_G + 0.5 \times n_S)\right)} \qquad [1]$$

Here, we define the variables as follows: n_{PET} represents the count of vehicles violating the PET metric, and n_{TTC} represents the count of vehicles violating the TTC metric. It is important to note that both n_{PET} and n_{TTC} will always be multiples of 2, as these metrics involve two vehicles. Additionally, we have n_G and $n_C$indicating the number of vehicles engaging in rapid acceleration or deceleration and overspeeding, respectively. The ratios for the different parameters are carefully selected to ensure that a single collision-prone situation is enough to label the entire frame as unsafe. If there is even one violation of the TTC metric, involving two vehicles, the situation is classified as risky regardless of the status of other metrics. However, a TTC violation is taken into account in conjunction with factors such as overspeeding, rapid acceleration, or deceleration. This is because TTC violations commonly occur in intersection scenarios. If the TTC value is violated along with either excessive speed or abrupt acceleration, the situation can be considered unsafe. The thresholds are shown in Table 3.

Table 3. Thresholds for Intersection Scenarios

TTC	PET	Acceleration	Deceleration
2 sec	1.5 sec	0.6 g	0.5 g

Node and Edge Features

A graph was created to represent each frame of both the training and test dataset splits. To create the SSG structure, we adopted the method proposed by Zipfl and Zöllner [32]. Once the structure was established, we assigned node and edge features. In this graph, each traffic participant corresponded to a node, and we represented each node embedding with a 9-dimensional feature vector.

The feature vector for each node consists of the following components: the first four elements encode the participant class using one-hot encoding, representing pedestrian, bike, truck/bus, or car. The remaining elements in the vector represent the magnitude of velocity, the *x*-component of velocity, the *y*-component of velocity, and the length and width of the participant. Equation 2 provides a clear representation of this feature vector.

$$\boldsymbol{u} = [\boldsymbol{isPed}, \boldsymbol{isBike}, \boldsymbol{isCar}, \boldsymbol{isTruckBus}, |\boldsymbol{v}|, \boldsymbol{vx}, \boldsymbol{vy}, \boldsymbol{length}, \boldsymbol{width}] \quad [2]$$

Additionally, the edges in the representation are characterized by a 6-dimensional feature vector. Within this vector, the elements at odd indices serve as one-hot representations indicating whether the edge is longitudinal or not, lateral or not, or intersecting or not (Figure 9). On the other hand, the even-indexed elements correspond to the distances between the two participants specifically for each edge type. It is important to highlight that these distances are not measured in terms of Euclidean distance, but rather they reflect the distances along the curve of the road. If that type of edge does not exist, the corresponding distance value in the feature vector is simply zero. Equation 3 provides a clear representation of this feature vector.

$$\boldsymbol{e} = [\boldsymbol{isLon}, |\boldsymbol{dlon}|, \boldsymbol{isLat}, |\boldsymbol{dlat}|, \boldsymbol{isInt}, |\boldsymbol{dint}|] \quad [3]$$

Creating the Dataset

The dataset was meticulously constructed by sequentially selecting approximately 2,000 timestamps for the training dataset and around 1,500 timestamps for the test dataset from one risky and one non-risky Virginia Beach intersection. The videos were chosen with the best lighting and under normal weather conditions, as the videos already have a low-resolution, and we did not want to introduce additional noise. The chosen intersections have their respective camera directly facing one of the roads leading to the intersection, which also covers vehicles entering and leaving the intersection. Because the videos were recorded at a frame rate of 15 frames per second, we obtained one frame (timestamp) approximately every 66.67 seconds. Given our specific focus on intersections, we determined that under-speeding had limited relevance for our analysis. Instead, we concentrated on consolidating data related to overspeeding, rapid acceleration, rapid deceleration, and the number of TTC and PET traffic rule violations. These various factors were combined to assign a binary label, either risky or non-risky, to each frame in the dataset.

Risk Estimation Using SSGs

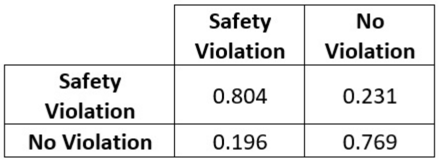
Next, we trained the GNN to estimate the risk of any intersection. A hyperparameter search was conducted for both selected models, and the best results obtained with the optimal hyperparameters are summarized in Table 4. This includes a total of approximately 80,000 frames of video. For each frame, we created a semantic graph and used them for classification. All models were trained for 100 epochs on a Linux machine with a 16-GB Nvidia V100 GPU. Figure 10 shows the confusion matrix of classifications for the GINEConv model.

Table 4. Comparison of Performance on the VT-CAST 2020 Dataset Against Two State-of-the-art GNN-based Methods

Performance Metric	TransformerConv [72]	GINEConv [73]
Accuracy	72.9%	79.85%

An important hyperparameter to consider when optimizing a GNN model is the number of layers. This parameter determines the extent to which a node can gather information from its neighboring nodes and the corresponding edges. When the number of layers is set to 1, the network can only aggregate information from its immediate neighbors, which constitutes a one-hop neighborhood.

On the other hand, if the number of layers is set to 2, a node can gather information not only from its direct neighbors but also from the neighbors of its neighbors, expanding the reach of information aggregation. In this work, we obtained the optimal number of layers as 3 for the GINEConv model, and 2 for the TransformerConv model. Although attention mechanisms from TransformerConv enable the model to selectively prioritize information from node and edge embeddings, it has been observed that convolutional approaches, GINEConv in particular, yield significantly superior performance. This discrepancy can be attributed to the relatively low dimensionality of the input data (node and feature embeddings).



	Safety Violation	No Violation
Safety Violation	0.804	0.231
No Violation	0.196	0.769

Figure 10. Confusion matrix for the GINEConv method evaluated on the VT-CAST 2020 dataset. The ground truth labels are on the horizontal axis, and the predicted labels are on the vertical axis.

Conclusions

In this project, we have proposed a new analysis method to study traffic intersection safety using a GNN and existing camera infrastructure. Traffic fatalities and crashes at intersections are a burning issue and need attention. With the ever-evolving traffic dynamics, higher traffic volume, and impending deployment of automated driving systems, it is important to develop a comprehensive method that can automatically analyze safety at intersection in real time. Over the last two decades, many researchers have focused on using camera-based data to study traffic at intersections. In this work, we specifically showed how information from infrastructure cameras can be used for traffic safety monitoring. These cameras have the capabilities to capture a plethora of information; however, we still lack constructive methods to convert this information to traffic safety-related features and practical applications for the benefit of the public.

Key Contributions

Use of computer vision: In this project, we first showed how modern computer vision methods can be used to process traffic videos in real time. As deep neural networks can operate in real time, several methods can be used to identify and track traffic objects. With the help of computer vision, this information can also be used to find out the kinematics of each of the objects. In recent times, many commercial systems are available that use high quality images (e.g., MioVision). However, these deployments can be costly. In this project, our focus has been traffic cameras that are already deployed in the intersection. We have shown that computer vision can enhance the potential of these cameras and be used for automatic detection and tracking of traffic participants.

Modeling of object interactions using graph: Next, we showed how the interaction of these objects can be modeled through a graph-based method. We have demonstrated a method that can help in modeling the traffic as well as the roadway network separately. This helps us to model all possible traffic actors, all possible traffic configurations, and all possible roadway designs. Therefore, this method introduces a generalized approach to model traffic that opens up enormous potential for research in the field of traffic modeling. Modeling traffic using graphs also provides key information about traffic characteristics and the impact of each actor (vehicle, pedestrians, etc.) through simple graph analysis (see Appendix E: Network Theory).

Safety prediction and analysis using GNN: Next, we proposed a new safety assessment method that uses a GNN and provides risk assessment of any intersection. Graphs were utilized to depict absolute and relative kinematic parameters. By examining the dynamic changes within these graphs, it became possible to forecast potential traffic characteristics and anomalies and identify risky situations. For instance, a contracting graph indicated the presence of traffic congestion, whereas an expanding graph indicated an increased separation distance between vehicles in a particular area. Modeling of traffic through graphs also shows the relative importance and vulnerability of each participant, including vulnerable road users. As we lacked enough annotations for safety critical events, we presented a semi-supervised method. Finally, we presented experimental results through real intersection traffic data on how safety features are computed using the proposed model. The code is made available at the GitHub repository referenced at the end of this report.

Discussion and Recommendations

In this project, we have demonstrated how traffic cameras, computer vision, and a GNN can be used to define safety for dynamic traffic in real time. The field of ML has shown progress in processing complex data structure efficiently. While the process shows tremendous promise in automating safety assessment, this section provides our recommendations to enhance the performance of this model. These recommendations relate to the sensor quality, effect of noise, environmental uncertainty, modeling assumptions, and practical limitations.

Image quality and noise: The current camera infrastructure, characterized by its low resolution of 320x240, falls significantly short of meeting the demands of modern imaging needs. This limitation becomes particularly evident as the cameras frequently lose focus in various situations. Moreover, the image quality deteriorates further under challenging conditions such as low-light environments during nighttime or adverse weather conditions like rain. As a result, the existing camera infrastructure struggles to provide clear and detailed visuals, hindering its effectiveness and reliability.

Upgrading the cameras to higher resolutions would greatly improve the clarity and level of detail in captured images and videos. This enhancement would enable better identification and recognition of traffic participants. Incorporating image sensors with improved low-light performance, such as larger pixel sizes or backside-illuminated (BSI) sensors, would significantly reduce noise and enhance image quality in dimly lit environments. This enhancement would enable cameras to capture detailed visuals even during nighttime or in poorly lit areas. Another improvement could be changing the video compression. Due to limited bandwidth, high compression is used. This significantly affects the performance of computer vision algorithms. Facilitating better compression algorithms may significantly improve performance.

Sensor field of view and completeness: Most of the intersection cameras capture the dynamics of the intersection partially. Often only one camera is installed at an intersection. This does not provide complete information of the traffic dynamics from all directions. This limits the capability of any automated method to identify risks at all sides of the signalized intersection. Also, the field of view for the cameras is often narrow. An ideal solution would deploying a system that can provide a bird's-eye view of the intersection. By employing drones to capture datasets at local intersections, we can acquire a refined dataset that offers improved quality, cleanliness, and minimal noise. This dataset would serve as a valuable resource for evaluating traffic behavior.

Adequate annotations of safety critical events: While sensor data is one major challenge for automated risk prediction, we lack enough annotated data that depicts safety. As most of our research is based on crash statistics and police accident reports, we lack data specifically indicating whether a given scenario is unsafe or prone to crashes. In this project, we relied on spatial and temporal metrics derived from traffic forecasting and proposed a semi-supervised approach. Nevertheless, these models are built on various assumptions, such as vehicle trajectory and maintenance, which means we lack an accurate representation of what constitutes an unsafe scenario. Furthermore, these metrics can limit the representation for all types of conflicts, focusing on those resulting from vehicles crossing the same area or following closely behind one another.

Advanced safety metric design: The current project used basic safety measures like TTC, PET, and speed behavior. In recent years, more advanced safety features including responsibility sensitive safety [52, 53] have been proposed. These measures can help in detailed safety scores.

Additional Products

Education and Workforce Development Products

The outcome of this project will be condensed into a guest lecture presented at VTTI to Dr. Zachary Doerzaph's graduate class on advanced vehicle technologies. Dr. Sarkar will deliver this lecture during Fall 2023. The team is planning to propose a workshop on traffic safety challenges in intersections using aerial imagery during the upcoming IEEE Intelligent Vehicle Symposium, 2024. Currently, it is planned in collaboration with Tsinghua University.

Technology Transfer Products

The project has produced a master's thesis, a new drone-based dataset, and a public codebase. One paper is also submitted to the Transportation Research Board, and it is currently under review. A journal submission has been planned in *IEEE Transaction on Intelligent Transportation Systems*, and it will be submitted in Fall 2023. Two chapters were developed through this project in the master's thesis of Akash Sonth [54].

Public GitHub repository of the code: https://github.com/VTTI/GNN-based-intersection-safety

Data Products

Finally, we developed a new drone dataset that covers four signalized intersections in Blacksburg, VA. This data also includes annotations of objects and signal states for all four directions. Videos are around 30 minutes long. The data also includes signal information from two different cameras. We thank Professor Hong Wang from Tsinghua University for providing annotations. Part of the data is released with this project. The final data will be updated by end of this year (more details in Appendix F: New Drone Data from Virginia, and discussion of the benefits and prior work is available in Appendix B: Drone-based Traffic Dataset). The project has generated one dataset. This dataset contains aerial video of four intersections in Blacksburg, VA.

References

1. NHTSA, Crash Factors in Intersection-Related Crashes- An On-Scene Perspective. US DOT, 2010.

2. Singh, S., Critical reasons for crashes investigated in the national motor vehicle crash causation survey. 2015.

3. Administration, N.H.T.S., Fatality analysis reporting system encyclopedia. NHTSA, 2009.

4. Virginia, D., 2019 Virginia Crash Facts. 2020.

5. NHTSA, Overview of Motor Vehicle Crashes in 2020. 2021.

6. Choi, E. H. (2010). Crash factors in intersection-related crashes: An on-scene perspective (No. HS-811 366).

7. https://highways.dot.gov/safety/intersection-safety/about.

8. Sarkar, A., Hickman, J. S., McDonald, A. D., Huang, W., Vogelpohl, T., & Markkula, G. (2021). Steering or braking avoidance response in SHRP2 rear-end crashes and near-crashes: A decision tree approach. Accident Analysis & Prevention, 154, 106055

9 Sarkar, A., (2022). A Comprehensive Safety Analysis for Gaze Fixation of Drivers to Outside Scene. In 13th International Conference on Applied Human Factors and Ergonomics

10 Sarkar, A., Jain, S., Sudweeks, J., & Perez, M. (2023). 2 Driver Attention Modeling Through Evidence Accumulation and Gaze Fixation. Towards Human-Vehicle Harmonization, 3, 13.

11 Sarkar, A., Alambeigi, H., McDonald, A., Markkula, G., & Hickman, J. (2021, September). Role of peripheral vision in brake reaction time during safety critical events. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 65, No. 1, pp. 695-699). Sage CA: Los Angeles, CA: SAGE Publications.

12 McDonald, A. D., Sarkar, A., Hickman, J. S., Alambeigi, H., Vogelpohl, T., & Markkula, G. (2021). Modeling driver behavior during automated vehicle platooning failures.

13 Klauer, S.G., et al., Distracted driving and risk of road crashes among novice and experienced drivers. New England journal of medicine, 2014. 370(1): p. 54-59.

14. Dingus, T.A., et al., Driver crash risk factors and prevalence evaluation using naturalistic driving data. Proc Natl Acad Sci U S A, 2016. 113(10): p. 2636-41.

15. Klauer, C., et al., The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data. 2006.

16. Guo, F., et al., Near Crashes as Crash Surrogate for Naturalistic Driving Studies. Transportation Research Record: Journal of the Transportation Research Board, 2010. 2147(1): p. 66-74.

17 Bhagat, H., Jain, S., Abbott, L., Sonth, A., & Sarkar, A. (2023, June). Driver gaze fixation and pattern analysis in safety critical events. In 2023 IEEE Intelligent Vehicles Symposium (IV) (pp. 1-8). IEEE.

18 Sonth, A., Sarkar, A., Bhagat, H., & Abbott, L. (2023, June). Explainable Driver Activity Recognition Using Video Transformer in Highly Automated Vehicle. In 2023 IEEE Intelligent Vehicles Symposium (IV) (pp. 1-8). IEEE

19 Lee, D.N., A Theory of Visual Control of Braking Based on Information about Time-to-Collision. Perception, 1976. 5(4): p. 437-459.

20 Mo, X., Xing, Y., & Lv, C. (2021). Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction. arXiv preprint arXiv:2106.07161.

21 Arnav Vaibhav Malawade, Shih-Yuan Yu, Brandon Hsu, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad Abdullah Al Faruque. Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction. IEEE Internet of Things Journal, 9(12):9379–9388, 2022

22 Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll. Graph neural networks for modelling traffic participant interaction. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 695–701. IEEE, 2019

23 Kamaldeep Singh Oberoi, Géraldine Del Mondo, Yohan Dupuis, and Pascal Vasseur. Spatial modeling of urban road traffic using graph theory. In Proceedings of Spatial Analysis and GEOmatics (SAGEO) 2017, pages 264–277, 2017

24 Martin Buechel, Gereon Hinz, Frederik Ruehl, Hans Schroth, Csaba Gyoeri, and Alois Knoll. Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles. In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 1471–1476. IEEE, 2017

25 Hongyi Zhang, Andreas Geiger, and Raquel Urtasun. Understanding high-level semantics by modeling traffic patterns. In Proceedings of the IEEE international conference on computer vision, pages 3056–3063, 2013

26 Xiwen Chen, Hao Wang, Abolfazl Razi, Brendan Russo, Jason Pacheco, John Roberts, Jeffrey Wishart, Larry Head, and Alonso Granados Baca. Network-level safety metrics for overall traffic safety assessment: A case study. IEEE Access, 2022

27 Eduardo Candela, Yuxiang Feng, Daniel Mead, Yiannis Demiris, and Panagiotis Angeloudis. Fast collision prediction for autonomous vehicles using a stochastic dynamics model. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pages 211–216. IEEE, 2021

28 Christoph Glasmacher, Robert Krajewski, and Lutz Eckstein. An automated analysis framework for trajectory datasets. arXiv preprint arXiv:2202.07438, 2022

29 Sheng Jin, Dian-hai Wang, Cheng Xu, and Dong-fang Ma. Short-term traffic safety forecasting using gaussian mixture model and Kalman filter. Journal of Zhejiang University SCIENCE A, 14(4):231–243, 2013

30 Ren Gang and Zhou Zhuping. Traffic safety forecasting method by particle swarm optimization and support vector machine. Expert Systems with Applications, 38(8): 10420–10424, 2011

31 Yujun Huang, Yunpeng Weng, Shuai Yu, and Xu Chen. Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. In 2019 18th IEEE International conference on trust, security and privacy in computing and communications/13th IEEE International conference on big data science and engineering (TrustCom/BigDataSE), pages 678–685. IEEE, 2019

32 Maximilian Zipfl and J Marius Zöllner. Towards traffic scene description: The semantic scene graph. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pages 3748–3755. IEEE, 2022

33 Chen Peng. Path planning in frenet frame. URL https://caseypen.github.io/posts/2021/01/FrenetFrame/

34 Bareiss, M. G. (2023). A Dataset of Vehicle and Pedestrian Trajectories from Normal Driving and Crash Events in One Year of Virginia Traffic Camera Data (Doctoral dissertation, Virginia Tech).

35 Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017

36 Winkowski, C., Sarkar, A., Hickman, J., Abbott, L., “Residual Network-Based Driver Gaze Classification In Naturalistic Driving Studies”, Transportation Research Record (Accepted)

37 V. Sundharam, Sarkar, A., Hickman, J., Abbott, L., " Characterization, Detection, And Segmentation of Work Zone scenes From Naturalistic Driving Data", Transportation Research Record

38 Papakis, I., Sarkar, A., Svetovidov, A., J. Hichman, L. Abbott, “A CNN-Based In-Vehicle Occupant Detection and Classification Method Using SHRP 2 Cabin Images”, Transportation Research Record: 0361198121998698.

39 Chen, Kai, et al. "MMDetection: Open MMLab detection toolbox and benchmark." arXiv preprint arXiv:1906.07155 (2019)

40 Papakis, Ioannis, Abhijit Sarkar, and Anuj Karpatne. "GCNNMatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization." arXiv preprint

arXiv:2010.00067 (2020)

41 Papakis, I., A. Sarkar, and A. Karpatne. A Graph Convolutional Neural Network Based Approach for Traffic Monitoring Using Augmented Detections with Optical Flow. in 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). 2021.

42 Zhou, Xingyi, et al. "Global tracking transformers." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022

43 Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023

44 Aharon, Nir, Roy Orfaig, and Ben-Zion Bobrovsky. "BoT-SORT: Robust associations multi-pedestrian tracking." arXiv preprint arXiv:2206.14651 (2022)

45 Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 1672–1679. IEEE, 2018

46 Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. IEEE Pervasive computing, 7(4):12–18, 2008

47 JOSM developers. JOSM (Java OpenStreetMap Editor). URL https://josm.openstreetmap.de/

48 Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." arXiv preprint arXiv:1903.02428 (2019)

49 Shi, Yunsheng, et al. "Masked label prediction: Unified message passing model for semi-supervised classification." arXiv preprint arXiv:2009.03509 (2020)

50 Hu, Weihua, et al. "Strategies for pre-training graph neural networks." arXiv preprint arXiv:1905.12265 (2019)

51 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graphneural networks? arXiv preprint:1810.00826, 2018.

52 Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2017). On a formal model of safe and scalable self-driving cars. arXiv preprint arXiv:1708.06374.

53 Sarkar, A., Krum, A., Hanowski, R., & Hickman, J. (2021, June). Responsibility Sensitive Safety Analysis of Truck Following in US Highway. In International Conference on Applied Human Factors and Ergonomics (pp. 119-126). Cham: Springer International Publishing.

54. Sonth, A. P. (2023). Enhancing Road Safety through Machine Learning for Prediction of

Unsafe Driving Behaviors (Master's Thesis, Virginia Tech).

55 Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In 2020 IEEE Intelligent Vehicles Symposium (IV), pages 1929–1934, 2020. doi: 10.1109/IV47402.2020.9304839

56 Tobias Fleck, Karam Daaboul, Michael Weber, Philip Schörner, Marek Wehmer, Jens Doll, Stefan Orf, Nico Sußmann, Christian Hubschneider, Marc René Zofka, Florian Kuhnt, Ralf Kohlhaas, Ingmar Baumgart, Raoul Zöllner, and J. Marius Zöllner. Towards large scale urban traffic reference data: Smart infrastructure in the test area autonomous driving baden-württemberg. In Intelligent Autonomous Systems 15 - Proceedings of the 15th International Conference IAS-15, Baden-Baden, Germany, June 11-15, 2018, pages 964–982, 2018

57 Maximilian Zipfl, Tobias Fleck, Marc René Zofka, and J. Marius Zöllner. From traffic sensor data to semantic traffic descriptions: The test area autonomous driving baden-württemberg dataset (taf-bw dataset). In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pages 1–7, 2020. doi: 10.1109/ITSC45102.2020.9294539

58 Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. arXiv:1910.03088 [cs, eess], 2019

59 Yanchao Xu, Wenbo Shao, Jun Li, Kai Yang, Weida Wang, Hua Huang, Chen Lv, and Hong Wang. Sind: A drone dataset at signalized intersection in China. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pages 2471–2478. IEEE, 2022

60 Ronald W Schafer. What is a savitzky-golay filter?[lecture notes]. IEEE Signal processing magazine, 28(4):111–117, 2011

61 Franz Pucher. Trajectory planning in the frenet space. URL https://fjp.at/posts/optimal-frenet/

62 Aoude, G.S., et al., Driver Behavior Classification at Intersections and Validation on Large Naturalistic Data Set. IEEE Transactions on Intelligent Transportation Systems, 2012. 13(2): p. 724-736.

63 Ali Razmpa. An Assessment of Post-Encroachment Times for Bicycle-Vehicle Interactions Observed in the Field, a Driving Simulator, and in Traffic Simulation Models. PhD thesis, Civil and Environmental Engineering, Portland State University, 2016.

64 Doerzaph, Z. R. (2007). Development of a threat assessment algorithm for intersection collision avoidance systems (Doctoral dissertation, Virginia Tech).

Appendix A: Object Detection and Tracking

Appendix A summarizes different object detector and tracker combinations and their testing results on the VT-CAST dataset and processing strategies using Oracle cloud service.

Object Detection and Tracking Algorithms

Table 5. Qualitative Comparison and Computational Complexity for Various Object Detection and Tracking Algorithms

Method	Performance	Limitations	Inference Speed
MMDetection + GCNNMatch	Combination performs excellently with the VDOT dataset, successfully tracking vehicles in various traffic scenarios.	Challenges arise when tracking very small or quickly moving vehicles.	12x
GTR (Global Tracking Transformer)	Decently performs on the VDOT dataset and can track objects in long sequences.	Struggles to track smaller objects consistently due to the lower quality of videos.	4x
YOLOv7 + BoT-SORT	YOLOv7 effectively detects small objects in lower resolution, making the pair efficient and accurate on the VDOT dataset.	Some smaller objects are not tracked properly.	1x

We used Oracle Cloud Infrastructure (OCI) to process the video data for object detection and tracking.

MMDetection + GCNNMatch

MMDetection [39] is a highly regarded open-source deep learning toolbox designed specifically for object detection tasks. Developed as part of the OpenMMLab project by the Multimedia Laboratory at CUHK, MMDetection offers a range of efficient and accurate detection methods that achieve state-of-the-art performance. This library has gained significant popularity within the artificial intelligence community due to its exceptional flexibility, extensibility, and comprehensive support for multiple detection frameworks. With its capabilities, MMDetection proves to be an ideal choice for VDOT's dataset, and it has been trained on the extensive Second Strategic Highway Research Program Naturalistic Driving Study dataset for object detection.

GCNNMatch (Graph Convolutional Neural Network Match) [40, 41] is an advanced tracking algorithm that leverages the power of GCNNs. The primary objective of GCNNMatch is to provide a robust mechanism for tracking objects across frames, effectively tackling challenges such as occlusions, changes in appearance, and unpredictable motion. By utilizing a graph convolution-based approach, GCNNMatch excels at handling variations in object appearance over time, as well as overcoming occlusions and other complexities encountered in video frames. Notably, GCNNMatch has demonstrated exceptional tracking performance on a variety of traffic scenarios captured in the VDOT dataset.

However, the study examining GCNNMatch's performance also identified certain limitations of the algorithm. For instance, it struggled to track vehicles that were extremely small or moving at high speeds. Despite its overall effectiveness, these specific scenarios posed challenges for GCNNMatch's tracking capabilities.

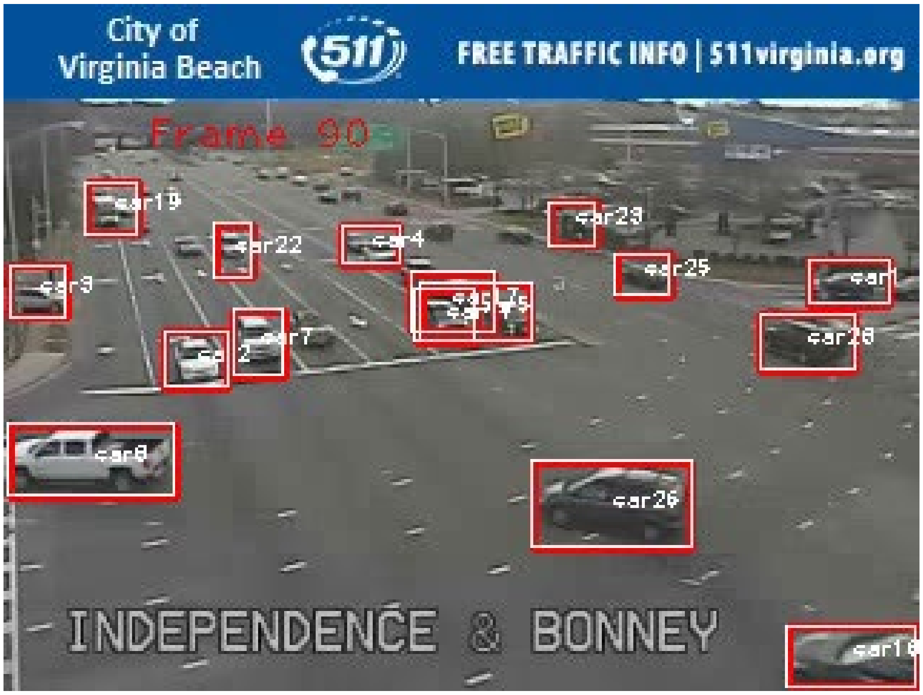


Figure 11. Video image. Sample result on a VDOT video demonstrating object detection with MMDetection [51], followed by tracking using GCNNMatch [52].

GTR

GTR [42], a cutting-edge transformer-based architecture, revolutionizes global multi-object tracking by effectively generating global trajectories for all objects in a short sequence of frames. The process begins with object detection in each frame, followed by encoding the detected objects into feature vectors. These feature vectors are then inputted into a global tracking transformer, which learns to associate the vectors across frames, ultimately producing accurate global trajectories for each object.

One of the notable advantages of GTR is its ability to be jointly trained with an object detector. This unique characteristic enables GTR to simultaneously learn object detection and tracking, distinguishing it from traditional methods that typically rely on separate models. As a result, GTR achieves enhanced efficiency and outperforms previous tracking approaches, evident by its state-of-the-art performance on the MOT17 benchmark. Furthermore, GTR exhibits proficiency in tracking objects in extended sequences, providing reliable results in long-duration scenarios.

While GTR demonstrates impressive performance on the VDOT dataset, it faces challenges when consistently tracking smaller objects due to the lower quality of videos. It is also a bit harder for the algorithm to maintain a high frames per second rate during longer video inference, taking a lot of time to process extended footage.

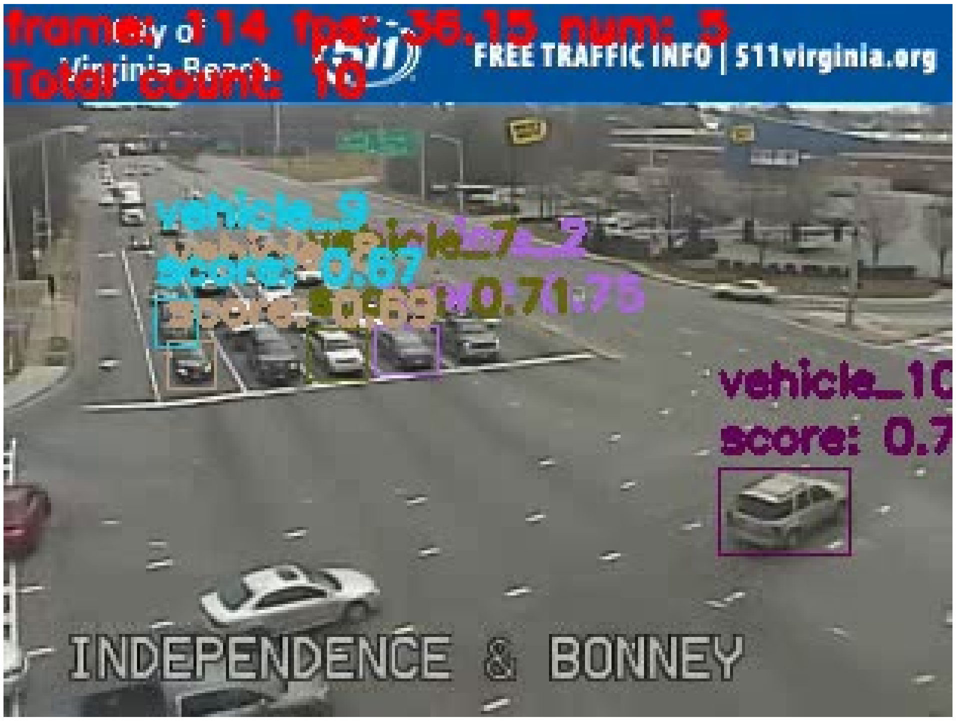


Figure 12. Video image. Sample result on a VDOT video demonstrating object detection and tracking using GTR [53].

YOLOv7 + BoT-SORT

BoT-SORT (Simple Online and Realtime Tracking) [44] is an advanced multi-object tracking algorithm that integrates object detection, association, and re-identification techniques. This state-of-the-art algorithm has demonstrated exceptional accuracy across various datasets.

The functioning of BoT-SORT begins with the detection of objects within each frame of a video using a pre-trained object detector, such as YOLOv7 [43]. Subsequently, the detected objects are associated with one another across frames utilizing a Bayesian approach. This approach takes into consideration the appearance, motion, and social relationships of the objects.

To perform object detection in each frame of the VDOT dataset, we employed the YOLOv7 pre-trained model. This model is highly effective in accurately detecting small objects even in lower resolution. The detected objects are then linked across frames using the BoT-SORT algorithm, which is widely regarded as the most efficient and accurate tracking method available.

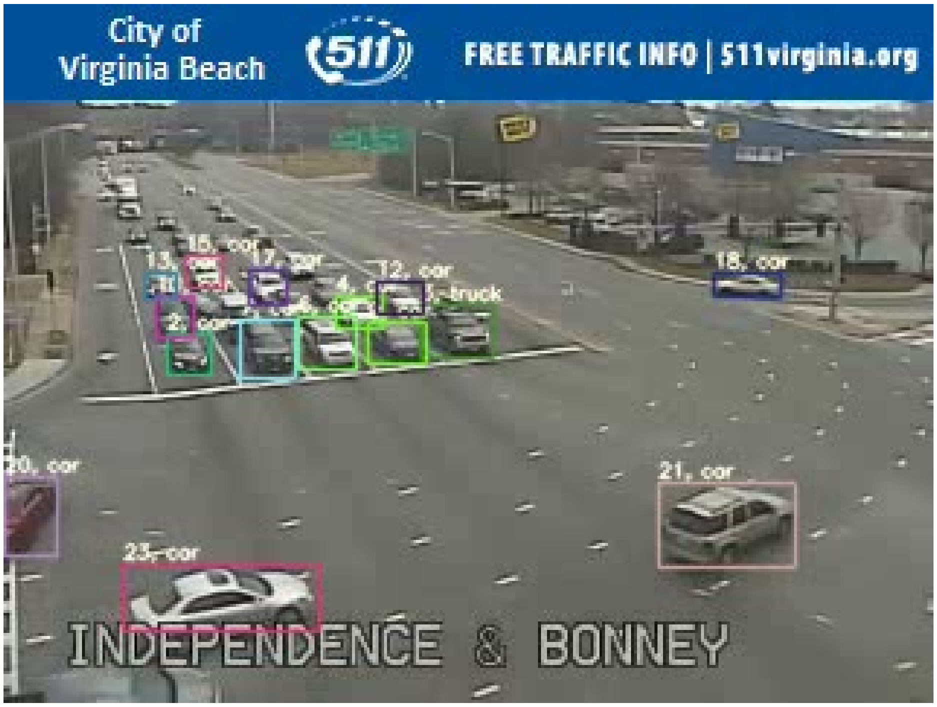


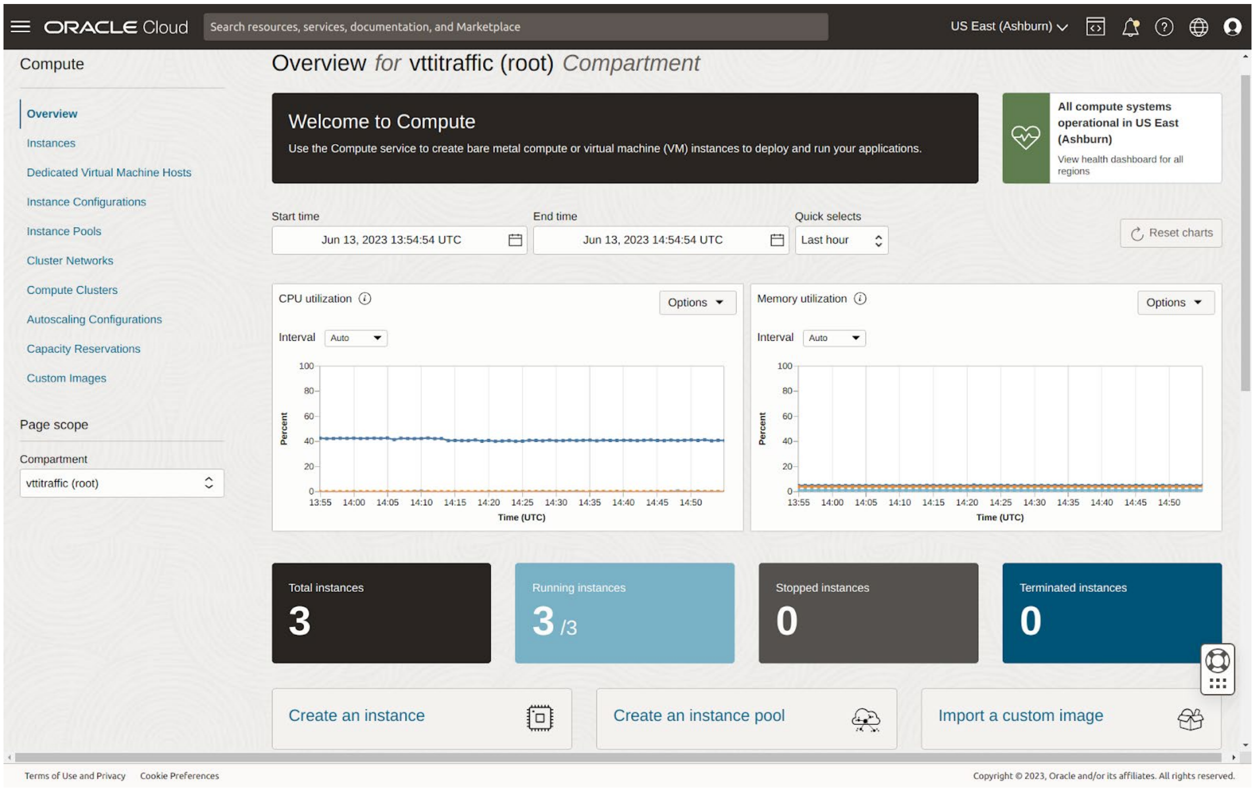
Figure 13. Video image. Sample result on a VDOT video demonstrating object detection with YOLOv7 [54], followed by tracking using BoT-SORT [55].

OCI

OCI proved instrumental in processing multiple videos by leveraging tracking algorithms, effectively reducing the computational load and minimizing the utilization of VTTI GPU servers. OCI stands as a robust cloud computing platform, offering an extensive range of global Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) solutions. It boasts high reliability, security, and cost-effectiveness, empowering users to develop and operate various applications and services. OCI adopts a pay-as-you-go pricing model and provides support for diverse programming languages and frameworks.

However, a few limitations were encountered while using OCI:

- Limited availability of instances equipped with Nvidia-based GPU capabilities.
- Instances with identical shapes are not permitted.
- At times, resources may be allocated to other users, resulting in the inability to create an instance with the required image and shape.



Overview *for* vttitraffic (root) *Compartment*

Welcome to Compute

Use the Compute service to create bare metal compute or virtual machine (VM) instances to deploy and run your applications.

All compute systems operational in US East (Ashburn)

View health dashboard for all regions

Start time: Jun 13, 2023 13:54:54 UTC

End time: Jun 13, 2023 14:54:54 UTC

Quick selects: Last hour

Interval: Auto

Compartment: vttitraffic (root)

Total instances 3

Running instances 3 /3

Stopped instances 0

Terminated instances 0

Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

Figure 14. Screenshot from the OCI webpage showcasing instance creation for computational purposes, along with convenient tracking of usage and expenses.

Appendix B: Drone-based Traffic Dataset

Video Monitoring of Traffic Scenes

Video data at a traffic scene captures the dynamics of the continuous traffic. Historically, video data at intersections has been captured in several ways. This includes roadside cameras, traffic surveillance cameras, dashboard cameras, and drone cameras. The data from different sources differs in many ways, including image resolution, frame rate, and field of view. Many states, including Virginia, have deployed several traffic cameras at intersections (Figure 5). These cameras are used for monitoring traffic, incident detection and monitoring, traffic enforcement, and law enforcement. These cameras generally have a fixed field of view but the capability to pan, tilt, and zoom. Recently, more advanced cameras have been deployed that can capture a 360-degree view of the intersection. Modern cameras and commercial solutions provide high resolution, a wide field of view, and high-speed cameras. In more recent times, drone cameras are being widely used. These cameras can capture a bird's-eye view of the full intersection. This helps to monitor the egress and ingress of traffic from all directions of the intersection (Figure 1).

Drone-based intersection datasets play a crucial role in advancing the research and development of autonomous vehicles and intelligent transportation systems. These datasets provide valuable real-world scenarios that help train and evaluate algorithms for perception, prediction, and decision-making at intersections. These datasets are also commonly referred to as trajectory datasets due to their focus on providing information about the movement paths different traffic participants follow. The distinct advantage of these datasets lies in their innate ability to portray each traffic participant as a singular point defined by its GPS coordinates. This "bird's-eye" perspective circumvents the potential errors introduced during the transformation from camera-based to GPS-based coordinates. Consequently, the resulting network seamlessly translates into a graph representation, with participants forming nodes and relationships being depicted through edges. This elegant graph structure can be readily embraced by graph neural network (GNN) methodologies, which natively integrate with such data. The absence of transformation-induced errors confers a marked advantage to these datasets. This pristine foundation leads to a significantly enhanced graph-based output, thereby culminating in the formulation of more refined and accurate GNN models.

inD [55] is a dataset of naturalistic vehicle trajectories recorded at intersections in Germany. The dataset includes data from four locations, and the positional error is typically less than 10 centimeters. The dataset is applicable to many tasks such as road user prediction, driver modeling, scenario-based safety validation of automated driving systems, or data-driven development of highly automated driving system components.

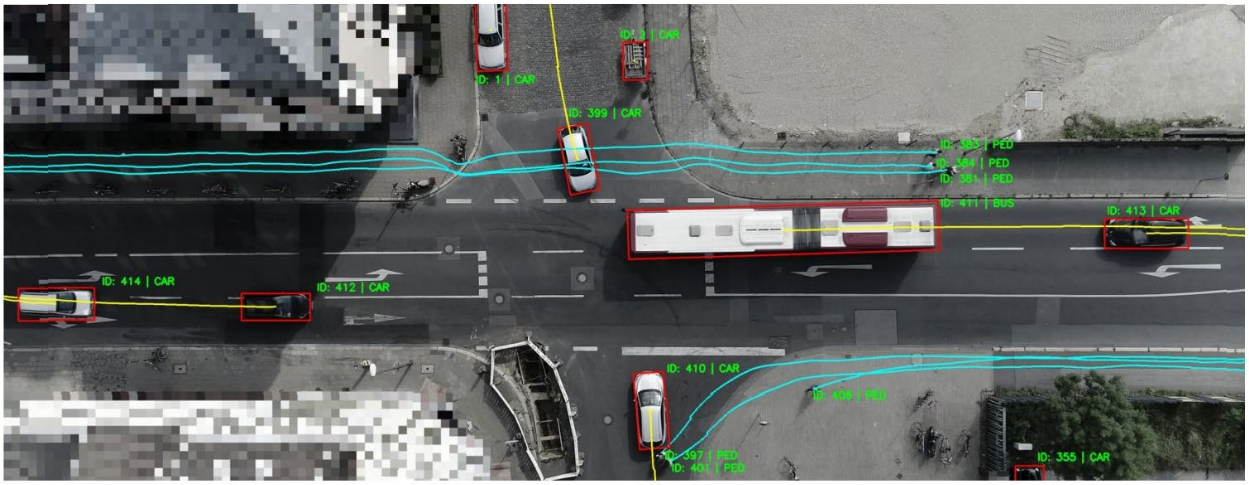


Figure 15. Annotated photo. Exemplar scenario showcasing the trajectories of traffic participants in the inD dataset [46].

TAF-BW [56, 57] is a dataset of vehicle trajectories recorded at intersections in Baden-Württemberg, Germany. The dataset includes data from 100 different intersections, and the positional error is typically less than 10 centimeters. The dataset is applicable to many tasks such as road user prediction, driver modeling, and traffic signal control.

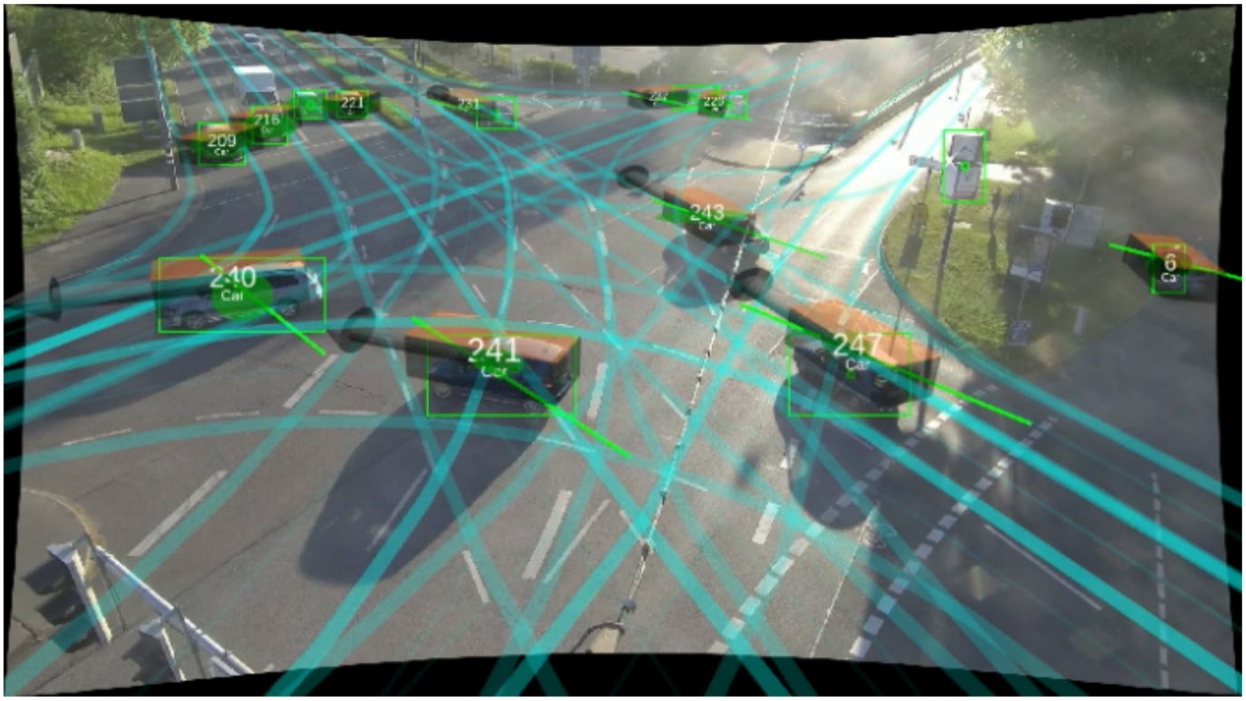


Figure 16. Annotated photo. Exemplar scenario showcasing the trajectories of traffic participants in the TAF-BW dataset [47, 48].

INTERACTION [58] is a dataset of vehicle trajectories recorded at intersections in the United States, China, Germany, and Bulgaria. The dataset includes data from a variety of intersection types, and the positional error is typically less than 10 centimeters. The dataset is applicable to many tasks such as road user prediction, driver modeling, and traffic signal control.

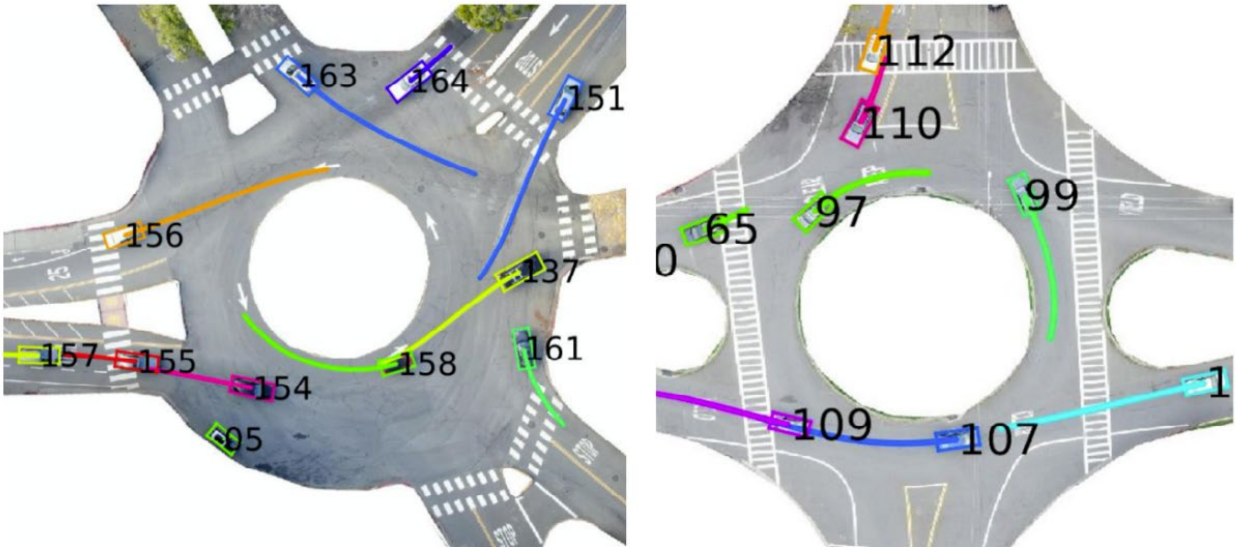


Figure 17. Annotated photo. Exemplar scenario showcasing the trajectories of traffic participants in the INTERACTION dataset [49].

The **SIND** [59] dataset was collected at a signalized intersection located in Tianjin, China, offering a comprehensive and detailed perspective on traffic dynamics. Captured via high-quality drones, the dataset comprises 4,000-pixel resolution video footage, enabling the extraction of valuable information such as the trajectories of various traffic participants, the status of traffic lights, and high-definition maps of the area.

Appendix C: Virginia Intersection Data

Selection of Intersections

- For each of the 48,450 intersections in the database, the total number of police reports was calculated for crashes that occurred within a radius from the intersection. This calculation was based on a unique "node" variable. The resulting value, known as "total crashes," indicated the overall number of crashes that took place at each intersection. This information was then utilized to rank the intersections based on their level of risk, ranging from the highest occurrence of crashes to the lowest, thus determining the safest intersections.
- A second ranked list was generated by considering crashes that caused either at least one fatality (identified by the variable "K_PEOPLE") or at least one severe injury (identified by the variable "A_PEOPLE"). This approach aimed to exclude minor crashes that only caused property damage or mild injuries. By focusing on more serious incidents, this ranking provided an independent estimate separate from fender benders and minor incidents.

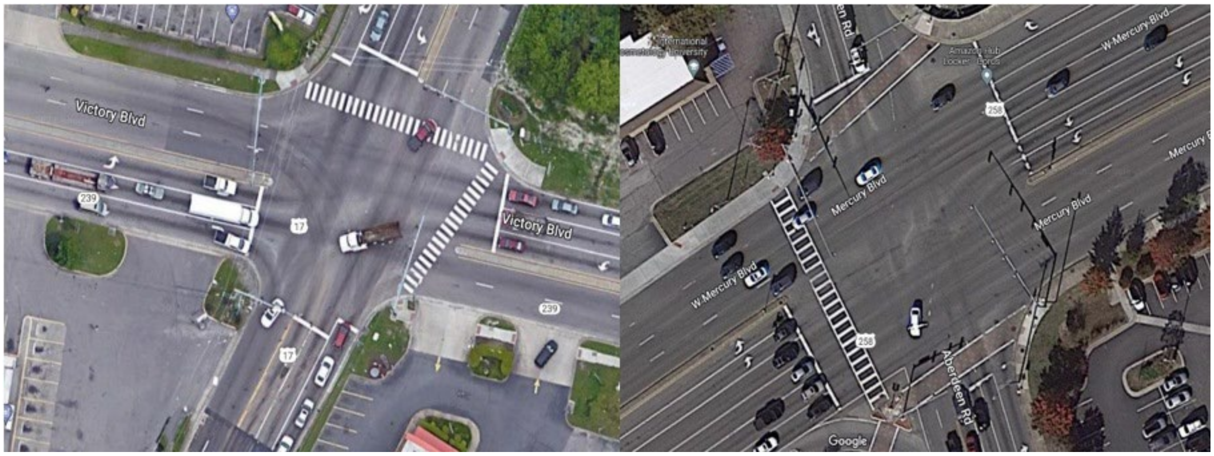


Figure 18. Photos. Satellite views of I1 (left) and I2 (right) obtained from Google Maps.

General Trend

- Higher frequency of severe crashes (56.4%) was observed between 12 and 7 p.m. across intersections (Figure 19).
- Severe crashes occurred more frequently during May-October.
- Fridays and Saturdays observed higher frequency of crashes during the week (Figure 20).
- Pedestrians accounted for 15.1% of all traffic fatalities, 47.9% of which came from crashes at intersections.
- Nighttime crashes (between dusk and dawn) were associated with a higher risk of pedestrian fatalities (73.7%), speeding-related crashes, and bicyclist fatalities.

- Pedestrian-related crashes were more often single-vehicle crashes, whereas non-pedestrian-related crashes involved multiple vehicles.
- Higher annual average daily traffic (AADT) was not found to be associated with a higher number of crashes. In fact, only 7 out of the top 20 riskiest intersections were ranked in the list of top 100 intersections based on AADT values.

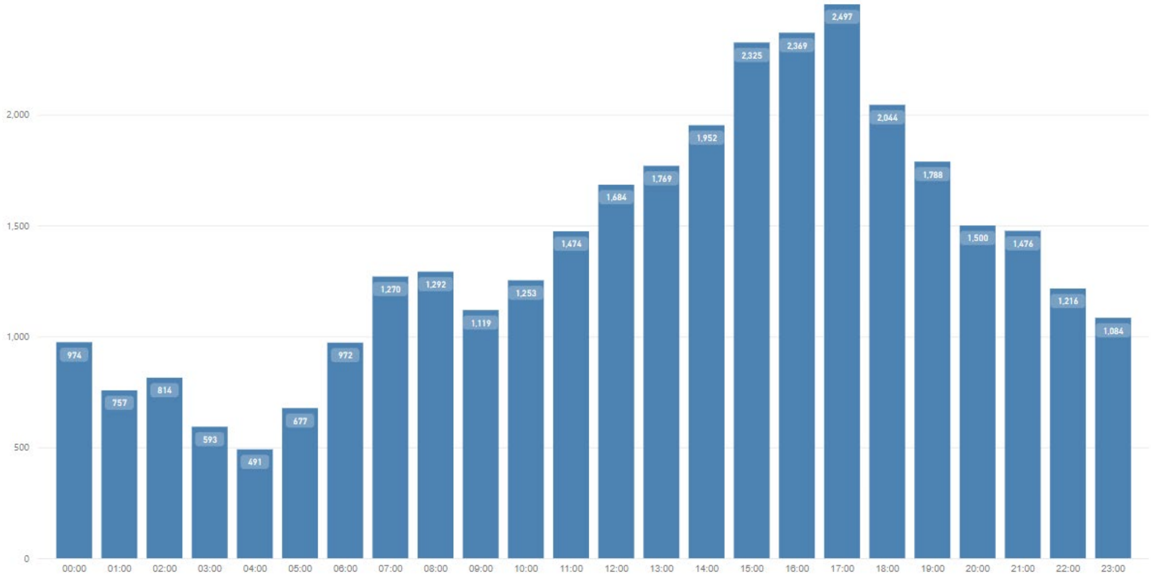


Figure 19. Graph. Crash distribution by time of day (*x*-axis denotes military time in hours).

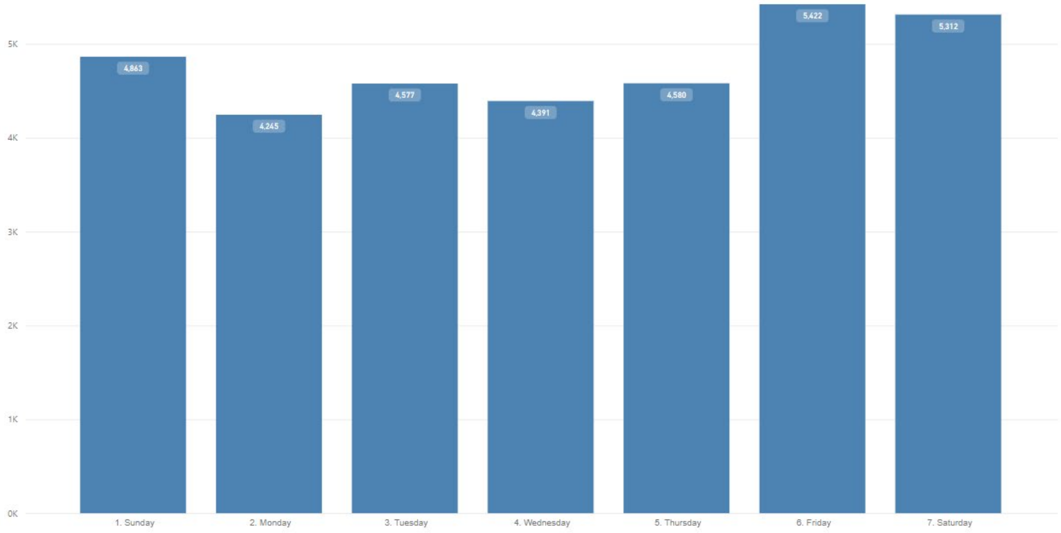


Figure 20. Graph. Crash distribution by day of the week.

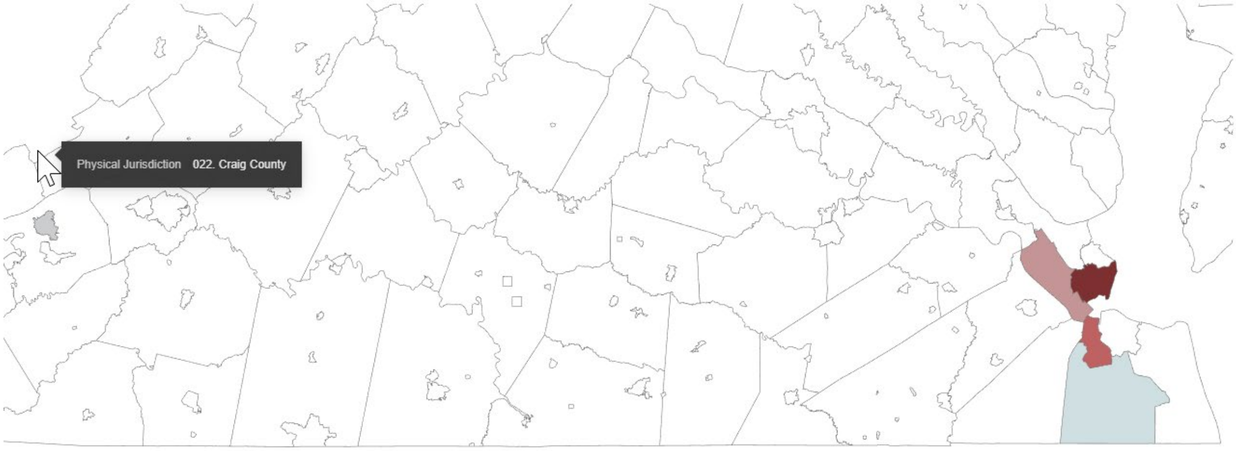


Figure 21. Map. Locations of selected intersections (Blacksburg, Virginia Beach, Newport News, Hampton).

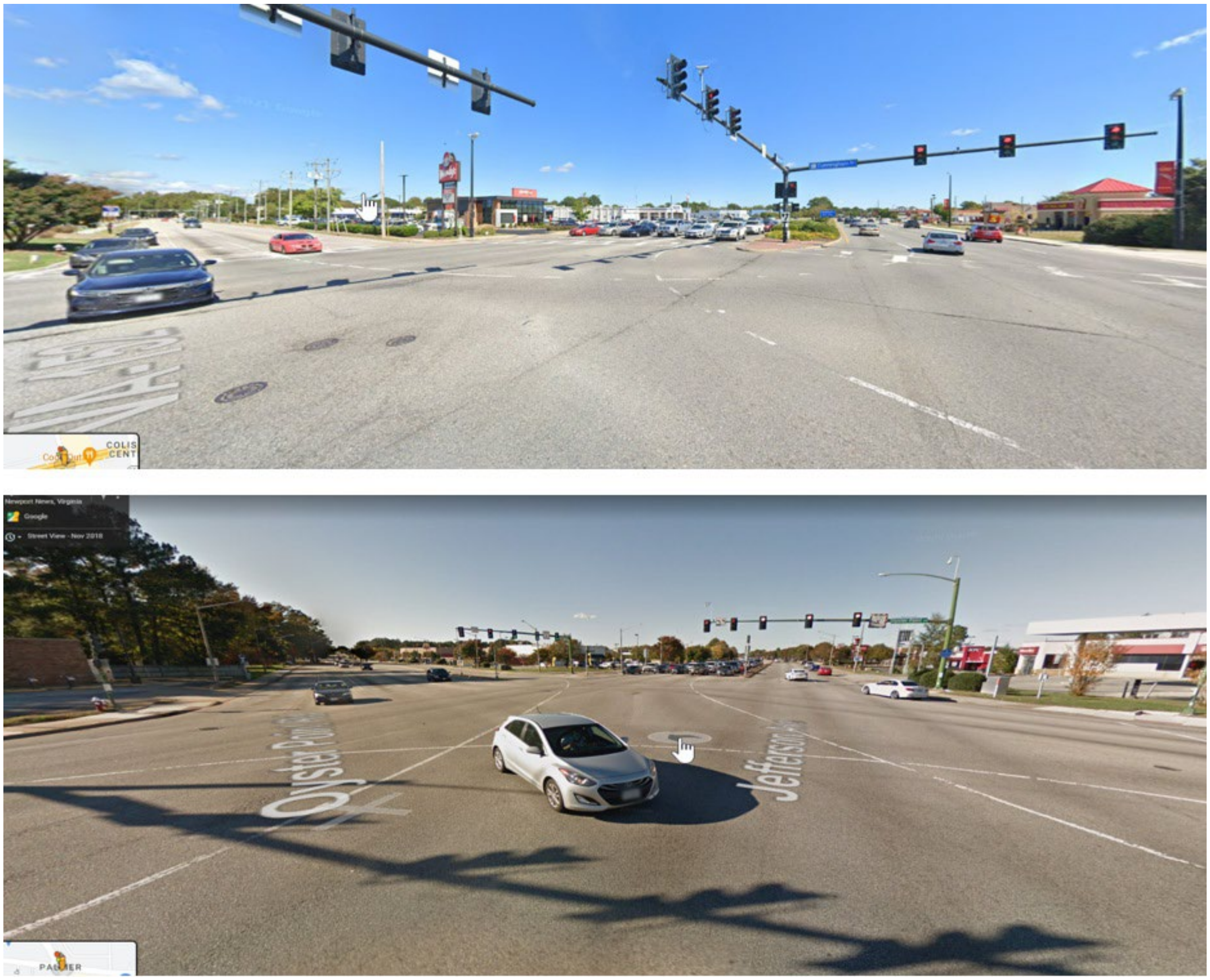


Figure 22. Photos. Street views of I4 (top) and I5 (bottom) from Google Maps (Hampton and Newport News, VA).

Identification and Collection of Traffic Videos from VDOT Traffic Cameras

In this work, we present our analysis of the VT-CAST (Traffic Cameras for Advanced Safety Technologies) 2020 dataset [34]. Traffic cameras spread throughout Virginia stream the live video feeds on the VDOT server. These streams are recorded in segments of 1-hour videos to form the dataset. The cameras are strategically placed at intersections, highways, and other roadways at several locations in Virginia. These cameras are positioned to capture a wide field of view and offer an oblique perspective that allows visibility of the surrounding road area. While the cameras do not provide a top-down view, they ensure comprehensive monitoring of traffic conditions. The cameras solely offer raw video feeds and do not provide specific information regarding the kinematics or movement patterns of the traffic participants in the observed scenario.

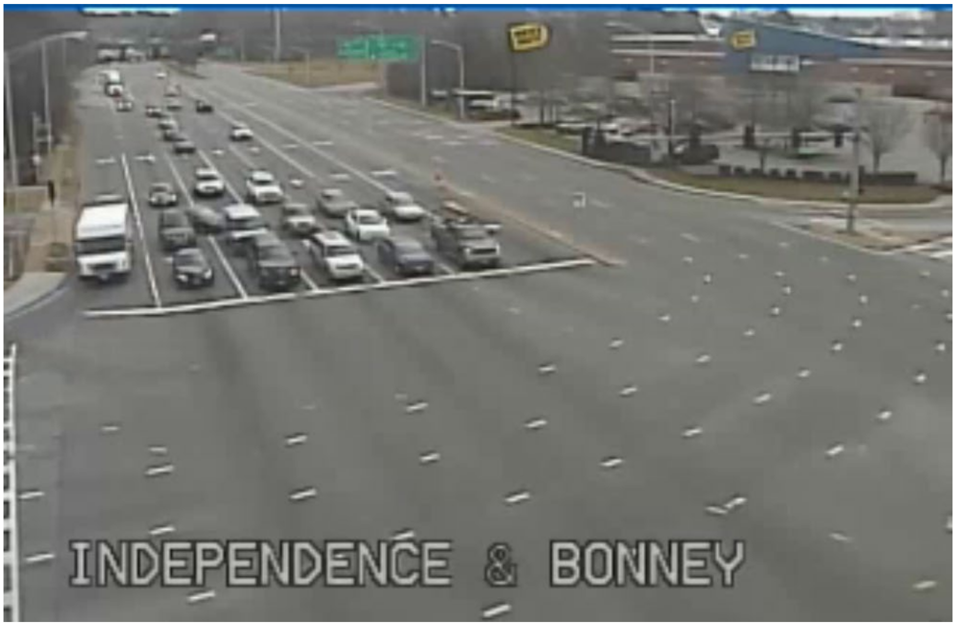


Figure 23. Video image. Sample frames from the VT-CAST 2020 dataset to demonstrate view range and video quality.

- A spreadsheet containing detailed information about VDOT traffic cameras was created. The spreadsheet includes the GPS coordinates and basic details of all 1,263 functional cameras.
- Annotations were employed to identify and eliminate cameras exhibiting inadequate data quality, including low resolution, rotating mechanisms, irrelevant field of view (e.g., parking lots), signal loss, or accidental changes in orientation caused by wind or other factors. As a result, a subset of 500 cameras with functional and pertinent data was obtained.
- Using GPS coordinates, the Euclidean distance was calculated for each possible pair of cameras and intersections. This resulted in a 500 x 48,450 matrix representing the distances

between cameras and intersections. From this matrix, the cameras that were closest to each of the 20 intersections of interest were determined.

- The traffic footage from each camera was carefully examined to verify data integrity and usability. This involved checking whether the camera captured a sufficient portion of the intersection’s traffic flow. Cameras that exhibited intermittent rotation or poor video quality were excluded from the analysis process.

Roughly 30 to 60 hours of video were recorded for each intersection based on recording availability and viable streaming quality.

Discussion

The five identified intersections (Dam Neck Rd - Virginia Beach, US-60 - Virginia Beach, Lynnhaven Pkwy - Virginia Beach, Independence Blvd - Virginia Beach, US-58 - Virginia Beach) rank among the worst crash locations in Virginia every year, as verified by multiple state, county, and independent reports. General trends found in crash distributions (e.g., days of week, pedestrian involvement) are further confirmed by statistics generated by VDOT, the Fatality Analysis Reporting System, and independent county departments of motor vehicles. Additionally, the presented analysis highlights subtle relationships that go beyond the state-generated reports. For example, roughly 53% of all crashes were found to have occurred during daylight hours. While it may imply that fewer crashes occur during the night, it should be noted that the average traffic volume flow during the night is significantly lower than that during the day—therefore, a lot more crashes occur during nighttime per unit flow of traffic, or per number of cars driven through an intersection. Furthermore, nighttime crashes result in a significantly higher percentages of fatalities compared to the former.

Appendix D: Technical Background

Pixel-to-GPS Transformation

Pixel-to-GPS conversion involves transforming an image captured by a camera into geographic coordinates using GPS data. This technique is widely used in the field of remote sensing and allows researchers to obtain precise geolocation information from satellite or aerial imagery.

In this work, we used the video feed from cameras situated at various intersections throughout Virginia by VDOT. We chose several intersections where there have been a high number of crashes according to statistics. We chose a frame with as few vehicles as possible so that the road structure and markings were clearly visible. Because we knew where these cameras are located, we also obtained the Google Earth (nadir) view of that location. Based on these two views (camera and Google Earth), the projection of pixel-to-GPS was achieved.

To accurately represent the intersectional areas, a rectangular Cartesian coordinate plane was generated by assuming parallel latitudes and longitudes within each 1,500-ft direction. The following approximations were derived:

1. A 1-degree change in latitude corresponds to approximately 69 miles.
2. For a 1-meter distance, the latitude experiences a change of approximately 0.0000089-degrees. A 1-degree change in longitude corresponds to around 55 miles.
3. In GPS coordinates, the fifth decimal place represents a distance of 1 meter or approximately 3.3 ft.

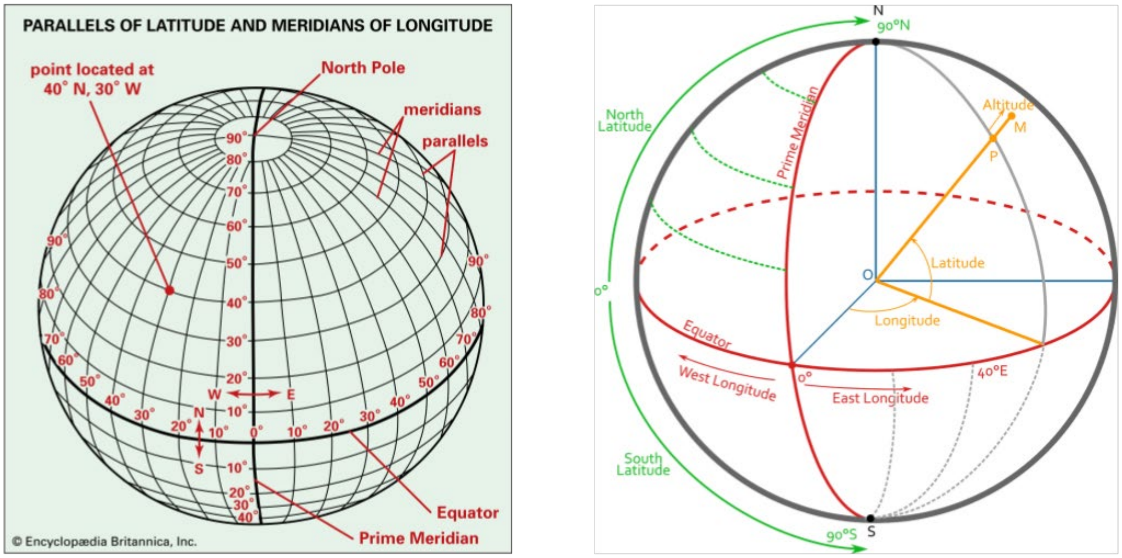


Figure 24. Diagrams. Parallel of latitude and meridians of longitude. Images credit: https://www.britannica.com/science/meridian-geography, https://docs.unrealengine.com/5.0/en-US/georeferencing-a-level-in-unreal-engine/

The process of pixel-to-GPS transformation involves annotating the image captured by the camera with key points that can be identified distinctively in the Google Earth view. Around 20-30 points are chosen that are spread around the entire image so that the projection is not biased towards a particular region of the image. Based on the pixel coordinates from the camera view and the GPS coordinates from the Google Earth view of these key points, a homography matrix is obtained. This homography matrix can now project any point visible in the camera to the corresponding GPS coordinate. One additional requirement is that the chosen point should be as close to the ground surface as possible. This is required since the projection matrix does not take height into account.

The accuracy of the transformation depends on several factors, including the quality of the GPS data, the stability of the camera's orientation, and the quality of the data captured by the camera. However, when these factors are carefully controlled, pixel-to-GPS perspective projection can produce highly accurate geolocation information, making it an essential tool for many applications in remote sensing, including mapping, monitoring, and land-use analysis.

To compute the homography matrix, we utilized the Random Sample Consensus (RANSAC) algorithm. This algorithm is widely used in computer vision and image processing for estimating a transformation matrix between sets of corresponding points in different images. It is particularly effective when dealing with data that contains outliers or errors.

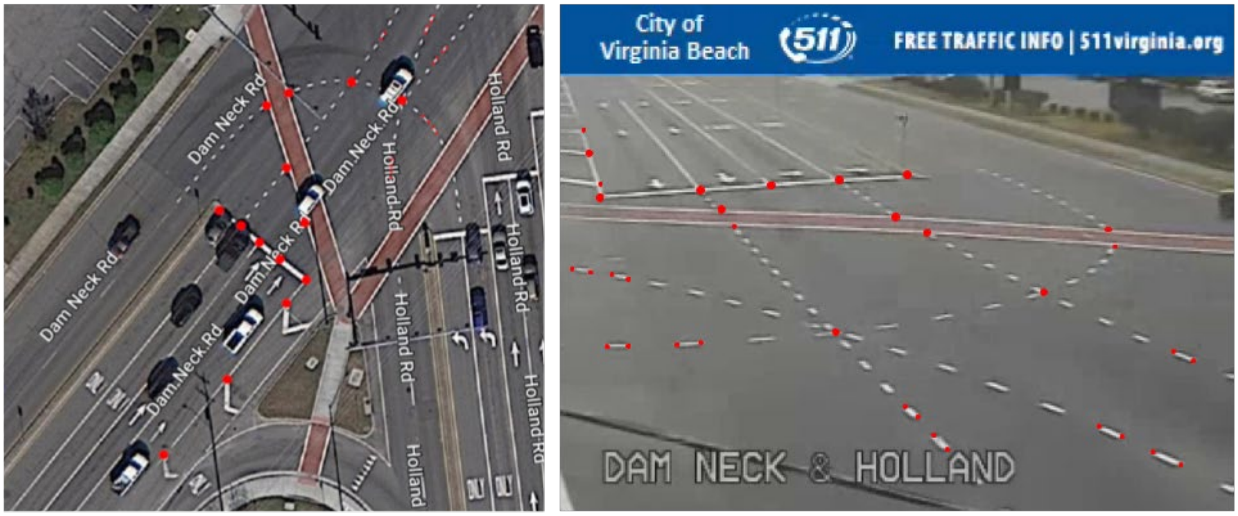


Figure 25. Video images. Matching points from Google Maps top-down view (left) and camera view (right) for homography computation.

The RANSAC algorithm works iteratively by randomly selecting subsets of the matching points and estimating a homography matrix based on those subsets. The obtained matrix is then used to evaluate the number of inliers, which are the points that align well with the estimated transformation. This iterative process helps filter out outliers and provides a robust estimation of the homography matrix.

Due to the lower quality of videos, the bounding box surrounding each participant tends to vary and exhibit jitter. This also affects the trajectory of the tracked point post transformation. To further enhance the smoothness of the participant's trajectory, we applied a Savitzky–Golay filter [60] on the tracked GPS point for each participant. This filtering technique helps reduce noise and irregularities in the participant's movement pattern, resulting in a more refined and consistent trajectory.

The GPS reference point is used to convert the GPS coordinates of all the participants in terms of relative distance (in meters) along the *x*-axis and *y*-axis directions. These axes are taken to be parallel to the image width and length.

Projection to Frenet Space

Now that we established the road layout, our next step was to represent the participants on the road. As shown in Figure 26, we additionally labeled the vehicles (alphabetically) to identify them in the creation of an SSG. In Frenet space, the centerline of each road segment corresponds to one of the axes. By transforming the variables (*x, y*) from Cartesian coordinates to (*s, d*) in Frenet space, we can accurately describe the position of a vehicle on the road.

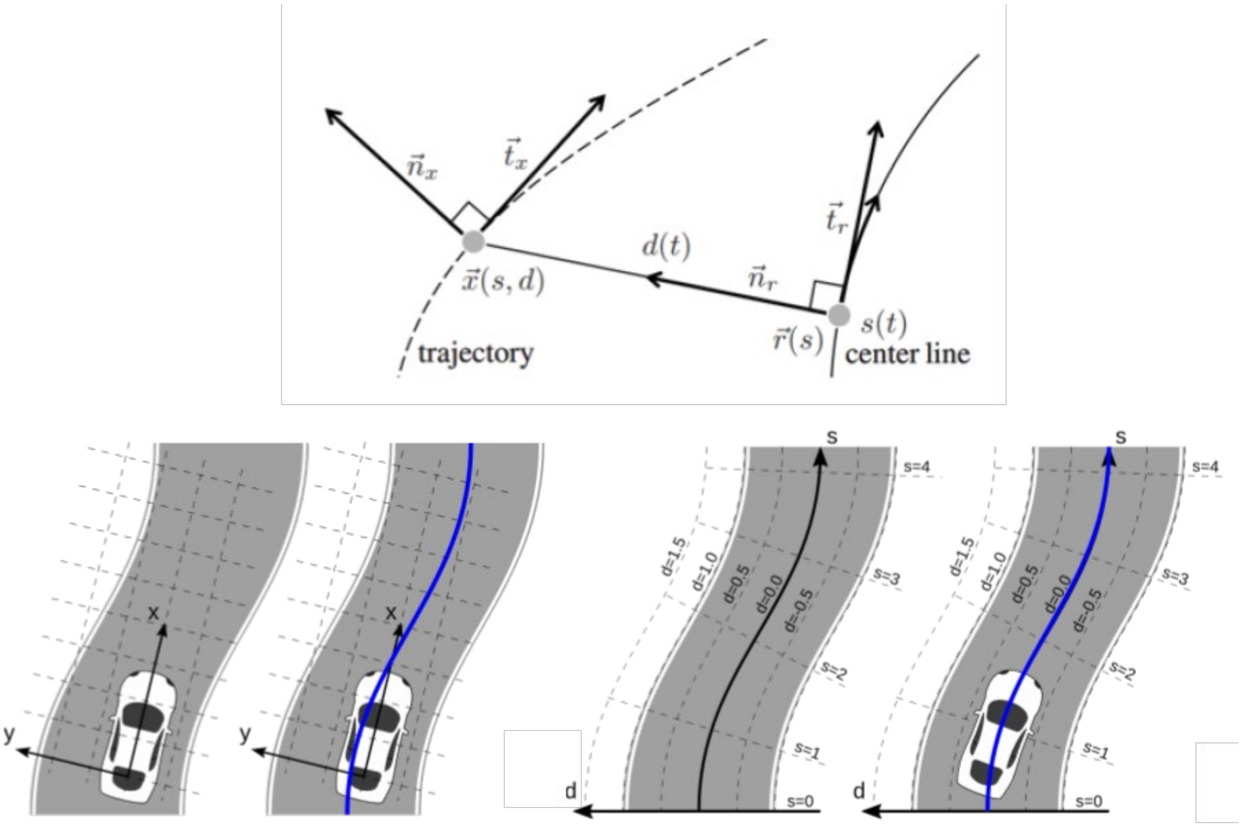


Figure 26. Diagrams. Transformation from Cartesian to Frenet coordinates. The figure on the bottom left shows an example vehicle with the origin of Cartesian coordinate frames aligned to it. The figure on the bottom right shows the same vehicle in Frenet space with the centerline of the road as the *s*-axis. The figure on top shows the transformation from Cartesian to Frenet space. Images credit: [33, 61].

In Frenet coordinates, *s* represents the distance along the road, also known as longitudinal displacement; *d* represents the side-to-side position on the road relative to the reference path, also known as lateral displacement.

To determine the Frenet coordinates for a given point in the vehicle frame, we need to find the closest point on the reference path. The run length from the reference path points provides us with the *s* coordinate of the transformed point. If the reference path is sufficiently smooth, meaning it is continuously differentiable, the vector at that point will be orthogonal to the reference path. The signed length of the perpendicular vector from the reference path determines the *d* coordinate.

Graph Neural Network

The graph neural network is specifically designed to capture relational information in a graph. This kind of neural network is different from traditional neural networks in multiple aspects:

1. Unlike images, graph data cannot be resized, reshaped, or padded to fit a predetermined input size. Consequently, conventional neural networks, which typically expect fixed-size inputs, face difficulties when handling graph data.
2. Isomorphism: Graphs that may appear dissimilar can actually possess identical structural properties. Consequently, algorithms designed to process graph data must exhibit invariance to permutations, enabling them to handle graphs that have different node ordering but equivalent structures.
3. Graphs exhibit a non-Euclidean structure, which contrasts with the grid-like structure of images or regular grid data. Consequently, machine learning techniques applied to graph data are often referred to as geometric deep learning, acknowledging the geometric nature of the data and the need for specialized approaches.

TransformerConv

TransformerConv is a novel attention-based model that was proposed in Shi et al. [49]. It is designed to be more efficient and effective than traditional attention mechanisms for graph-structured data. The graph transformer operator works by first computing a local attention score for each pair of nodes in the graph. This score is based on the features of the two nodes and the edges between them. The local attention scores are then used to compute a global attention score for each node. This score is used to weight the features of the neighboring nodes when updating the features of the current node. The graph transformer operator has several advantages over traditional attention mechanisms. First, it is more efficient because it only computes local attention scores, rather than global attention scores. Second, it is more effective because it can capture long-range dependencies between nodes in the graph. Here is a more detailed explanation of the graph transformer operator: The process begins with the computation of local attention scores. Each node in the graph is assigned a score based on its own features and those of its neighboring nodes. These scores are subsequently normalized, and the attention score for each node pair is obtained by multiplying the scores of the two involved nodes. Next, the global attention score is calculated for each node by summing up the weighted local attention scores of its neighbors. The weights are determined using a softmax function to ensure their sum is equal to 1. Finally, the features of a node are updated using the global attention score. This entails computing a weighted sum of the features of the node's neighbors, where the weights correspond to the global attention scores.

GINEConv

GINEConv (Graph Isomorphism Network with Edges) [50] is a graph convolutional neural network (GCNN) that uses a novel message passing mechanism to aggregate node features. The message passing mechanism is based on the Graph Isomorphism Network (GIN) [51], but it also incorporates edge features. This allows GINEConv to learn more complex representations of graphs than GNNs that do not use edge features.

The GINEConv architecture consists of several key components. First, the layer begins by computing a per-edge representation, which is derived from both the node features and the edge features. This process enables the layer to capture the intricate relationships between nodes and edges in the graph. Subsequently, the layer aggregates these per-edge representations, combining them into a comprehensive per-node representation. By doing so, the layer effectively summarizes the information obtained from neighboring edges and nodes for each specific node. Finally, to generate the desired output node features, the layer applies a neural network to the per-node representation, leveraging its ability to learn complex patterns and relationships within the data. Overall, the GINEConv architecture adeptly transforms the initial input features into refined and informative node representations.

Traffic Safety Metrics

Accelerating or Decelerating Rapidly

Rapid acceleration presents a multitude of risks and hazards that demand recognition and mitigation [36]. These hazards encompass the potential loss of control, reduced traction, and an extended stopping distance. Similarly, rapid deceleration, commonly known as sudden braking, entails its own set of dangers, such as whiplash injuries, compromised balance, and an increased likelihood of rear-end collisions. Safeguarding individuals necessitates a meticulous assessment of acceleration or deceleration, treating it as the rate of speed change. Consequently, this parameter can be expressed in terms of the g-force experienced by each participant, ensuring a comprehensive evaluation of safety measures.

We have established specific thresholds for acceleration and deceleration based on the generally accepted safe limits observed in most vehicles [11,12,16, 19, 62]. For acceleration, we have set the threshold at 0.6 g, indicating the capability to reach a speed of 60 mph from a standstill in less than 5 seconds. This ensures quick and efficient acceleration without compromising safety.

Similarly, for deceleration, we have set a threshold of 0.5 g, which corresponds to the maximum force experienced during safe braking. This allows for effective and controlled deceleration, ensuring the vehicle can come to a stop swiftly while maintaining safety standards.

Time to Collision

The time to collision (TTC) is a critical measurement used to assess the probability of a collision between two traffic participants. It provides a quantitative indication of the time required for two participants to collide if their current trajectories remain unchanged. The TTC metric plays a vital role in predicting and preventing crashes by enhancing the effectiveness of collision avoidance systems. Typically, TTC is computed based on the relative distance and velocities of the two participants. By estimating these parameters, it becomes possible to determine the time remaining until the participants reach critical proximity that could result in a collision.

The calculation of the TTC parameter is applicable when two participants are moving longitudinally, either in the same direction or opposite directions. The information about the longitudinal relationship between a pair of participants is obtained from the SSG. The TTC value is usually compared to a predefined safe threshold. If the calculated TTC falls below this threshold, it is considered dangerous, indicating an increased likelihood of a crash.

The purpose of establishing a safe threshold for TTC is to define a clear criterion that helps identify critical situations where there is insufficient time remaining to take appropriate actions before a potential collision. By defining this threshold, it becomes possible to prompt timely responses and implement necessary measures to avoid crashes.

One limitation of TTC as a safety measure in intersection scenarios is its tendency to generate a significant number of false violations, which may not necessarily indicate unsafe driving behavior. This issue arises because vehicles often decelerate when approaching a stop signal, which can mistakenly trigger TTC calculations suggesting a potential collision with the vehicle ahead. The fundamental assumption underlying TTC is that vehicles maintain their velocities, which does not hold true in situations where vehicles slow down to stop or accelerate to cross intersections, such as when encountering traffic signals.

In this study, we selected a threshold value of 2 seconds for TTC based on research on safe following distances and speed [11, 19]. While TTC is an important safety metric, it alone cannot provide accurate analysis, particularly in the case of intersections. Hence, we adopted an approach by incorporating factors such as vehicle speed, braking, and TTC to identify collision-prone situations more effectively.

Post-encroachment Time

Post-encroachment time (PET) is a crucial metric used to assess the risk of collisions in traffic scenarios. It represents the time interval between one traffic participant leaving an encroached region and another participant entering the same space thereafter. PET directly measures the amount of time available for road users to react to potential hazards, making it a valuable indicator of collision probability.

To evaluate PET values, this study focused on pairs of participants that exhibit either a lateral or intersecting relationship, as determined by the SSG. Intersecting relationships commonly occur at traffic intersections or when one road merges into another. Lateral relationships, on the other hand, arise when two participants occupy adjacent lanes and may potentially cross paths if one or both participants decide to change lanes.

Unlike previous metrics, the calculation of PET cannot be obtained through a straightforward formula. It necessitates the identification of instances where a current participant intersects with a point previously traversed by another participant. Crucially, this calculation considers the requirement that the two participants share a lateral or intersecting relationship. By accounting for these factors, PET provides a comprehensive assessment of collision risk in traffic scenarios.

A threshold of 1.5 seconds has been established for the PET metric, taking into account previous research on PET values for medium and high-risk scenarios [63]. If the estimated PET value of two vehicles involved in a situation is less than or equal to 1.5 seconds, it is considered unsafe.

Appendix E: Network Theory

Network theory is a field of study that focuses on analyzing and understanding complex systems composed of interconnected elements. It has found numerous applications in various domains, including transportation and traffic analysis. The structure of these networks, referred to as topology, holds important information and can be useful in uncovering insights about traffic patterns and flow. The goal of network analysis is to effectively manage the complexity of the network and extract meaningful information about traffic behavior and performance.

Networks can be used to depict various types of data. The nodes in these networks can symbolize different components such as vehicles or pedestrians, and the edges between them represent the connections or relationships between them.

The topological properties of these networks, including the arrangement of nodes and edges, play a crucial role in identifying key structures and relationships within the network. These properties can be applied to the entire network or specific nodes and edges to better understand the traffic flow and optimize transportation systems.

Node-level Features

Node features provide information regarding the structure and position of nodes in the network. By analyzing these node-level features, we can gain insights into the role and importance of individual nodes in the network, as well as identify patterns and trends in the network's overall structure and behavior.

Node Degree

Node degree refers to the number of edges connected to the node. This parameter is a critical factor that impacts various aspects of network analysis, including the centrality of a node. In directed networks, nodes have two different degrees: out-degree and in-degree. Out-degree represents the number of edges originating from a node, while in-degree refers to the number of edges ending at a node. Nodes with high degrees are more connected to other nodes and can have a greater influence on the network as a whole. On the other hand, nodes with low degrees are less connected to other nodes and may have a smaller impact on the network as a whole.

Centrality

There are various forms of centrality, each capturing a unique concept. Centrality can be computed for nodes and edges and provides a measure of their importance in terms of connectivity or information flow within the network. The degree of a node significantly impacts many centrality measures, such as degree centrality. However, more advanced forms of centrality, such as betweenness centrality, consider a wider range of factors and therefore result in a reduced influence of node degree.

Eigenvector centrality, also known as eigencentrality or prestige score, is a measure of the influence of a node in a network. High-scoring nodes contribute more to the score of a node in consideration. This is a recursive formulation, as the score of a node is dependent on the scores of its neighboring nodes, and so on. Rewriting this in a matrix form and finding the eigenvector corresponding to the largest eigenvalue gives the centrality. The following equation shows the formulation and the matrix representation:

$$\mathrm{c_v} = \frac{1}{\lambda} \sum_{u \in N(v)} c_u \Leftrightarrow \quad \lambda \ \mathrm{c} = \mathrm{Ac}$$

Here, A is the adjacency matrix such that A_{uv} = 1 if u is a neighbor of v; λ is the normalization constant, and the eigenvector c_{max} corresponding to the largest eigenvalue λ_{max} is used for centrality.

A node with high eigenvector centrality is connected to other nodes that are also highly connected and important in the network. This means that the node has a high influence in the network and can spread information or influence throughout the network more effectively. Conversely, a node with low eigenvector centrality is connected to nodes that are less important in the network and therefore has less influence on the network as a whole.

Betweenness centrality is used to determine the importance of a node within a network based on its ability to connect other nodes. Specifically, it is calculated as the number of shortest paths that pass through a node divided by the total number of shortest paths in the network. This results in a score between 0 and 1. Nodes with high betweenness centrality (close to 1) are considered to be critical in maintaining the flow of information or resources throughout the network because they act as mediators or connectors. Nodes with low betweenness centrality (close to 0) lie on fewer of the shortest paths between pairs of nodes in the network and therefore are considered to have less control over the flow of information or resources.

Closeness centrality is used to determine the importance of a node within a network based on its ability to reach other nodes in the network. Specifically, it calculates the reciprocal of the sum of the shortest distances between a node and all other nodes in the network. Nodes with high closeness centrality are considered to be important because they have a higher ability to receive and disseminate information or resources within the network. Conversely, nodes with low closeness centrality have a longer path to reach other nodes in the network and therefore have a lower ability to receive and disseminate information or resources.

Clustering Coefficient

The clustering coefficient is a measure used to determine the degree to which nodes in a network are connected to each other. Specifically, it quantifies the extent to which a node's neighbors are themselves connected to each other. Nodes with high clustering coefficients are said to form clusters or cliques, where each node is connected to multiple other nodes in the cluster. The calculation of clustering coefficient involves determining the fraction of a node's neighbors that are also neighbors of each other. This value is then averaged over all nodes in the network to give a score between 0 and 1, with higher scores indicating that the node is part of a tightly knit group or cluster within the network, where there are many connections between nodes. A low score indicates that the node may not be part of a tightly knit group or cluster within the network and there are fewer connections between nodes.

Link/edge-level Features

Link-level features provide information about individual edges or links in a graph and can be used to understand the relationships between nodes in the network. By analyzing these link-level features, we can gain insights into the patterns and trends of relationships between nodes in the network and also identify influential edges and understand their role in the network's overall behavior.

Local Neighborhood Overlap

Local neighborhood overlap measures the similarity between the sets of neighbors of two nodes in a network. If two nodes in a graph have many neighbors in common, then they are likely to have a stronger relationship or influence on each other.

Jaccard's coefficient measures the similarity between two sets of data. It is defined as the ratio of the number of common neighbors of two nodes to the total number of neighbors of the two nodes combined. A high Jaccard's coefficient between two nodes signifies that they have a significant number of common neighbors, indicating that they are likely to have a stronger relationship or influence on each other. Conversely, a low Jaccard's coefficient between two nodes signifies that they have few common neighbors, indicating that they may have a weaker relationship or influence on each other.

Adamic-Adar index is based on the idea that the importance of a shared neighbor between two nodes is inversely proportional to the number of neighbors that the shared neighbor has in the network. The Adamic-Adar index between two nodes is calculated by summing the inverse logarithm of the degrees of all the shared neighbors of the two nodes. The degree of a node is the number of edges that it has in the network. The formula for calculating the Adamic-Adar index is shown below:

$$\sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{\log\left(\deg(u)\right)}$$

Here, $\deg(u)$ refers to the degree of node u that is a common neighbor of nodes v_1 and v_2.

A high Adamic-Adar index between two nodes indicates that they have many common neighbors with low degree, which implies that their similarity is based on connections to less important nodes in the network. In contrast, a low Adamic-Adar index between two nodes indicates that they have few common neighbors with low degree, which implies that their similarity is based on connections to more important nodes in the network.

Global Neighborhood Overlap

Global neighborhood overlap measures the similarity between the sets of neighbors of all pairs of nodes in a network. It quantifies the extent to which nodes in the network have similar sets of neighbors and therefore reflects the overall structure and connectivity of the network.

Katz index evaluates the relative importance of each node based on its direct and indirect connections to other nodes. It counts the number of walks of all lengths between a pair of nodes and can be formulated as follows:

$$S = (I - \beta A)^{-1} - I$$

Here, S is the Katz index matrix, β is the discount factor, I is an identity matrix, and A is the adjacency matrix.

Safety Features from Graph

Table 6 and Table 7 present the node and link features, respectively, for one of the intersections in Virginia. This particular intersection has a documented history of a high frequency of crashes. The node and edge features provided in the tables were obtained from a 50-minute video feed captured during the afternoon. The distance margin for creating the graph was chosen as 10 meters. Based on the obtained graph, these features were calculated. The details of all measures in the tables can be found in Appendix D: Technical Background.

Table 6. Average Values for Various Node-level Features such as Degree, Centrality, and Clustering Coefficient Computed for One of the Videos from the VDOT Database

Node	Count	Degree	Centrality			Clustering coefficient
			Eigenvector	Betweenness	Closeness	
Car	8177	2.48659	0.18796	0.03496	0.24433	0.32658
Truck/Bus	3110	2.57098	0.21295	0.02449	0.25161	0.3795
Pedestrian	120	1.46054	0.10737	0.016178	0.18812	0.25529
Bicycle	23	4.81395	0.19966	0.05233	0.41798	0.42497
Motorcycle	23	2.6279	0.24813	0.05588	0.3292	0.40337

Table 6 presents some interesting characteristics about the traffic scene; it is evident that cars, trucks, and motorcyclists typically have an average degree of approximately 2.5. This suggests that, on average, they are surrounded by two to three other vehicles or traffic participants. On the other hand, bicyclists have a higher degree because they share edges with both pedestrians and vehicles. However, due to the narrower cycling lanes, there tend to be more traffic participants within a 10-meter proximity to cyclists. Consequently, cyclists are generally at a higher risk, as they have a greater number of traffic participants in proximity.

Moreover, among all the road users, bicyclists experience the least amount of safety. This is primarily due to their higher exposure to nearby traffic participants and the inherent risks associated with sharing the road with vehicles. In contrast, pedestrians typically walk on sidewalks and thus have the fewest number of connections or edges with other participants. Additionally, it appears that there are only a limited number of pedestrians captured in the video data.

Eigenvector centrality measures a traffic participant's significance within a network. In Table 6, it is evident that pedestrians hold the lowest influence on the network, while other participants exhibit relatively comparable levels of influence. Notably, motorcyclists emerge as the group with the most substantial impact.

Closeness centrality is a measure that quantifies how close a traffic participant is to other participants within a network. It is calculated based on the distances between the participant and all other participants. The values of closeness centrality follow a similar trend as the average degree for different traffic participants in the network.

The clustering coefficient is a measure that indicates the tendency of nodes in a network to form clusters or groups. It is observed that bicycles exhibit the highest tendency to form clusters, followed by motorcycles. In contrast, pedestrians have the least inclination to form such clusters.

Table 7. Average Values for Various Link-level Features such as Jaccard's Coefficient, Adamic-Adar Index, and Katz Index Computed for One of the Videos from the VDOT Database

Node 1	Node 2	Average	
		Jaccard	Adamic-Adar
Bicycle	Car	0.20755	0.85407
Bicycle	Pedestrian	0.05555	0.2103
Bicycle	Truck/Bus	0.19010	0.89363
Car	Car	0.2341	0.82552
Car	Motorcycle	0.21041	0.72338
Car	Pedestrian	0.191915	0.5831
Car	Truck/Bus	0.25405	0.89782
Motorcycle	Pedestrian	0.03809	0.16156
Motorcycle	Truck/Bus	0.2	0.63092
Pedestrian	Truck/Bus	0.12771	0.43341
Truck/Bus	Truck/Bus	0.29414	0.99186

Based on the local-overlap data (Jaccard's coefficient and Adamic-Adar index) presented in Table 7, motorcycles and pedestrians, and bicycles and pedestrians have the least number of edges with common traffic participants. This could be because most of the edges of traffic participants are with cars given the high number of cars in the video. The values for the various other node types are very similar. It is additionally observed that, in the case when two trucks/buses are present in the frame, there are more common traffic participants than usual.

The Katz index is typically used in network analysis to measure similarity or overlap between nodes based on their connectivity patterns within a network. It is commonly applied to social networks, where nodes represent individuals and edges represent relationships between them. In such networks, the Katz index can capture the degree of similarity or overlap in terms of shared connections.

However, in a traffic network where nodes represent traffic participants and edges represent proximity, the concept of similarity or overlap between nodes may not be meaningful in the same way. In this context, the focus is on proximity and interaction between participants rather than shared connections.

In traffic networks, other measures such as traffic flow, congestion, shortest paths, or centrality measures like betweenness centrality or closeness centrality may be more relevant for understanding the dynamics and efficiency of the network.

Appendix F: New Drone Data from Virginia

Traffic Datasets

Datasets play a major role in any research. Traffic-related research has been historically carried out by using crash reports and statistics. In recent years, we have seen more emergence of advanced sensor-based datasets. These datasets are primarily camera data with traffic videos. The videos are generally collected in two ways: using infrastructure cameras and using drones. In an ideal situation, we should receive traffic information from all directions of traffic, including entry and egress. Also, we should collect data from diverse intersections to cover the large spectrum of behaviors in intersections. Most of the infrastructure camera-based studies to date were very small scale and targeted one to six intersections [64]. Traffic cameras from departments of transportation are often a good source of information with wide geographical coverage. In this project, our primary target was traffic camera data. Alternatively, drone-based videos can capture a wide area around the intersection, depicting a much better understanding of the interplay of traffic from multiple directions. Along with the trajectory data, these datasets typically include GPS coordinates of the intersection location and, in some cases, the road layout based on the OpenStreetMap format. Additionally, certain trajectory datasets may also include traffic signal information, further enhancing their usefulness for analyzing traffic scenarios. Some of the most popular intersection datasets are inD [55], TAF-BW [56, 57], INTERACTION [58], and SIND [59]. In this project, we specifically explored traffic camera data from Virginia.

New Drone-based Traffic Dataset

The intersection datasets we discussed primarily focus on Europe and China, while only considering very few scenarios from the United States. However, it is crucial to note that the U.S.-based datasets we examined do not include traffic signal information. This omission is significant because various states, as well as certain cities, have varying regulations regarding the permissibility of making a right turn on a red signal. Furthermore, driving styles exhibit significant disparities across states, cities, and even college towns. To address these limitations and the inadequate quality of the VDOT videos, we put forth the proposition of creating a new dataset.

We began with various intersections in Blacksburg, VA, where a 4,000-pixel resolution DJI drone was flown above the intersection at an altitude of 100-120 m. Two phone cameras were used at diagonally opposite sides of the intersection to capture the traffic signal information for two sides from each camera. We also used a light flash visible in all three cameras to synchronize all three videos. Our dataset collection strategy thus also provides us with the traffic signal information that not many public datasets provide. Finally, these videos have been provided to Tsinghua University, who provided the annotations in the form of a lanelet2 map (OpenStreetMap format) and the position and kinematic information of various traffic scenario participants.

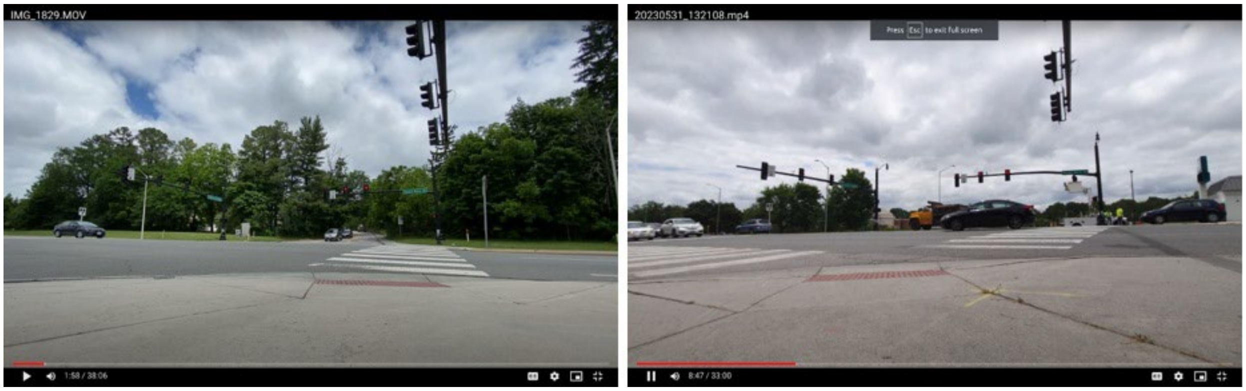


Figure 27. Photos. Two mobile phones placed on diagonally opposite corners of the street such that they capture data from two of the traffic signals.

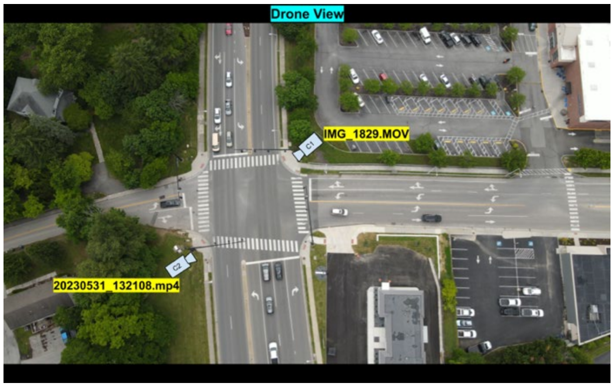


Figure 28. Photo. A frame from the drone view annotated with the mobile phone camera location.