Enhancing Road Safety through Machine Learning for Prediction of Unsafe Driving Behaviors

Akash Sonth

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

Lynn Abbott, Co-chair

Abhijit Sarkar, Co-chair

Creed Jones

July 19, 2023

Blacksburg, Virginia

Keywords: traffic safety, graph neural network, distraction monitoring, transformers

Enhancing Road Safety through Machine Learning for Prediction of Unsafe Driving Behaviors

Akash Sonth

(ABSTRACT)

Road accidents pose a significant threat, leading to fatalities and injuries with far-reaching consequences. This study addresses two crucial challenges in road safety: analyzing traffic intersections to enhance safety by predicting potentially risky situations, and monitoring driver activity to prevent distracted driving accidents. Focusing on Virginia's intersections, we thoroughly examine traffic participant interactions to identify and mitigate conflicts, employing graph-based modeling of traffic scenarios to evaluate contributing parameters. Additionally, we leverage graph neural networks to detect and track potential crash situations from intersection videos, offering practical recommendations to enhance intersection safety. To understand the causes of risky behavior, we specifically investigate accidents resulting from distracted driving, which has become more prevalent due to advanced driver assistance systems in semi-autonomous vehicles. For monitoring driver activity inside vehicles, we propose the use of Video Transformers on challenging secondary driver activity datasets, incorporating grayscale and low-quality data to overcome limitations in capturing overall image context. Finally, we validate our predictions by studying attention modules and introducing explainability into the computer vision model. This research contributes to improving road safety by providing comprehensive analysis and recommendations for intersection safety enhancement and prevention of distracted driving accidents.

Enhancing Road Safety through Machine Learning for Prediction of Unsafe Driving Behaviors

Akash Sonth

(GENERAL AUDIENCE ABSTRACT)

Road accidents are a serious problem causing numerous deaths and injuries each year. By studying driver behavior, we can uncover common causes of accidents like distracted driving, impaired driving, speeding, and not following traffic rules. New vehicle technologies aim to assist drivers, raising concerns about driver attentiveness. It is crucial for car manufacturers to develop systems that can detect and prevent accidents, especially in semi-autonomous vehicles. This study focuses on intersections in Virginia and examines driver behavior within vehicles to identify and prevent dangerous situations. We create models of different traffic scenarios using graphs/networks and utilize machine learning to identify potential accidents. Our objective is to provide practical recommendations for improving intersection safety. Existing datasets and algorithms for recognizing driver activities often fail to capture common distractions like eating, drinking, and phone use. To address this, we introduce two challenging datasets specifically designed to capture distracted driving activities. Finally, we try to understand the predictions bade by the chosen deep learning model by visualizing the inner workings.

Dedication

Dedicated to familia

Acknowledgments

I am thrilled and honored to convey my profound gratitude to my esteemed supervisors, Prof. Lynn Abbott and Dr. Abhijit Sarkar. Their unwavering guidance and steadfast support throughout my research journey have filled me with immense joy and an overwhelming sense of pride. Additionally, I extend my heartfelt appreciation to my committee member, Prof. Creed Jones, for his invaluable feedback that significantly contributed to the improvement of my thesis. The exceptional expertise and mentorship provided by these individuals have been priceless, and I am genuinely grateful for their unwavering presence and support.

I would also like to extend my deepest appreciation to Calvin Winkowski for providing me with the remarkable opportunity to begin my path as a Student Research Assistant at the prestigious Virginia Tech Transportation Institute (VTTI). Furthermore, I extend my heartfelt thanks to Dr. Abhijit Sarkar and VTTI for their generous provision of funding through the Graduate Student Research Assistantship. Their financial support has played a pivotal role in enabling me to pursue my research goals and has been instrumental in my academic and professional development.

I am sincerely grateful to the Graduate School and the Bradley Department of Electrical & Computer Engineering at Virginia Tech for providing me with the opportunity to pursue my Master's degree. I consider myself incredibly fortunate to be affiliated with Virginia Tech and the Virginia Tech Transportation Institute.

My deepest appreciation and gratitude go to my exceptional labmates, Hirva Bhagat, Sandesh Jain, Suren Thapa, and Han Xu, for their invaluable assistance and unwavering support throughout the various stages of my projects. Their dedication and collaboration have made my time at VTTI truly meaningful and fulfilling. I am genuinely grateful to have had the privilege of collaborating with such outstanding individuals.

I am indebted to my parents, Meenakshi Sonth and Prakash Sonth, as well as my sister, Neeti Sonth, for their blessings, affectionate encouragement, and unwavering cooperation throughout every aspect of my life. Without their constant support, it would have been impossible for me to reach the position I am in today.

I am immensely grateful to the College of Engineering for awarding me the first-author fellowship grant, which provided me with the invaluable opportunity to register my paper for the IEEE IV conference. Additionally, I would like to express my sincere gratitude to the Center for Human-Computer Interaction for granting me the travel grant, which immensely supported my conference-related travel and covered various expenses. Their generous support has been instrumental in enabling me to participate in this significant event, and I am truly thankful for their invaluable contributions.

I am thankful to the various employees at VTTI without whom my research could not have been successful. I would also like to extend my thanks to Maximilian Zipfl from the FZI Research Center for Information Technology for his invaluable assistance, insightful perspectives, and helpful suggestions.

My heartfelt gratitude goes to my roommates, Dhanush Dinesh, Aaditaya Roshan, and Siddhant Taliyan, for making my time in Blacksburg truly unforgettable. Their positive influence and shared experiences have enriched my journey, and I will forever cherish the memories we have created together. I would also like to express my sincere appreciation to both the new friends I have made and the cherished old friends back in India. To all the individuals mentioned here, as well as those whose names may have inadvertently been omitted, I am filled with a profound sense of appreciation.

Contents

List of Figures

List of Tables

List of Abbreviations

AV Autonomous Vehicle

CNN Convolutional Neural Network

GNN Graph Neural Network

OSM OpenStreetMap

SHRP Strategic Highway Research Program

VTTI Virginia Tech Transportation Institute

Chapter 1

Introduction

1.1 Background and motivation

Globally, roads serve as common pathways for various modes of transportation, including cars, buses, trucks, motorcycles, mopeds, pedestrians, animals, taxis, and other travelers. The accessibility provided to various destinations by motor vehicles has played a crucial role in fostering economic and social progress in numerous nations.

The unfortunate reality is that each year, vehicular accidents result in millions of fatalities and injuries. Road traffic accidents pose a serious and urgent threat, given their profound and far-reaching impacts, such as the tragic loss of lives, debilitating injuries, and extensive property damage.

Every year, approximately 1.35 million individuals lose their lives in road accidents worldwide [30]. On a daily basis, nearly 3,700 people fall victim to fatal crashes involving various modes of transportation such as cars, buses, motorcycles, bicycles, trucks, or pedestrians. Shockingly, more than half of these casualties consist of pedestrians, motorcyclists, or cyclists. These accidents have emerged as the eighth leading cause of death across all age groups globally, and they claim the lives of children and young adults between the ages of 5 and 29 more frequently than any other cause, surpassing even HIV/AIDS-related deaths [2]. In terms of economic impact, it is projected that between 2015 and 2030, fatal and non-fatal crash injuries will incur a cost of approximately 1.8 trillion USD (measured in 2010) on the global economy. To put this in perspective, this amount is equivalent to an annual tax of 0.12% on the gross domestic product (GDP) of the entire world [13].

The Virginia Highway Safety Office conducted a comprehensive analysis of car accidents that occurred in the state in 2018 [67]. The study revealed several significant factors contributing to these accidents. Among the primary causes, speeding and aggressive driving were responsible for nearly 20% of all reported accidents. Distracted driving emerged as the primary cause in approximately 18.5% of the accidents. Additionally, driving under the influence of alcohol or drugs accounted for 5.4% of the accidents.

Several other common causes were also identified, including following too closely and rear-end accidents, failure to yield, side-impact accidents, fatigued driving, falling asleep at the wheel, as well as accidents caused by defective vehicles and vehicle parts. These factors collectively contribute to the occurrence of car accidents in Virginia.

The most common reasons for collisions are distraction, misinterpretation of the environment, and lack of reaction time. If the driver is warned 0.5 s earlier, 60% collisions could be avoided. This reduction is raised to 90% if warned 1.5 s in advance [8]. By actively monitoring traffic scenes and promptly identifying anomalies or potential risks, an advanced surveillance system could play a crucial role in preventing numerous accidents. The pivotal component of this prevention system would be its capability to accurately calculate a risk score for every situation and effectively communicate this crucial information, thereby mitigating the likelihood of any potential collisions. This proactive approach holds immense potential for reducing collisions between multiple traffic participants. However, it is important to note that this system may not be able to detect incidents where a vehicle deviates from the road or collides with stationary objects due to the driver's lack of attention.

Of all the road accidents, those due to distracted driving are steadily increasing, with a significant portion being attributed to mobile phone use while driving [63]. The National Highway Traffic Safety Administration (NHTSA) defines distracted driving as any activity that diverts attention from driving [64]. The Centers for Disease Control and Prevention (CDC) has identified three major types of distractions - (a) visual: taking one's eyes off the road, (b) manual: taking one's hands off the driving wheel, and (c) cognitive: taking one's mind off driving - with different driver behaviors having varying impacts on each category [14]. Over-reliance on Advanced Driver Assistance Systems (ADAS) in modern semi-autonomous vehicles is a prevalent cause of drivers not paying attention to the road, thus highlighting the increasing importance of research on driver attention.

The classification of autonomous vehicles into six levels of autonomy, as defined by the Society of Automotive Engineers (SAE) [59], provides a framework to understand their capabilities. At Level 0 (L0), traditional cars lack automation, relying entirely on the driver's control. At Level 5 (L5), vehicles encompass full automation capable of handling all driving tasks without human intervention, but are not yet a reality. These distinct levels of autonomy underscore the need for drivers to remain vigilant and avoid complacency, particularly when operating vehicles that offer varying degrees of automation.

The highest commercially available autonomous capability currently lies partially at Level 3 (L3). Level 3 denotes conditional automation, where the vehicle can autonomously handle all driving tasks under specific conditions, typically on highways. However, the driver must still remain prepared to take over control when alerted. Studies have indicated that drivers often become distracted and engage in secondary activities when operating highly automated vehicles. Therefore, it is crucial to comprehend the driver's exact posture, level of attentiveness, and readiness to regain control in order to ensure safe vehicle operations. Camera-based methods have proven to be useful in assessing these factors. The readiness assessment encompasses a complex process and may involve multiple components, such as monitoring the driver's gaze attention, cognitive alertness, response time to assume control of the steering wheel, engagement in secondary activities, and time required to assume control of the braking systems [56].

As the market sees a rise in commercial semi-autonomous vehicles, ensuring safe driving will require the integration of traffic anomaly and driver distraction detection techniques. These components are vital not only for the overall safety of the vehicles but also for activating features in ADAS, particularly Collision Avoidance Systems (CAS) that necessitate planning evasive actions. Although the transition to full automation will occur gradually, drivers will still need to take control in uncertain situations. Consequently, driving will transform into a shared activity between humans and machines in the foreseeable future.

To facilitate this transition, it will be crucial to monitor the traffic scenario for potential risky maneuvers and convey pertinent information to drivers. Furthermore, the use of Driver Monitoring Systems becomes essential in alerting distracted drivers. These systems will play a pivotal role in monitoring traffic safety and facilitating a smoother and more dependable transition to autonomous driving mode.

1.2 Problem Statement and Research Questions

Traffic scenes are complex and constantly changing environments where various elements such as vehicles, pedestrians, and cyclists interact within the framework of road infrastructure, regulations, and traffic signals. These interactions offer valuable insights into the likelihood of accidents or unfavorable incidents. Typically, intersection infrastructure includes traffic signals equipped with cameras that monitor vehicles following the corresponding signals. However, manually analyzing the raw video frames from these cameras can be a tedious and time-consuming process. To enhance the efficiency of analysis and extract meaningful information, it is essential to present this intricate data in a streamlined and user-friendly format.

As emphasized in Section 1.1, the issue of distracted driving has emerged as a significant contributor to the rising number of road accidents in recent times. There are several traffic cameras (that are basically video cameras) installed at various traffic signals, intersections, and highway segments that can be used to monitor traffic conditions and enforce traffic laws. While traffic cameras play a vital role in monitoring traffic violations, their ability to identify the root causes of these infractions is limited. Therefore, it is crucial to adopt a proactive approach to preventing potential collisions caused by distracted driving, rather than solely relying on identifying accidents after they have occurred. In light of this, we propose the installation of an in-cabin camera that monitors the driver and alerts them when distracted. By implementing preventive measures, we can effectively mitigate the risks associated with distracted driving before they escalate into actual accidents.

In recent years, the computer vision community has made enormous progress in video understanding. Activity recognition (alternatively action recognition) is an active subfield in computer vision that identifies human actions from videos. Now, these methods are developed on datasets that contain hundreds of actions that a human can perform including running, playing guitar, etc. Drivers' secondary behavior in comparison is a much-limited set of classes and often not included in the large-scale activity recognition pipeline. Also, secondary behaviors are performed by drivers while seated behind a steering wheel, hence, restricting the movement of the driver significantly compared to the diverse sets in traditional activity recognition datasets like Kinetics-400 [40].

To address these problems, the following research questions will be investigated:

- Can a traffic scenario be effectively represented in a simplified format while still preserving the necessary and valuable information?
- What are the current limitations in sensing and how do they translate to annotation and modeling?
- How well can the progress in activity recognition be translated to identify secondary behavior for drivers?
- Do we have adequate resources including dataset and associated annotations for determining driver distractions?

By exploring these research questions, we aim to enhance the analysis process of traffic scenes, enable effective identification of secondary behaviors, improve safety assessments, and contribute to the development of machine learning techniques in the context of traffic modeling.

1.3 Scope and Objectives

In this study, our primary focus is on assessing the likelihood of hazardous scenarios that may result in collisions. The objective is to enhance safety by proactively preventing vehicle-vehicle or vehicle-pedestrian accidents, as well as near-crash incidents. Additionally, we investigate the prediction of various driver distractions that could divert their attention from the task of driving, potentially leading to risky situations.

Presently, there is a lack of datasets encompassing both in-cabin drivers' videos and the corresponding traffic scenarios observed from a "bird-eye" view. Consequently, we address this limitation by independently examining two aspects: driver distraction within the cabin and potential collision situations in a traffic scenario based on the video feed from intersection traffic cameras.

Our contributions are summarized as follows:

- Represented traffic scenarios at two intersections based on the video feed from traffic cameras into an intuitive graph-based representation that captures essential details of the traffic participants such as position, orientation, and interactions.
- Developed a new approach using Graph Neural Networks to assess traffic scenarios for anomalous situations that could possibly lead to a crash.
- Updated two challenging datasets to support research involving the recognition of drivers' activities.
- Comparison is made between state-of-the-art CNN models and more recent video transformer-based models that utilize attention modules.

Below are publications related to this thesis.

- Sonth, A., Sarkar, A., Bhagat, H., Abbott, L., 2023. "Explainable Driver Activity Recognition Using Video Transformer in Highly Automated Vehicle." In Proceedings of the IEEE Intelligent Vehicles Symposium.

Chapter 2

Related Work

2.1 Traffic scene representation

In this section, we explore the existing literature for representing traffic scenes, distinct from utilizing raw video frames. We focus on two types of data sources: cameras positioned at traffic intersections or drones capable of capturing a bird's-eye view of the scenario, and cameras mounted on ego-vehicles facing the road. In the latter case, the traffic scenes are represented from the perspective of the ego-vehicle.

Traffic scenes are complex situations where multiple participants interact and respond to each other while following road structures, rules, and signs/signals. The modeling of traffic scenarios has made significant progress by adopting a graph-based approach. This approach shows great potential in simplifying and improving the analysis of intricate traffic scenarios by representing participants as nodes and their interactions as edges in a graph. This representation enables quicker analysis, facilitates the utilization of graph theory algorithms, and enables the integration of contextual information. As a result, this advancement has the power to transform traffic management, enhance safety measures, and optimize transportation systems in the ever-growing congestion of urban environments.

A straightforward method for modeling a wide range of objects, including vehicles, buildings, pedestrians, bicycles, and roadside elements, is presented in Mo et al.'s work [48]. In this approach, each object type functions as a node within a network. 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 various aspects, such as topological relations, relative orientation, relative trajectories, relative speed, metric relations, and order relations. Additionally, vehicles also establish connections with vertical structures (such as lane dividers, signboards, traffic signals, and guard rails), road segments (including zebra crossings, edge lines, stop lines, center lines, etc.), roadsides, intersections, and road markings. By incorporating these diverse relationships, the model provides a comprehensive representation of the intricate interactions between vehicles and their surroundings.

A scene graph is created based on the video feed of a road-facing camera in the ego-vehicle [46]. Mask R-CNN [35] is utilized for object identification within an image. Additionally, the authors employ an inverse perspective mapping technique, commonly referred to as a "birds-eye view" projection. This projection aids in estimating the positions of objects in relation to the ego car, thereby facilitating the creation of three distinct types of pair-wise relationships between objects: proximity, directional, and belonging relations. Consequently, this process transforms each image into a comprehensive scene graph, where individual nodes are initialized using one-hot vectors as embeddings.

Diehl et al. [19] construct a graph to represent car features such as position or velocity, where nodes represent the vehicles. The edges of the graph are determined based on the Euclidean distance of a vehicle with all nearby vehicles.

In [50], the nodes are depicted by various objects, such as vehicles, buildings, pedestrians, bicycles, roadside elements, vertical structures (like lane dividers, signboards, traffic signals, guard rails), road markings (such as zebra crossing, edge line, stop line, center line), road segments, and intersections. These objects are interconnected through different relationships, including topological (Region Connected Calculus 8 [17]), relative orientation, relative trajectories, relative speed, quantitative distance, and order relations. To effectively handle the size of the graph, 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 together, forming sub-graphs 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.

In [11], the authors propose a method that utilizes ontologies to provide semantic descriptions of traffic scenes. These descriptions are then utilized for situation classification. By incorporating knowledge of road regulations, the authors are able to define rules and facilitate the classification process. In a separate study [74], the authors present a generative model aimed at creating 3D urban scenes that offer insights into the structure, participants, and traffic patterns. The authors employ a small amount of real data to train the model on the underlying patterns and rules. Once trained, the model can generate new scenes based on the learned information.

2.2 Traffic forecasting and safety analysis

Chen et al. [15] utilized high-resolution video feeds obtained from traffic cameras operated by the Arizona Department of Transportation. These video feeds were subjected to a transformation process, converting them into a "bird-eye" perspective. The transformed view was then used for object detection and tracking purposes. The authors focused specifically on analyzing small road segments where vehicles were moving in the same direction, primarily on highways rather than at intersections. However, it should be noted that the selection of all vehicles in the region for calculating safety metrics may include vehicles that are not necessarily relevant or required for the analysis.

The authors in [62] propose a computationally inexpensive method based on image processing to predict vehicle speeds using monocular video feed obtained from fixed cameras. They employ optical flow techniques to distinguish moving vehicles from the stationary background. By tracking the centroids of the vehicles, they monitor their movement throughout the video footage. To convert the speed measurements from pixels per second to kilometers per hour, a transformation factor is applied.

In [12], the researchers utilized a Graph Neural Network (GNN) to improve the accuracy of vehicle trajectory prediction. By leveraging this advanced technique, they aimed to anticipate the future paths of vehicles more effectively. Subsequently, they employed the predicted trajectories to calculate stochastic positions and assessed whether these positions would intersect with any other vehicles, potentially leading to collisions.

In [32], three characteristic scores are defined for a traffic scenario: interaction, anomaly, and relevance. Interaction represents the interplay of road users, anomaly highlights unusual phenomena, and relevance is the product of interaction and anomaly. These scores are combined across four layers: temporal punctual, track, spatial region, and overall. The scores are obtained using trajectory data (velocity, acceleration, position, and orientation), meta data (road user dimensions, type, and region usage permissions), and a semantic map (infrastructure information like lanes, driving paths, speed limits, and walkways).

Vehicle information such as position and velocity is utilized to create features associated with the corresponding nodes in [19]. The graph's edges are formed by considering all vehicles in the proximity of each vehicle. To exploit the graph structure, a Graph Attention Network

(GAT) model is employed. The layers of the GAT model compute the next representation for each node using an attention mechanism that takes into account information from all of its neighbors. By applying attention mechanisms, the model can selectively focus on relevant interactions and assign varying degrees of importance to different neighbors.

Another system described in [46] aims to determine the likelihood of a collision involving the ego-vehicle by analyzing real-time video feed from a road-facing camera fixed inside the ego-vehicle. The system consists of two main steps: scene-graph creation and spatiotemporal embedding modeling. A scene graph, which visually represents the surrounding environment, is constructed. Each node in the graph is assigned a color based on its attention score, indicating its relative importance in assessing the collision probability. Nodes with higher attention scores are considered more critical in predicting collisions. Multi-Relational Graph Convolutional Network (MR-GCN) layers are employed to update the embeddings. This is followed by Graph pooling and Graph Readout to obtain a single embedding for each frame. A Long Short-Term Memory (LSTM) model aggregates the embeddings across multiple timestamps, resulting in a final crash prediction classification that assesses the probability of a collision occurring.

In the study by [48], a comprehensive framework is proposed for predicting trajectories of multiple agents. The framework takes into account various factors such as individual dynamics, interactions, and road structure. To overcome the challenge of coordinate shifting and its impact on trajectory prediction, the authors adopt an approach where each agent is placed in its own coordinate system. This ensures that the recorded states of an agent can always be converted to its own coordinate system without affecting other agents, thereby eliminating any discrepancies caused by coordinate shifting.

The authors of [38] introduce a novel approach for estimating risk based on time-to-collision using a Gaussian Mixture Model. Additionally, they employ a Kalman filter to estimate the proportion of exposure to traffic conflicts. In another study for traffic safety [29], a forecasting model for scenario safety is presented, which utilizes particle swarm optimization and support vector machine techniques. The authors emphasize that this model is superior to a neural network approach.

The paper [37] discusses recent advancements in spatio-temporal forecasting using deep learning techniques, specifically convolutional and recurrent neural networks. Previous studies have utilized graph convolutional networks (GCNs) to capture spatial dependencies, but these models have a constraint where the importance weights of neighboring nodes remain fixed during information aggregation. This limitation leads to the neglect of diverse neighboring information. To overcome this constraint, the authors propose a novel mechanism called rank influence learning (RIL) that dynamically adjusts the weights of neighboring nodes during aggregation in GCNs. They introduce a new model, called diffusion convolutional recurrent neural network with rank influence learning.

2.3 Driver Action Recognition

The field of action recognition has undergone significant research advancements in recent years, focusing on developing network architectures that effectively capture both spatial and temporal information in videos. One notable architecture is the SlowFast network [24], which comprises two parallel pathways: a “slow” pathway that processes one frame out of every 16 frames, and a “fast” pathway that processes alternate frames. Although the SlowFast network aims to capture both semantic and temporal information, it is not efficient in extracting both simultaneously.

Another network architecture, known as the Temporal Spatial Network (TSN) [70], also incorporates two parallel pathways. The spatial pathway receives random RGB frames from video clips as input, while the temporal pathway receives optical flow fields from the same clips. The use of optical flow fields allows for more effective extraction of temporal information compared to the SlowFast approach. However, capturing all temporal information between frames often results in significant noise.

Recently, the Temporal Adaptive Network (TANet) [45] has been introduced, employing a two-level adaptive modeling approach. This approach breaks down the video-specific temporal kernel into two components: a location-sensitive importance map and a location-invariant aggregation kernel. By doing so, TANet improves the discrimination of temporal information from a local perspective while also capturing the temporal relationships with a global view of the video sequence.

The Video Swin Transformer [44] is a recent network architecture with higher model complexity compared to most CNN-based methods. Its parallel processing capability significantly speeds up computation time. Furthermore, it can process both spatial and temporal information using just one pathway, making it a promising approach for action recognition tasks.

To support driver action recognition, several datasets have been established for training and evaluating driver action recognition models. The Drive&Act dataset [47], recorded in a driving simulator with a multi-view multi-modal camera setup, provides extensive all-around data but may not be suitable for practical use in commercial vehicles. In contrast, datasets like the State Farm Distracted Driver Detection dataset [49] and the Distracted Driver Dataset [**?**] only contain still images of drivers, lacking temporal information. Other similar datasets include the LISA Driver Gaze Dataset [68] and the Driver Attention Dataset [60].

The SHRP2 dataset [20] is a large-scale dataset with relatively low resolution but contains a wealth of information, including secondary behaviors. Papakis et al. [52] present their work on in-vehicle occupant detection using this dataset, while Osman et al. [51] perform binary classification to detect the presence of secondary activities. This information is then used to predict one of three activity labels: "cellphone talking", "cellphone texting", and "passenger interaction." They achieve an accuracy of 99.89% on the binary classification task and 82.3% accuracy in determining one of the three activities. This work is the first to estimate driver activities within the major 10 classes while also providing explainability using the SHRP2 dataset.

2.4 Transformers in Computer Vision

Transformers, originally introduced by Vaswani et al. [66], have emerged as the dominant deep learning architecture in the field of Natural Language Processing (NLP) and are increasingly being utilized in computer vision tasks. Their ability to capture long-range dependencies while enabling parallel computation has made them highly valuable. One notable application in computer vision is the Vision Transformer (ViT) proposed in [21], which has shown promising results. However, ViTs face certain limitations, such as their quadratic computational complexity with respect to input image size, which hinders their effectiveness with high-resolution images. Moreover, the fixed token scale in ViT does not cater well to vision tasks involving elements of varying scales. To address these limitations, [43] introduced the Swin (Shifted Window) Transformer, which incorporates two innovative concepts: hierarchical feature maps and shifted window attention.

Video Transformers aim to capture spatiotemporal dependencies in video data. One key advantage of Video Transformers is their ability to handle input sequences of varying lengths, which is crucial for video data where sequence length can vary significantly. Additionally, the transformer architecture enables efficient parallel computation, making it well-suited for processing large-scale video datasets. Recent studies by [7], [4], [23] have demonstrated that Video Transformers achieve state-of-the-art performance on benchmark datasets for video classification [1] and action recognition tasks [40], [33]. Moreover, Video Transformers have been successfully applied to object detection in videos as well [75].

Chapter 3

Traffic Modeling

Significant progress has been made in modeling traffic scenarios through the adoption of a graph-based approach. Graphs provide a versatile framework for representing and analyzing complex relationships among different elements in a system. A graph-based representation enables the depiction of 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. Pedestrians and cyclists can also be incorporated into the graph as separate nodes, allowing for 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. The graph-based modeling thus enables faster analysis and extraction of relevant information.

Second, the graph-based representation allows for the application of 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 aid in identifying 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.

3.1 Dataset

In this work, we present our analysis on the VT-CAST (Traffic Cameras for Advanced Safety Technologies) 2020 dataset [6]. 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 for 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.

One major issue of the video feed from these cameras is the low quality of the videos, which are typically recorded at a resolution of 320×240 pixels. This low resolution introduces several limitations that impact the effectiveness of the system.

- Distant vehicles and pedestrians: Due to the low resolution, objects located far away from the camera are not clearly visible. Objects located far away from the camera suffer from reduced visibility, resulting in difficulties for our detection and tracking system. As a consequence, our approach fails to identify these distant objects accurately, potentially compromising safety and leading to an incomplete analysis of the traffic situation.

- Jittering bounding boxes: The low-resolution videos can cause bounding boxes (used to enclose and track objects of interest) to exhibit significant jittering or instability. This instability can make it challenging to accurately identify and track participants, such as vehicles or pedestrians, leading to less reliable data for further analysis.

- Bad weather conditions and low-lighting: In adverse weather conditions or low-light environments, the already diminished quality of videos presents additional obstacles in extracting valuable information. The scarcity of discernible details and contrasting elements poses significant challenges for object recognition and tracking algorithms, thereby diminishing the accuracy and dependability of any subsequent analysis.

- Feasibility of 3D object tracking: While 3D object detection and tracking could potentially provide more accurate heading information for traffic participants, implementing this approach is not feasible given the low-resolution videos. Creating precise 3D bounding boxes typically requires more detailed and higher-resolution data, which is not available with the current video feed quality.

3.2 Modeling traffic scenarios

In this section, we will discuss the methodology employed to create a graph-based representation of a traffic scenario.

3.2.1 Object detection and tracking

Object detection and tracking play a crucial role in determining the locations of participants through the identification of bounding boxes. However, in scenarios where the video quality is poor, the object detection process is prone to generating both false positives and false negatives. To overcome this challenge, it becomes necessary to explore and evaluate different algorithms to obtain the most effective model.

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. However, when the video quality is low, it becomes more difficult to distinguish between true detections and false detections.

False positives refer to instances where the object detection algorithm erroneously identifies an object that is not actually present. On the other hand, false negatives occur when the algorithm fails to detect an object that is indeed present in the video. These inaccuracies can significantly impact the reliability and effectiveness of the overall tracking system.

In this work, we used the widely popular YOLOv7 for object detection, followed by BoT-SORT (Simple Online and Realtime Tracking) [3] for object tracking.

BoT-SORT (Simple Online and Realtime Tracking) [3] is an advanced multi-object tracking algorithm that seamlessly integrates object detection, association, and re-identification techniques. With its exceptional accuracy across diverse datasets, BoT-SORT stands at the forefront of state-of-the-art algorithms in this domain.

The functioning of BoT-SORT starts with the identification of objects within each frame of a video. This is accomplished by leveraging a powerful pre-trained object detector, such as

YOLOv7 [69]. Notably, YOLOv7 excels in accurately detecting even small objects within lower-resolution frames. Once the objects are detected, BoT-SORT employs a Bayesian approach to associate them across frames, taking into account their appearance, motion, and social relationships.

In order to accomplish object detection, we rely on the robust YOLOv7 pre-trained model. This model has proven to be highly effective in identifying objects, even when dealing with lower-resolution frames. Subsequently, the detected objects are linked across frames utilizing the BoT-SORT algorithm, which is widely acclaimed as the most efficient and accurate tracking method currently available. Results for object detection using YOLOv7 on a sample frame are shown in Fig. 3.1.

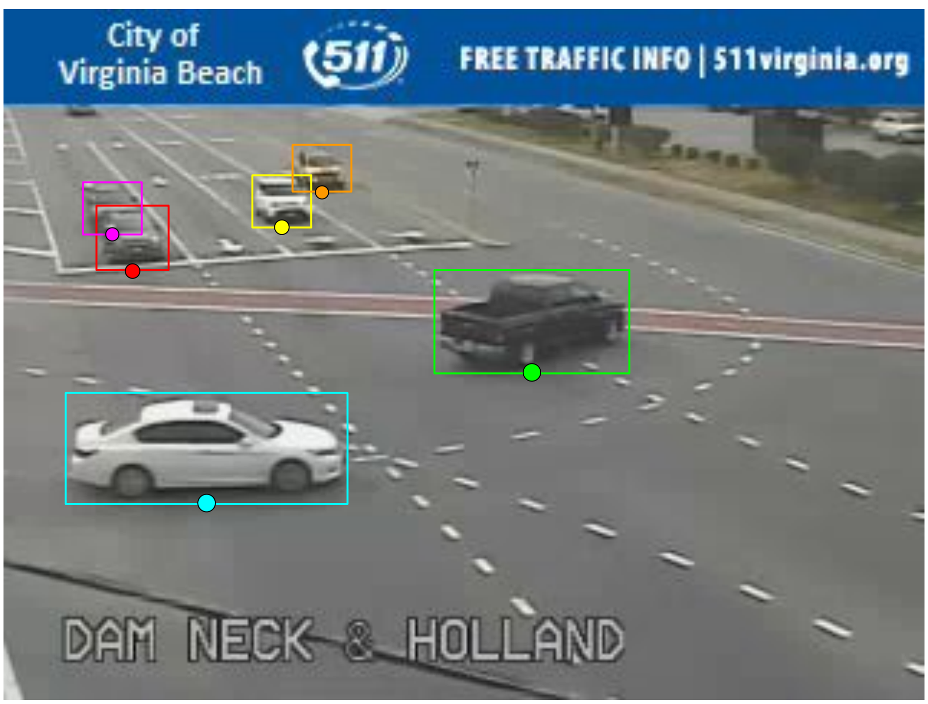

Figure 3.1: Sample frame from the VT-CAST 2020 dataset showcasing object detection using YOLOv7. Every traffic participant is enclosed within a bounding box, with their representative point denoted by the center of the bottom edge of each box.

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 difficult to construct a 3D bounding box with a ground plane. Therefore, we obtain an estimation of each participant's position by considering the midpoint of the lowermost edge of the bounding box (shown in Fig. 3.1). 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.

3.2.2 Pixel-to-GPS transformation

Converting the pixel coordinates to GPS is a process that involves transforming an image captured by a camera into geographic coordinates using GPS (Global Positioning System) data. This technique is widely used in the field of remote sensing and allows researchers to obtain geolocation information from satellite or aerial imagery. Figure 3.2 shows how GPS location is represented in the geographic coordinate system.

In this study, we utilize the video feed obtained from cameras strategically positioned at different intersections across Virginia by the Virginia Department of Transportation (VDOT). Our focus is on two intersections- one which has exhibited a comparatively high frequency of crashes, and the other which has had a low frequency of crashes. To ensure optimal visibility of the road structure and markings, we carefully select a frame that minimizes the presence of vehicles. In addition to the camera footage, we incorporate the Google Earth (nadir) view of the same area, leveraging the known locations of these cameras. By combining these two perspectives, we perform pixel-to-GPS transformation to establish geospatial references.

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

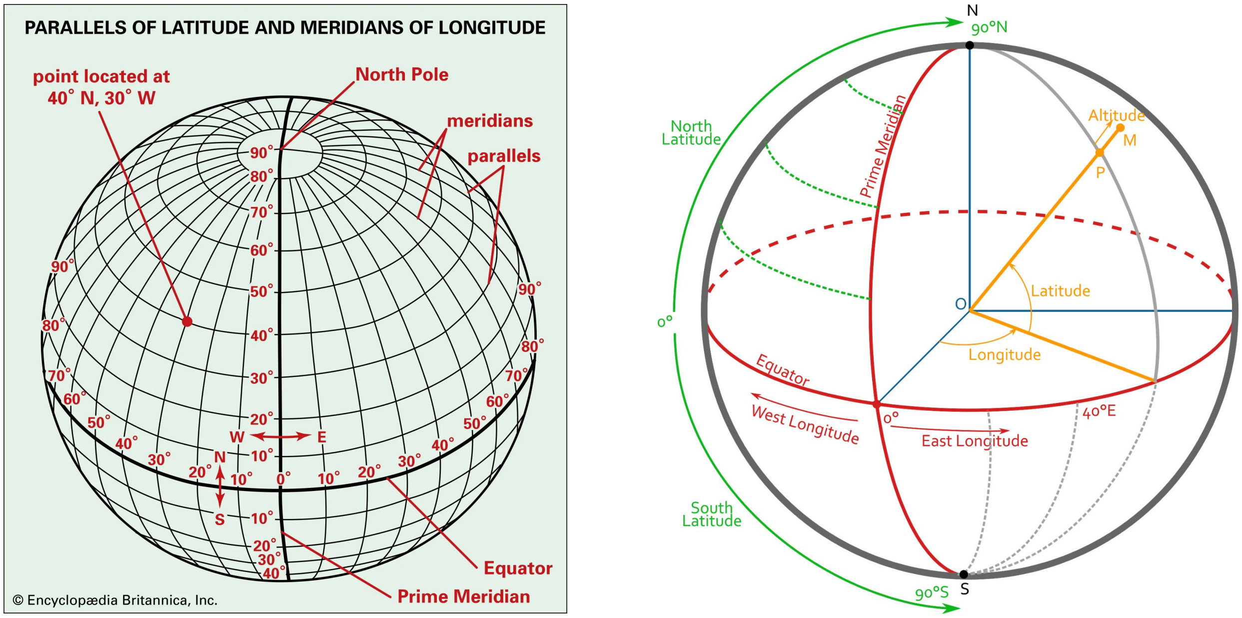

Figure 3.2: Exploring the geographic coordinate System: Latitude angles range from 0 to 90 degrees north and south of the equator, while longitude angles span from 0 to 180 degrees east and west of the prime meridian. Images credit: [10, 22]

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 map 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 homography matrix does not take height into account.

The accuracy of pixel-to-GPS perspective projection 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 projection can produce highly accurate geolocation information, making it an essential tool for many applications in remote sensing, including mapping and monitoring.

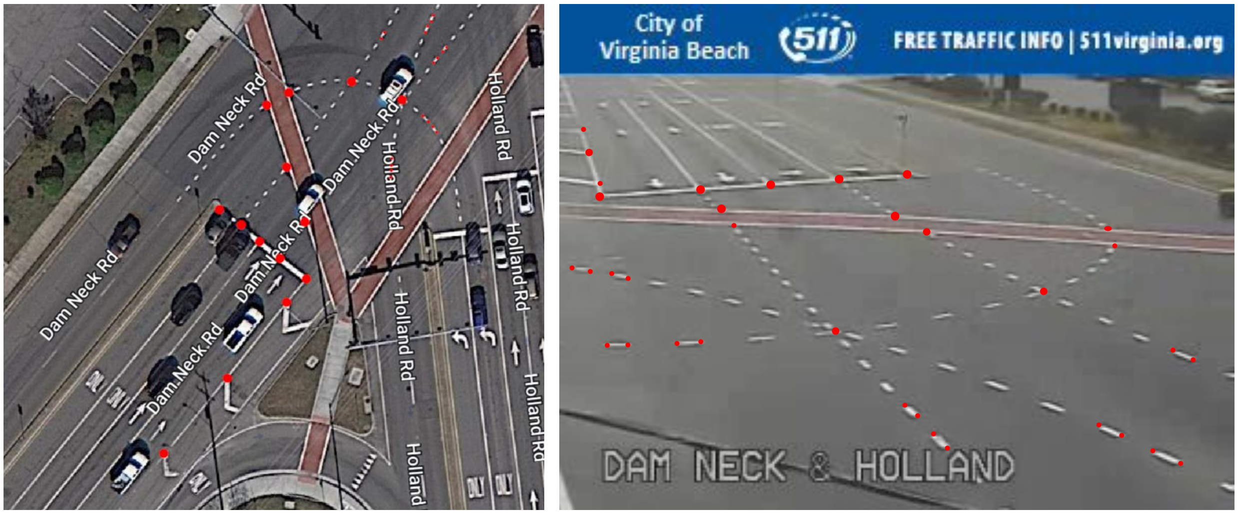

Figure 3.3: Matching points are chosen manually in Google Maps "bird-eye" view (left) and camera view (right) for homography computation.

To compute the homography matrix, we utilize the Random Sample Consensus (RANSAC) algorithm [26]. 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.

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. The below equation is used for transformation after the homography matrix has been obtained:

$$\begin{bmatrix} wx_d \\ wy_d \\ w \end{bmatrix} = H \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \tag{3.1}$$

Here, x_s and y_s are the x and y coordinates in the source (camera) coordinate system, and x_d and y_d are the x and y coordinates in the destination (geographic) coordinate system. The homography matrix is represented by H and is a 3×3 matrix, and w is a scaling factor.

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 apply a Savitzky–Golay filter [58] 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.

3.2.3 OpenStreetMap for road layout

OpenStreetMap (OSM) [34] is an inclusive and collaborative mapping initiative that strives to develop a broad, freely accessible map of the entire world. Its primary objective is to provide an open-source platform for individuals and organizations to contribute geospatial data, encompassing essential elements like roads, buildings, and points of interest. Contributors can utilize a variety of methods, including GPS devices, aerial imagery, and manual editing, to contribute their data to the project. In this work, we use the JOSM (Java OpenStreetMap Editor) software [39] for creating maps.

Lanelet2 [53] is an adaptable and all-encompassing library designed to handle and manipulate map data specifically in the context of autonomous driving scenarios. As an integral part of ROS2 (Robot Operating System 2), Lanelet2 focuses on processing high-definition

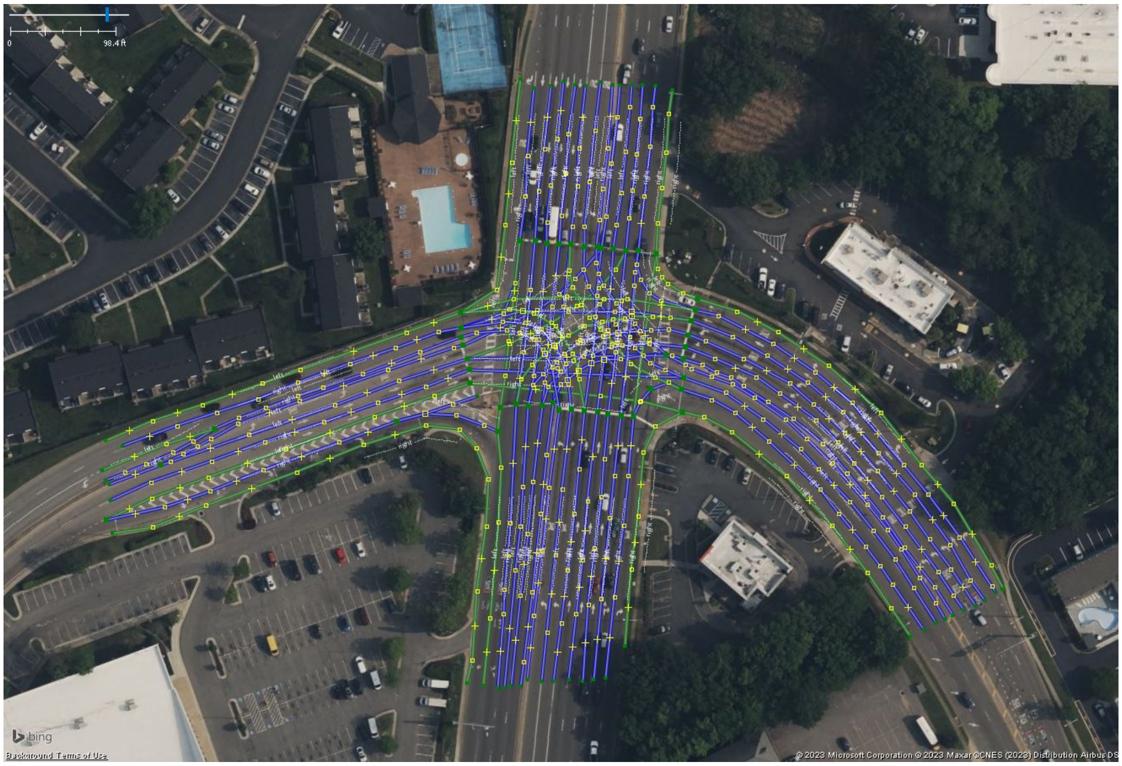

Figure 3.4: Mapping created manually in the OpenStreetMap format for one of the intersections in Virginia with a high history of crashes. Blue lines delineate the boundaries between roads and road surfaces, while green lines indicate the boundaries between non-road areas and road surfaces.

(HD) maps, taking into consideration not only the geometric attributes of the road but also semantic information such as lane regulations and traffic rules.

The core data structure in Lanelet2 is the "lanelet", which represents an individual drivable lane on the road. Fig. 3.4 shows the lanelet map created for an intersection in Virginia that has experienced a higher frequency of traffic accidents. The map has been overlayed on Bing Maps aerial imagery to better visualize the lanelets according to the actual road structure. 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 property 'virtual' means that the boundary is between two sets of lanes moving in the same or opposite direction. The property 'road boundary' 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 traffic rules associated with them. This comprehensive annotation approach enables the Lanelet2 model to provide a nuanced and thorough understanding of the road environment.

3.2.4 Semantic scene graph

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

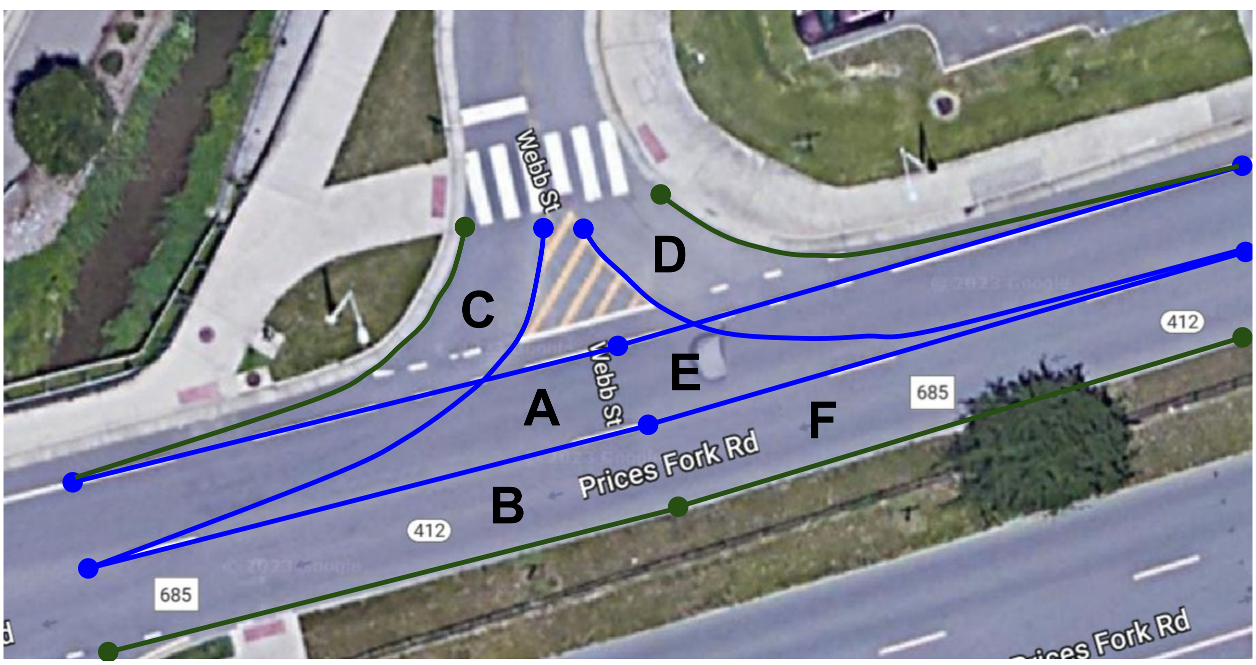

Figure 3.5: Google Earth view for a road intersection in Blacksburg, VA. Intersection labeled with the different road segments called “lanelets” in the OpenStreetMap format

A semantic scene graph, on the other hand, is a specialized type of scene graph utilized in computer vision to represent a 3D scene in terms of its semantics. In a semantic scene graph, 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, a semantic scene graph also captures functional and semantic relationships between objects. For instance, it might indicate that a ‘person’ node is “adjacent” to a ‘car’ node or that a ‘truck’ node is “behind” a ‘car’ node. This supplementary semantic information can be leveraged to support a broad array of applications, including object recognition, scene comprehension, and task planning.

Lanelet map and the road graph

In Fig. 3.5, we present the creation of a lanelet map by following the established protocol for generating standard drone-based trajectory datasets such as [9, 27, 72, 73]. This mapping has been overlayed over the Google Earth view for better visualization.

To construct the directed road graph depicted in Figure 3.6 automatically, we follow the methodology outlined by Zipfl et al. [76]. Each lanelet has been assigned a unique character for easy identification within the road graph.

The graph consists of three types of edges, distinguished by color. Consecutive edges indicate that two road segments / lanelets / nodes are positioned consecutively, with one following the other. These edges are unidirectional since traffic can only flow in one direction through two consecutive road segments. Adjacent edges 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 represent road segments that overlap or cross over each other, sharing a specific area of the road. These are also bidirectional in nature.

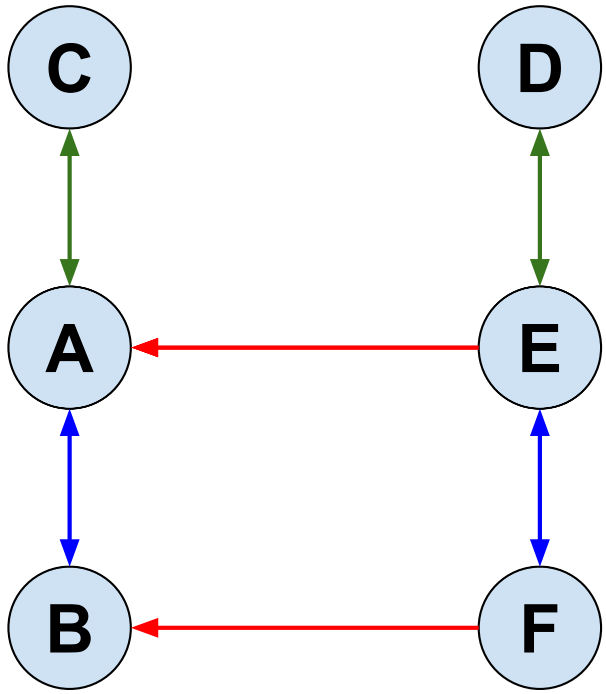

Figure 3.6: Road graph for the intersection shown in Fig. 3.5. Each node represents an individual road segment (lanelet). Different colored edges indicate edges representing various relations- Consecutive, adjacent and overlapping

Projection to Frenet space

Now that we have established the road layout, our next step is to represent the participants on the road. In Fig. 3.7, we additionally label the vehicles (alphabetically) to identify them in the creation of a semantic scene graph. We utilize "Frenet Coordinates" [5] (Fig. 3.8), which offer a more intuitive representation of a position on a road compared to the traditional Cartesian Coordinates. 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.

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

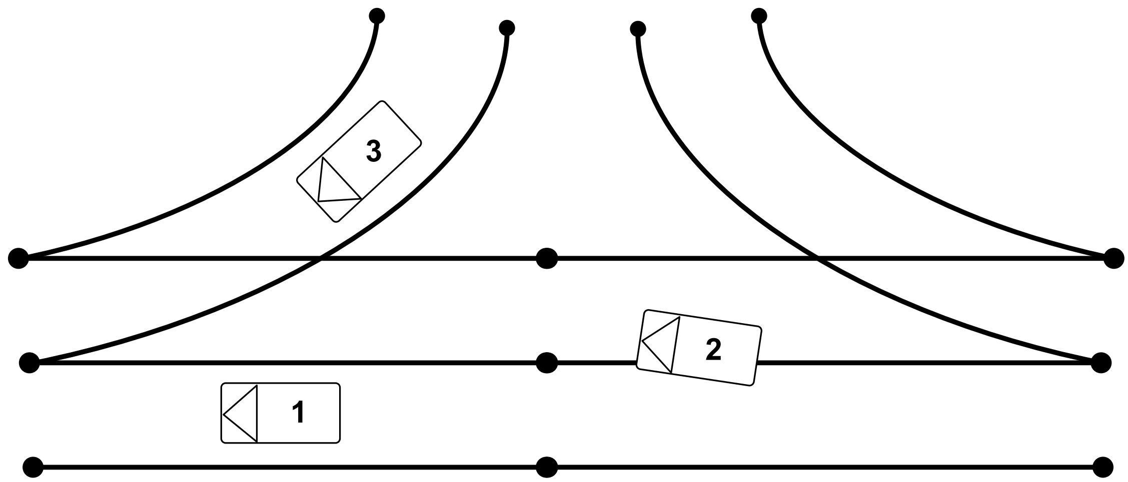

Figure 3.7: Example scenario demonstrating multiple vehicles along with their position and heading with respect to the road structure shown in Fig. 3.5

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.

By utilizing these concepts, we can represent the positions of vehicles on the road using Frenet Coordinates.

Spatial abstraction

The 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

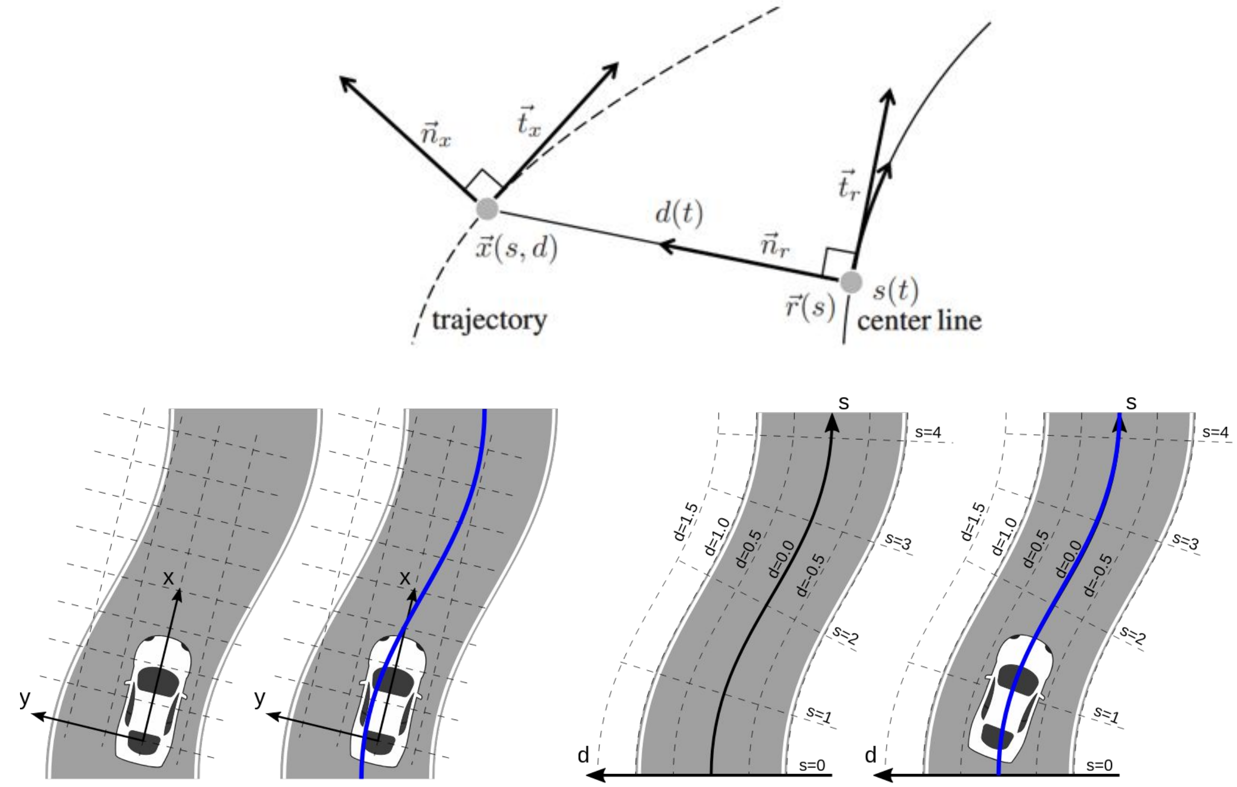

Figure 3.8: Transformation from Cartesian to Frenet coordinates. The figure on the bottom-left shows an example vehicle with the origin of cartesian coordinate frames aligned to it. The figure on the bottom-right shows the same vehicle in Frenet space with the centerline of the road as the 's' axis. Figure on the top shows the transformation from Cartesian to Frenet space. Images credit: [16, 28]

is changing lanes. This scenario is particularly common at intersections where multiple road segments meet and overlap. To accurately represent these projections, each participant's projection is aligned with the corresponding road segment's Frenet coordinates.

Semantic scene graph creation

In the proposed system, despite participants having multiple projection identities, each participant is consolidated into a single object 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

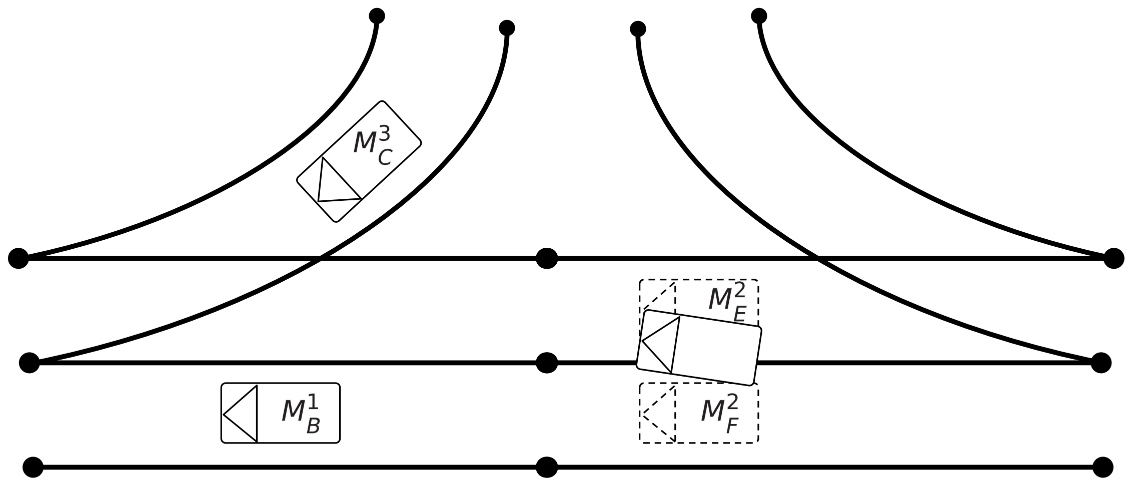

Figure 3.9: Projecting traffic participants onto their corresponding lanelets. Vehicle '2' has multiple projections as it is partially present in multiple lanelets

two projection identities m_a^i and m_b^i exist. If all edges in p_{ab} are labeled as "consecutive," it indicates a longitudinal relation between the two traffic participants i and j.

When vehicles travel in adjacent lanes, they are linked in the scene graph through a lateral relationship. This means that when a path p_{ab} exists between v_{road}^a and v_{road}^b 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 $\kappa_{rel} = lateral$.

Intersecting traffic participants i and j ($\kappa_{rel} = intersecting$) 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. Figure 3.10 illustrates the final semantic scene graph created based on the traffic scenario depicted in Figure 3.7. The corresponding graph's adjacency matrix representation is shown in Table 3.1. This method is carried out for the different intersections and various scenarios automatically through the Python implementation provided by Zipfl et al. [76].

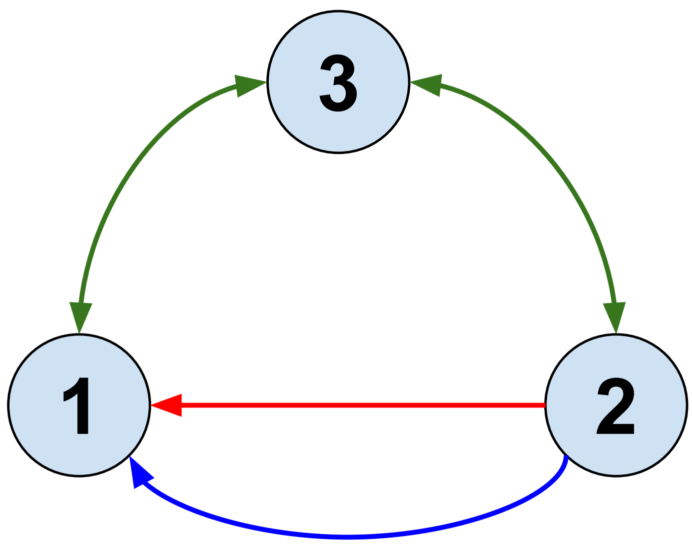

Figure 3.10: Semantic scene graph for the example scenario shown in Fig. 3.7. Each node represents an individual traffic participant from the scene. Different colored edges indicate edges representing various semantic relations- Longitudinal, lateral and intersecting

Table 3.1: Adjacency matrix representation for the graph structure in Fig. 3.10. The row at the top corresponds to the tail of the edge, i.e., the starting node for the directed edge. The column at the left corresponds to the head of the edge, i.e., the ending node for the directed edge. Each element in the matrix represents the edge embedding which would store the semantic and relational information. Detailed information regarding the edge embeddings are presented in Section 4.3.1

Head ↓ || Tail →	1	2	3
1	-	-	$e_{1\to3}$
2	$e_{2\to1}$	-	$e_{2\to3}$
3	$e_{3\to1}$	$e_{3\to2}$	-

Chapter 4

Safety Analysis

4.1 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 [42]. These properties can be applied to the entire network or specific nodes and edges to better understand the traffic flow and optimize transportation systems.

4.1.1 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 it. 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:

$$c_v = \frac{1}{\lambda} \sum_{u \in N(v)} c_u \Leftrightarrow \lambda c = Ac \tag{4.1}$$

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.

4.1.2 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, as well as 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:

$$\sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{log(deg(u))} \tag{4.2}$$

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:

$$S = (I - \beta A)^{-1} - I \tag{4.3}$$

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

4.1.3 Results

In order to assess the traffic density of a specific traffic scene, we employ a simplified graph representation. This approach is adopted due to the presence of multiple edges with distinct properties between two nodes in the semantic scene graph, where network theory concepts fail to differentiate between these various edge types. As a result, we employ a preliminary method that establishes 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) based on their Euclidean distance as shown here:

$$e_{uv} = \begin{cases} 1, & \text{if } dist(u, v) \leq margin \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

This graph construction method is further illustrated in Fig. 4.1.

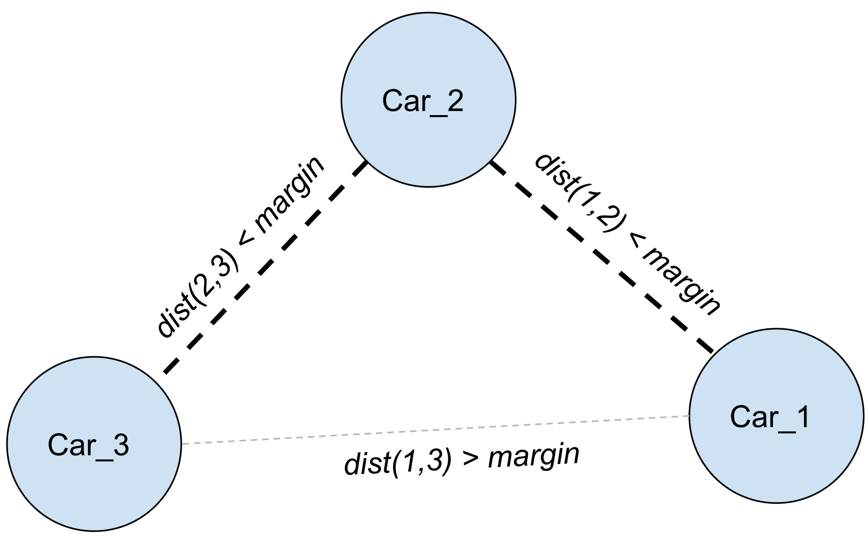

Figure 4.1: Euclidean distance based graph construction method: Edges exist between the pairs ‘Car_1’ and ‘Car_2’, and ‘Car_2’ and ‘Car_3’ as the Euclidean distance between them is less than the set margin. There is no edge between the nodes ‘Car_1’ and ‘Car_3’ as the distance margin is violated.

Tables 4.1 and 4.2 present the node and link features, respectively, for one of the unsafe 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 at the intersection. The margin was chosen as 10 meters for creating the graph. Based on the obtained graph, these features were calculated.

Table 4.1: 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	Closenness	
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	0.3293	0.40337

Based on the data presented in Table 4.1, 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 2-3 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 close 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. Upon analyzing the table, 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.

Betweenness centrality quantifies the extent to which a traffic participant facilitates connections among other participants. In the context at hand, most of the betweenness centrality values are in close proximity to zero. Bicycles and motorcycles exhibit the highest betweenness centrality, indicating their significant role in connecting other participants. Conversely, pedestrians exhibit the lowest betweenness centrality value, suggesting a comparatively lesser influence on facilitating connections among traffic participants.

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 4.2: 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

Node1	**Node2**	**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 4.2, 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.

4.2 Traffic safety metrics

Overspeeding or driving too slow

Engaging in driving practices that significantly exceed or fall below the designated speed limit poses inherent dangers [65]. To evaluate the severity of such deviations, we employ a simple method of calculating the variance between the prescribed speed limit and the actual speed of the vehicle. This straightforward calculation allows us to gauge the extent of the violation and its potential consequences on road conditions.

The roads leading to the selected Virginia Beach intersection have a prescribed speed limit of 45 mph (miles per hour). In order to identify instances of deliberate reckless driving and violation of the speed limit, we have established a speed threshold of 55 mph. Given that the chosen intersection is equipped with traffic signals, we do not focus on very slow vehicles. This is because vehicles approaching the intersection are expected to slow down and come to a stop in accordance with the signalization. Thus, our focus is on monitoring situations where vehicles exceed the speed limit by driving at speeds greater than 55 mph.

Accelerating or decelerating rapidly

Rapid acceleration presents a multitude of risks and hazards that demand recognition and mitigation [19]. 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 [57]. For acceleration, we have set the threshold at 0.6g, indicating the capability to reach a speed of 60 mph from a standstill in less than 5 seconds. This ensures a quick and efficient acceleration without compromising safety.

Similarly, for deceleration, we have set a threshold of 0.5g, 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 (TTC)

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 accidents 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 semantic scene graph. 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 an accident.

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 before a potential collision for appropriate actions to be taken. By defining this threshold, it becomes possible to prompt timely responses and implement necessary measures to avoid accidents.

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 work, we choose the threshold value for TTC as 2 seconds based on studies concerning safe following distances and speed [18]. It is to be noted that only considering TTC as the safety metric would lead to incorrect analysis, especially for intersections. We therefore, take into consideration both the vehicle speed and braking along with TTC to consider a situation as collision-prone (explained in Section 4.3.1).

In this study, we selected a threshold value of 2 seconds for Time-to-Collision (TTC) based on research on safe following distances and speed [18]. 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 (explained in Section 4.3.1).

Post-Encroachment Time (PET)

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 focuses on pairs of participants that exhibit either a lateral or intersecting relationship, as determined by the semantic scene graph. 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 (Post-Encroachment Time) metric, taking into account previous research on PET values for medium and high-risk scenarios [54]. 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.

4.3 Graph Neural Networks

Graph Neural Networks (GNNs) have gained widespread usage across various domains due to their ability to learn from data modeled as 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. However, the unique properties of graph data pose challenges for traditional neural networks, making them less suitable for this type of data. Some key limitations are:

- 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.

- 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.

- 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.

GNNs are built upon the fundamental concept of message-passing layers. While there exist various versions of GNNs, their differences primarily lie in the AGGREGATE and UPDATE functions employed within the message passing layer. These functions play a crucial role in information aggregation and updating representations throughout the graph network.

4.3.1 Building the data

In 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. On the contrary, 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. Consequently, we obtained a frame (timestamp) at approximately every 66.67 seconds, which enabled us to carry out a thorough examination of the intersections' traffic behavior and safety patterns.

Given our specific focus on intersections, we determined that underspeeding had limited relevance for our analysis. Instead, we concentrated on consolidating data related to overspeeding, rapid acceleration, rapid deceleration, as well as the number of traffic rule violations concerning Time-to-Collision (TTC) and Post-Encroachment Time (PET). These various factors were combined to assign a binary label, either "Risky" or "Non-risky", to each frame in the dataset. The below equation 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 "Non-risky".

$$c_f = \min(1, \text{floor}(0.5 \times n_{\text{PET}} + 0.25 \times n_{\text{TTC}} + 0.5 \times n_{\text{G}} + 0.5 \times n_{\text{S}})) \tag{4.5}$$

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_{S} 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 Time to Collision (TTC) metric, involving two vehicles, the situation is classified as "Risky" regardless of the status of other metrics. However, 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.

A graph is created to represent each frame of both the training and test dataset splits. To create the Semantic Scene Graph structure, we adopt the method as proposed by [76] and described in Chapter 3. Once the structure is established, we proceed to assign node and edge features. In this graph, each traffic participant corresponds to a node, and we represent 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, as well as the length and width of the participant. The below equation provides a clear representation of this feature vector.

$$e_{i \to j} = [isPed, isBike, isCar, isTruckBus, |v|, v_x, v_y, length, width] \tag{4.6}$$

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. 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. The below equation provides a clear representation of this feature vector.

$$u_i = [isLon, |d_{lon}|, isLat, |d_{lat}|, isInt, |d_{int}|] \tag{4.7}$$

4.3.2 Model selection and evaluation

Selecting the appropriate Graph Neural Network (GNN) capable of integrating node and edge embedding information is of utmost significance. In our research, we relied upon the GNN cheatsheet offered by PyG [25] as a valuable reference for selecting the different models.

TransformerConv

TransformerConv is a novel attention-based model that was proposed in [61]. 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) [36] 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) [71], 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. Firstly, 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.

Results

A hyperparameter search was conducted for both selected models, and the best results obtained with the optimal hyperparameters are summarized in Table 4.3. All models were trained for 100 epochs on a Linux machine with a 16 GB Nvidia V100 GPU.

Table 4.3: Comparison of performance on the VT-CAST 2020 dataset against two state-of-the-art GNN-based methods

	TransformerConv [61]	**GINEConv** [36]
Accuracy	72.9%	**79.85%**

An important hyperparameter to consider when optimizing a Graph Neural Network (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. However, if the number of layers becomes excessively high, the network may suffer from oversmoothing, where it aggregates information from all nodes in the network. Conversely, having too few layers prevents nodes from capturing deeper interactions effectively.

In this work, we obtained the optimal number of layers as 3 for the GINEConv model, and 2 for the TransformerConv model. This represents the optimal value at which peak performance was achieved. Deviating from this chosen number of layers, either by increasing or decreasing it, would result in a decline in performance.

Although attention mechanisms enable the model to selectively prioritize information from node and edge embeddings, it has been observed that convolutional approaches yield significantly superior performance. This discrepancy can be attributed to the relatively low dimensionality of the input data (node and feature embeddings).

Table 4.4: 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

Predicted ↓ || Ground-truth →	Risky	Non-risky
Risky	0.804	0.231
Non-risky	0.196	0.769

The confusion matrix presented in Table 4.4 provides valuable insights into the distribution of predictions. To ensure a fair comparison, the elements in the confusion matrix have been normalized by dividing them by the sum of the predictions in the corresponding column. This normalization is based on the total number of videos with ground truth matching the column label.

Given that each video contains approximately 40,000 frames, the inference results encompass the analysis of around 80,000 frames (two videos, one classified as safe and the other as unsafe, based on statistical information). The numbers within the confusion matrix reveal a notable presence of false positive and false negative classifications. Such occurrences may be attributed, in part, to the dataset's inherent noise, stemming from resolution issues and accumulating through various stages of object detection, homography transformation, and safety metric estimation.

To enhance the accuracy of the model's predictions, a potential solution could involve manually annotating risky scenarios rather than relying solely on numerical and temporal tracking-based methods. By incorporating this approach, we may significantly improve the model's performance in this specific scenario.

Chapter 5

Driver Distraction Monitoring

While implementing risk analysis for transportation systems is crucial, understanding the underlying causes of risky behavior in vehicles can pose challenges. One possible explanation for such behavior is that the vehicle in question may be an emergency vehicle urgently trying to reach its destination, which could account for overspeeding and breaking traffic rules. Another possibility is that the vehicle is experiencing mechanical failures or technical issues that are leading to dangerous driving. Additionally, reckless driving may be attributed to the driver's thrill-seeking tendencies or even involvement in criminal activities. Incidents involving reckless driving and component failure often result in damage, injuries, and even loss of life for law-abiding drivers. Therefore, it becomes imperative to alert other participants in such scenarios about potential collisions that could be prevented. In such cases, our objective is to analyze the situation and provide warnings to all parties involved a few seconds before an imminent unwanted event occurs.

The system proposed in Chapter 4 focused on assessing the risk level of various scenarios, particularly in cases where the cause is unknown. One specific type of accident that has witnessed a surge in recent times is those resulting from distracted driving. It is important to note that these drivers do not intentionally flout rules or seek to cause accidents. We strongly believe that by addressing the issue of distracted driving at its core, we can effectively prevent such accidents.

To accomplish this objective, we propose a highly efficient approach that entails the use of a camera system to monitor the driver's behavior. This system is designed to promptly issue appropriate alerts to the driver whenever necessary, thereby proactively mitigating the risks associated with distracted driving. In this chapter, we introduce a methodology for estimating secondary driver activity that distracts the driver from the primary activity of driving. We utilize a low-resolution monocular video feed captured by a camera positioned to face the driver.

5.1 Approach

In this section, we present a detailed description of our advanced action recognition model for driver distraction detection, which is built upon a transformer neural network framework. This model showcases exceptional performance in identifying and classifying various driver distractions.

The Video Swin Transformer [44] is a transformer neural network model, specifically designed for handling video recognition tasks like action recognition. Building upon the success of the Swin Transformer [43], which initially focused on spatial image understanding tasks such as object detection and instance segmentation, the Video Swin Transformer extends its capabilities to handle temporal information as well. By incorporating temporal understanding into the original Swin Transformer architecture, the Video Swin Transformer enables the effective analysis of video sequences. This powerful spatiotemporal model can efficiently process and recognize actions within videos. It leverages the strength of the Swin Transformer, which is known for its ability to capture long-range dependencies and contextual information in images, and extends it to video domains.

There are several variants of the Video Swin Transformer based on the depth and size of the network. In this paper, we use the Swin-B (Base) version as it has a marginally lower

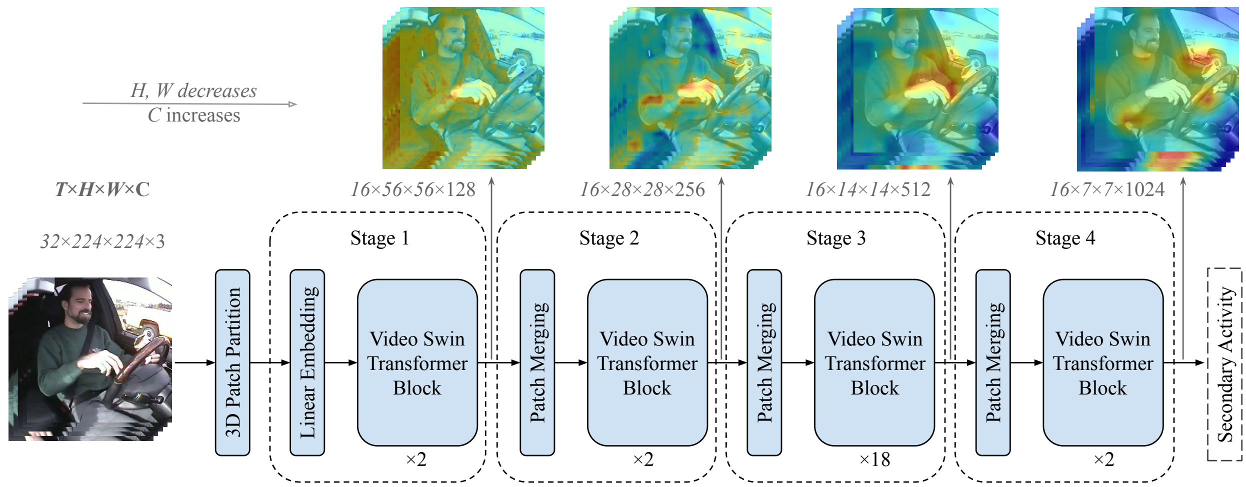

Figure 5.1: Overall architecture of the Video Swin-B Transformer. Here, T denotes the temporal dimension, which represents the sequence of frames in a clip. The dimensions H, W, and C refer to the height, width, and number of channels of each frame in the clip, respectively. Attention maps from the intermediate layers are overlayed over the original frames for better understanding and visualization. The early layers in a network are seen to identify textual and low-level features, whereas the deeper layers in a network learn high-level object and part-based features. The final layer discards spatial information and acts as a learned global pooling operation.

accuracy than Swin-L (Large) on the Kinetics-400 [40] action recognition dataset, while having substantially less complexity. A preview of the architecture is shown in Fig. 5.1. The 3D patch partitioning layer splits the input 3D tensor into 16×56×56 3D tokens, each comprising a 96-dimensional feature. These features are then projected to 128 dimensions using a linear embedding layer. The architecture is made up of four stages, each containing one or more pairs of consecutive Video Swin Transformer Blocks. The first block in each set is made up of a 3D window-based multi-head self-attention (3D W-MSA) module, followed by a feed-forward network (FFN). Before each MSA module and FFN, layer normalization (LN) is applied, and a residual connection is applied after each module. The second block in each set takes as input the output from the residual connection following the FFN layer of the first block. The architecture of the second block is very similar to that of the first block, except that the 3D W-MSA module is replaced by a 3D shifted window-based multi-head self-attention (3D SW-MSA) module.

The model pre-trained on Kinetics-400 was used as a starting point. The classifier layer (i.e. the layer after the Stage-4 module in Fig. 5.1) was replaced by a new layer with randomized weights and the number of output nodes was set equal to the number of classes (S). All the layers were set as trainable, and the entire model was trained on our dataset (Section 5.2.1). During training, 32 frames from the input clip are sampled at a time with a stride of 2, and then resized to 224×224. All the hyperparameters are kept the same as for the when the model was trained on the Kinetics-400 dataset.

During inference, we follow the procedure from Liu et al. [44] by using 4×3 views. A video is uniformly sampled in the temporal dimension as 4 clips, and for each clip, the shorter spatial side is scaled to 224 pixels. Three crops of size 224×224 are taken that cover the longer spatial axis. Taking multiple crops from the input image also allows for higher spatial resolution. The final score is computed as the average score over all the views. For the purpose of visualization, the attention maps are extracted from various intermediate layers of the network. Different layers and channels provide varied information for a set of frames, and they collectively determine the class of a video clip.

5.2 Experiments

5.2.1 Datasets

This section introduces the details of two datasets that we used to demonstrate our results. The VTTIMLP01 dataset is an RGB dataset with 640×480 resolution that was collected in a testing facility. The SHRP2 dataset is comparatively more challenging in nature. The dataset contains videos sampled from more than 3000 vehicles with unconstrained camera extrinsics. The video frames from the face view camera are grayscale in nature and with a resolution of 320×240 which can be considered low-quality.

VTTIMLP01 (VTTI Machine Learning Pose) [41]

This dataset consists of videos of 25 participants performing eight activities on the VTTI "smart road", which is a restricted experimental facility. Each driver was asked to perform six secondary behaviors, and two baseline driving behaviors in static and dynamic experimental conditions. The secondary behaviors were eating, drinking, talking on a cell phone, texting on a cell phone, changing the radio, and adjusting the sun-visor. The static scenario signifies that the car was stationary when the participants were performing the eight tasks. We also added two classes 'nothing' and 'reaching'. The class 'nothing' corresponds to not being involved in any secondary activity. The class 'reaching' refers to the action performed by the driver when they extend their hand or arm in order to obtain an object. This particular action often precedes engaging in additional activities commonly associated with "eating" or "drinking". The baseline driving behavior included driving with one hand and driving with two hands. During the driving scenario, the driver was driving on the VTTI "smart road" while performing the tasks. An in-vehicle experimenter instructed the driver to perform the tasks. During the tasks, the drivers were not restricted to performing any tertiary tasks including gazing outside, talking to the passenger, or pointing at things at the outside scene. The collection was repeated for two different vehicles. A driver-facing camera was used to capture the behavior of the driver as shown in Fig. 5.1.

We annotated this dataset following the protocol used by standard action recognition datasets such as Kinetics-400 [40] and Something-Something V2 [33]. Each video is manually annotated to define the start and the end time of each secondary behavior. For this dataset, we segmented the full videos into smaller segments of 6 seconds and 10 seconds. We found that creating a dataset with the length of each video clip as 6 seconds leads to better performance than when each clip's length is 10 seconds (demonstrated in the next subsection).

The training set consists of actions performed by 23 drivers, whereas the testing is composed of actions by the remaining 2 drivers. The classes for the secondary behavior in this dataset along with the distribution are shown in Table 5.1.

Table 5.1: Number of clips for each activity label in the Training and Testing dataset split for the VTTIMLP01 and SHRP2 datasets

Class	VTTIMLP01		SHRP2	
	Training	**Testing**	**Training**	**Testing**
No secondary action	734	36	600	1400
Reaching for something	98	7	521	60
Cellphone talking	424	34	476	53
Sun visor adjusting	393	26	-	-
Center stack adjusting	387	27	547	63
Eating	736	75	249	29
Drinking	727	52	123	16
Cellphone texting/using	455	34	600	61
Personal hygiene	-	-	468	54
Smoking	-	-	163	19
Interaction	-	-	1320	2356
Total	*3954*	*291*	*4994*	*4111*

SHRP2 (Strategic Highway Research Program 2) [20]

This dataset contains data collected from real-world drivers and vehicles. It is considered the largest naturalistic dataset available for public use. The diversity of this data is much larger than the VTTIMLP01 dataset. It is a low-quality dataset with videos of drivers during both day and night time. The SHRP2 dataset also includes manual annotations of nearly 42000 event epochs from crash, near crash, and baseline events. In this work, we specifically looked at the secondary behavior based annotations. For each epoch, the manual annotator identifies the engagement of the driver from a total of 56 secondary behaviors [55]. Multiple activities are also annotated with their start and end times. One additional challenge for this dataset is that the duration of each of the annotations and the durations of actions are not equal. They generally vary between less than one second to 6 seconds. We chose segments between 2 seconds and 6 seconds. Also, this dataset had high class imbalance. We therefore combined several fine-grained activity labels into 10 labels following the recommendations by Sarkar [55]. The classes for the secondary behavior in this dataset along with the distribution are shown in Table 5.1.

Upon analyzing the table, it becomes evident that the training dataset of SHRP2 still exhibits a significant class imbalance. Specifically, the 'Interaction' class contains a large number of samples, while the 'Eating,' 'Drinking,' and 'Smoking' classes are severely underrepresented with very few samples. To address this issue, a weighted loss function is employed during the training process. This loss function aims to mitigate misclassifications by assigning higher class weights to the minority classes and lower class weights to the majority classes. The formulation of this approach is defined as follows:

$$w^l = \frac{n_{samples}}{n_{classes} \times n^l_{samples}} \tag{5.1}$$

Here, w^l is the weight assigned to the output node corresponding to class l. $n_{classes}$ is the number of classes in the dataset, and $n^l_{samples}$ is the number of samples in class l. These weights are multiplied by their respective nodes' softmax activation before computing the classification loss.

5.2.2 Results

In this section, we compare the performance of the Video Swin Transformer against state-of-the-art CNNs for action recognition. The Top-1 and Top-3 accuracy for all the models on SHRP2 and VTTIMLP01 datasets are summarized in Table 5.2. Since the number of classes is at most 10 in either dataset, we chose to use the Top-3 accuracy metric instead of the commonly used Top-5 accuracy. It can be seen that Video Swin has a Top-1 performance that is better or comparable to state-of-the-art CNNs, while the Top-3 accuracy is better in all cases.

Table 5.2: Comparison of performance on VTTIMLP01 and SHRP2 datasets against state-of-the-art methods

Method	**Accuracy on VTTIMLP01**		**Accuracy on SHRP2**	
	Top-1	**Top-3**	**Top-1**	**Top-3**
SlowFast [24]	71.48%	87.97%	59.86%	83.36%
TSN [70]	76.19%	88.54%	53.54%	77.48%
TAM [45]	**80.41%**	90.72%	68.38%	90.85%
Video Swin [44]	80.20%	**92.78%**	**71.49%**	**94.65%**

The confusion matrices for all classes on both SHRP2 and VTTIMLP01 datasets are shown in Fig. 5.2. The class label 'Nothing' corresponds to when no secondary activity is being performed and the driver is solely focused on driving. The elements in the confusion matrix have been divided by the sum of the predictions in the corresponding row (i.e, the total number of videos with ground-truth as the row label). This normalization greatly facilitates the comparison of classwise performance.

After analyzing both datasets, it is evident that there is significant confusion between the classes 'Reaching', 'Smoking', 'Centerstack adjusting', 'Eating', and 'Drinking' with the class 'Nothing'. These particular classes may only involve activity for a few frames, while the driver predominantly appears to be engaged in driving for the majority of the video frames.

In the SHRP2 dataset, some confusion arises between the classes 'Nothing' and 'Interaction'. 'Interaction' refers to instances where the driver is merely moving their mouth to speak with passengers while maintaining focus on driving. Since this activity does not involve any external objects, it becomes challenging to distinguish it from other activities. The absence or presence of an object significantly aids in identifying most of the other activities.

The classes 'Cellphone talking' and 'Cellphone texting/using' stand out as highly distinguishable in both datasets. They involve the presence of an external object along with noticeable changes in the hand pose and possibly the head pose.

It is worth noting that the 'Reaching' class in the VTTIMLP01 dataset experiences a high misclassification rate. This is partly because no explicit recordings were made for this class, unlike in the SHRP2 dataset. Consequently, there is considerable confusion between 'Reaching' and 'Eating' since reaching for something is typically associated with the act of eating. Additionally, there is some misclassification between 'Reaching' and 'Nothing' due to the dataset labeling, which indicates that any action starts after the driver has finished reaching for the object and acquired it.

Comparitive analysis of video length

Next, we perform a study to see how the duration of the input sequence affects the performance of both CNN and Transformer based methods. Table 5.3 shows the comparison using different clip lengths for VTTIMLP01.

The dataset videos are created by trimming the video such that the activity is centered in the clip. We see that creating either dataset with the length of each video as 6 seconds leads to better performance than when the length is 10 seconds. Both of the secondary activity datasets (SHRP2 and VTTIMLP01) have a few hundred videos for each label which

Figure 5.2: Confusion matrices for the SHRP2 (left) and VTTIMLP01 (right) datasets for all classes- the ground truth labels are on the vertical axis, and the predicted labels are on the horizontal axis. (Labels on the respective horizontal and vertical axes match.)

is much lower than what more common action recognition datasets have. Moreover, there is an imbalance in the number of videos for different classes. During training and testing, long videos are processed by splitting each video into multiple fixed-sized clips by the model. The predictions from the clips are then combined to produce a prediction for the video. Reducing the video size to 6 seconds increases the number of distinct videos by almost double, thus leading to better training. We cannot reduce the dataset length any further as then each clip would be really short for the network to learn any useful information from the dataset.

Table 5.3: Performance comparison on using 6-second clips vs. 10-second clips on VT-TIMLP01

Method	**VTTIMLP01 (10 sec)**		**VTTIMLP01 (6 sec)**	
	Top-1	**Top-3**	**Top-1**	**Top-3**
SlowFast [24]	61.90%	82.99%	71.48%	87.97%
TSN [70]	71.14%	86.39%	76.19%	88.54%
TAM [45]	72.79%	89.52%	**80.41%**	90.72%
Video Swin [44]	74.91%	90.56%	80.20%	**92.78%**

5.2.3 Explainable action recognition

Graph Neural Networks (GNNs) excel at analyzing graph structures by leveraging aggregation and message-passing mechanisms among neighboring nodes. This allows for a clear understanding of how GNNs perform graph analysis. However, the same level of clarity cannot be attributed to Transformers when it comes to extracting information from video sequences. It is not immediately apparent what specific elements the neural network focuses on to extract relevant information for action recognition. In this section, we aim to shed light on the attention layers of transformer neural networks within the context of secondary action recognition.

Figure 5.3: Sequentially from left to right- The figure shows the shifting of attention from the mouth to the hand as the driver moves one of his hands to pick up the bag of chips

Vision transformers preserve spatial context in most of the layers up to the final layers. They capture the texture and low-level information similar to CNNs [31]. As we go deeper into the network, the Transformer network is able to collate information to understand high-level information. Finally, the last layer captures global information as it proceeds to the classification layer. This is considered to be analogous to the global pooling operation in CNNs. Attention maps at various steps of the network can be seen in Fig. 5.1. The attention is shown in a heatmap scale where red color indicates the highest attention, and blue color indicates the least attention.

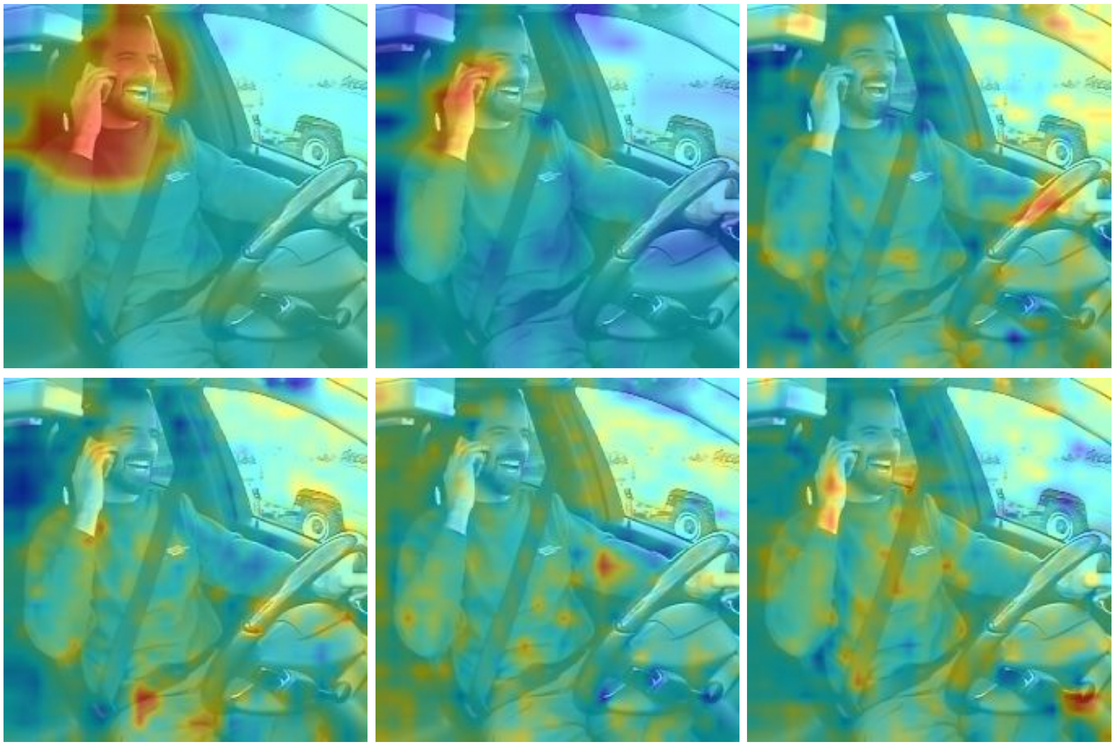

Figure 5.4: Variation in attention across various channels from the same layer of the network for class 'Cellphone talking'. The top row shows the network focusing on the hand, cellphone (object), face, and steering wheel. Bottom row shows attention towards various joints (wrist, fingers, elbow, hip, and knee) of the driver that help in determining the pose.

Following the ideas of [31], we only visualize the attention maps after Stage 2 and Stage 3. In Fig. 5.3 we can see the variation of the attention with time for one channel of the network. Initially, the model has some attention to the mouth and hand area and classifies the activity as 'eating' as there is motion in the mouth. But with time, the driver moves his hands to reach for the packet of chips and the model now classifies the activity as 'reaching' as there is also no mouth movement. There is also some attention on a part of the steering wheel which shows that the model is trying to determine if the driver's hand is on or off the wheel.

Fig. 5.4 shows the attention for various channels for the same frame of a video clip with the class label 'Cellphone talking'. The first image in the top row has attention spread around the face and hand area. The network is able to understand that there is movement in the mouth and perhaps the head, while the hand is very close to the face. The second attention map particularly focuses on the hand and phone indicating the presence of an external object, and that one of the hands is used for this activity. The third image further adds to it by indicating an area of the steering wheel where the hand should have been present. The second and third attention maps would thus most likely contribute to determining that one hand is off the steering wheel. The bottom row focuses more on various body joints which are crucial in determining body pose. The first image in the bottom row focuses on one of the hands (elbow joint, wrist joint, palm) and the limb-hip joint. The second image shows spread-out attention over the torso area with high attention on the keypoints of the other hand. Finally, the last image shows strong attention over the torso, shoulders, and the hand holding the cellphone. The upper body pose is crucial in determining if the body pose is off normal. Although only the upper limbs are visible in the body, the lower body would hardly change pose to a large extent.

Chapter 6

Conclusion and Future Directions

The development of road infrastructure on a global scale has led to a significant increase in the number of accidents, which has a detrimental impact on the safety of cyclists and pedestrians. Unfortunately, they are the ones who bear the brunt of this situation, often lacking adequate safety measures.

One major contributing factor to these accidents is driver distraction, which has emerged as the primary cause of roadway incidents and fatalities in recent years. This trend has raised concerns about the evolving nature of distraction in advanced automobiles. As automation and driving assistance systems continue to advance, it is crucial for the research community to refocus its efforts on understanding driver attention.

This work aims to address the issue of secondary behaviors and their complexity. Our initial focus was on evaluating the maturity level of current computer vision methods in automatically identifying various secondary behaviors. We compared the performance of a transformer-based architecture with state-of-the-art convolutional neural networks (CNNs) for secondary driver activity recognition, utilizing two challenging naturalistic driving datasets. The results of our study clearly demonstrate the effectiveness of our approach, leading to the development of multiple high-performing secondary action recognition models.

It is important to note that different secondary behaviors have varying impacts on drivers' attention. To design an effective distraction mitigation system and accurately determine the driver's takeover time in safety-critical situations within highly automated vehicles, it is necessary to understand each of these secondary behaviors individually. This work is groundbreaking in that it introduces explainability in driver action recognition tasks and utilizes tools to highlight the complexity of secondary behaviors.

Furthermore, we have shown that traffic scenarios can be represented in a simpler manner using graphs while retaining the relevant information. By employing a distance-based edge construction method, we can analyze traffic density and flow more effectively. Semantic scene graphs enable us to leverage road structure information and the relative positions of vehicles to one another within the scene.

In summary, this work addresses the adverse effects of increased accidents resulting from road infrastructure development. It emphasizes the importance of understanding driver distraction and secondary behaviors, presenting a successful approach to secondary action recognition. Additionally, it introduces the concept of using graphs and semantic scene graphs for analyzing traffic scenarios, providing valuable insights into traffic density and flow patterns.

There is significant potential for future research in this field, particularly in the area of traffic analysis and surveillance. To further explore this potential, conducting analyses of multiple intersections at various times of the day and under different weather conditions is recommended. Such a comprehensive examination of the variations in traffic density would provide valuable insights into the dynamics of traffic flow.

In order to enhance both the analysis capabilities and safety during surveillance for violators, it is advisable for the Virginia Department of Transportation (VDOT) to consider installing high-quality cameras at intersections. Unclear footage often proves insufficient as evidence when apprehending suspects, thus improving camera quality would greatly contribute to the effectiveness of monitoring efforts. High-resolution cameras can capture clearer and more detailed images, improving the accuracy and reliability of the evidence collected.

Additionally, the creation of mapping systems for multiple intersections would be beneficial. By mapping and creating graph representations of the different intersection videos, researchers can gain a better understanding of the complex interactions that occur. Manual annotation of the scenes can provide more robust data, allowing for a deeper analysis of the interactions that may not be fully captured by pre-defined safety metrics. This approach would enable researchers to uncover nuanced patterns and identify potential areas for improvement in traffic management and safety measures.

Overall, by conducting thorough analyses of traffic patterns, implementing high-quality surveillance cameras, and leveraging manual annotation techniques, future research in this field can yield valuable insights that contribute to improved traffic management and safety.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint:1609.08675*, 2016.

[2] Ibrahim Abubakar, Taavi Tillmann, and Amitava Banerjee. Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the global burden of disease study 2013. *Lancet*, 385(9963):117–171, 2015.

[3] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint:2206.14651*, 2022.

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[5] G Arreaga, R Capovilla, and J Guven. Frenet–serret dynamics. *Classical and Quantum Gravity*, 18(23):5065, 2001.

[6] Max G. Bareiss. *A Dataset of Vehicle and Pedestrian Trajectories from Normal Driving and Crash Events in One Year of Virginia Traffic Camera Data*. PhD thesis, Biomedical Engineering, Virginia Tech, 2023.

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint:2102.05095*, 2021.

[8] United States. National Transportation Safety Board. *Special Investigation Report: Vehicle-and Infrastructure-based Technology for the Prevention of Rear-end Collisions*, volume 1. 2001.

[9] 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 *IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934, 2020.

[10] The Editors of Encyclopaedia Britannica. latitude and longitude, March 2023. URL https://www.britannica.com/science/latitude.

[11] 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 *IEEE Intelligent Vehicles Symposium (IV)*, pages 1471–1476, 2017.

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

[13] CDC (Centers for Disease Control and Prevention). Road traffic injuries and deaths—a global problem, March 2023. URL https://www.cdc.gov/injury/features/global-road-safety/index.html.

[14] Centers for Disease Control and Prevention. Distracted driving, March 2023. URL https://www.cdc.gov/transportationsafety/distracted_driving/index.html.

[15] 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.

[16] Chen Peng. Path planning in frenet frame, May 2023. URL https://caseypen.github.io/posts/2021/01/FrenetFrame/.

[17] George Christodoulou, Euripides GM Petrakis, and Sotirios Batsakis. Qualitative spatial reasoning using topological and directional information in owl. In *IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 596–602, 2012.

[18] Erez Dagan, Ofer Mano, Gideon P Stein, and Amnon Shashua. Forward collision warning with a single camera. In *IEEE Intelligent Vehicles Symposium*, pages 37–42, 2004.

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

[20] Thomas A. Dingus, Jonathan M Hankey, Jonathan F Antin, Suzanne E Lee, Lisa Eichelberger, Kelly E Stulce, Doug McGraw, Miguel Perez, and Loren Stowe. *Naturalistic driving study: Technical coordination and quality control*. Number SHRP 2 Report S2-S06-RW-1. 2015.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[22] Unreal Engine. Georeferencing a level, March 2023. URL https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Georeferencing/.

[23] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[24] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[25] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint:1903.02428*, 2019.

[26] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[27] Tobias Fleck, Karam Daaboul, Michael Weber, Philip Sch”orner, Marek Wehmer, Jens Doll, Stefan Orf, Nico Sußmann, Christian Hubschneider, Marc Ren’e Zofka, Florian Kuhnt, Ralf Kohlhaas, Ingmar Baumgart, Raoul Z”ollner, and J. Marius Z”ollner. Towards large scale urban traffic reference data: Smart infrastructure in the test area autonomous driving baden-w”urttemberg. In *Intelligent Autonomous Systems 15 - Proceedings of the 15th International Conference*, pages 964–982, 2018.

[28] Franz Pucher. Trajectory planning in the frenet space, May 2023. URL https://fjp.at/posts/optimal-frenet/.

[29] 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.

[30] Geneva: World Health Organization. Global status report on road safety 2018, April 2023. URL https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/.

[31] Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. What do vision transformers learn? a visual exploration. *arXiv preprint:2212.06727*, 2022.

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

[33] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

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

[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] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande,

and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint:1905.12265*, 2019.

[37] Yujun Huang, Yunpeng Weng, Shuai Yu, and Xu Chen. Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. In *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, 2019.

[38] 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.

[39] JOSM developers. JOSM (Java OpenStreetMap Editor), April 2023. URL https://josm.openstreetmap.de/.

[40] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint:1705.06950*, 2017.

[41] Ujwal Krothapalli, Loren Stowe, Zachary R Doerzaph, and Andy Petersen. Application of proximity sensors to in-vehicle data acquisition systems. 2018.

[42] Jure Leskovec. Stanford cs224w: Machine learning with graphs, Winter 2022/23. URL http://cs224w.stanford.edu.

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows.

In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[44] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[45] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. TAM: Temporal adaptive module for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13708–13718, 2021.

[46] 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.

[47] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&Act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2801–2810, 2019.

[48] Xiaoyu Mo, Yang Xing, and Chen Lv. Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction. *arXiv preprint:2106.07161*, 2021.

[49] Anna Montoya, Dan Holman, Taylor Smith, and Wendy Kan. State Farm Distracted Driver Detection, 2016. URL https://kaggle.com/competitions/state-farm-distracted-driver-detection.

[50] 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.

[51] Osama A Osman, Mustafa Hajij, Sogand Karbalaieali, and Sherif Ishak. A hierarchical machine learning classification approach for secondary task identification from observed driving behavior data. *Accident Analysis & Prevention*, 123:274–281, 2019.

[52] Ioannis Papakis, Abhijit Sarkar, Andrei Svetovidov, Jeffrey S Hickman, and A Lynn Abbott. Convolutional neural network-based in-vehicle occupant detection and classification method using second strategic highway research program cabin images. *Transportation Research Record*, 2675(8):443–457, 2021.

[53] 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 *21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1672–1679. IEEE, 2018.

[54] 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.

[55] Abhijit Sarkar. Visual dictionary of human action in vehicular environment using computer vision. *Human Factors in Transportation*, 60(60), 2022.

[56] Abhijit Sarkar, Sparsh Jain, Jeremy Sudweeks, and Miguel Perez. 2 driver attention modeling through evidence accumulation and gaze fixation. *Towards Human-Vehicle Harmonization*, 3:13, 2023.

[57] Donald Sawicki. Police radar information center, March 2023. URL https://copradar.com/chapts/references/acceleration.html.

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

[59] Sergej S Shadrin and Anastasiia A Ivanova. Analytical review of standard sae j3016 «taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles» with latest updates. *Avtomobil'. Doroga. Infrastruktura.*, (3 (21)):10, 2019.

[60] Yuan Shen, Niviru Wijayaratne, Pranav Sriram, Aamir Hasan, Peter Du, and Katherine Driggs-Campbell. CoCAtt: A cognitive-conditioned driver attention dataset. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 32–39, 2022.

[61] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint:2009.03509*, 2020.

[62] Akash Sonth, Harshavardhan Settibhaktini, and Ankush Jahagirdar. Vehicle speed determination and license plate localization from monocular video streams. In *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, pages 267–277. Springer, 2020.

[63] Timothy Stewart. Overview of motor vehicle crashes in 2020. Technical Report No. DOT HS 813 266, 2022.

[64] United States Department of Transportation. Distracted driving, December 2022. URL https://www.nhtsa.gov/risky-driving/distracted-driving.

[65] A Vadeby, B Turner, K Machata, P Hollo, and T Shelton. Speed and crash risk. In *ITF (International Transport Forum)*, 2018.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[67] Virginia Highway Safety Office. 2018 virginia traffic crash facts, March 2023. URL https://www.dmv.virginia.gov/safety/crash_data/crash_facts/crash_facts_18.pdf.

[68] Sourabh Vora, Akshay Rangesh, and Mohan Manubhai Trivedi. Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis. *IEEE Transactions on Intelligent Vehicles*, 3(3):254–265, 2018.

[69] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[70] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[71] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint:1810.00826*, 2018.

[72] 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.

[73] 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*, 2019.

[74] 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.

[75] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. TransVOD: end-to-end video object detection with spatial-temporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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

Appendices

Appendix A

Semantic Scene Graph

We present another example of semantic scene graph creation in complex scenarios such as in Fig. A.1. The road segments (lanelets) have been labeled for reference.

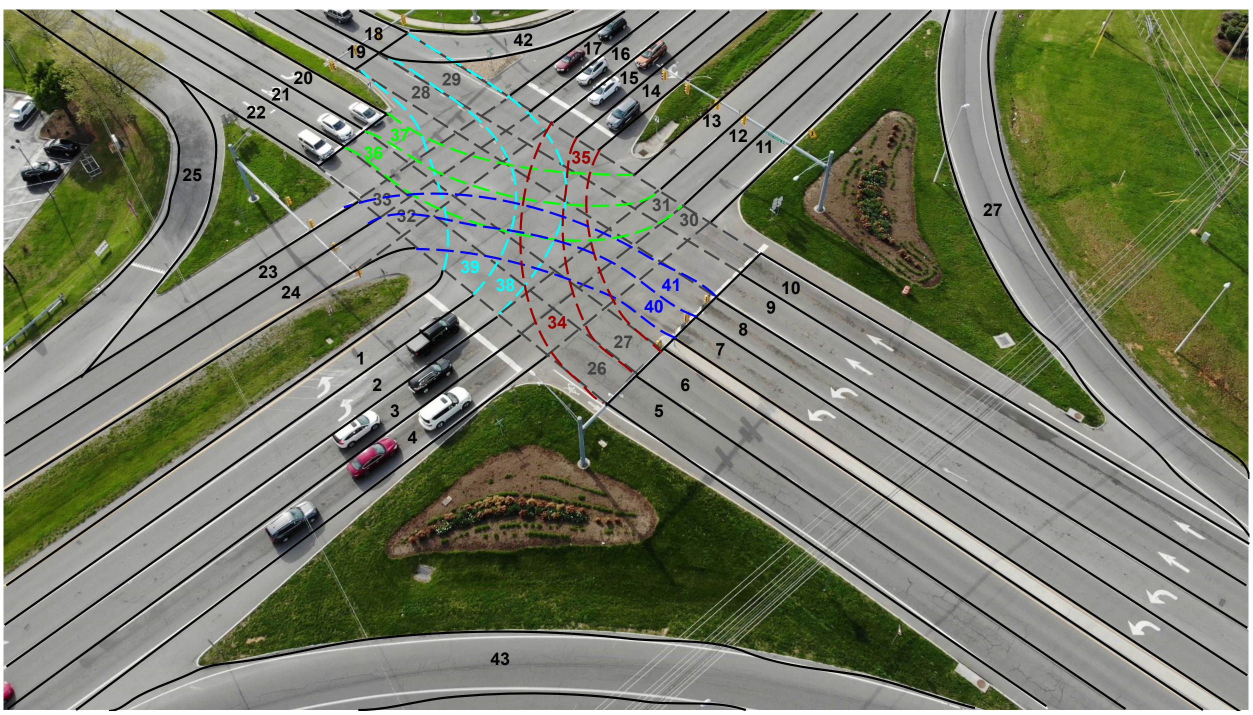

Figure A.1: Drone footage for a road intersection in Blacksburg, VA. Intersecion labeled with the different road segments called “lanelets”. (Turning lanes have been colored differently for clarity.)

The road graph based on the road intersection in Fig. A.1 has been shown in Fig. A.2. Since the selected intersection has several road segments leading into and going out of the intersection, the resulting road graph has several nodes.

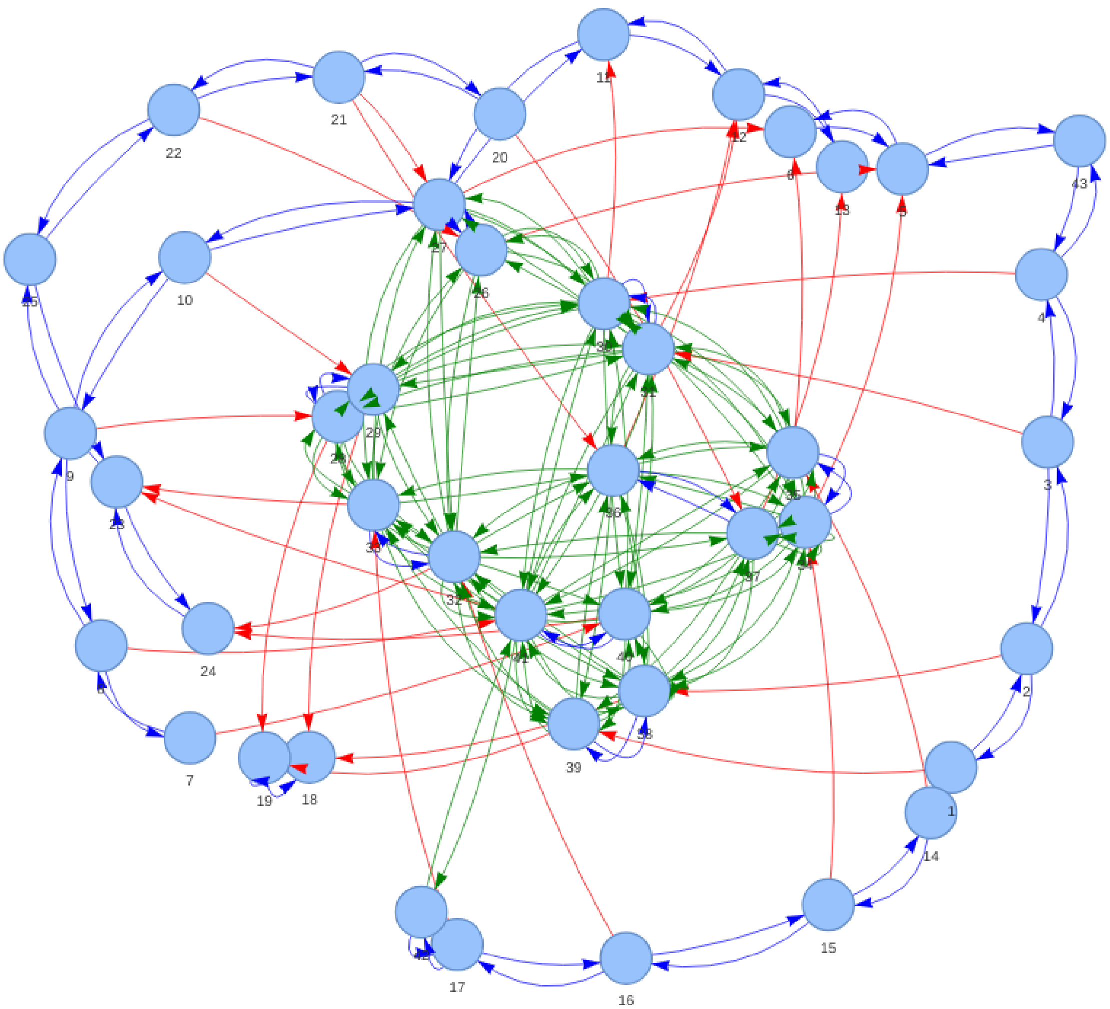

Figure A.2: Road graph for the intersection shown in Fig. A.1. Different colored edges indicate different types of edges- Consecutive, adjacent and overlapping

In Fig. A.3 we additionally label the vehicles by alphabets. Object detection is first used to obtain the bounding boxes of various traffic participants. Each traffic participant is then further represented by the centroid of the bounding boxes.

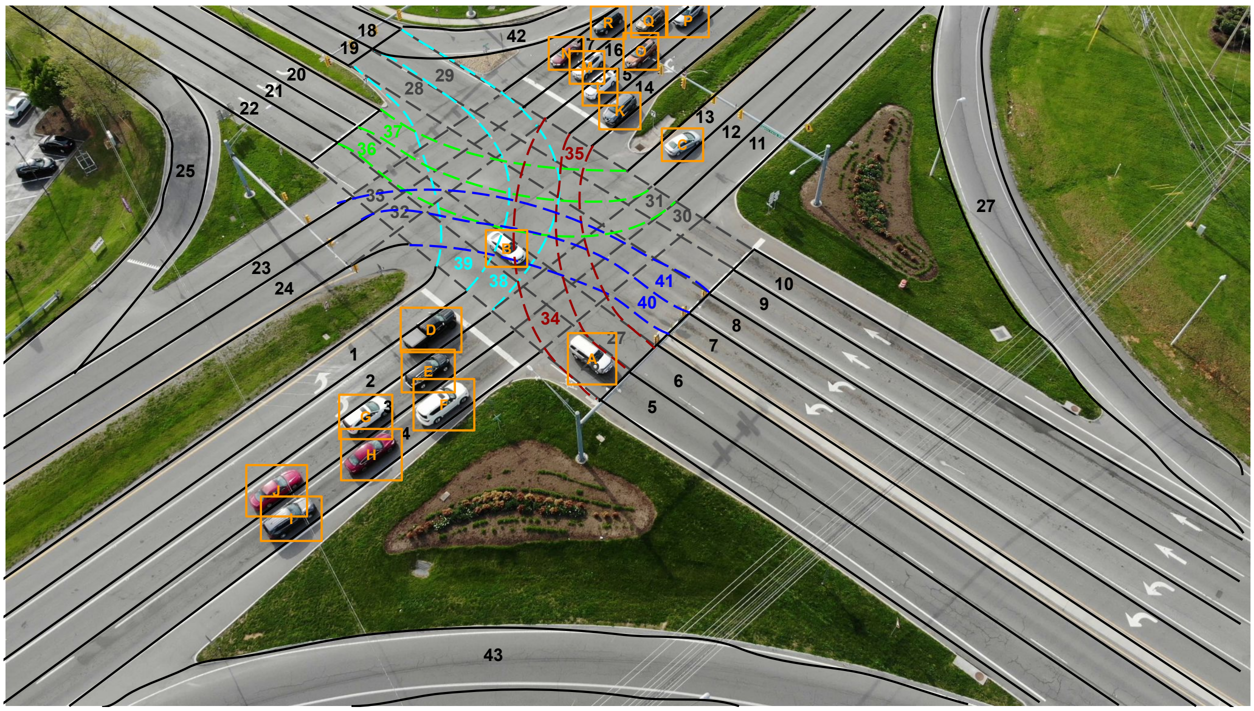

Figure A.3: Vehicles labeled for one of the frames from the intersection drone footage from Blacksburg, VA. We assume that each traffic participant only has projections in the lanelet(s) the bounding box centroid lies in.

Finally, in Fig. A.4 we create the semantic scene graph where each node is a traffic participant, and the edges represent various semantic relationships.

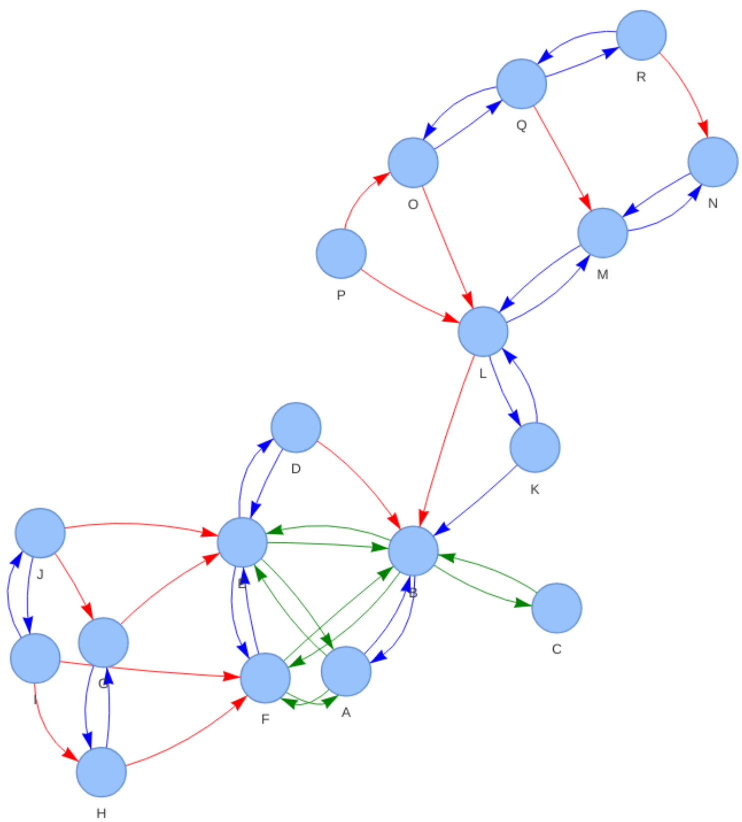

Figure A.4: Semantic scene graph for the frame shown in Fig. A.3. Different colored edges indicate different types of edges- Longitudinal, lateral and intersecting