SJSU SAN JOSÉ STATE UNIVERSITY

MTI MINETA TRANSPORTATION INSTITUTE

Developing a Computer Vision-Based Decision Support System for Intersection Safety Monitoring and Assessment of Vulnerable Road Users

Arash Jahangiri, PhD

Anagha Katthe

Aryan Sohrabi

Xiaobai Liu, PhD

Shashank Pulagam

Vahid Balali, PhD

Sahar Ghanipoor Machiani, PhD

CSUTC California State University Transportation Consortium

SAN DIEGO STATE UNIVERSITY

MINETA TRANSPORTATION INSTITUTE

Founded in 1991, the Mineta Transportation Institute (MTI), an organized research and training unit in partnership with the Lucas College and Graduate School of Business at San José State University (SJSU), increases mobility for all by improving the safety, efficiency, accessibility, and convenience of our nation's transportation system. Through research, education, workforce development, and technology transfer, we help create a connected world. MTI leads the four-university. MTI leads the four-university California State University Transportation Consortium funded by the State of California through Senate Bill 1.

MTI's transportation policy work is centered on three primary responsibilities:

Research

MTI works to provide policy-oriented research for all levels of government and the private sector to foster the development of optimum surface transportation systems. Research areas include: bicycle and pedestrian issues; financing public and private sector transportation improvements; intermodal connectivity and integration; safety and security of transportation systems; sustainability of transportation systems; transportation / land use / environment; and transportation planning and policy development. Certified Research Associates conduct the research. Certification requires an advanced degree, generally a Ph.D., a record of academic publications, and professional references. Research projects culminate in a peer-reviewed publication, available on TransWeb, the MTI website (http://transweb.sjsu.edu).

Education

The Institute supports education programs for students seeking a career in the development and operation of surface transportation systems. MTI, through San José State University, offers an AACSB-accredited Master of Science in Transportation Management and graduate certificates in Transportation Management, Transportation Security, and High-Speed Rail Management that serve to prepare the nation's transportation managers for the 21st century. With the active assistance of the California Department of Transportation (Caltrans), MTI delivers its classes over a state-of-the-art videoconference network throughout the state of California and via webcasting beyond, allowing working transportation professionals to pursue an advanced degree regardless of their location. To meet the needs of employers seeking a diverse workforce, MTI's education program promotes enrollment to under-represented groups.

Information and Technology Transfer

MTI utilizes a diverse array of dissemination methods and media to ensure research results reach those responsible for managing change. These methods include publication, seminars, workshops, websites, social media, webinars, and other technology transfer mechanisms. Additionally, MTI promotes the availability of completed research to professional organizations and journals and works to integrate the research findings into the graduate education program. MTI's extensive collection of transportation-related publications is integrated into San José State University's world-class Martin Luther King, Jr. Library.

REPORT 20-08

DEVELOPING A COMPUTER VISION-BASED DECISION SUPPORT SYSTEM FOR INTERSECTION SAFETY MONITORING AND ASSESSMENT OF VULNERABLE ROAD USERS

Arash Jahangiri, PhD
Anagha Katthe
Aryan Sohrabi
Xiaobai Liu, PhD
Shashank Pulagam
Vahid Balali, PhD
Sahar Ghanipoor Machiani, PhD

March 2020

TECHNICAL REPORT DOCUMENTATION PAGE

1. **Report No.** 20-08	2. **Government Accession No.**	3. **Recipient's Catalog No.**
4. **Title and Subtitle** Developing a Computer Vision-Based Decision Support System for Intersection Safety Monitoring and Assessment of Vulnerable Road Users		5. **Report Date** March 2020
		6. **Performing Organization Code**
7. **Authors** Arash Jahangiri, PhD Anagha Katthe Aryan Sohrabi Xiaobai Liu, PhD Shashank Pulagam Vahid Balali, PhD Sahar Ghanipoor Machiani, PhD		8. **Performing Organization Report** CA-MTI-1853
9. **Performing Organization Name and Address** Mineta Transportation Institute College of Business San José State University San José, CA 95192-0219		10. **Work Unit No.**
		11. **Contract or Grant No.** ZSB12017-SJAUX
12. **Sponsoring Agency Name and Address** State of California SB1 2017/2018 Trustees of the California State University Sponsored Programs Administration 401 Golden Shore, 5th Floor Long Beach, CA 90802		13. **Type of Report and Period Covered** Final Report
		14. **Sponsoring Agency Code**

15. **Supplemental Notes**

16. **Abstract**

Vision-based trajectory analysis of road users enables identification of near-crash situations and proactive safety monitoring. The two most widely used sur-rogate safety measures (SSMs), time-to-collision (TTC) and post-encroachment time (PET)—and a recent variant form of TTC, relative time-to-collision (RTTC)—were investigated using real-world video data collected at ten signalized intersections in the city of San Diego, California. The performance of these SSMs was compared for the purpose of evaluating pedestrian and bicyclist safety. Prediction of potential trajectory intersection points was performed to calculate TTC for every interacting object, and the average of TTC for every two objects in critical situations was calculated. PET values were estimated by observing potential intersection points, and frequencies of events were estimated in three critical levels. Although RTTC provided useful information regarding the relative distance between objects in time, it was found that in certain conditions where objects are far from each other, the interaction between the objects was incorrectly flagged as critical based on a small RTTC. Comparison of PET, TTC, and RTTC for different critical classes also showed that several interactions were identified as critical using one SSM but not critical using a different SSM. These findings suggest that safety evaluations should not solely rely on a single SSM, and instead a combination of different SSMs should be considered to ensure the reliability of evaluations. Video data analysis was conducted to develop object detection and tracking models for automatic identification of vehicles, bicycles, and pedestrians. Outcomes of machine vision models were employed along with SSMs to build a decision support system for safety assessment of vulnerable road users at signalized intersections. Promising results from the decision support system showed that automated safety evaluations can be performed to proactively identify critical events. It also showed challenges as well as future directions to enhance the performance of the system.

17. **Key Words** proactive safety monitoring, safety surrogate measures, visual trajectory analysis, signalized intersection safety, vulnerable road users	18. **Distribution Statement** No restrictions. This document is available to the public through The National Technical Information Service, Springfield, VA 22161		
19. **Security Classif. (of this report)** Unclassified	20. **Security Classif. (of this page)** Unclassified	21. **No. of Pages** 56	22. **Price**

Form DOT F 1700.7 (8-72)

040520

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

While the number of people using eco-friendly modes of travel such as walking and bicycling is growing, statistics show an increasing trend for crashes involving pedestrians and bicyclists. According to the fatality analysis reporting system (FARS) encyclopedia, around 6,000 pedestrian fatalities and 800 bicyclist fatalities were recorded in the U.S. in 2017. A fundamental task when encouraging and promoting eco-friendly transportation modes is to improve bicyclist and pedestrian safety and create safer communities. A preliminary and important step towards improving safety is identifying situations where bicycle and pedestrian crashes are more likely to occur. With the advent of technology in several domains, such as transportation, communication, computer vision, and machine learning, our cities are now transforming into "smart cities" where video cameras, smartphones, and other sensory devices can be leveraged to analyze and produce transportation insights. This report focuses on utilizing vision-based safety monitoring to identify critical intersections for walking and bicycling. Since video cameras can be found in many locations and especially at intersections, it would be beneficial to use this existing infrastructure for safety assessments by analyzing road user interactions in video data obtained from these cameras.

The traditional approach to identifying high-risk locations is to use historical roadway crash data and measure safety based on frequency of crashes and exposure (e.g., population exposed to crashes expressed in many different forms) data. However, this traditional approach requires a long period of time since roadway crashes are rare events. Even a long period of time may not produce sufficient data, especially if an infrequent crash type is being studied (e.g., a crash between bicyclists making left turns from an approach and vehicles going through the intersection from the opposite approach). In addition, changes may occur over long periods of time such as design improvements, demand variation, and so on, that potentially could impact the results of safety evaluations. Given these shortcomings, the traditional safety assessment is considered a reactive approach. A more efficient way of assessing safety is to utilize measures known as surrogate safety measures (SSMs) that enable a proactive safety evaluation approach. The present project investigated the two most widely used SSMs, time-to-collision (TTC) and post-encroachment time (PET), alongside a recent variant form of TTC, relative time-to-collision (RTTC). Given two interacting road users, TTC estimates the time that it will take each road user to reach a predicted object trajectory intersection point. RTTC is the difference between the time it takes the first and second user to reach the predicted intersection point. PET calculates the difference in time from when the first road user reaches the observed trajectory intersection point to when the second user reaches that point.

An essential part of safety assessment using SSMs is obtaining detailed information regarding how vehicles, pedestrians, and bicyclists interact with each other at intersections. Video data annotation was conducted to obtain detailed information about the object trajectories of all road users. Adopting these SSMs, the safety of vulnerable road users was assessed using real-world video data collected at ten signalized intersections in the city of San Diego, California. Although RTTC provided useful information regarding the relative distance between objects in time, it was found that in certain conditions where objects are far from each other, the interaction between the objects was incorrectly flagged as critical based on a small RTTC. Comparison of PET, TTC, and RTTC for different critical classes also

showed that several interactions were identified as critical using one SSM but non-critical using a different SSM. These findings suggest that safety evaluations should not solely rely on a single SSM, and instead, different SSMs should be considered to ensure the reliability of evaluations.

This project also developed a decision support system to automatically assess pedestrian and bicycle safety at intersections by applying computer vision algorithms and SSMs. The proposed system can identify high-risk pedestrian and bicycle locations and proactively measure the safety effects of countermeasures at intersections. To train and develop machine vision models, data from a candidate intersection were utilized. Object detection and tracking models were developed to identify object trajectories for vehicles, pedestrians, and bicyclists. At every time frame, key variables such as speed, direction of travel, and location were extracted from machine vision model outputs. All three SSMs were implemented in the decision support system, and consequently, near-crash situations were identified. The decision support system architecture is presented in Figure 2.

Although many intersections are equipped with video cameras—which can provide detailed data for proactive safety monitoring—the existing infrastructure is not typically leveraged for these types of analyses. As a result, local agency staff are not able to accurately and proactively identify high-risk intersections or assess which facilities are in the greatest need of improvement. The vision-based safety monitoring system developed in this project shows promising results and can support proactive safety evaluation of vulnerable road users at intersections. This system was capable of detecting and tracking objects such as vehicles, pedestrians, and bicyclists with relatively adequate accuracy. This provides an opportunity to proactively answer the question of whether or not intersections with a certain treatment are safer than similar intersections without that treatment, or whether safety has improved after the implementation of a certain countermeasure.

The performance of the decision support system can be enhanced in several ways. Using a higher-resolution camera to record the videos at intersections could aid in improving detection for vulnerable road users, which in turn would also improve the tracking results. The tracking module can further be improved by estimating the positions of occluded objects accurately. Furthermore, machine vision models can be enhanced by utilizing more data. Additional data would add more information for model training and consequently would make them more intelligent. Finally, it is a challenge to use data from one intersection in model training to predict behavior in another intersection. Using more data from different intersections in model training could lead to better generalizability.

I. INTRODUCTION

According to the National Highway Transportation Safety Administration (NHTSA) reports, the annual average number of pedestrian and bicyclist roadway fatalities in the U.S. has been around 4,800 and 720, respectively, over the last decade; the annual average number of pedestrian and bicyclist injuries has hovered around 65,000 and 50,000, respectively. Between 2009 and 2016, the number of pedestrian and bicyclist fatalities saw a marked trend upward. Taken together, the overall percentage of pedestrian and bicycle crashes now accounts for 18% of total fatalities, up from 13% only a decade ago. In 2017, the number of pedestrian and bicyclist fatalities were nearly 6,000 and 800, respectively. This alarming trend urgently needs attention from researchers and practitioners.

Accurate estimation of bicycle and pedestrian volumes, known as exposure data, is an essential part of safety assessments. However, most existing bicycle and pedestrian networks are not equipped to routinely collect count data such as is typically collected for vehicular networks (e.g., via loop detectors). Given this lack of bicycle and pedestrian exposure data, local agency staff are not able to accurately assess which facilities are at the highest need of improvement. In the era of big data, there is an opportunity to obtain exposure data from video data to support estimation of pedestrian and bicyclist volumes and conduct in-depth safety assessments.[1] In addition, visual analysis of roadway user interactions provides detailed information on road user trajectories that enables advanced safety monitoring for the identification of near-crash situations.

The present study aims to explore the utilization of advanced computer vision techniques to proactively conduct safety assessment at intersections for bicycling and walking by applying SSMs. This study develops a decision support system that can be used to identify high pedestrian and bicycle crash risk intersections and to measure the safety effects of countermeasures, such as traffic calming strategies. The remainder of the report is organized as follows. Past studies are reviewed in the literature review section. Next, the decision support system development is presented, including data collection, visual analysis of road users, and proactive safety monitoring. Visual analysis consists of machine vision models such as object detection and tracking. These models aid in obtaining the trajectories of vehicles, bicyclists, and pedestrians. Proactive safety monitoring discusses the SSMs adopted in this study. This shows how trajectories can be utilized to identify near-crash situations, which in turn leads to proactive safety assessment. Subsequently, visual analysis results as well as safety evaluation results are presented, followed by the conclusions section.

II. LITERATURE REVIEW

Traditionally, road accident statistics and historical crash data have been used as indicators to evaluate the level of road safety. Crash data are often used to identify necessary safety improvements and assess the present conditions. However, in order to utilize this traditional approach, one should collect crash occurrences for at least a few years since crashes occur infrequently. Over a long period of data collection, infrastructure design, traffic signal timing, pedestrian and bicyclist activities may change such that biases are produced in safety assessment results. Also, crash data are not reliable, as they can be erroneous and need long observation times.[2]

Traffic conflict techniques (TCTs) can be adopted to assess safety for road users, and they have been suggested as a substitute for the analysis of historical crash data.[3,4] The main idea of TCT is to examine conflict points at intersections when road users are interacting with each other. The conflict points are locations where object (e.g., vehicle, pedestrian, bicyclist) trajectories intersect. TCTs investigate these locations by analyzing object trajectories. The severity and frequency of these conflict points is determined by safety measures known as SSMs. Various safety measures have been studied, and among them, TTC and PET are the most widely used measures (Chen et al. 2017; Gettman and Head 2001; Allen, Shin, and Cooper 1978).[5,6,7] Below, we present a brief overview of SSMs that have been used in the past.

SURROGATE SAFETY MEASURES (SSMS)

Frequency and severity of crashes have been used as indicators for safety evaluations. The idea behind SSMs is to identify unsafe situations (i.e., near crashes) and measure the frequency and severity of these events, yielding more observations compared to actual crash events, as crashes are rare events. As a result, the time period required to perform safety evaluations can be dramatically reduced, enabling proactive safety assessments.

SSMs form the core concept of TCTs. A traffic conflict is a perceptible situation where two or more road users proceed towards each other in time and space so far that there is a risk of collision provided their movements remain unchanged. In order for SSMs to be useful for safety applications, two conditions have to be satisfied (A. Tarko et al. 2009).[8]

- A non-crash event should be observed and perceived as a potential crash event.

- A practical method should be used for transforming a non-crash event, based on its future potential crash location, to a crash frequency and/or severity (A. Tarko et al. 2009) [9].

In order to measure the criticality, severity, and frequency of different types of road user conflicts, SSMs can be classified into three categories: temporal proximal indicators, spatial proximal indicators, and deceleration-based proximal indicators (see Table 1). The main variables used to define unsafe situations in these three categories (temporal, spatial, and deceleration-based) are time, distance, and deceleration, respectively.

Table 1.   List of Traffic Conflict Measures Classified Based on Type of Proximity (Mahmud et al. 2017; Zheng, Ismail, and Meng 2014) [10] [11]

Type	Traffic conflict measures
Temporal	TTC (Hayward, n.d.) [12], PET (Allen, Shin, and Cooper 1978) [13], Time to accident (TA) (Yang, Ozbay, and Bartin 2010) [14], Time to stop line, Time exposed time-to-collision (TET) (Mahmud et al. 2017) [15], Time integrated time-to-collision (TIT) (Mahmud et al. 2017) [16], Time to line crossing (Van Der Horst 1990) [17], Gap time (GT) (Gettman and Head 2001) [18], Initially attempted PET (Gettman and Head 2001) [19], Encroachment time (Gettman and Head 2001) [20], Time headway (H) (Vogel 2003) [21], Crash index (CI) (Ozbay et al. 2008) [22], Modified time-to-collision (MTTC) (Ozbay et al. 2008) [23], Time advantage (Laureshyn, Svensson, and Hydén 2010) [24], Time to departure (A. P. Tarko 2012) [25], Braking time (Zheng, Ismail, and Meng 2014; Lu et al. 2012) [26] [27], Safety surrogate histogram (SSH) (Ghanipoor Machiani and Abbas 2016) [28]
Spatial	Proportion of stopping distance (PSD) (Allen, Shin, and Cooper 1978) [29], Potential index for collision with urgent deceleration (PIUCD), Unsafe density (UD) (Barceló et al. 2003) [30], Margin to collision, Difference of space distance and stopping distance (DSS) (Svensson 1998) [31]
Deceleration-based	Deceleration rate to avoid crash (DRAC) (Svensson 1998) [32], Crash potential index (CPI), Criticality index function (CIF) (Svensson 1998) [33]

The most commonly used SSMs for conflict assessment include but are not limited to TTC, RTTC, time exposed time to collision (TET), time integrated time to collision (TIT), and PET. (Ozbay et al. 2008).[34] In all of these measures, time has the key role in defining a critical event (i.e., measures belong to the temporal category). Comparing to distance and deceleration, time provides a clearer and more intuitive picture of unsafe situations.

Time-to-collision (TTC)

The concept of was first proposed as a safety measure more than four decades ago by Hayward (Hayward, n.d.).[35] He defined at an instant t as the "time taken by the two road users to collide, provided the collision course and speed difference are the same," and the measure can be calculated using Equation 1 (Nadimi, Behbahani, and Shahbazi 2016).[36]

Equation 1: $$\text{TTC} = (X_L - X_F - l_L)/(V_F - V_L)$$

Here, X_L = Position of leading vehicle

X_F = Position of following vehicle

V_F = Velocity of the following vehicle

V_L = Velocity of the leading vehicle

l_L = Length of the vehicle

Equation 1 was modified (Ozbay et al. 2008)[37] to account for acceleration and deceleration variations, and the modified TTC safety indicator was termed MTTC. It should be noted that the aforementioned study (Ozbay et al. 2008)[38] was only focused on interactions

between vehicles on freeways and limited to rear-end collisions. The safety indicator was validated using traffic simulation models and real accident data. It was found that only MTTC as an indicator was not enough to determine the severity of the events. MTTC is estimated using Equation 2.

$$\text{MTTC} = \frac{-\Delta V \pm \sqrt{\Delta V^2 + 2\Delta a d}}{\Delta a}$$

Equation 2:

Here, Δa = Relative acceleration; ΔV = Relative velocity; d = Relative distance

MTTC is calculated using Equation 2 under the assumption of constant acceleration for the two road users. However, situations can arise wherein the following vehicle has a higher acceleration compared to the leading vehicle, and in such cases MTTC is calculated considering change in acceleration in the equation. In order to accommodate change in acceleration in the new equation , linear equations of motion were adopted, and positions of leading and following vehicles were computed considering linear acceleration.[39, 40] The equations can be found in Behbahani and Nadimi.[41]

The limitation of constant acceleration were addressed in Behbahani and Nadimide[42] by developing a theoretical formula such that the kth derivative of position is constant. TTC was then calculated with respect to velocity, acceleration, and jerk, which are the first, second, and third derivatives of position, respectively. Nevertheless, the equations can only be applied for rear-end situations.

Relative Time-to-collision (RTTC)

The formula for calculating for rear-end and side-impact collisions varies. In the former case, it is necessary to define the objects as "following" and "leading" in order to apply the mathematical formula presented earlier, whereas in the latter case, it is not possible to have similar object definitions as they are not traveling on the same path. For side impact or angle collisions, it becomes important to first determine the potential conflict point and later calculate time to this intersection point (TTX), which is the time taken by individual road users to reach the potential conflict point. [43 44]

For side impact collisions, the potential conflict point should be determined based on objects' directions of travel, which often change, and thus direction of travel at every instant is calculated with respect to a reference point. When two objects (i.e. 1 and 2) are moving with directions θ_1 and θ_2 from location (x_1, y_1) and (x_2, y_2), respectively, the conflict point (x_+, y_+) is determined using Equation 3 and Equation 4.

$$x_+ = \frac{(y_2 - y_1) - (x_2 \tan\theta_2 - x_1 \tan\theta_1)}{(\tan\theta_1 - \tan\theta_2)}$$

Equation 3:

$$y_+ = \frac{(x_2 - x_1) - (y_2 \cot\theta_2 - y_1 \cot\theta_1)}{(\cot\theta_1 - \cot\theta_2)}$$

Equation 4:

The time taken by each object (i.e. 1 and 2) to reach the potential conflict point (i.e. TTX_1 and TTX_2) is calculated using Equation 5 and Equation 6, respectively (Behbahani and Nadimi 2015).[45]

Equation 5: $\quad \text{TTX}_1 = \frac{|\vec{r_+} - \vec{r_1}|}{|\vec{v_1}|} \, \text{sign}\,((\vec{r_+} - \vec{r_1}).\vec{v_1})$

Equation 6: $\quad \text{TTX}_2 = \frac{|\vec{r_+} - \vec{r_2}|}{|\vec{v_2}|} \, \text{sign}\,((\vec{r_+} - \vec{r_2}).\vec{v_2})$

$\vec{v_1}$ and $\vec{v_2}$ are the respective velocities of two objects, and $\vec{r_n}$ is the vector representation of coordinate (x_n, y_n). The sign function determines whether the object has passed the intersection. When the sign function returns a negative value, it indicates that the object has already passed the intersection. The difference in TTXs between two objects is known as RTTC, as shown in Equation 7.

Equation 7: RTTC = TTX_1 - TTX_2

When RTTC is zero, a collision is going to occur if directions of travel, speed, and acceleration are maintained. This is because the time taken for both road users to reach the potential conflict point is equal. When RTTC is close to zero the objects are in a dangerous situation as they are too close to each other. Hence, a minimum value for RTTC is considered in order to identify potential critical events (e.g., a critical event exist if RTTC is less than one second). Assigning a threshold value helps to identify not only the critical events but also near-critical events. To determine traffic conflict severity, different threshold values can be investigated. The threshold values vary across different studies from 1.0 s to 5.0 s.[46][47][48]

Time Exposed Time to Collision (TET)

This measure is defined as the summation of all the moments (in seconds) in which the driver approaches the leading vehicle with time to collision less than a predefined threshold value of time to collision (1.5 and/or 3 seconds). Thus, as the TET decreases, the situation becomes safer. However, TET may not be a good indicator of collision risk as it does not indicate the potential risk conditions for different critical events. In other words, after measuring TET, the number of critical events is unknown. In order to account for the frequency of critical events, time integrated time to collision (TIT) was proposed as a new measure.[49][50]

Time Integrated Time to Collision (TIT)

TIT calculates the number of critical events with TTC below a predefined threshold. It expresses the level of safety, or relative probability of a conflict, using the integral of the time to collision profile of drivers (in s^2). A graph of versus time period illustrates how time to collision for a specific vehicle changes over time as shown in Figure 1.
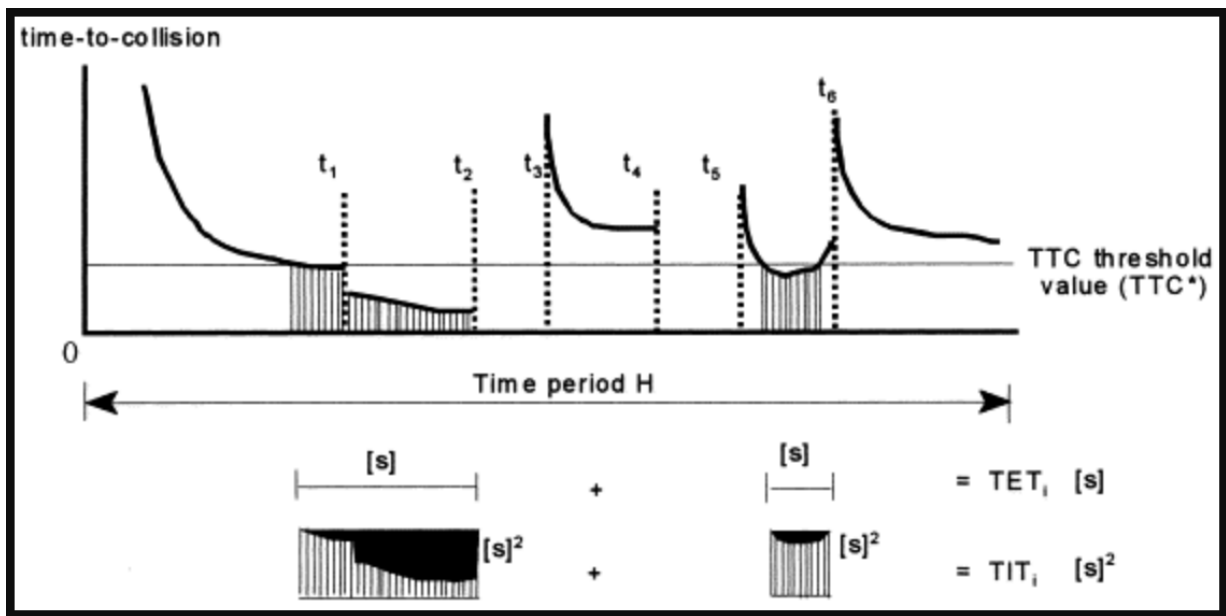
Figure 1. Time to Collision Profile of Driver–Vehicle Combination (Minderhoud and Bovy 2001b)

To better understand the variations in TTC for a specific vehicle, Figure 1 plots the TTC curve for a vehicle that interacts with other vehicles over a time period H. At t = 0, the vehicle approaches a slow-moving vehicle, so the TTC value starts decreasing until t = t_1. At that moment, the vehicle decides to slow down and change lanes but after changing the lane, faces another vehicle at a shorter distance, and TTC decreases further between t = t_1 and t = t_2. At t = t_2, the driver is very close to a conflict, as seen by the low TTC value, so the vehicle reduces the speed. There is no TTC information during this period of speed reduction, and the similar pattern continues afterwards.

For TTC values below a certain threshold, TIT is calculated as the product of time to collision and the time difference for two instants, say t_5 and t_6 (see Figure 1) (Yang, Ozbay, and Bartin 2010; Minderhoud and Bovy 2001a). [51] [52] To calculate TIT for an object '*i*' in discrete time '*t*', Equation 8 is used.

Equation 8: $TIT = \sum_{t=0}^{T}[TTC^* - TTC_i(t)] \cdot \tau_{sc}$

Here, TTC* = Threshold value, $TTC_i(t)$ = TTC value for leading vehicle, τ_{sc} = Discrete time moments, T= Time interval ($t_2 - t_1$)

For objects *i* =1…N in continuous time '*t*' where N = number of vehicles.

Post-Encroachment Time (PET)

PET was defined as the time difference between the moment when an offending road user leaves an area of potential collision and the moment of arrival of a conflicted road user possessing the right of way (Behbahani and Nadimi 2015). [53]

Equation 9: $\mathrm{PET} = \frac{(X_L - X_F - l_L)}{(V_F)}$

Here, X_L = Position of leading vehicle

X_F = Position of following vehicle

V_F = Velocity of the following vehicle

l_L = Length of the vehicle

Equation 9 is applicable only for rear-end collisions. Unlike TTC, PET cannot be calculated continuously at any time, and the interaction between two road users needs to be observed completely in order to determine the PET value. Thresholds that have been used for PET are similar to the values investigated for TTC: 1.0 s to 5.0 s (Zheng, Ismail, and Meng 2014; Peesapati, Hunter, and Rodgers 2013). [54][55]

Comparing TTC and PET as the most widely used measures, the former determines potential conflict points by predicting object trajectories at different times, and thus a series of TTC values is obtained and can be used for safety evaluations. The latter only deals with a single potential conflict point, which can be determined by observing actual object trajectories. Research has been performed combining these two SSMs to evaluate safety, and a new term called Mixed Index (MI) has been used (Nadimi, Behbahani, and Shahbazi 2016). [56] This index is based on a fuzzy inference system, and accounts for several parameters such as clearance, speed, and relative speed. By combining TTC and PET at each instant, Nadimi et al. (2016) (Nadimi, Behbahani, and Shahbazi 2016) [57] utilized regression analysis to estimate the MI. A study on the I-80 freeway was conducted, and comparison analysis was performed for the values of TTC, PET, and MI; it was found that this measure is most suitable for rear-end collisions. TTC has various mathematical formulae for rear-end conflicts that have been proposed in the past; however, for side impact or cross-angle crashes, prediction of trajectories and collision points still needs to be more efficient and accurate. The latest research on determination of conflict points is based on prediction of the trajectories of two road users based on collision probability, vehicle dynamics, and learned path (Shirazi and Morris 2017). [58] Shirazi and Morris (2017) (Shirazi and Morris 2017) [59] utilized learned paths to correct the predictions by using a Kalman filter on current velocity. The conflict point was obtained by estimating the shortest distance between the coordinate points on the predicted trajectories. Although many studies have focused on developing different SSMs, the suitability of SSMs as an alternative to crash data needs more research due to several limitations and difficulties and the lack of consensus among researchers on which measure or combination of measures should be used.

III. DEVELOPING A DECISION SUPPORT SYSTEM

A vision-based decision support system for proactive safety evaluation consists of several steps as presented in Figure 2. The system developed in this study essentially integrates the process of adopting machine vision models, obtaining critical variables, and evaluating the safety of vulnerable road users by employing SSMs. The system first collects video data at locations of interest, and it then conducts visual analysis of road users, which consists of annotating video data and developing object detection and tracking models. Road users such as vehicles, pedestrians, and bicyclists are referred to as objects in general. Object trajectories can then be constructed either by utilizing annotated data or outputs of machine vision models (i.e., detection and tracking models). Object trajectory data include key variables such as speed, direction of travel, and location, which are utilized to identify near-crash situations for pedestrians and bicyclists based on SSMs. Subsequently, the safety of vulnerable road users can be proactively assessed to produce insights. Proactive safety evaluation can be utilized to measure the effectiveness of a certain treatment implemented at an intersection comparing to a similar intersection without the treatment. It can also be utilized to conduct before after studies to see if a certain countermeasure has improved the safety.
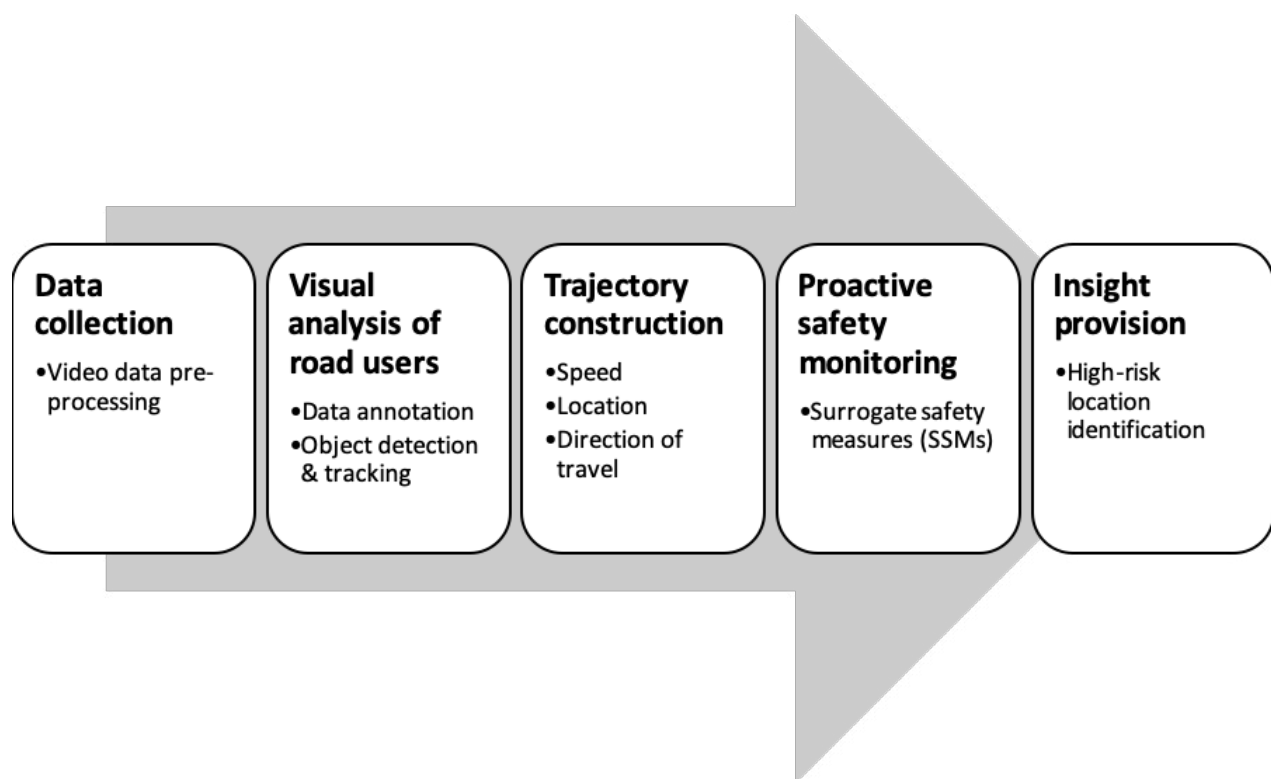
Figure 2.   Decision Support System Architecture for Safety Assessment

DATA COLLECTION

Video data from a total of ten signalized intersections in the city of San Diego have been utilized (see Figure 3 and Table 2). Data collection was conducted using a video camera at each study site for a duration of 24 hours. The data extraction was performed by reviewing these videos for morning, afternoon, and evening peak hours. Subsequently, the data were

reduced to 5-minute videos for each intersection which were obtained by amalgamating several video slots that were shortlisted based on maximum activity of vulnerable road users manually observed in those frames. The raw data collected were in the form of video files which were later annotated using a tool called Vatic. Utilizing the annotated video data, machine vision models were developed (see next section) to extract location and kinetic information of objects interacting at the intersections.
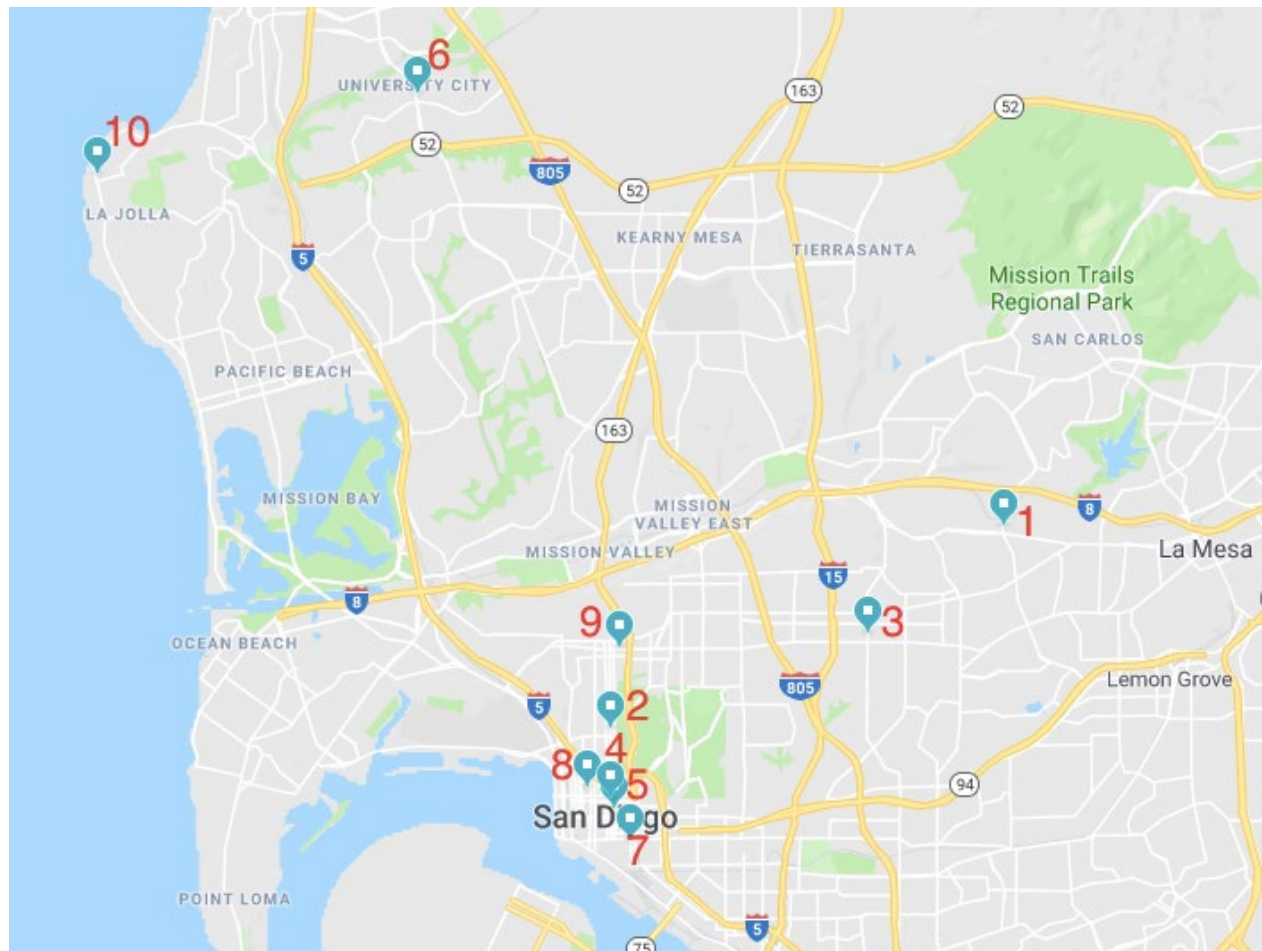
Figure 3. Study Intersections

Table 2.   List of Ten Study Intersections

Intersection number	Name of the Intersection
1	College Ave & Montezuma Rd
2	5th Ave & Laurel St
3	Fairmount Ave & University Ave
4	5th Ave & B St
5	Sixth Ave & Broadway
6	Genesee Ave & Governor Dr
7	10th Ave & J St
8	Union St & Ash St
9	7th Ave & Robinson Ave
10	La Jolla Blvd & Pearl St

VISUAL ANALYSIS OF ROADWAY USERS AT INTERSECTIONS

The goal of this task is to detect objects (e.g., pedestrians, vehicles, bicycles) in video frames and track them over time. The outcomes of these perception tasks can be used to analyze the behaviors and interactions of various roadway users.

Figure 4 summarizes the visual analysis workflow used in this project. It includes seven major components. The first one is data annotation, where a semi-automated toolbox, Vatic annotation tool (Vondrick, Patterson, and Ramanan 2013a) [60], was used to annotate bicycles, cars, and pedestrians in videos. With these data, three different modules are developed to detect objects in video frames, associate detections over consecutive frames to get their tracks, register detections in 3D space (3D localization), estimate scene geometry (perspective transformation), and extract 3D object trajectories. The trajectory results are used to quantify intersection safety evaluations.
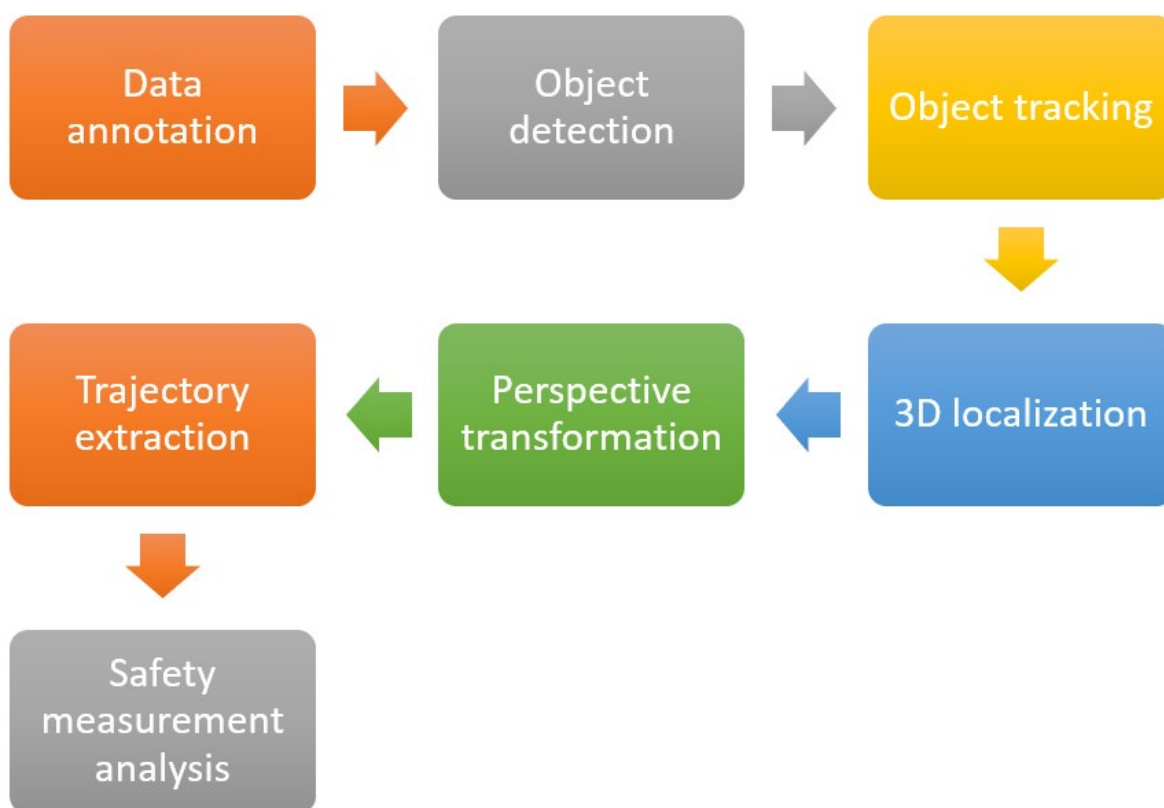
Figure 4.   Visual Analysis Workflow (Zhang, n.d.) [61]

Data Annotation

The objective of this module is to annotate bicycles, cars, and pedestrians in frames to train an object detection model in the next step and to annotate footprints of each object in frames to train a Convolutional Neural Network (CNN) model for estimating the footprint of objects in the 3D localization step.

Methods

In order to train the object detector, identified video data as described in the data collection section were manually annotated with Vatic (Vondrick, Patterson, and Ramanan 2013b) [62] for each of the 10 intersections. Cumulatively, about 50 minutes of video data were annotated from all 10 intersections.

Similarly, in training the CNN model for 3D localization, two videos each of length 5 minutes were annotated with Vatic (Vondrick, Patterson, and Ramanan 2013b) [63] from intersections 2 and 3. However, a problem arises, as Vatic does not directly support annotating points. Hence, the Vatic code was modified to enable annotating footprint points by choosing the point to be the center of the annotated bounding box.

Implementation

The Vatic code used for annotating short videos is located in this github repository: https://github.com/cvondrick/vatic (Vondrick [2011] 2019). [64]

Object Detection

The objective of this module is to detect each object given a video frame and to output its bounding box, indicated by the four coordinates (i.e., the bounding box around an object is defined by four points) for the location where the object is detected, as well as a label indicating whether it is a bicycle, car, or pedestrian.

Methods

To evaluate the performance of the object detection module, Mean Average Precision (mAP) metric was used. A bounding box is considered to be truly positive if it has an Intersection over Union (IOU) overlap bigger than or equal to 0.5 with a ground truth bounding box. IOU is calculated by dividing the overlap area between bounding boxes by the total area of the union of the two bounding boxes (see Equation 10) (Rosebrock 2016). [65] Precision is the fraction of predictions that are truly positive (see Equation 11). Recall is the fraction of true positives predicted correctly by the algorithm (See Equation 12). If an algorithm predicts three positive data samples, among which two are truly positive, the precision is 0.66 (2/3), as only 2 out of the 3 predicted are truly positive, and the recall is 1, as the algorithm predicted both the positive examples correctly. Average Precision (AP) can be calculated by computing the sum of maximum precision values at each recall value changing from 0 to 1 (i.e., recall value of 0.1, 0.2, …, 1) and averaging the obtained total across 11 observations as presented in Equation 13. For instance, if the detection module detects three bicycles, among which both the first and third one are truly bicycles, the precision values at each prediction are 1, 0.5, and 0.67, and the recall values are 0.50, 0.50, and 1 as calculated in Table 3. The maximum precision at each recall value can be identified as shown in Table 3. This example only has two recall values (i.e., 0.5 and 1) and thus only two maximum precision values can be identified. For the recall values at which maximum precision is unavailable, the next best (highest) maximum precision that is available is used. For example, since maximum precision at recall value of 0.6 is unavailable, the next

best maximum precision that is available (i.e., 0.67) was used at this recall level as shown in Table 4. Then, the average precision for this example is 1/11 (1 * 6 + 0.67 * 5), which is 0.85 from the values in Table 4. Mean average precision is then calculated by computing the mean of all the Average Precision scores of different classes (i.e., bicycles, cars, and pedestrians) as presented in Equation 14 (Tan 2019; Hui 2019; Sahoo 2017; "What You Wanted to Know about Mean Average Precision - FastML" n.d.). [66 67 68 69]

Table 3.   Calculation of Precision and Recall Values

Prediction	Ground Truth	Precision	Recall	Max precision
Bicycle	Bicycle	1/1 = 1	1/2 = 0.50	1
Bicycle	Not Bicycle	1/2 = 0.50	1/2 = 0.50	1
Bicycle	Bicycle	2/3 = 0.67	2/2 = 1	0.67

Table 4.   Maximum Precision Value at Each Recall Value Based on Table 3

Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Max precision	1	1	1	1	1	1	0.67	0.67	0.67	0.67	0.67

Equation 10: IOU = Area of overlap / Area of union

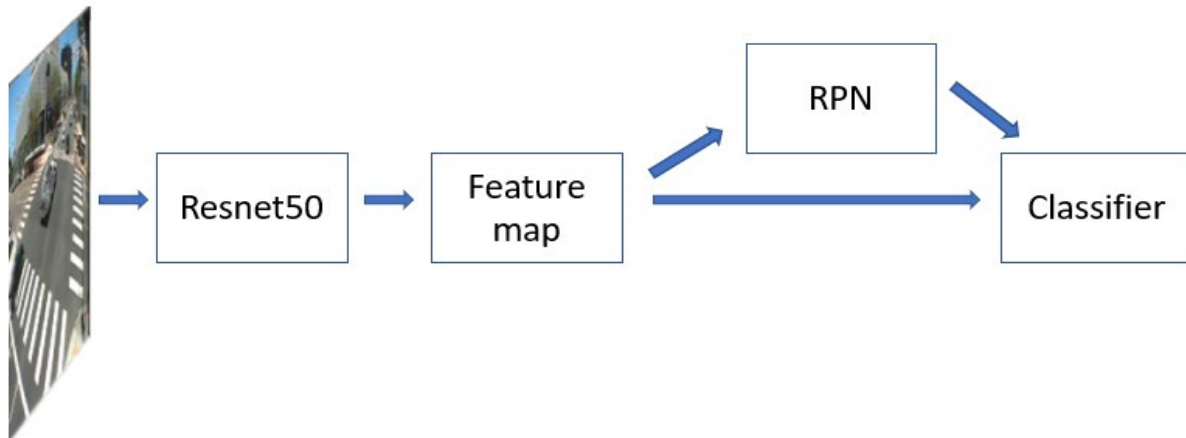Equation 11: Precision = TP / (TP + FP)

Equation 12: Recall = TP / (TP + FN)

Equation 13: AP = $AP = \frac{1}{11}\sum_{r=0}^{1} precision(r)$

Here, TP = True Positives, FP = False Positives, FN = False Negatives, and r = Recall.

Equation 14: $mAP = \frac{\sum_{i=1}^{n} AP_n}{n}$

Here, n represents the number of different classes, which in this project is three: bicycles, cars, and pedestrians.

Figure 5. Faster R-CNN Network Architecture (Ren et al. 2015) [70]

Each road user was detected using the Faster R-CNN model, which is a detection model that uses a CNN to extract features from an input image and feeds them to its two major components: the RPN (Region Proposal Network) and the classifier (see Figure 5). The RPN regresses and outputs bounding box proposals, where an object might be located in the image, to the classifier. The classifier then uses the feature map and classifies each bounding box proposal either as a non-road user, bicycle, car, or pedestrian (Ren et al. 2015).[71] Resnet50, a pretrained network upon the ImageNet dataset, was used as the backbone convolutional feature extractor for the model (He et al. 2015).[72]

The Faster R-CNN model is better suited to the task of detecting objects for this project, as it achieves good performance upon Pascal VOC benchmark and saves computation time by using anchor boxes instead of performing selective searches over images at different scales and ratios. In addition, the convolutional layers are shared between RPN and classifier to speed up bounding box proposals (Ren et al. 2015).[73] As the Faster R-CNN model is both accurate and computationally less expensive than some of its counterpart object detection methods, it is the method that was utilized to perform object detection in the present project.

Implementation

The object detection module was implemented using Keras 2.1.5 (Chollet [2015] 2019),[74] Tensorflow 1.6 (Abadi et al., n.d.),[75] Python 3.6 (van Rossum and Drake 2011),[76] and GeForce GTX 1070 TI GPU. The code used for implementing the object detection module is located in this repository: https://github.com/kbardool/keras-frcnn (Bardool [2017] 2019).[77]

Object Tracking

The objective of this module is to obtain a trajectory for each object in the video given the detection results and output the track ID, the bounding box prediction, and the label associated with the detected object for each frame.

Methods

The five metrics used for evaluating the performance of the tracking module are Multiple Object Tracking Accuracy (MOTA), ID switches (IDs), Mostly Tracked (MT), Partially Tracked (PT), and Mostly Lost (ML).

Equation 15: $$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mm_t)}{\sum_t g_t}$$

MOTA metric measures the accuracy of the tracking algorithm. First, the tracking algorithm error is computed by dividing the sum of the number of missed tracks m_t, the number of false positive track associations fp_t, and the number of mismatches mm_t at all timesteps by the total number of ground truth tracks g_t at all timesteps (see Equation 15). Then, this error is subtracted from 1 to compute the tracking accuracy (Heindl [2017] 2019; Bernardin, Elbs, and Stiefelhagen, n.d.). [78, 79] IDs metric measures how many times an already existing track is reinitialized as another track and is an important metric to look at because the ideal tracking algorithm must reinitialize tracks as little as possible. In the presence of ID switches, an object trajectory may not be traced for longer period of time, which leads to poor safety analysis results. The remaining three metrics (i.e., MT, PT, and ML) measure how many objects are tracked for 80% of the lifespan, 20% to 80% of the lifespan, and less than 20% of the lifespan, respectively, and they are important in indicating how many objects the algorithm tracks successfully for a longer time (Heindl [2017] 2019). [80] The higher the MT and PT count and the lower the ML count, the better the tracking algorithm performance.

After detecting roadway users, all the detection results were transferred to a high-speed tracking algorithm called the Simple Online Realtime Tracking (SORT) algorithm (Bewley et al. 2016). [81] SORT algorithm performs data association between detection results frame by frame to identify tracks, create new tracks if a new object has entered, and stop tracks if an object has disappeared. The Kalman filter predicts the bounding box positions of the tracks, which are then matched to detections using the Hungarian algorithm to assign each track to its detection with maximal IOU. The tracks, whose maximal IOU with any bounding box is less than a threshold, are considered false positives and suppressed.

In the original algorithm, a track is terminated and assigned a new track ID if it is not associated with a detection for more than one frame, which leads to numerous reassignments. To reduce reassignments, the original algorithm was modified to refrain from terminating tracks immediately by preserving the tracks in memory for 600 frames and predicting the positions of the object for upcoming frames with the Kalman filter to handle missing detections and occlusions (Bewley et al. 2016; Durant n.d.). [82 83] This modified tracking algorithm is suited for this project as it achieves real-time performance and avoids ID reassignments by handling occlusions and missing detections robustly.

Implementation

SORT tracker was implemented in Python 3.6 using filterpy 1.4.1 (Labbe [2014] 2019), [84] Pandas 0.24.2 (McKinney, n.d.), [85] Numpy 1.16.4 (Van Der Walt, Colbert, and Varoquaux

2011), [86] and SciPy 1.3.0 (Jones, Oliphant, and Peterson 2001). [87] The tracking code was implemented using the code from these two repositories: https://github.com/abewley/sort (abewley [2016] 2019) [88] and https://github.com/cfotache/pytorch_objectdetecttrack (cfotache [2018] 2019). [89]

3D Localization

The objective of this module is to obtain a two-dimensional (2D) footprint estimate of where an object might be touching the ground given the bounding box coordinates, object category, frame number, and cropped image of the object.

Methods

Mean Squared Error () is the metric used for evaluating the performance of footprint models, as it is a distance-based metric that can find how far off the coordinate predictions are from the ground truth. It is calculated by taking the mean of the square of the sum over the pixel differences between the predictions and the ground truth footprint annotations (see Equation 16) (Wang and Bovik 2009). [90]

Equation 16: $$\text{MSE} = \frac{\left(\sum_{i=1}^{n} \sqrt[2]{(x_{true}-x_{predicted})^2 + (y_{true}-y_{predicted})^2}\right)^2}{n}$$

To obtain a footprint estimate for a bounding box, three different models were trained. One model is a simple regression model trained to regress footprint coordinate given the bounding box coordinates of each object. The model performs well when it is used for the same intersections upon which it is trained. However, it performs poorly for others, as the camera angles differ and it lacks image context to look at features that may help it estimate the footprint more accurately (Zhang, n.d.). [91]

The second model is a convolution neural network built from scratch. This model receives a cropped image, along with bounding box coordinates, as input, but it also performs poorly, as the annotated data set for training the model is not large enough (Zhang, n.d.). [92]

The third model is a CNN model that uses resnet50 and performs global average pooling to obtain a vector of relevant features from the image, which are in turn fed into a two-layer neural network to obtain a footprint for the bounding box. To improve robustness for the CNN model, nine images were added as noise to the training data by offsetting the cropped image between -30% to 30% per training epoch (Zhang, n.d.). [93] This model performs well not only for the intersections used for training but also for intersections that are not used in training (see Table 9 in results).

Implementation

All three models were implemented using Keras 2.1.5 (Chollet [2015] 2019), [94] Tensorflow 1.6 (Abadi et al., n.d.), [95] Python 3.6 (van Rossum and Drake 2011), [96] and GeForce GTX 1070 TI GPU.

Perspective Transformation

The objective of this module is to transform each footprint of an object in an intersection frame by frame into its corresponding bird's eye view coordinate by using the homography matrix, which is computed by picking 4 corresponding points between the bird's eye view from Google Maps and the street view of the intersection.

Methods

Google Maps is used for collecting the bird's eye view of all 10 intersections. To ensure a scale consistency of 0.25 ft/pixel across intersections, a square with a diagonal of length 350 feet was captured using a 1,000 pixel ×1,000 pixel square for each intersection (see Figure 6). One representative image of the street view of each intersection was used to pick 4 corresponding points from the pedestrian crosswalks of the intersection. The points specific to each intersection were stored in text files and reloaded to compute the homography matrix on the fly and perform the perspective transformation upon the footprints of objects.

Figure 6.   Top-down Views for Two Intersections

Street view cameras are not great for estimating speeds of moving objects. Hence, a perspective transformation was performed to transform the street view to the bird's eye view of the intersection. A bird's eye view helps in estimating the speeds of moving objects much better than street view does. Therefore, safety analysis can be performed more accurately (Zhang, n.d.). [97]

Implementation

Matlab 2019a ("MATLAB Documentation" n.d.) [98] was used to pick the corresponding points between the two views. Opencv (*Opencv-Python: Wrapper Package for OpenCV Python Bindings.* (version 4.1.0.25) n.d.) [99] and Python (van Rossum and Drake 2011) [100]

were used to compute the homography matrices based on the 4 corresponding points and transform the footprint to the corresponding bird's eye view coordinate.

PROACTIVE SAFETY MONITORING

In this study, TTX and PET, the two popular measures, were utilized to assess the safety of vulnerable road users at signalized intersections, and SSM performance comparison was also conducted to see if same unsafe events can be identified by different SSMs. In addition, RTTC, which seems to be the most recent variant of TTC, was also employed and was compared to the other two measures. The SSMs' outputs for all possible interaction types were investigated, including vehicle–pedestrian, vehicle–bicycle, pedestrian–bicycle, and bicycle–bicycle interaction. Since the traditional one-dimensional TTC equation cannot be used to determine frequency for cross angle or side-impact conflicts, an advanced method was adopted for calculation of TTX (Miller and Qingfeng Huang 2002). [101] TTX can be calculated continuously at every time frame, and thus the potential trajectory intersection point is constantly changing. Prediction of trajectory intersection is based on the angle and velocity of any two objects interacting with each other.

The procedure for PET calculation adopted in this project is as follows. Based on the trajectory of the road users obtained from video analysis, the time frame of the intersection point was identified. This was considered to be the observed trajectory intersection point. The time difference (measured by counting video frames divided by frames per second) between when the first object passes this intersection point and when the second object reaches it was defined as PET. In other words, PET was calculated by observing the path travelled by the objects, whereas TTX was calculated by predicting the path the object might travel if it continued in the same speed and direction.

Estimating TTC and RTTC

The sequential steps to analyze safety using this measure are as follows:

- Prediction of Trajectory Intersection

- Calculation of Time to Intersection (TTX)

- Estimation of RTTC

Prediction of Trajectory Intersection

Predicting the path of travel requires data from every road user in the form of velocity, location coordinates, and direction. After video processing analysis, location coordinates for all the objects in all video frames, direction of travel, and velocity at every instant were obtained. From the known coordinates (x, y) for every road user and direction of travel (θ) at every instant, the trajectory intersection point was estimated for every frame. The trajectory intersection point calculation was carried out for every frame to record all critical events.

Figure 7-a is an example of vehicle–bicycle interaction. Here, the car with coordinates (x_1, y_1)

and bicycle (x_2, y_2) are moving towards each other in space and time. The direction of travel for vehicle and bicycle is determined by θ_1 and θ_2, respectively. Coordinates obtained from the output of video processing analysis were used to calculate the velocity for the vehicle and bicycle. Using Equation 3 and Equation 4, the potential trajectory intersection (potential collision) point (TIP_1) is calculated. If the vehicle and/or bicycle changes their course of action and direction, as shown in Figure 7-a, the new potential trajectory intersection point (TIP_2) is calculated for the updated direction of travel (θ) and (v) velocities (). This is an iterative process, and potential trajectory intersection points are recalculated every time there is change in location and direction.
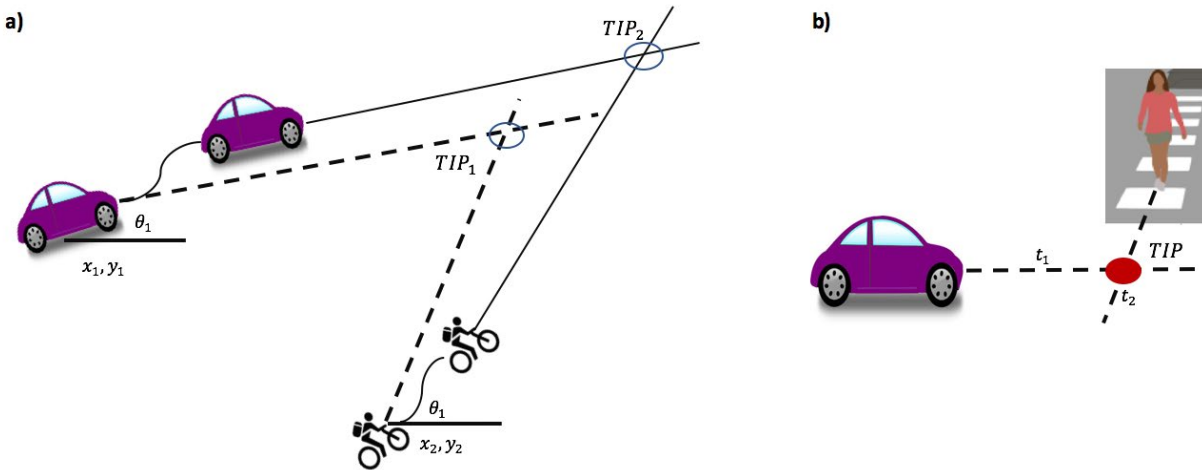
Figure 7.   a) Vehicle–Bicycle Route Interaction b) Vehicle–Pedestrian Route Interaction

Calculation of Time to Intersection (TTX)

Considering Figure 7-a, the time taken by the vehicle to reach the potential TIP_1 (i.e., TTX_1) is calculated by Equation 5. Similarly, the time taken by the bicycle to reach the potential TIP_1 (i.e., TTX_2) is calculated by Equation 6. Since there are two values (i.e., TTX_1 and TTX_2) for a single interaction, the authors opted to use the average of the two as a single measure (TTX_{avg}) to assess estimated time to potential collisions.

Calculation of Relative Time-to-collision (RTTC)

The difference between the TTX values is calculated using Equation 7 to obtain $RTTC$. In the past, studies have shown that the most critical events are obtained when $RTTC$ is at a minimum. It is important to know that $RTTC$ on its own may not be suitable to show all true critical events. There might be cases with very small but with large TTX_{avg}, which would indicate the event is not truly critical as the objects are far from each other.

Estimating PET

The procedure for calculation of PET used was the approach earlier presented where the difference in arrival and departure time of respective objects is noted. Since PET was calculated based on observed trajectories, it was considered as a baseline to compare with the results obtained from the $RTTC$ and TTX_{avg} analysis. The comparison was made

to see whether and how the frequency of critical events identified by RTTC and TTX_{avg} is the same as PET.

In Figure 7-b, both car (x_1, y_1) and pedestrian (x_2, y_2) are moving with their respective speed and direction. It is observed that the trajectory of car and pedestrian meet in future. Assuming it will take t_1 seconds for the vehicle to reach the observed TIP first, and t_2 seconds for the pedestrian to reach the observed TIP second, PET was calculated by finding the difference in time $(t_2 - t_1)$.

Identification of Critical Events

No matter which SSM is applied, it is important to consider certain thresholds to identify critical situations. Threshold values of 1.5, 3.0, and 5.0 s were considered in this project that is consistent with the literature. Therefore, three classes of event criticality were considered: class I: 0–1.5 s; class II: 1.5–3 s; class III: 3–5 s.

IV. RESULTS

MACHINE VISION MODELING RESULTS

Object Detection

Originally, the detection algorithm resizes the shorter side of the image to 600 pixels. To improve detection for small objects (i.e., pedestrians and bicycles far away from the street camera), the image was upscaled (Strelnikov and Oleksandr 2019). [102] The upscaling enables the detector to look at finer features that may otherwise be lost and detect objects more accurately. Hence, the detector was also trained by upscaling the shorter side from 600 to 800 pixels. Upscaling the image improved the overall Mean Average Precision score by 13.28% (see Table 5 and Table 6). Each object detector was trained upon the data, split into 1,000 batches, for 200 epochs. Horizontal flipping was used to augment and double the size of training data.

To evaluate the performance of the detection module, each annotated video clip was split into 80% for training and 20% for testing. Once the object detector was trained, it was run upon testing data for each video clip to obtain detection results. Subsequently, the mAP score was calculated cumulatively for all the test data and individually for the test data by intersection using the detection results and the ground truth bounding boxes. The code used for calculating the score can be found in this repository: https://github.com/rafaelpadilla/Object-Detection-Metrics (Padilla [2018] 2019). [103]

Table 5.   mAP for Detection Model Trained with Image Resized to 600 px

Intersection	Bicycle	Car	Pedestrian	mAP
College Avenue and Montezuma Road	73.78%	67.22%	13.74%	51.58%
5th Avenue and Laurel St	51.73%	74.95%	34.62%	53.77%
Fairmount and University Avenue	50.63%	72.38%	26.84%	49.95%
5th Avenue and B St	61.27%	80.40%	39.28%	60.32%
Sixth Avenue and Broadway	58.37%	81.67%	46.03%	62.06%
Genesee Avenue and Governor Drive	51.73%	74.95%	34.62%	53.77%
10th Avenue and J St	43.95%	80.05%	31.98%	52.00%
Union St and Ash St	77.41%	73.29%	47.04%	65.91%
7th Avenue and Robinson Avenue	17.19%	72.11%	14.07%	34.46%
La Jolla and Pearl St	26.16%	75.18%	14.74%	38.69%
Overall	49.82%	69.80%	28.65%	49.42%

Table 6.   mAP for Detection Model Trained with Image Resized to 800 px

Intersection	Bicycle	Car	Pedestrian	mAP
College Avenue and Montezuma Road	81.40%	79.39%	43.56%	68.15%
5th Avenue and Laurel St	63.61%	88.53%	50.83%	67.65%
Fairmount and University Avenue	57.02%	90.05%	48.22%	65.10%
5th Avenue and B St	69.22%	90.72%	48.17%	69.37%
Sixth Avenue and Broadway	66.76%	88.51%	64.55%	73.27%
Genesee Avenue and Governor Drive	34.69%	72.85%	44.68%	50.74%
10th Avenue and J St	50.53%	93.33%	50.11%	64.66%
Union St and Ash St	84.29%	93.94%	72.26%	83.50%
7th Avenue and Robinson Avenue	28.63%	82.77%	37.32%	49.57%
La Jolla and Pearl St	40.13%	88.32%	27.36%	51.94%
Overall	58.46%	82.90%	46.74%	62.70%

Object Tracking

The modified tracking algorithm performed better than the original one as it preserved tracks in memory for a longer time. It decreased tracking accuracy; however, it reduced ID switches and mostly lost objects significantly and improved the number of mostly tracked and partially tracked objects (see Table 7 and Table 8).

Both the tracking algorithms, the original and the modified versions of SORT, ran upon the detections obtained for the test data from the best object detector. The tracking metrics for different intersections have been computed using the non-occluded tracks from the annotated data and the tracking results from both algorithms using this repository: https://github.com/cheind/py-motmetrics (Heindl [2017] 2019). [104]

Table 7.   Tracking Results for Different Intersections Using Original SORT Algorithm

Intersection	MOTA	IDs	MT	PT	ML
College Avenue and Montezuma Road	57.2%	221	346	98	206
5th Avenue and Laurel St	56.5%	131	138	47	50
Fairmount and University Avenue	55.6%	193	238	56	141
5th Avenue and B St	45.6%	220	233	101	51
Sixth Avenue and Broadway	40.6%	165	182	59	80
Genesee Avenue and Governor Drive	40.1%	293	300	164	265
10th Avenue and J St	34.5%	143	82	41	23
Union St and Ash St	66.6%	80	290	40	67
7th Avenue and Robinson Avenue	29.1%	67	105	30	17
La Jolla and Pearl St	45.6%	74	181	33	37
Overall	48.5%	1587	2095	669	937

Table 8.   Tracking Results for Different Intersections Using Modified SORT Algorithm

Intersection	MOTA	IDs	MT	PT	ML
College Avenue and Montezuma Road	54.4%	87	352	97	201
5th Avenue and Laurel St	53.7%	53	147	39	49
Fairmount and University Avenue	53.2%	134	235	66	134
5th Avenue and B St	44.6%	107	238	100	47
Sixth Avenue and Broadway	36.8%	54	188	59	74
Genesee Avenue and Governor Drive	39.2%	95	310	177	242
10th Avenue and J St	32.3%	69	87	38	21
Union St and Ash St	64.3%	46	295	36	66
7th Avenue and Robinson Avenue	23.7%	27	105	31	66
La Jolla and Pearl St	44%	15	184	32	35
Overall	46.3%	687	2141	675	885

3D Localization

As can be seen from the results, the fourth model (see column headers in Table 9), which uses data augmentation and resnet50 for extracting features, is the best model as it is generalizable across unseen intersections and produces better MSE scores than other models upon noisy data. Noisy data were used to decide the model for estimating footprints as sometimes the bounding boxes used for estimating the position of the object in the frame are not accurate.

Table 9.   MSE (Pixel Level) for Different Models

	Regression model	CNN model	Resnet+FC model	Resnet+FC model with augmentation
Training	58.3	189.8	105.0	121.8
Test	137.8	346.3	181.6	209.5
Noise Test	763.2	890.3	587.9	237.0

Future Model Enhancement

Even though upscaling improved detection for most objects, the detector still does not identify all objects accurately. Recently, researchers from Carnegie Mellon University improved the Mean Average Precision for pedestrian detection from 33.6% to 75.44% by using a larger image resolution (Ruzicka and Franchetti 2018). [105] Perhaps, using a higher resolution camera to record the videos at intersections could help in improving detection for vulnerable road users such as pedestrians and bicycles, which in turn would also improve the tracking results. The tracking module can further be improved by estimating the positions of occluded objects accurately. The 3D localization module can still be improved as the footprint estimation works upon non-occluded objects only as the model needs an image context, which prohibits using occluded objects while outputting trajectories. If the model can be generalized more to estimate footprints for the occluded objects, then the object

trajectories can become more accurate and lead to better safety analysis results. Finally, machine vision models can be enhanced by utilizing more data. Additional data would add more information when models are trained and thus would make them more intelligent.

SAFETY EVALUATION RESULTS

This section is divided into two parts: first, safety analysis was conducted using annotated video data from all ten intersections. From visual analysis of road users, only outcomes of data annotation task were employed. Second, video data from a candidate intersection was utilized to conduct a safety analysis. From visual analysis of road users, the trained object detection and tracking models were employed to automate the trajectory construction process. All three SSMs (TTX_{avg}, RTTC, and PET) were implemented and compared in both parts.

Safety Analysis with Annotated Trajectory Construction

From video data annotation outcomes of the visual analysis step, the potential trajectory intersection points for vehicle–pedestrian, vehicle–bicycle, pedestrian–bicycle, and bicycle–bicycle interactions were identified. Subsequently, SSMs of TTX_{avg}, RTTC, and PET were calculated as presented in the following manner: total interactions, frequency of critical events within safety classes, and comparison of SSMs.

Total Interactions

Figure 8 shows close interactions between vehicles and vulnerable road users for all ten intersections for each SSM less than the threshold of 5 seconds. Interactions with SSM greater than 5 seconds are considered non-critical which is consistent with previous studies. The frequency of interactions per conflict type (i.e., vehicle–bicycle, vehicle–pedestrian, bicycle–pedestrian, and bicycle–bicycle) are illustrated in this figure. It can be seen that vehicle–bicycle and vehicle–pedestrian interactions are more common in all intersections while bicycle–pedestrian and bicycle–bicycle interactions are observed in small numbers.
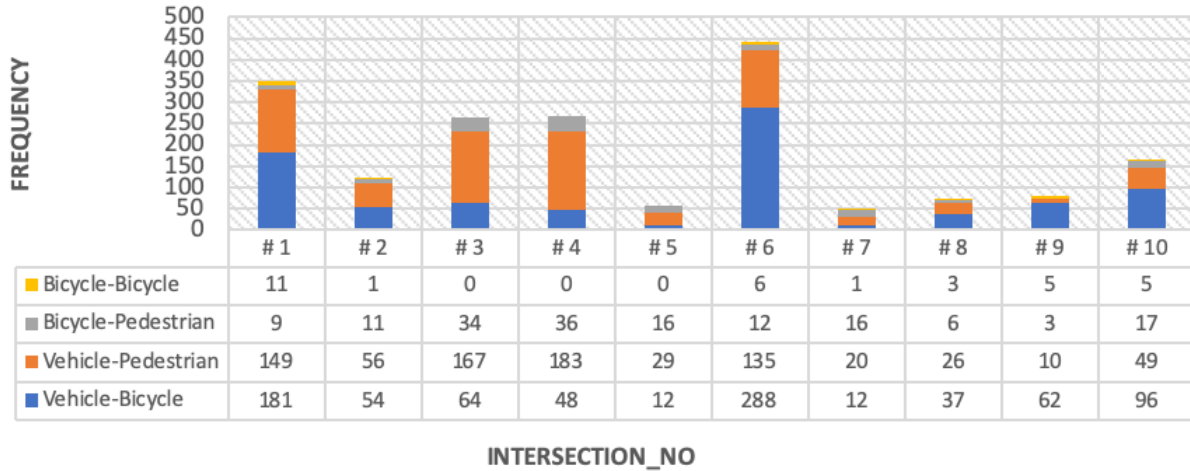
Intersection 6 shows a high number of vehicle–bicycle close encounters, quantified by all the three safety measures, indicating a greater exposure for bicycles at that intersection. This results in high number of critical cases (e.g., 267, 237, and 288 critical cases based on PET, TTX_{avg}, and RTTC, respectively), making intersection 6 a critical intersection for bicycles. Similarly, intersections 3, 4, and 6 show a high number of vehicle–pedestrian close encounters as evident in Figure 8 (e.g., 167, 183, and 135 critical cases based on RTTC), making these three critical intersections for pedestrians.

Looking at intersections 5, 8, and 9, the frequencies of close encounters are similar based on all three safety measures. There are some intersections where the number of close encounters identified by PET is more than those identified by RTTC and vice versa. For instance, the number of close encounters for vehicle–bicycle interactions as identified by RTTC at intersection 1 is 181, while the number estimated by PET is only 60. Similarly, close encounters for vehicle–pedestrian interactions as identified by RTTC at intersection 1 is 149, while the number estimated by PET is only 17. RTTC identifies a large number of vehicle–
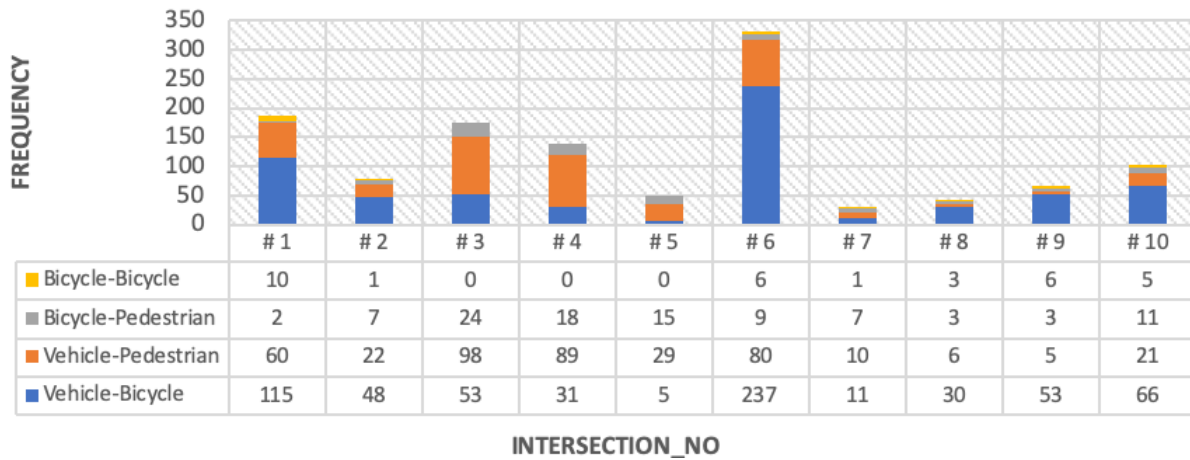
pedestrian and vehicle–bicycle close encounters compared to the other two measures. This is explained by a large number of interactions which have a potential conflict point but do not cross each other's trajectories, thus resulting in a low $RTTC$ value but no PET value. This discrepancy between $RTTC$ and PET results could be attributed to the fact that pedestrians and bicyclists may change their direction of travel more than vehicles. Also, the values of TTX_1 and TTX_2 can be high and close to each other in some cases, resulting in low $RTTC$ values but a high TTX_{avg} value, which in turn could lead to more cases identified by $RTTC$ compared to TTX_{avg}.
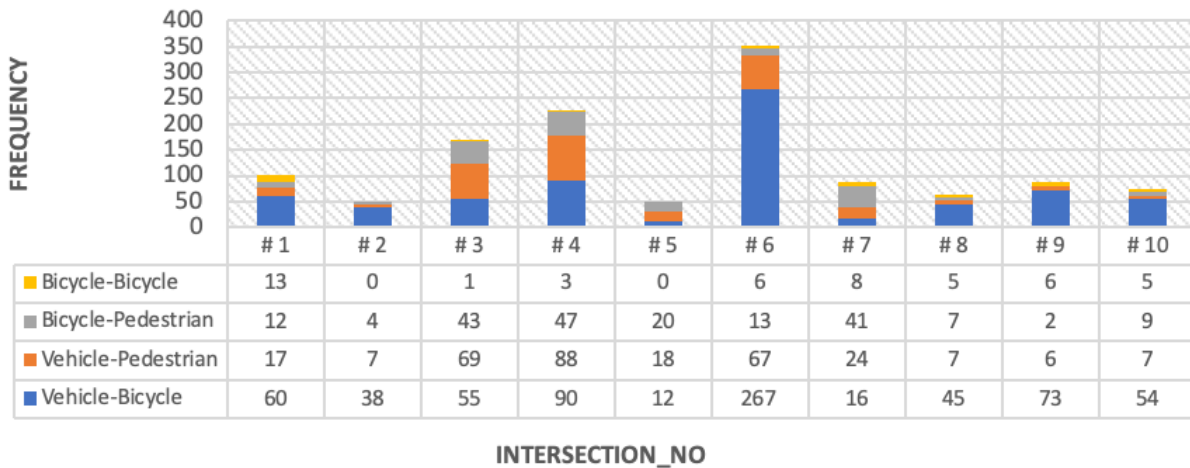
Interactions for RTTC <= 5 seconds

	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
Bicycle-Bicycle	11	1	0	0	0	6	1	3	5	5
Bicycle-Pedestrian	9	11	34	36	16	12	16	6	3	17
Vehicle-Pedestrian	149	56	167	183	29	135	20	26	10	49
Vehicle-Bicycle	181	54	64	48	12	288	12	37	62	96

INTERSECTION_NO

Interactions for TTXavg <= 5 seconds

	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
Bicycle-Bicycle	10	1	0	0	0	6	1	3	6	5
Bicycle-Pedestrian	2	7	24	18	15	9	7	3	3	11
Vehicle-Pedestrian	60	22	98	89	29	80	10	6	5	21
Vehicle-Bicycle	115	48	53	31	5	237	11	30	53	66

INTERSECTION_NO

Interactions for PET <= 5 seconds

	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
Bicycle-Bicycle	13	0	1	3	0	6	8	5	6	5
Bicycle-Pedestrian	12	4	43	47	20	13	41	7	2	9
Vehicle-Pedestrian	17	7	69	88	18	67	24	7	6	7
Vehicle-Bicycle	60	38	55	90	12	267	16	45	73	54

INTERSECTION_NO

Figure 8. Total Number of Interactions between Vehicle–Bicycle, Vehicle– Pedestrian, Bicycle–Pedestrian and Bicycle–Bicycle for All Ten Intersections

It should be noted that if the trajectories of two users do not cross, no value can be calculated. However, a crossing point was assumed if the users get very close to each other (i.e., a threshold of 50 pixels was used). The number of close encounters for vehicle–bicycle interactions as determined by at intersection 4 is 48, while the number estimated by is 90. This is explained by observing several trajectories (i.e., watching in video data) that are in the same direction (one object following the other) or opposite directions. An observed trajectory intersection point results in a value—although the trajectories do not cross, they get very close to each other at one point—while the trajectories in such cases may not result in an value as they are almost parallel.

Frequency of Critical Events Within Safety Classes

Figure 9 shows number of vehicle–bicycle, vehicle–pedestrian, bicycle–pedestrian and bicycle–bicycle interactions within class I (≥0 and < 1.5 seconds), class II (>1.5 and <3.0 seconds), and class III (>3.0 and <5.0 seconds) for safety measures. Unlike PET, which calculates a single value for each interaction, TTX_{avg} and $RTTC$ estimate a series of values for each interaction. In order to determine the frequency of critical events, the minimum value for TTX_{avg} and $RTTC$ from the series of values obtained for each interaction was considered. It was observed that results estimated by the three safety measures for all 10 intersections and each class cannot be generalized across different measures. In some cases, the frequency estimated by PET in a certain class is high, and in other cases, frequencies estimated by TTX_{avg} or $RTTC$ are high.

Class I (i.e., highly critical cases) contains a larger number of events identified by $RTTC$ compared to TTX_{avg} or PET. This was observed for all intersections and each interaction type. The histograms for each intersection and interaction type always show higher $RTTC$ events for class I.
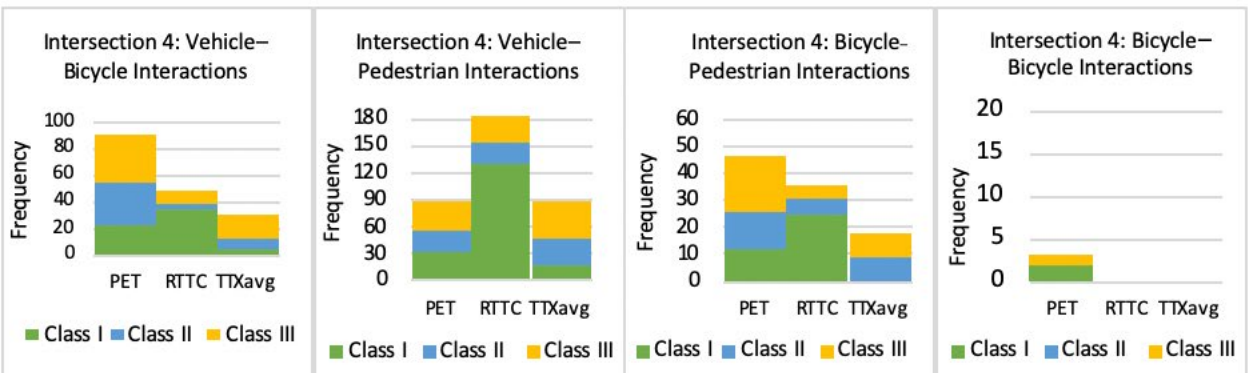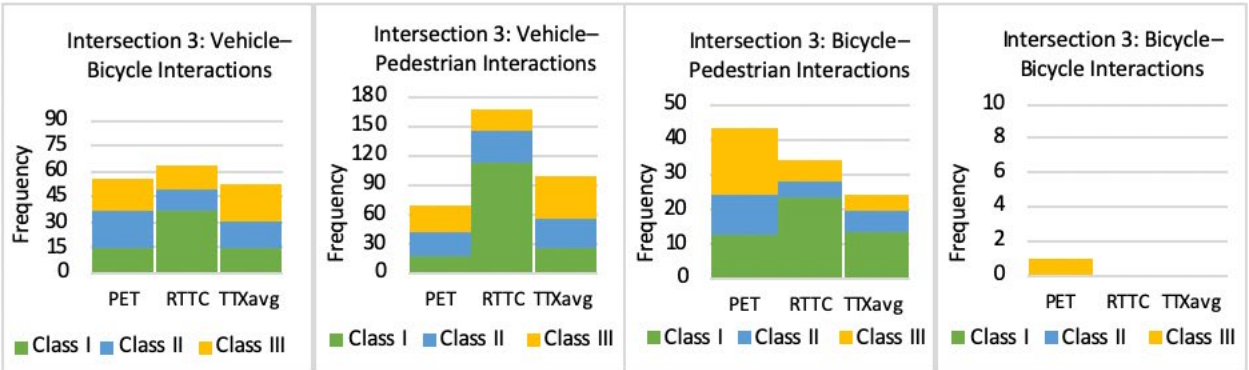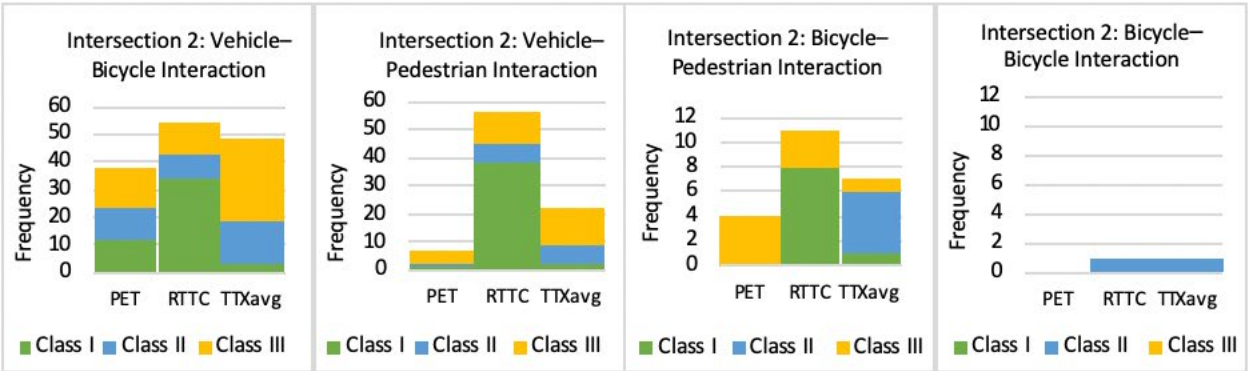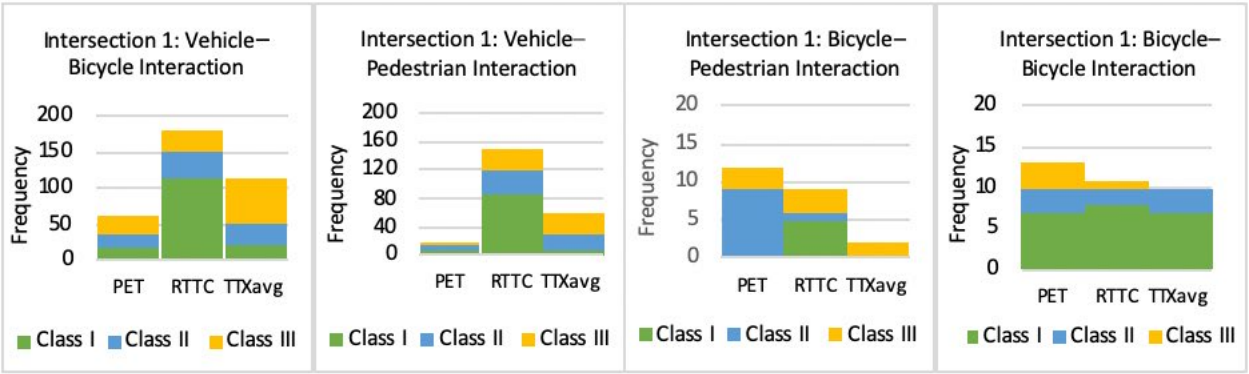
Class II (i.e., moderately critical cases) mostly shows a large number of events estimated by PET; however, some intersections also indicate high number of events estimated by TTX_{avg}. $RTTC$ events in class II are comparatively lower than the other two measures for vehicle–bicycle, bicycle–pedestrian, and bicycle–bicycle interactions. This can be explained by reasoning in two ways. First, as explained in the previous section, objects going in parallel (either in the same or opposite directions) could lead to PET values, but it may not be feasible to calculate $RTTC$ and TTX_{avg} values as trajectories do not collide. Second, in some cases, it was observed that two trajectories cross each other such that the time taken by one of the objects to reach the predicted trajectory intersection point is a low value (say, less than 0.5 seconds) and time taken by the other road user is comparatively high value (say, greater than 3.5 and less than 5.5 seconds). This gives a class II TTX_{avg} value, however, resulting in class III values. The values estimated for PET for such interaction types were also found to satisfy class II range. This is one of the reasons for a dip in $RTTC$ values for class II at all intersections except intersection 1.
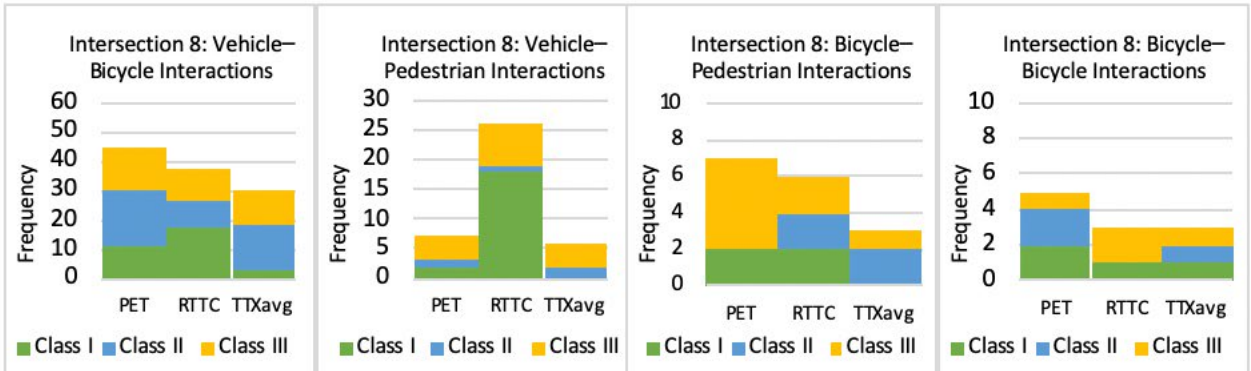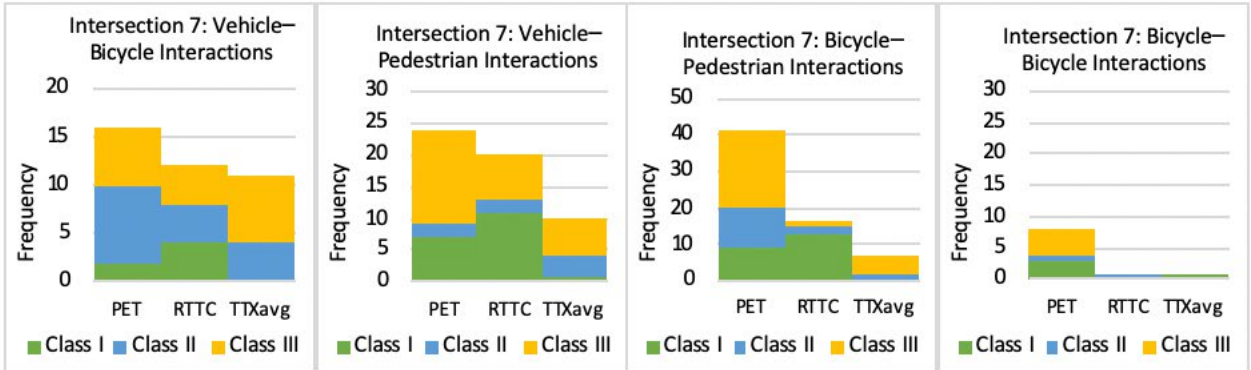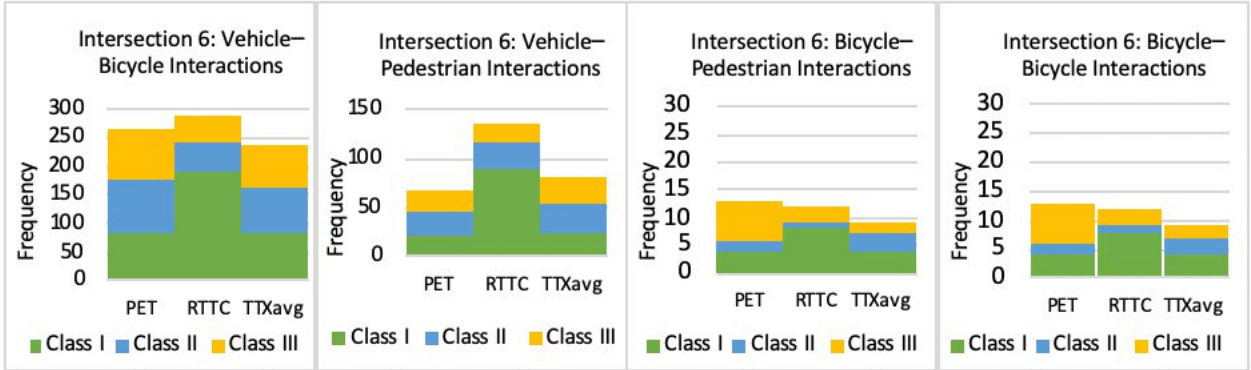
Class II, vehicle–pedestrian interactions, continues to show high $RTTC$ values when compared to the other two measures.

Class III follows the same pattern as class II in estimation of frequency of events by all

three safety measures for vehicle–bicycle, vehicle–pedestrian, bicycle–pedestrian, and bicycle–bicycle.

It should be noted that in several intersections, specific interaction types did not have sufficient number of observations to reveal a clear pattern. Especially, bicycle–bicycle and bicycle–pedestrian interactions had significantly fewer observations than vehicle–bicycle and vehicle–pedestrian interactions at these intersections.

Intersection 1: Vehicle–Bicycle Interaction
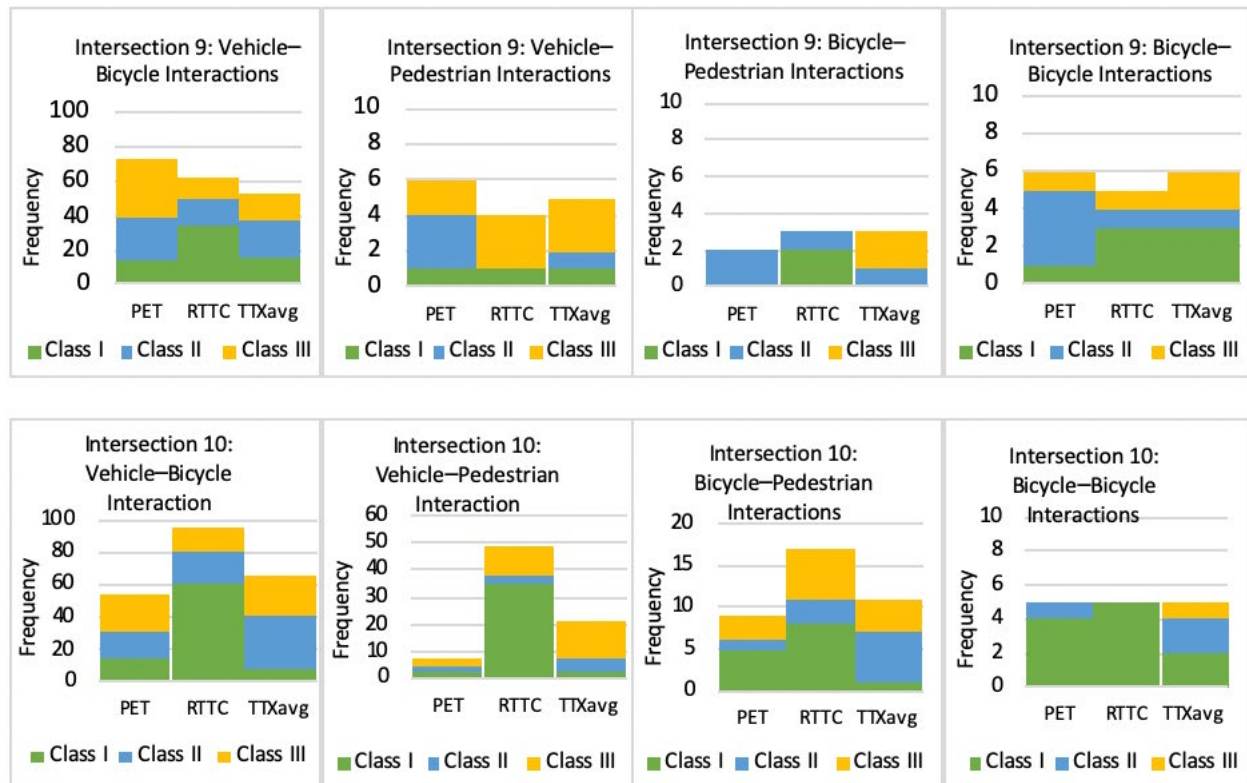
Intersection 1: Vehicle–Pedestrian Interaction

Intersection 1: Bicycle–Pedestrian Interaction

Intersection 1: Bicycle–Bicycle Interaction

Intersection 2: Vehicle–Bicycle Interaction

Intersection 2: Vehicle–Pedestrian Interaction

Intersection 2: Bicycle–Pedestrian Interaction

Intersection 2: Bicycle–Bicycle Interaction

Intersection 3: Vehicle–Bicycle Interactions

Intersection 3: Vehicle–Pedestrian Interactions

Intersection 3: Bicycle–Pedestrian Interactions

Intersection 3: Bicycle–Bicycle Interactions

Intersection 4: Vehicle–Bicycle Interactions

Intersection 4: Vehicle–Pedestrian Interactions

Intersection 4: Bicycle-Pedestrian Interactions

Intersection 4: Bicycle–Bicycle Interactions

Intersection 5: Vehicle–Bicycle Interactions

Intersection 5: Vehicle–Pedestrian Interactions

Intersection 5- Bicycle-Pedestrian Interactions

Intersection 5: Bicycle–Bicycle Interactions

Intersection 6: Vehicle–Bicycle Interactions

Intersection 6: Vehicle–Pedestrian Interactions

Intersection 6: Bicycle–Pedestrian Interactions

Intersection 6: Bicycle–Bicycle Interactions

Intersection 7: Vehicle–Bicycle Interactions

Intersection 7: Vehicle–Pedestrian Interactions

Intersection 7: Bicycle–Pedestrian Interactions

Intersection 7: Bicycle–Bicycle Interactions

Intersection 8: Vehicle–Bicycle Interactions

Intersection 8: Vehicle–Pedestrian Interactions

Intersection 8: Bicycle–Pedestrian Interactions
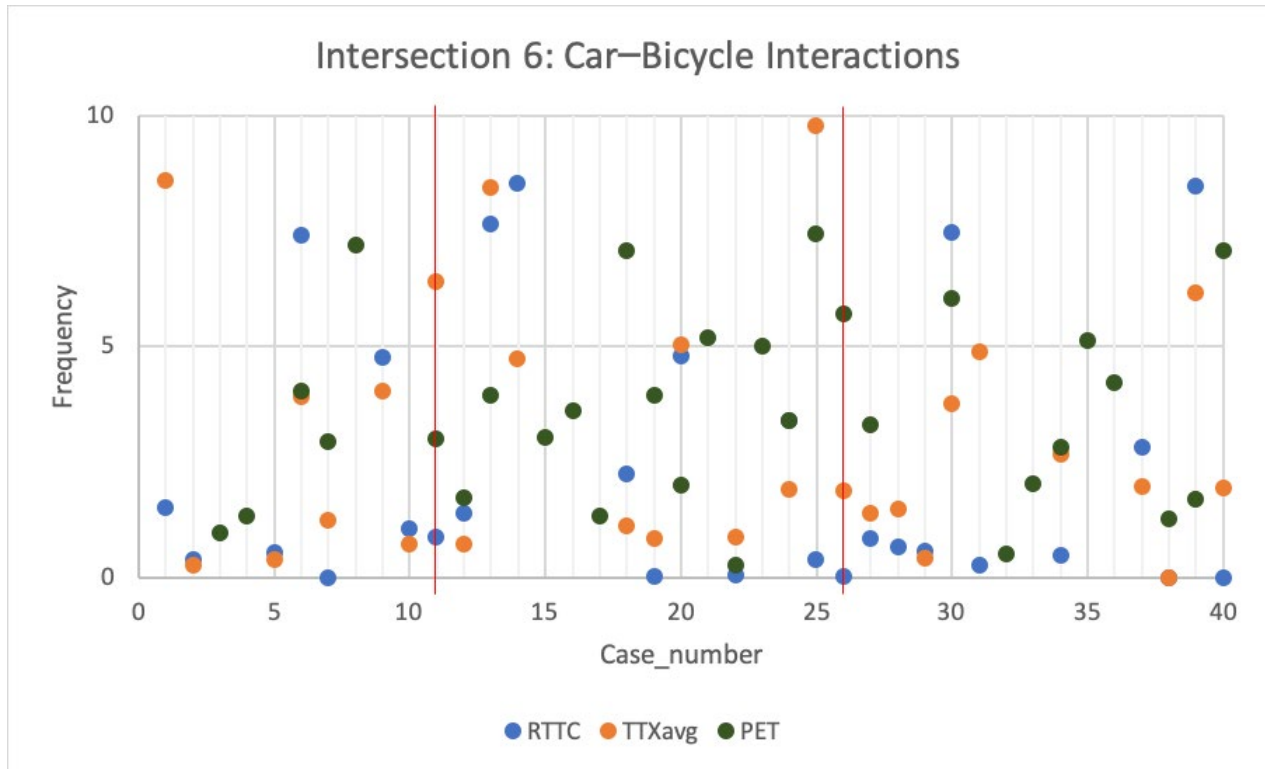
Intersection 8: Bicycle–Bicycle Interactions

Figure 9. **Frequency of Events for TTXavg, RTTC, and PET for Each Safety Class and Each Type of Interaction for All Ten Intersections**

Comparison of Surrogate Safety Measures

Figure 10 is a scatter plot of 40 vehicle–bicycle interactions (out of 747 total interactions) and their corresponding safety measurements at the intersection of Genesee Ave & Governor Dr (intersection 6). As mentioned earlier, the values for TTX_{avg} and RTTC are the minimum values considered from a series of values obtained for each interaction.

The comparison of three safety measures together was made to better see how these measures differ in identifying critical events. Considering the three safety measures as either critical (less than or equal to 5 seconds) or non-critical (greater than 5 seconds), there are different combinations possible when considering all three for an interaction. An interaction can have critical PET and RTTC but non-critical TTX_{avg}. For example, as shown in Figure 10 by a vertical red line, the vehicle–bicycle event (i.e., interaction 11) has critical PET and RTTC but non-critical TTX_{avg}. This is a case where vehicle and bicycle are moving in the same direction, but the encounter is a non-critical event as the TTX_{avg} value is higher than the threshold. Looking at interaction 26, the car–bicycle interaction has critical TTX_{avg} and RTTC, but it has been identified as non-critical based on its PET value. This is an interesting case where a near potential trajectory intersection point is predicted and hence a critical TTX_{avg} and RTTC value is obtained; however, the event was flagged as non-critical based on its PET value. The frequency of different events identified as critical or non-critical based on the three measures is presented in Table 10. For example, 48 interactions were flagged as critical based on their PET values while they were identified as non-critical

events based on both $RTTC$ and TTX_{avg} values. There are also cases for which SSM values were not available (N/A); for example, there is no $RTTC$ value for objects going in parallel.

Figure 10. Comparison of All Three Safety Indicators for Vehicle–Bicycle Interaction for Intersection

Table 10. Frequency of Critical and Non-Critical Combinations of SSMs

Critical	Non-Critical	N/A	Frequency
PET	RTTC, TTXavg	-	48
PET, RTTC	TTXavg	-	22
PET, TTXavg	RTTC	-	5
RTTC, TTXavg	PET	-	51
RTTC	PET, TTXavg	-	5
RTTC	TTXavg	PET	39
TTXavg	RTTC	PET	4
-	TTXavg, RTTC	PET	203
TTXavg, RTTC		PET	63
TTXavg	PET, RTTC	-	6
-	PET	RTTC, TTXavg	90
PET	-	RTTC, TTXavg	85
RTTC, TTXavg, PET	-	-	108
-	RTTC, TTXavg, PET	-	18

Table 11. One-on-One Comparison of SSMs for Different Interaction Types

VEHICLE–BICYCLE INTERACTIONS

RTTC	I	II	III
III	0	116 ~14%	30 ~4%
II	24 ~3%	106 ~13%	21 ~3%
I	131 ~16%	339 ~42%	41 ~5%
	I	II	III
	TTXavg		

RTTC	I	II	III
III	7 ~2%	25 ~9%	21 ~7%
II	13 ~5%	19 ~7%	16 ~6%
I	56 ~20%	70 ~25%	54 ~19%
	I	II	III
	PET		

TTXavg	I	II	III
III	8 ~2%	12 ~4%	19 ~6%
II	46 ~14%	68 ~21%	72 ~23%
I	37 ~12%	41 ~13%	16 ~5%
	I	II	III
	PET		

VEHICLE–PEDESTRIAN INTERACTIONS

RTTC	I	II	III
III	0	80 ~11%	39 ~5%
II	13 ~2%	89 ~12%	22 ~3%
I	75 ~10%	363 ~48%	68 ~9%
	I	II	III
	TTXavg		

RTTC	I	II	III
III	7 6~%	4 ~4%	10 ~9%
II	5 ~5%	3 ~3%	6 ~6%
I	24 ~22%	28 ~26%	21 ~19%
	I	II	III
	PET		

TTXavg	I	II	III
III	10 ~8%	7 ~6%	10 ~8%
II	20 ~16%	18 ~14%	24 ~19%
I	11 ~9%	13 ~10%	12 ~10%
	I	II	III
	PET		

BICYCLE–PEDESTRIAN INTERACTIONS

RTTC	I	II	III
III	0	20 ~13%	9 ~6%
II	6 ~4%	14 ~9%	2 ~1%
I	19 ~13%	68 ~45%	14 ~9%
	I	II	III
	TTXavg		

RTTC	I	II	III
III	5 ~7%	5 ~7%	5 ~7%
II	4 ~5%	4 ~5%	6 ~8%
I	11 ~15%	14 ~19%	21 ~28%
	I	II	III
	PET		

TTXavg	I	II	III
III	2 ~2%	3 ~4%	6 ~7%
II	14 ~17%	16 ~20%	22 ~27%
I	4 ~5%	7 ~8%	8 ~10%
	I	II	III
	PET		

BICYCLE–BICYCLE INTERACTIONS

RTTC	I	II	III
III	0	3 ~16%	1 ~16%
II	3 ~16%	3 ~16%	0
I	12 ~16%	8 ~16%	2 ~16%
	I	II	III
	TTXavg		

RTTC	I	II	III
III	0	3 ~11%	1 ~4%
II	2 ~7%	2 ~7%	0
I	14 ~50%	6 ~21%	0
	I	II	III
	PET		

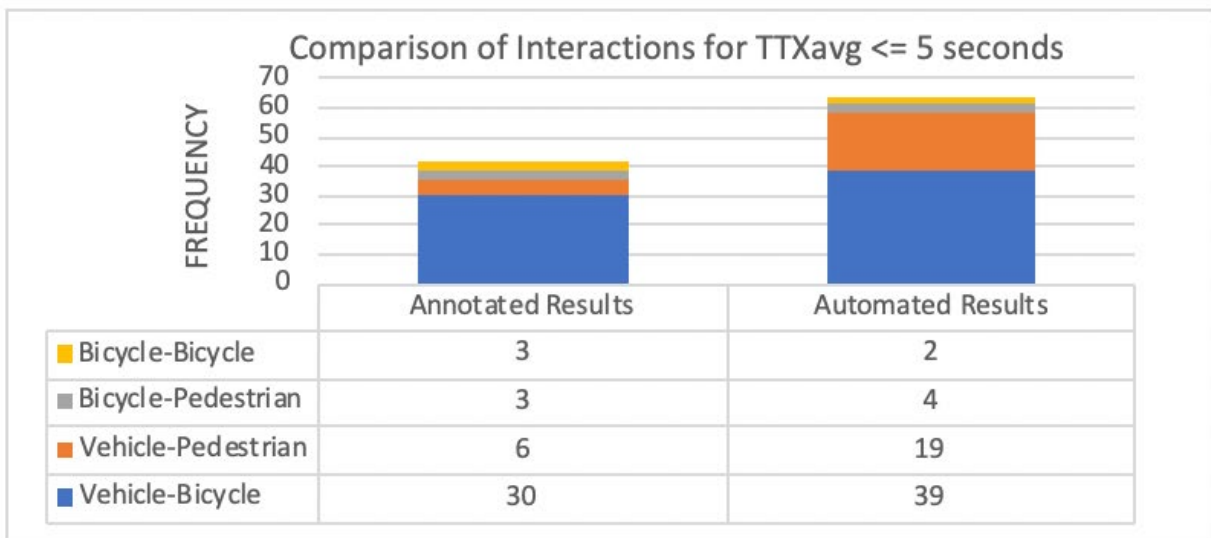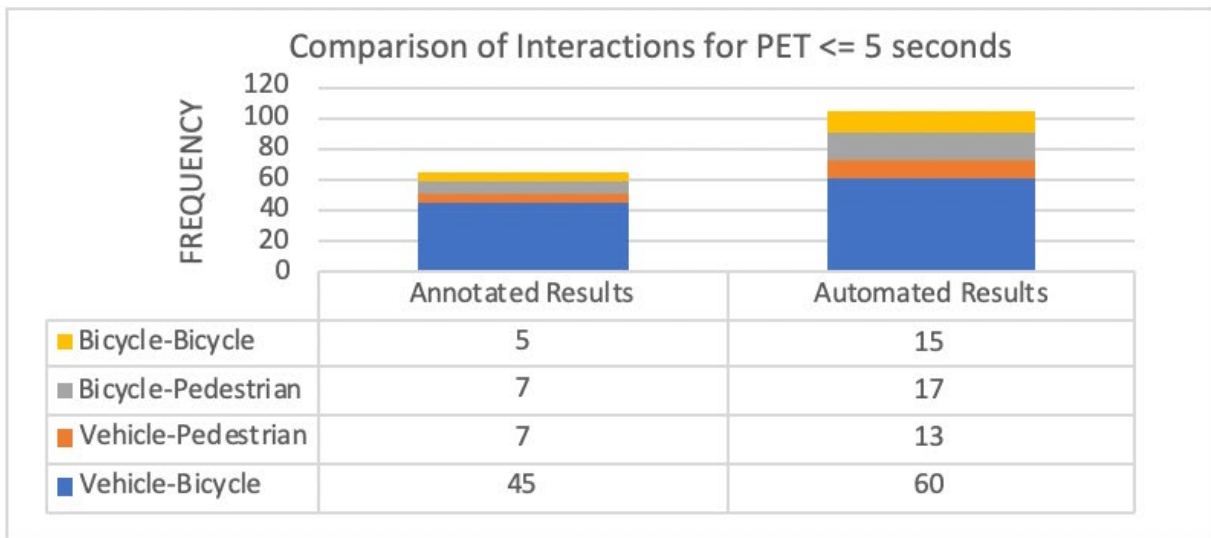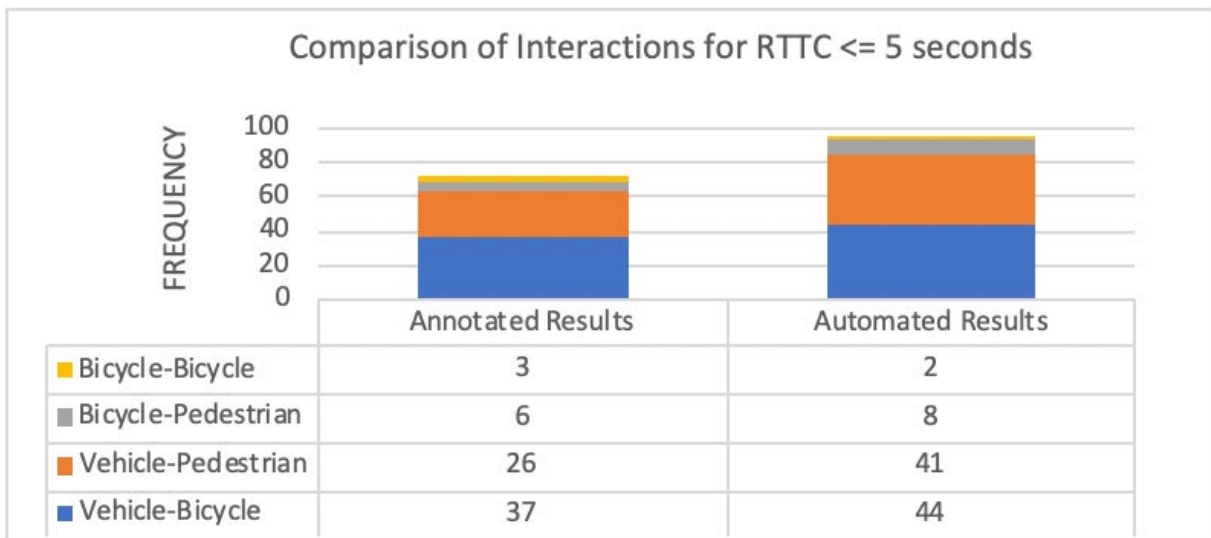| TTXavg	I	II	III
III	1 ~4%	1 ~4%	0
II	4 ~14%	6 ~21%	1 ~4%
I	11 ~40%	4 ~14%	0
	I	II	III
	PET		

When the values for all three measures (TTX_{avg}, $RTTC$, and PET) for particular interactions are below the critical value (<5 seconds) then such events can be flagged as most critical events as all three SSMs are in agreement. For this particular intersection as shown in Table 10, total of 108 vehicle–bicycle interactions out of 747 were identified as critical based on all thee SSMs. Similar analyses can be done for other intersections and interaction types.

Table 11 shows comparison of two safety measures at a time ($RTTC$ vs. TTX_{avg}; $RTTC$ vs. PET; and TTX_{avg} vs. PET) considering class I, class II, and class III events for different interaction types. In the form of several matrices, this table shows frequency and percentage of agreement of two safety measures using data from all 10 intersections combined. For class I, class II, and class III, the frequency and percentage of events that are in agreement with two safety measures (i.e., same outputs by both measures) at a time was calculated. It is found that TTX_{avg} and PET performed more similarly than other safety measure combinations (i.e., for vehicle–bicycle, bicycle–pedestrian, and bicycle–bicycle interactions, the same results were obtained by and in 39%, 32%, and 61% of the total cases, respectively). For vehicle–pedestrian interactions, $RTTC$ and PET performed more similarly (i.e., both measures led to the same result in 34% of the total cases).

Safety Analysis with Automated Trajectory Construction

The results presented in the previous section were obtained using annotated data, which means all objects (e.g., pedestrians, bicyclists, and vehicles) were tagged using the Vatic software to build object trajectories. The decision support system developed in this project is capable of automatic safety evaluation by utilizing the trained detection and tracking models that were presented earlier. Therefore, object trajectories are constructed automatically. The idea is to use raw video files as inputs and feed them into the decision support system to contribute to safety evaluation results. A comparison was made to observe how well the automated safety evaluation can perform. Data from a candidate intersection (intersection 8) was used for this part of the analysis.

Figure 11 shows comparison of safety evaluation results obtained using annotated data and automatically constructed trajectories. It was observed that in general the frequencies of critical events in both approaches (i.e., annotated vs. automated) are close. For example, 37 events were identified as critical based on $RTTC$ in the "annotated" approach whereas 44 events were flagged as critical in the "automated" approach. However, in some cases, the automated approach was not able to provide a good estimation. Looking at vehicle–pedestrian events, TTX_{avg} identified 6 critical cases in the annotated approach, but it identified 19 critical cases in the automated approach. As discussed in the future model enhancement section earlier, there are several ways to improve the performance of machine vision models, which in turn would enhance the performance and reliability of the automated safety evaluation.

Comparison of Interactions for RTTC <= 5 seconds

	Annotated Results	Automated Results
Bicycle-Bicycle	3	2
Bicycle-Pedestrian	6	8
Vehicle-Pedestrian	26	41
Vehicle-Bicycle	37	44

Comparison of Interactions for PET <= 5 seconds

	Annotated Results	Automated Results
Bicycle-Bicycle	5	15
Bicycle-Pedestrian	7	17
Vehicle-Pedestrian	7	13
Vehicle-Bicycle	45	60

Comparison of Interactions for TTXavg <= 5 seconds

	Annotated Results	Automated Results
Bicycle-Bicycle	3	2
Bicycle-Pedestrian	3	4
Vehicle-Pedestrian	6	19
Vehicle-Bicycle	30	39

Figure 11. Comparison of Annotated and Automated Results

Figure 12 shows the number of vehicle–bicycle, vehicle–pedestrian, bicycle–pedestrian,

and bicycle–bicycle interactions within class I (≥0 and <1.5 seconds), class II (>1.5 and <3.0 seconds), and class III (>3.0 and <5.0 seconds) for safety measures TTX_{avg}, RTTC, and PET obtained from the "annotated" and "automated" safety analysis approaches. As shown in this figure, in general, the number of encounters estimated by the automated approach is higher than those by the annotated approach for all classes and SSMs. This figure also shows that for conflict types with more observations, the results from the automated approach are closer to those of the annotated approach. For example, the histograms of vehicle–bicycle interactions have a higher number of observations in this figure (comparing to the other interaction types) and it can be seen that the histograms of the automated approach are very similar to the histograms of the annotated approach. This is a positive indication that suggests more observations could potentially enhance the performance of the automated results.
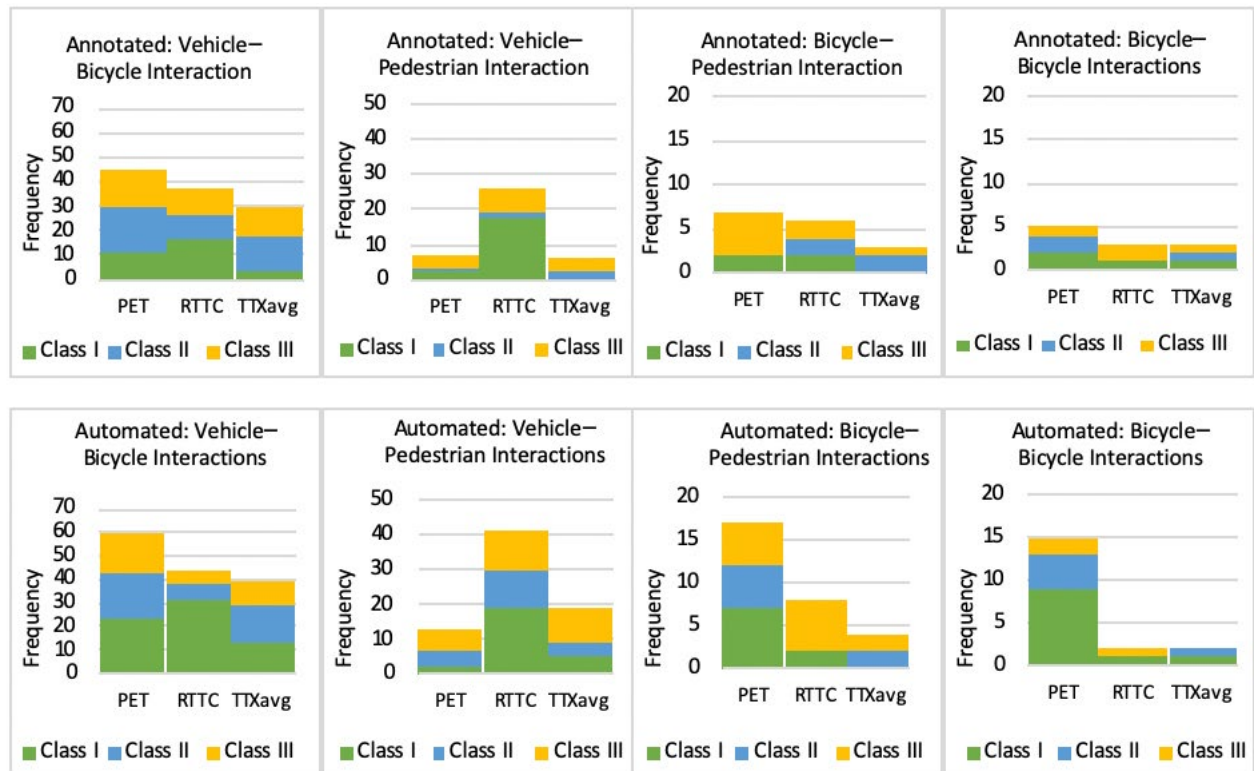
Figure 12. Comparison of Safety Measures for Annotated and Automated Results

V. CONCLUSIONS

The traditional safety assessment approach requires a long period of time since roadway crashes are rare events. Even a long period of time may not produce enough data, especially if an infrequent crash type is being studied. In addition, changes such as geometric design alterations, demand increase, and signal timing adjustments could potentially impact the results of safety evaluations. Therefore, the traditional approach is considered a reactive approach since a significant number of crashes need to occur before a preventative action is considered. This reactivity reduces the potential for an examination of the safety effects of a recently implemented safety countermeasure such as a traffic calming strategies or advanced dilemma zone protection systems. An effective solution is to adopt SSMs to proactively assess safety.

In this study, the two most widely used SSMs, TTC and PET, as well as a recent variant form of TTC, RTTC, were estimated using real-world video data collected at ten signalized intersections in the city of San Diego, California. Prediction of potential object trajectory intersection points was performed to estimate TTC for every interacting object, and the average of TTC for every two objects in critical situations was calculated. PET values were estimated by observing potential intersection points, and frequency of the critical events were estimated based on three critical classes. According to the results, although RTTC provided useful information regarding the relative distance between objects in time, it was found that in certain conditions where objects are far from each other, the interaction between the objects was incorrectly flagged as critical based on a small RTTC. Comparison of PET, TTC, and RTTC for different critical classes also showed that several interactions were identified as critical using one SSM but non-critical using a different SSM. These findings suggest that safety evaluations should not solely rely on a single SSM, and instead a combination of different SSMs should be considered to ensure the reliability of evaluations.

Furthermore, this project built a decision support system that utilizes outcomes of machine vision models and compares the performance of the SSMs for the purpose of evaluating pedestrian and bicyclist safety at signalized intersections. Data from a candidate intersection were used to train object detection and tracking models. It was shown that the decision support system can automatically assess safety of vulnerable road users using different SSMs in short periods of time with relatively decent performance. Comparing to the ground truth, the decision support system identified similar number of critical events for certain interaction types (e.g., 44 critical vehicle–bicycle events were identified by the system comparing to 37 ground truth critical events). This development provides an opportunity to proactively and automatically answer the question of whether intersections with certain treatments are safer than similar intersections without the treatments or whether safety has improved after implementation of a certain countermeasure.

ABBREVIATIONS AND ACRONYMS

SSM	Surrogate Safety Measure
FARS	Fatality Analysis Reporting System
NHTSA	National Highway Transportation Safety Administration
TCT	Traffic Conflict Technique
TTC	Time-To-Collision
PET	Post-Encroachment Time
MTTC	Modified Time To Collision
RTTC	Relative Time To Collision
TET	Time Exposed Time To Collision
TIT	Time Integrated Time To Collision
MI	Mixed Index
TTX	Time To Intersection
CNN	Convoluted Neural Networks
mAP	Mean Average Precision
IOU	Intersection over Union
AP	Average Precision
RPN	Region Proposal Network
VOC	Visual Object Classes
MOTA	Multiple Object Tracking Accuracy
MT	Mostly Tracked
PT	Partially Tracked
ML	Mostly Lost
SORT	Simple Online Realtime Tracking
MSE	Mean Squared Error
TIP	Trajectory Intersection Point

ENDNOTES

1. Jahangiri, Arash, Mahdie Hasani, Ipek N. Sener, Sirajum Munira, Justin Owens, Bruce Appleyard, Sherry Ryan, Shawn M. Turner, and Sahar Ghanipoor Machiani. 2019. "Data Mining to Improve Planning for Pedestrian and Bicyclist Safety." Safe-D National UTC. https://www.vtti.vt.edu/utc/safe-d/index.php/projects/data-mining-to-improve-planning-for-pedestrian-and-bicyclist-safety/.

2. Chen, Peng, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. 2017. "Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos." Journal of Advanced Transportation 2017. https://doi.org/10.1155/2017/5202150. [Accessed: 02-Oct-2018].

3. Tarko, Andrew, Gary Davis, Nicolas Saunier, Tarek Sayed, and Simon Washington. 2009. "Surrogate Measures of Safety - A White Paper." Transportation Research Board, Washington, DC.

4. Gettman, Douglas, and Larry Head. 2003. "Surrogate Safety Measures from Traffic Simulation Models." Transportation Research Record 1840 (1): 104–115.

5. Chen, Peng, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. 2017. "Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos." Journal of Advanced Transportation 2017. https://doi.org/10.1155/2017/5202150. [Accessed: 02-Oct-2018].

6. Gettman, Douglas, and Larry Head. 2003. "Surrogate Safety Measures from Traffic Simulation Models." Transportation Research Record 1840 (1): 104–115.

7. Allen, Brian L, B Tom Shin, and Peter J Cooper. 1978. "Analysis of Traffic Conflicts and Collisions." Transportation Research Record, no. 667: 67–74.

8. Tarko, Andrew, Gary Davis, Nicolas Saunier, Tarek Sayed, and Simon Washington. 2009. "Surrogate Measures of Safety - A White Paper." Transportation Research Board, Washington, DC.

9. Ibid.

10. Mahmud, S. M. Sohel, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. 2017. "Application of Proximal Surrogate Indicators for Safety Evaluation: A Review of Recent Developments and Research Needs." IATSS Research 41 (4): 153–63. https://doi.org/10.1016/j.iatssr.2017.02.001.

11. Zheng, Lai, Karim Ismail, and Xianghai Meng. 2014. "Traffic Conflict Techniques for Road Safety Analysis: Open Questions and Some Insights." Canadian Journal of Civil Engineering 41 (7): 633–41. https://doi.org/10.1139/cjce-2013-0558.

12. Hayward, J. C. 1972. "Near-Miss Determination through Use of a Scale of Danger."

Highway Research Record 384: 24–34.

13. Allen, Brian L, B Tom Shin, and Peter J Cooper. 1978. "Analysis of Traffic Conflicts and Collisions." Transportation Research Record, no. 667: 67–74.

14. Yang, Hong, Kaan Ozbay, and Bekir Bartin. 2010. "Application of Simulation-Based Traffic Conflict Analysis for Highway Safety Evaluation." In Proceedings of the 12th WCTR, Lisbon, Portugal. Vol. 4.

15. Mahmud, S. M. Sohel, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. 2017. "Application of Proximal Surrogate Indicators for Safety Evaluation: A Review of Recent Developments and Research Needs." IATSS Research 41 (4): 153–63. https://doi.org/10.1016/j.iatssr.2017.02.001.

16. Ibid.

17. Van Der Horst, Ara. 1990. "A Time-Based Analysis of Road User Behaviour in Normal and Critical Encounters." PhD Dissertation, Delft, Netherlands: Delft University of Technology. https://trid.trb.org/view/355354,.

18. Gettman, Douglas, and Larry Head. 2003. "Surrogate Safety Measures from Traffic Simulation Models." Transportation Research Record 1840 (1): 104–115.

19. Ibid.

20. Ibid.

21. Vogel, Katja. 2003. "A Comparison of Headway and Time to Collision as Safety Indicators." Accident Analysis & Prevention 35 (3): 427–33. https://doi.org/10.1016/S0001-4575(02)00022-2.

22. Ozbay, Kaan, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. 2008. "Derivation and Validation of New Simulation-Based Surrogate Safety Measure." Transportation Research Record: Journal of the Transportation

23. Ibid.

24. Laureshyn, Aliaksei, Åse Svensson, and Christer Hydén. 2010. "Evaluation of Traffic Safety, Based on Micro-Level Behavioural Data: Theoretical Framework and First Implementation." Accident Analysis & Prevention 42 (6): 1637–1646.

25. Tarko, Andrew P. 2012. "Use of Crash Surrogates and Exceedance Statistics to Estimate Road Safety." Accident Analysis & Prevention 45: 230–40. https://doi.org/10.1016/j.aap.2011.07.008.

26. Zheng, Lai, Karim Ismail, and Xianghai Meng. 2014. "Traffic Conflict Techniques for Road Safety Analysis: Open Questions and Some Insights." Canadian Journal of Civil

Engineering 41 (7): 633–41. https://doi.org/10.1139/cjce-2013-0558.

27. Lu, Guangquan, Miaomiao Liu, Yunpeng Wang, and Guizhen Yu. 2012. "Quantifying the Severity of Traffic Conflict by Assuming Moving Elements as Rectangles at Intersection." Procedia - Social and Behavioral Sciences, 8th International Conference on Traffic and Transportation Studies (ICTTS 2012), 43: 255–64. https://doi.org/10.1016/j.sbspro.2012.04.098.

28. Ghanipoor Machiani, Sahar, and Montasir Abbas. 2016. "Safety Surrogate Histograms (SSH): A Novel Real-Time Safety Assessment of Dilemma Zone Related Conflicts at Signalized Intersections." Accident Analysis & Prevention 96: 361–70. https://doi.org/10.1016/j.aap.2015.04.024.

29. Allen, Brian L, B Tom Shin, and Peter J Cooper. 1978. "Analysis of Traffic Conflicts and Collisions." Transportation Research Record, no. 667: 67–74.

30. Barceló Bugeda, Jaime, André-Gilles Dumont, Lídia Montero Mercadé, Josep Perarnau, and Alexandre Torday. 2003. "Safety Indicators for Microsimulation-Based Assessments." In Transportation Research Board 82nd Annual Meeting, 1–18. TRB.

31. Svensson, Ase. 1998. "A Method for Analysing the Traffic Process in a Safety Perspective." PhD Dissertation, Lund, Sweden: University of Lund, Lund Institute of Technology, Department of Traffic Planning and Engineering. https://portal.research.lu.se/portal/files/4456913/1653512.pdf.

32. Ibid.

33. Ibid.

34. Ozbay, Kaan, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. 2008. "Derivation and Validation of New Simulation-Based Surrogate Safety Measure." Transportation Research Record: Journal of the Transportation Research Board 2083 (1): 105–13. https://doi.org/10.3141/2083-12.

35. Hayward, J. C. 1972. "Near-Miss Determination through Use of a Scale of Danger." Highway Research Record 384: 24–34.

36. Nadimi, Navid, Hamid Behbahani, and HamidReza Shahbazi. 2016. "Calibration and Validation of a New Time-Based Surrogate Safety Measure Using Fuzzy Inference System." Journal of Traffic and Transportation Engineering (English Edition) 3 (1): 51–58. https://doi.org/10.1016/j.jtte.2015.09.004.

37. Ozbay, Kaan, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. 2008. "Derivation and Validation of New Simulation-Based Surrogate Safety Measure." Transportation Research Record: Journal of the Transportation Research Board 2083 (1): 105–13. https://doi.org/10.3141/2083-12.

38. Ibid.

39. Behbahani, Hamid, and Navid Nadimi. 2015. "A Framework for Applying Surrogate Safety Measures for Side Swipe Collisions." International Journal for Traffic & Transport Engineering 5 (4): 371–83. https://doi.org/10.7708/ijtte.2015.5(4).03.

40. Saffarzadeh, Mahmoud, Navid Nadimi, Saber Naseralavi, and Amir Reza Mamdoohi. 2013. "A General Formulation for Time-to-Collision Safety Indicator." Proceedings of the Institution of Civil Engineers - Transport 166 (5): 294–304. https://doi.org/10.1680/tran.11.00031.

41. Behbahani, Hamid, and Navid Nadimi. 2015. "A Framework for Applying Surrogate Safety Measures for Side Swipe Collisions." International Journal for Traffic & Transport Engineering 5 (4): 371–83. https://doi.org/10.7708/ijtte.2015.5(4).03.

42. Ibid.

43. Chen, Peng, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. 2017. "Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos." Journal of Advanced Transportation 2017. https://doi.org/10.1155/2017/5202150.

44. Laureshyn, Aliaksei, Åse Svensson, and Christer Hydén. 2010. "Evaluation of Traffic Safety, Based on Micro-Level Behavioural Data: Theoretical Framework and First Implementation." Accident Analysis & Prevention 42 (6): 1637–1646.

45. Behbahani, Hamid, and Navid Nadimi. 2015. "A Framework for Applying Surrogate Safety Measures for Side Swipe Collisions." International Journal for Traffic & Transport Engineering 5 (4): 371–83. https://doi.org/10.7708/ijtte.2015.5(4).03.

46. Chen, Peng, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. 2017. "Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos." Journal of Advanced Transportation 2017. https://doi.org/10.1155/2017/5202150.

47. Zheng, Lai, Karim Ismail, and Xianghai Meng. 2014. "Traffic Conflict Techniques for Road Safety Analysis: Open Questions and Some Insights." Canadian Journal of Civil Engineering 41 (7): 633–41. https://doi.org/10.1139/cjce-2013-0558.

48. Peesapati, Lakshmi N., Michael P. Hunter, and Michael O. Rodgers. 2013. "Evaluation of Postencroachment Time as Surrogate for Opposing Left-Turn Crashes." Transportation Research Record: Journal of the Transportation Research Board 2386 (1): 42–51. https://doi.org/10.3141/2386-06.

49. Yang, Hong, Kaan Ozbay, and Bekir Bartin. 2010. "Application of Simulation-Based Traffic Conflict Analysis for Highway Safety Evaluation." In Proceedings of the 12th WCTR, Lisbon, Portugal. Vol. 4.

50. Minderhoud, Michiel M., and Piet H. L. Bovy. 2001. "Extended Time-to-Collision Measures for Road Traffic Safety Assessment." Accident Analysis & Prevention 33 (1): 89–97. https://doi.org/10.1016/S0001-4575(00)00019-1.

51. Yang, Hong, Kaan Ozbay, and Bekir Bartin. 2010. "Application of Simulation-Based Traffic Conflict Analysis for Highway Safety Evaluation." In Proceedings of the 12th WCTR, Lisbon, Portugal. Vol. 4.

52. Minderhoud, Michiel M., and Piet H. L. Bovy. 2001. "Extended Time-to-Collision Measures for Road Traffic Safety Assessment." Accident Analysis & Prevention 33 (1): 89–97. https://doi.org/10.1016/S0001-4575(00)00019-1.

53. Behbahani, Hamid, and Navid Nadimi. 2015. "A Framework for Applying Surrogate Safety Measures for Side Swipe Collisions." International Journal for Traffic & Transport Engineering 5 (4): 371–83. https://doi.org/10.7708/ijtte.2015.5(4).03.

54. Zheng, Lai, Karim Ismail, and Xianghai Meng. 2014. "Traffic Conflict Techniques for Road Safety Analysis: Open Questions and Some Insights." Canadian Journal of Civil Engineering 41 (7): 633–41. https://doi.org/10.1139/cjce-2013-0558.

55. Peesapati, Lakshmi N., Michael P. Hunter, and Michael O. Rodgers. 2013. "Evaluation of Postencroachment Time as Surrogate for Opposing Left-Turn Crashes." Transportation Research Record: Journal of the Transportation Research Board 2386 (1): 42–51. https://doi.org/10.3141/2386-06.

56. Nadimi, Navid, Hamid Behbahani, and HamidReza Shahbazi. 2016. "Calibration and Validation of a New Time-Based Surrogate Safety Measure Using Fuzzy Inference System." Journal of Traffic and Transportation Engineering (English Edition) 3 (1): 51–58. https://doi.org/10.1016/j.jtte.2015.09.004.

57. Nadimi, Navid, Hamid Behbahani, and HamidReza Shahbazi. 2016. "Calibration and Validation of a New Time-Based Surrogate Safety Measure Using Fuzzy Inference System." Journal of Traffic and Transportation Engineering (English Edition) 3 (1): 51–58. https://doi.org/10.1016/j.jtte.2015.09.004.

58. Shirazi, Mohammad Shokrolah, and Brendan Tran Morris. 2017. "Investigation of Safety Analysis Methods Using Computer Vision Techniques." Journal of Electronic Imaging 26 (5): 051404. https://doi.org/10.1117/1.JEI.26.5.051404.

59. Shirazi, Mohammad Shokrolah, and Brendan Tran Morris. 2017. "Investigation of Safety Analysis Methods Using Computer Vision Techniques." Journal of Electronic Imaging 26 (5): 051404. https://doi.org/10.1117/1.JEI.26.5.051404.

60. Vondrick, Carl, Donald Patterson, and Deva Ramanan. 2013. "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling." International Journal of Computer Vision 101 (1): 184–204. https://doi.org/10.1007/s11263-012-0564-1.

61. Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable Road Users." MS Thesis, San Diego: San Diego State Universit.

62. Vondrick, Carl, Donald Patterson, and Deva Ramanan. 2013. "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling." International Journal of Computer Vision 101 (1): 184–204. https://doi.org/10.1007/s11263-012-0564-1.

63. Vondrick, Carl, Donald Patterson, and Deva Ramanan. 2013. "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling." International Journal of Computer Vision 101 (1): 184–204. https://doi.org/10.1007/s11263-012-0564-1.

64. Vondrick, Carl. 2012. Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces. HTML. https://github.com/cvondrick/vatic.

65. Rosebrock, Adrian. 2016. "Intersection Over Union (IoU) for Object Detection." PyImageSearch (blog). 2016. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/..

66. Tan, Ren Jie. 2019. "Breaking Down Mean Average Precision (MAP)." Medium (blog). 2019. https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52.

67. Hui, Jonathan. 2018. "MAP (Mean Average Precision) for Object Detection." Medium (blog). 2018. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

68. Sahoo, Pallavi. 2017. "Mean Average Precision." Medium (blog). 2017. https://medium.com/@pds.bangalore/mean-average-precision-abd77d0b9a7e.

69. Zając, Zygmunt. 2012. "What You Wanted to Know about Mean Average Precision." FastML (blog). 2012. http://fastml.com/what-you-wanted-to-know-about-mean-average-precision/

70. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In Advances in Neural Information Processing Systems 28, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 91–99. Curran Associates, Inc. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-ne works.pdf.

71. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In Advances in Neural Information Processing Systems 28, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 91–99. Curran Associates, Inc. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-

networks.pdf.

72. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

73. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In Advances in Neural Information Processing Systems 28, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 91–99. Curran Associates, Inc. http://papers.nips. cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

74. Chollet, Francois. 2015. Deep Learning for Humans. Python. Keras. https://github. com/keras-team/keras F. Chollet, Deep Learning for humans. Contribute to keras-team/keras development by creating an account on GitHub. Keras, 2019.

75. Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. "Tensorflow: A System for Large-Scale Machine Learning." In 12th USENIX Symposium on Operating Systems Design and Implementation, 265–283.

76. Van Rossum, Guido, and Fred L. Drake. 2011. The Python Language Reference Manual. Network Theory Ltd.

77. Bardool, Kevin. 2017. Keras Implementation of Faster R-CNN. Python. https://github. com/kbardool/keras-frcnn.

78. Heindl, Christoph. 2017. Benchmark Multiple Object Trackers (MOT) in Python. Python. https://github.com/cheind/py-motmetrics.

79. Bernardin, Keni, Alexander Elbs, and Rainer Stiefelhagen. 2006. "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment." In Sixth IEEE International Workshop on Visual Surveillance, in Conjunction with ECCV, 90:91. Citeseer.

80. Heindl, Christoph. 2017. Benchmark Multiple Object Trackers (MOT) in Python. Python. https://github.com/cheind/py-motmetrics.

81. Bewley, Alex, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. "Simple Online and Realtime Tracking." In 2016 IEEE International Conference on Image Processing (ICIP), 3464–3468. IEEE.

82. Bewley, Alex, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. "Simple Online and Realtime Tracking." In 2016 IEEE International Conference on Image Processing (ICIP), 3464–3468. IEEE.

83. Durant, Thibault. 2018. "Tracking Things in Object Detection Videos." Move Lab (blog). 2018. https://www.move-lab.com/blog/tracking-things-in-object-detection-videos.

84. Labbe, Roger. 2014. FilterPy Documentation. Python. https://github.com/rlabbe/filterpy.

85. McKinney, Wes. 2011. "Pandas: A Foundational Python Library for Data Analysis and Statistics." Python for High Performance and Scientific Computing 14 (9).

86. Van Der Walt, Stefan, S. Chris Colbert, and Gaël Varoquaux. 2011. "The NumPy Array: A Structure for Efficient Numerical Computation." Computing in Science & Engineering 13 (2): 22–30. https://doi.org/10.1109/MCSE.2011.37.

87. Jones, Eric, Travis Oliphant, and Pearu Peterson. 2001. SciPy: Open Source Scientific Tools for Python. https://www.scipy.org/.

88. Bewley, Alex. 2016. Simple, Online, and Realtime Tacking of Multiple Objects in a Video Sequence. Python. https://github.com/abewley/sort.

89. Fotache, Chris. 2018. PyTorch Object Detection and Tracking. Jupyter Notebook. https://github.com/cfotache/pytorch_objectdetecttrack.

90. Wang, Z., and A. C. Bovik. 2009. "Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures." IEEE Signal Processing Magazine 26 (1): 98–117. https://doi.org/10.1109/MSP.2008.930649.

91. Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable Road Users." MS Thesis, San Diego: San Diego State University.

92. Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable Road Users." MS Thesis, San Diego: San Diego State University.

93. Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable Road Users." MS Thesis, San Diego: San Diego State University.

94. Chollet, Francois. 2015. Deep Learning for Humans. Python. Keras. https://github.com/keras-team/keras.

95. Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. "Tensorflow: A System for Large-Scale Machine Learning." In 12th USENIX Symposium on Operating Systems Design and Implementation, 265–283.

96. Van Rossum, Guido, and Fred L. Drake. 2011. The Python Language Reference Manual. Network Theory Ltd.

97. Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable

Road Users." MS Thesis, San Diego: San Diego State University.

98. MATLAB. 2019. R2019b. Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com.

99. Opencv, n.d. Opencv-Python: Wrapper Package for OpenCV Python Bindings. (version 4.1.0.25).

100. Van Rossum, Guido, and Fred L. Drake. 2011. The Python Language Reference Manual. Network Theory Ltd.

101. Miller, R., and Qingfeng Huang. 2002. "An Adaptive Peer-to-Peer Collision Warning System." In Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367), 1:317–21 vol.1. https://doi.org/10.1109/VTC.2002.1002718.

102. Strelnikov, Ilya, and Cherepanov Oleksandr. 2019. "Small Objects Detection Problem." Mc.Ai (blog). 2019. https://mc.ai/small-objects-detection-problem/.

103. Padilla, Rafael. 2018. Most Popular Metrics Used to Evaluate Object Detection Algorithms. Python. https://github.com/rafaelpadilla/Object-Detection-Metrics.

104. Heindl, Christoph. 2017. Benchmark Multiple Object Trackers (MOT) in Python. Python. https://github.com/cheind/py-motmetrics.

105. Ruzicka, Vit, and Franz Franchetti. 2018. "Fast and AccurateObject Detection in High Resolution 4K and 8K Video Using GPUs." In 2018 IEEE High Performance Extreme Computing Conference (HPEC), 1–7. Waltham, MA: IEEE. https://doi.org/10.1109/HPEC.2018.8547574.

BIBLIOGRAPHY

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. "Tensorflow: A System for Large-Scale Machine Learning." In *12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283.

Allen, Brian L, B Tom Shin, and Peter J Cooper. 1978. "Analysis of Traffic Conflicts and Collisions." *Transportation Research Record*, no. 667: 67–74.

Barceló Bugeda, Jaime, André-Gilles Dumont, Lídia Montero Mercadé, Josep Perarnau, and Alexandre Torday. 2003. "Safety Indicators for Microsimulation-Based Assessments." In *Transportation Research Board 82nd Annual Meeting*, 1–18. TRB.

Bardool, Kevin. 2017. *Keras Implementation of Faster R-CNN*. Python. https://github.com/kbardool/keras-frcnn.

Behbahani, Hamid, and Navid Nadimi. 2015. "A Framework for Applying Surrogate Safety Measures for Side Swipe Collisions." *International Journal for Traffic & Transport Engineering* 5 (4): 371–83. https://doi.org/10.7708/ijtte.2015.5(4).03.

Bernardin, Keni, Alexander Elbs, and Rainer Stiefelhagen. 2006. "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment." In *Sixth IEEE International Workshop on Visual Surveillance, in Conjunction with ECCV*, 90:91. Citeseer.

Bewley, Alex. 2016. *Simple, Online, and Realtime Tacking of Multiple Objects in a Video Sequence*. Python. https://github.com/abewley/sort.

Bewley, Alex, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. "Simple Online and Realtime Tracking." In *2016 IEEE International Conference on Image Processing (ICIP)*, 3464–3468. IEEE.

Chen, Peng, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang. 2017. "Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos." *Journal of Advanced Transportation* 2017. https://doi.org/10.1155/2017/5202150.

Chollet, Francois. 2015. *Deep Learning for Humans*. Python. Keras. https://github.com/keras-team/keras.

Durant, Thibault. 2018. "Tracking Things in Object Detection Videos." *Move Lab* (blog). 2018. https://www.move-lab.com/blog/tracking-things-in-object-detection-videos.

Fotache, Chris. 2018. *PyTorch Object Detection and Tracking*. Jupyter Notebook. https://github.com/cfotache/pytorch_objectdetecttrack.

Gettman, Douglas, and Larry Head. 2003. "Surrogate Safety Measures from Traffic Simulation Models." *Transportation Research Record* 1840 (1): 104–115.

Ghanipoor Machiani, Sahar, and Montasir Abbas. 2016. "Safety Surrogate Histograms (SSH): A Novel Real-Time Safety Assessment of Dilemma Zone Related Conflicts at Signalized Intersections." *Accident Analysis & Prevention* 96: 361–70. https://doi.org/10.1016/j.aap.2015.04.024.

Hayward, J. C. 1972. "Near-Miss Determination through Use of a Scale of Danger." *Highway Research Record* 384: 24–34.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Heindl, Christoph. 2017. *Benchmark Multiple Object Trackers (MOT) in Python*. Python. https://github.com/cheind/py-motmetrics.

Heinisuo, Olli-Pekka. 2019. *OpenCV on Wheels*. Python. https://github.com/skvark/opencv-python.

Hui, Jonathan. 2018. "MAP (Mean Average Precision) for Object Detection." *Medium* (blog). 2018. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

Jahangiri, Arash, Mahdie Hasani, Ipek N. Sener, Sirajum Munira, Justin Owens, Bruce Appleyard, Sherry Ryan, Shawn M. Turner, and Sahar Ghanipoor Machiani. 2019. "Data Mining to Improve Planning for Pedestrian and Bicyclist Safety." Safe-D National UTC. https://www.vtti.vt.edu/utc/safe-d/index.php/projects/data-mining-to-improve-planning-for-pedestrian-and-bicyclist-safety/.

Jones, Eric, Travis Oliphant, and Pearu Peterson. 2001. *SciPy: Open Source Scientific Tools for Python*. https://www.scipy.org/.

Labbe, Roger. 2014. *FilterPy Documentation*. Python. https://github.com/rlabbe/filterpy.

Laureshyn, Aliaksei, Åse Svensson, and Christer Hydén. 2010. "Evaluation of Traffic Safety, Based on Micro-Level Behavioural Data: Theoretical Framework and First Implementation." *Accident Analysis & Prevention* 42 (6): 1637–1646.

Lu, Guangquan, Miaomiao Liu, Yunpeng Wang, and Guizhen Yu. 2012. "Quantifying the Severity of Traffic Conflict by Assuming Moving Elements as Rectangles at Intersection." *Procedia - Social and Behavioral Sciences*, 8th International Conference on Traffic and Transportation Studies (ICTTS 2012), 43: 255–64. https://doi.org/10.1016/j.sbspro.2012.04.098.

Mahmud, S. M. Sohel, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. 2017.

"Application of Proximal Surrogate Indicators for Safety Evaluation: A Review of Recent Developments and Research Needs." *IATSS Research* 41 (4): 153–63. https://doi.org/10.1016/j.iatssr.2017.02.001.

MATLAB. 2019. *R2019b*. Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com.

McKinney, Wes. 2011. "Pandas: A Foundational Python Library for Data Analysis and Statistics." *Python for High Performance and Scientific Computing* 14 (9).

Miller, R., and Qingfeng Huang. 2002. "An Adaptive Peer-to-Peer Collision Warning System." In *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, 1:317–21 vol.1. https://doi.org/10.1109/VTC.2002.1002718.

Minderhoud, Michiel M., and Piet H. L. Bovy. 2001. "Extended Time-to-Collision Measures for Road Traffic Safety Assessment." *Accident Analysis & Prevention* 33 (1): 89–97. https://doi.org/10.1016/S0001-4575(00)00019-1.

Nadimi, Navid, Hamid Behbahani, and HamidReza Shahbazi. 2016. "Calibration and Validation of a New Time-Based Surrogate Safety Measure Using Fuzzy Inference System." *Journal of Traffic and Transportation Engineering (English Edition)* 3 (1): 51–58. https://doi.org/10.1016/j.jtte.2015.09.004.

Opencv, n.d. *Opencv-Python: Wrapper Package for OpenCV Python Bindings.* (version 4.1.0.25)

Ozbay, Kaan, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. 2008. "Derivation and Validation of New Simulation-Based Surrogate Safety Measure." *Transportation Research Record: Journal of the Transportation Research Board* 2083 (1): 105–13. https://doi.org/10.3141/2083-12.

Padilla, Rafael. 2018. *Most Popular Metrics Used to Evaluate Object Detection Algorithms*. Python. https://github.com/rafaelpadilla/Object-Detection-Metrics.

Peesapati, Lakshmi N., Michael P. Hunter, and Michael O. Rodgers. 2013. "Evaluation of Postencroachment Time as Surrogate for Opposing Left-Turn Crashes." *Transportation Research Record: Journal of the Transportation Research Board* 2386 (1): 42–51. https://doi.org/10.3141/2386-06.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In *Advances in Neural Information Processing Systems 28*, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 91–99. Curran Associates, Inc. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Rosebrock, Adrian. 2016. "Intersection Over Union (IoU) for Object Detection."
PyImageSearch (blog). 2016. https://www.pyimagesearch.com/2016/11/07/
intersection-over-union-iou-for-object-detection/.

Ruzicka, Vit, and Franz Franchetti. 2018. "Fast and AccurateObject Detection in High
Resolution 4K and 8K Video Using GPUs." In *2018 IEEE High Performance
Extreme Computing Conference (HPEC)*, 1–7. Waltham, MA: IEEE. https://doi.
org/10.1109/HPEC.2018.8547574.

Saffarzadeh, Mahmoud, Navid Nadimi, Saber Naseralavi, and Amir Reza Mamdoohi.
2013. "A General Formulation for Time-to-Collision Safety Indicator." *Proceedings
of the Institution of Civil Engineers - Transport* 166 (5): 294–304. https://doi.
org/10.1680/tran.11.00031.

Sahoo, Pallavi. 2017. "Mean Average Precision." *Medium* (blog). 2017. https://medium.
com/@pds.bangalore/mean-average-precision-abd77d0b9a7e.

Shirazi, Mohammad Shokrolah, and Brendan Tran Morris. 2017. "Investigation of Safety
Analysis Methods Using Computer Vision Techniques." *Journal of Electronic
Imaging* 26 (5): 051404. https://doi.org/10.1117/1.JEI.26.5.051404.

Strelnikov, Ilya, and Cherepanov Oleksandr. 2019. "Small Objects Detection Problem."
Mc.Ai (blog). 2019. https://mc.ai/small-objects-detection-problem/.

Svensson, Ase. 1998. "A Method for Analysing the Traffic Process in a Safety
Perspective." PhD Dissertation, Lund, Sweden: University of Lund, Lund Institute
of Technology, Department of Traffic Planning and Engineering. https://portal.
research.lu.se/portal/files/4456913/1653512.pdf.

Tan, Ren Jie. 2019. "Breaking Down Mean Average Precision (MAP)." *Medium* (blog).
2019. https://towardsdatascience.com/breaking-down-mean-average-precision-
map-ae462f623a52.

Tarko, Andrew, Gary Davis, Nicolas Saunier, Tarek Sayed, and Simon Washington. 2009.
"Surrogate Measures of Safety - A White Paper." Transportation Research Board,
Washington, DC.

Tarko, Andrew P. 2012. "Use of Crash Surrogates and Exceedance Statistics to
Estimate Road Safety." *Accident Analysis & Prevention* 45: 230–40. https://doi.
org/10.1016/j.aap.2011.07.008.

Van Der Horst, Ara. 1990. "A Time-Based Analysis of Road User Behaviour in Normal
and Critical Encounters." PhD Dissertation, Delft, Netherlands: Delft University of
Technology. https://trid.trb.org/view/355354,.

Van Der Walt, Stefan, S. Chris Colbert, and Gaël Varoquaux. 2011. "The NumPy Array:
A Structure for Efficient Numerical Computation." *Computing in Science &*

Engineering 13 (2): 22–30. https://doi.org/10.1109/MCSE.2011.37.

Van Rossum, Guido, and Fred L. Drake. 2011. *The Python Language Reference Manual*. Network Theory Ltd.

Vogel, Katja. 2003. "A Comparison of Headway and Time to Collision as Safety Indicators." *Accident Analysis & Prevention* 35 (3): 427–33. https://doi.org/10.1016/S0001-4575(02)00022-2.

Vondrick, Carl. 2012. *Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces*. HTML. https://github.com/cvondrick/vatic.

Vondrick, Carl, Donald Patterson, and Deva Ramanan. 2013. "Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling." *International Journal of Computer Vision* 101 (1): 184–204. https://doi.org/10.1007/s11263-012-0564-1.

Wang, Z., and A. C. Bovik. 2009. "Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures." *IEEE Signal Processing Magazine* 26 (1): 98–117. https://doi.org/10.1109/MSP.2008.930649.

Yang, Hong, Kaan Ozbay, and Bekir Bartin. 2010. "Application of Simulation-Based Traffic Conflict Analysis for Highway Safety Evaluation." In *Proceedings of the 12th WCTR, Lisbon, Portugal*. Vol. 4.

Zając, Zygmunt. 2012. "What You Wanted to Know about Mean Average Precision." *FastML* (blog). 2012. http://fastml.com/what-you-wanted-to-know-about-mean-average-precision/.

Zhang, Shenchao. 2019. "Vision-Based Intersection Safety Evaluation For Vulnerable Road Users." MS Thesis, San Diego: San Diego State University.

Zheng, Lai, Karim Ismail, and Xianghai Meng. 2014. "Traffic Conflict Techniques for Road Safety Analysis: Open Questions and Some Insights." *Canadian Journal of Civil Engineering* 41 (7): 633–41. https://doi.org/10.1139/cjce-2013-0558.

ABOUT THE AUTHORS

ARASH JAHANGIRI

Dr. Jahangiri, PhD (Civil Engineering, Virginia Tech, 2015), is an Assistant Professor in the San Diego State University (SDSU) Department of Civil Engineering and the Co-Director of SDSU Smart Transportation Analytics Research (STAR) Lab. His research studies the behavior of current and future transportation systems and aims to improve safety, mobility, and sustainability by utilizing data-driven approaches, emerging technologies, and simulation.

ANAGHA KATTHE

Anagha is currently pursuing her second year as a master's student in Civil Engineering at San Diego State University with a focus on Transportation Engineering. She received her bachelor's degree in Civil Engineering from DSCE (Bangalore, India) in July 2014. She worked for three years as an Assistant Transportation Engineer in Bangalore before starting her graduate studies. Her research interests include intelligent transportation systems, connected vehicles, and traffic safety. She likes to keep herself updated by attending transportation conferences and events.

ARYAN SOHRABI

Aryan is a second-year master's student studying Computational Science with an emphasis in data science at San Diego State University. He received his bachelor's in Computer Science from McMaster University (Ontario, Canada) in May 2018. His areas of interest are high performance computing, artificial intelligence, and big data.

XIAOBAI LIU

Dr. Liu is an Associate Professor of Computer Science at SDSU and directs the SDSU Computer Vision Laboratory. His research concentrates on developing a unified framework for image and video parsing. He has contributed to new approaches to hierarchical representations of visual data, statistical knowledge discovery, and scalable stochastic inference.

SHASHANK PULAGAM

Shashank obtained his bachelor's degree in Computer Science and Engineering from University of California, Irvine in June 2018 and is currently a second-year master's student in Computer Science at San Diego State University. He is interested in Artificial Intelligence and has previous experience in this field from taking classes in Computer Vision, Data Mining, and Machine Learning as well as his involvement in machine vision projects.

VAHID BALALI

Vahid Balali, PhD, is an Assistant Professor in the Department of Civil Engineering and Construction Engineering Management at California State University Long Beach. He

is a member of National Center for Transportation, Green Technology, and Education (TransGET). Dr. Balali's research focuses on the visual data sensing and analytics, virtual design and construction for civil infrastructure and interoperable system integration, and smart cities in transportation for sustainable decision-making.

SAHAR GHANIPOOR MACHIANI

Dr. Machiani, PhD (Civil Engineering, Virginia Tech, 2014), is an Assistant Professor in the San Diego State University (SDSU) Department of Civil Engineering, an Associate Director (SDSU Director) of Safe-D National University Transportation Center, and the Co-Director of SDSU Smart Transportation Analytics Research (STAR) Lab. She has expertise in traffic safety, driver behavior modeling, and infrastructure-based safety systems.

PEER REVIEW

San José State University, of the California State University system, and the Mineta Transportation Institute (MTI) Board of Trustees have agreed upon a peer review process required for all research published by MTI. The purpose of the review process is to ensure that the results presented are based upon a professionally acceptable research protocol.