Titre: Title:	Evaluation of Deep Learning-Based Object Detection and Tracking Methods for Transportation Applications
Auteur: Author:	Liu Qingwu
Date:	2026
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Qingwu, L. (2026). Evaluation of Deep Learning-Based Object Detection and Tracking Methods for Transportation Applications [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/76890/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Evaluation of Deep Learning-Based Object Detection and Tracking Methods for Transportation Applications

QINGWU LIU
Département des génies civil, géologique et des mines

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie civil

Mai 2026

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Evaluation of Deep Learning-Based Object Detection and Tracking Methods for Transportation Applications

présentée par **Qingwu LIU**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Owen WAYGOOD, président
Nicolas SAUNIER, membre et directeur de recherche
Guillaume-Alexandre BILODEAU, membre et codirecteur de recherche
Martin TRÉPANIER, membre
Mohamed ZAKI, membre externe

DEDICATION

Ne te laisse jamais troubler par lavenir. Tu le rencontreras, si nécessaire, avec les mêmes armes de la raison qui tarment aujourdhui contre le présent.

Never let the future disturb you. You will meet it, if you have to, with the same weapons of reason which today arm you against the present.

- Marcus Aurelius, Meditations

ACKNOWLEDGEMENTS

I am sincerely grateful to my supervisor, Professor Nicolas Saunier, for him giving me the opportunity to pursue my studies at Polytechnique Montréal. I deeply appreciate his extensive knowledge, his rigorous research attitude, and his passion in the transportation research field. My sincere thanks also go to my co-supervisor, Professor Guillaume-Alexandre Bilodeau, for his invaluable guidance in the domain of computer vision and his continuous support throughout my studies.

I am deeply thankful to my family: my parents, my sister, and my nephews for their unconditional love, understanding, and constant encouragement. Their unwavering support has been a source of strength and motivation throughout my academic journey.

My heartfelt gratitude goes to my best friend, Sergio Vivó Sánchez. Whenever I feel lost, he is always offering me supports. Thank you for being willing to get to know my personality and for your constant support. I truly value our friendship.

I am also grateful to my friends I met in and out of CIRRELT. To Daniel Martínez, Mario José Basallo Triana, Jorge Mortes Alcaraz, Judith López, Huan Liu, Dario Vezzali, Fabio Mercurio, Man Shi, David Tremblet, Miao Feng, Mehdi Miah, Bingxu Zhu, Wang Luo, Jean Ertault, Yifei Qin, Hongshan He, Flore Caye, Pedram Farghadani-Chaharsooghi, Bruno Pacheco, Baik Seung Min, Felipe Torres, Alfaima Solano, Daniel Gracia de Vicuña, David Pinzon, Nagisa Sugishita, Imène Belabbas, Ismail Sevim, Étienne Boucher, Ali Rouhani, Luis Rojo, Benedetta Ferrari, Ehsan Mirzaei, Ando Rakotonandrasana, Jasone Ramirez, Prakash Gawas, Warley Alimeda, Fangxiang Peng and Jianbin Lin, for their accompanies and supports.

I would like to thank colleagues working in CIRRELT. To Guillaume Michaud, Pierre Girard, Lucie Nathalie Cournoyer, Nathalie Andrade, Khalid Laaziri, for their endless help and support both in my academic work and in my daily life. I would like to thank the volunteers for the data collection. Without your helps, I would not finish my research. I am also grateful for the supports provided by my neighbors, Jésus Angel Vaca, Juan Ramon Morienago, Romina Andole and Elias Andole, thanks for the parties, cycling memories in each summer and their supports every year.

I would like to thank my dog Monster. He has been with me through every happy and tough moment, and I promise that he will be happy and loved in every second of his life.

Finally, I would like to thank Montréal, it is my honor doing my PhD here.

RÉSUMÉ

Les méthodes de détection et de suivi dobjets basées sur lapprentissage profond ont connu un succès significatif en vision par ordinateur. Dans le domaine des transports, ces méthodes peuvent être utilisées pour extraire diverses mesures de trafic, telles que les comptages de véhicules, la vitesse et les indicateurs de sécurité. Certaines de ces mesures nécessitent lextraction de trajectoires à partir de données vidéo, ce qui peut être réalisé automatiquement par des méthodes de détection et de suivi. Cependant, leurs performances lorsquelles sont appliquées à des applications en transport restent encore insuffisamment comprises.

Cette thèse porte sur lévaluation des performances des méthodes de détection et de suivi dobjets pour des applications en transport, avec un accent sur lanalyse de la sécurité et lanalyse du comportement de traversée des piétons. Bien que ces méthodes atteignent souvent une grande précision sur des jeux de données de référence publics, elles présentent encore des erreurs telles que des non-détections, des faux positifs, des fragmentations de trajectoires et des changements didentité. Malgré leur utilisation répandue, peu de travaux évaluent la manière dont ces erreurs influencent les analyses de transport en aval. Cette lacune motive la question de recherche centrale de cette thèse : *Dans quelle mesure les méthodes de détection et de suivi dobjets basées sont-elles performantes pour les applications en transport ?*

La première étude évalue lapplicabilité des méthodes de détection et de suivi pour le calcul dun indicateur de sécurité, le temps avant collision (Time-to-collision, TTC), à partir dun jeu de données acquis par caméra embarquée. Les trajectoires des usagers de la route, incluant les véhicules, les piétons et les cyclistes, sont extraites de plusieurs méthodes de détection et de suivi, et les distributions de TTC obtenues sont comparées à celles dérivées des trajectoires de référence. Les résultats indiquent que, bien que les tendances générales du TTC soient capturées, des écarts importants apparaissent pour les faibles valeurs de TTC. Cela met en évidence la sensibilité de ces méthodes aux erreurs de perception et leurs limites pour des analyses critiques en matière de sécurité. Létude est ensuite étendue à un jeu de données obtenues de caméras fixes en bord de route et à deux indicateurs de sécurité, le TTC et le temps post-empiètement (Post-encroachment Time, PET). Une méthodologie dappariement des interactions est proposée afin de permettre une comparaison plus rigoureuse de ces deux indicateurs entre interactions prédites et de référence. Les résultats montrent que des écarts importants persistent au niveau des interactions, illustrant davantage les limites des méthodes actuelles de vision par ordinateur pour le calcul fiable dindicateurs de sécurité à ce niveau.

La deuxième étude traite d'une limitation importante dans lapplication des méthodes de

détection et de suivi aux analyses en transport: ces méthodes restent limitées dans leur capacité à détecter et suivre avec précision les groupes de piétons sous-représentés. En particulier, les piétons utilisant des aides à la mobilité (assisting and walking devices, AWD) sont rarement pris en compte dans les jeux de données existants, malgré leur importance pour des systèmes de transport inclusifs. Afin de combler cette lacune, un jeu de données de piétons utilisant des AWD a été construit à partir dun processus de collecte et dannotation des données, sous le nom de Polytechnique Montréal Mobility Aids Dataset (PMMA), et mis à disposition de la communauté scientifique Sept algorithmes de détection basés sur lapprentissage profond et trois méthodes de suivi sont évalués sur PMMA afin danalyser leurs performances sur ces catégories sous-représentées. Les résultats établissent une base de référence systématique pour les piétons utilisant des aides à la mobilité.

La dernière étude analyse les différences de comportement de traversée entre les piétons avec et sans aides à la mobilité à partir de données réelles nouvellement collectées. Bien que des modèles de détection aient été entraînés sur ce jeu de données, leur précision reste insuffisante pour une analyse comportementale fiable. Par conséquent, des trajectoires extraites manuellement sont utilisées pour lanalyse des comportements de traversée. Les résultats mettent en évidence des différences claires entre les catégories de piétons, notamment en termes de vitesse de traversée. Toutefois, des recherches supplémentaires sont nécessaires pour confirmer ces observations, compte tenu de quantité limitée de données disponibles pour les piétons utilisant des AWD.

Dans lensemble, cette thèse évalue les performances des méthodes de vision par ordinateur dans deux applications en transport, à savoir lanalyse de la sécurité et lanalyse du comportement de traversée des piétons. Les résultats révèlent un écart persistant entre les performances de ces méthodes sur des jeux de données de référence et leur efficacité dans des applications réelles en transport, soulignant la nécessité de poursuivre les recherches afin daméliorer leur précision.

ABSTRACT

Object detection and tracking methods have achieved significant success in computer vision. In the field of transportation, these methods can be used to extract various traffic measures, such as traffic counts, speed, and safety indicators. Some of these measures require trajectories from video data, which can be automatically extracted from video data by these detection and tracking methods. However, their performance when applied to these transportation application tasks remain insufficiently understood.

This dissertation deals with the performance evaluation of object detection and tracking for transportation applications, with a focus on safety analysis and pedestrian crossing behavior analysis. While these methods often achieve high accuracy on public benchmarks, they still exhibit errors such as missed detections, false positives, trajectory fragmentation, and identity switches. Despite their widespread use, there is limited work assessing how such errors propagate to downstream transportation analyses. This gap motivates the central research question of this dissertation: *How well do object detection and tracking methods perform for transportation application?*

The first study evaluates the applicability of detection and tracking pipelines for computing a safety indicator, time to collision (TTC), using a vehicle-mounted camera dataset. Trajectories of road users, including vehicles, pedestrians and cyclists, are extracted from multiple detection and tracking methods, and the distributions of TTC from these methods are compared with those derived from ground-truth trajectories. The results indicate that, while general TTCs are captured, substantial deviations occur for low TTC values. This highlights the sensitivity of these methods to perception errors and their limitations for safety-critical analyses. The study is then extended to a roadside camera dataset and two safety indicators, TTC and post-encroachment time (PET). An interaction matching methodology is proposed to enable a more rigorous comparison of these two indicators between predicted and ground-truth interactions. The results show that substantial discrepancies persist at the interaction level, further illustrating the limits of current computer vision methods for reliable safety indicators computations in the interaction level.

The second study addresses a key limitation in applying detection and tracking methods to transportation analysis, as these methods remain limited accurately detect and track underrepresented pedestrian groups. In particular, pedestrians with assisting and walking devices (AWDs) are rarely captured in existing datasets, despite their importance for inclusive transportation systems. Therefore, to address this limitation, a dataset on pedestrians

with AWDs was constructed through data collection and annotation, named as the Polytechnique Montréal mobility aids dataset (PMMA) and released as a resource for the research community. Seven deep learning-based detection algorithms and three tracking methods are bench-marked on PMMA to assess their performance on these underrepresented categories. The results establish a systematic baseline for pedestrians with AWDs.

The last study analyzes differences crossing behaviors between pedestrians with and without AWDs using newly collected real-world data. Although detection models are trained on this dataset, their accuracy is insufficient for reliable behavioral analysis. Instead, manually extracted trajectories are used for crossing behavior analysis. The results show clear differences across pedestrian categories, particularly in crossing speed. However, further research is needed to confirm these findings given the limited amount of data for pedestrians with AWDs.

Overall, this dissertation evaluates the performance of computer vision methods in two transportation applications, including safety analysis and pedestrian crossing behavior analysis. The results reveal a persistent gap between the performance of these methods on benchmark datasets and in real-world transportation applications, highlighting the need for further research to improve their accuracy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AoI	Areas of Interest
AP	Average Precision
AR	Average Recall
AssA	Association Accuracy
AWDs	Assisting and Walking Devices
BB	Bounding Box
BEV	Bird-Eye View
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
DFT	Detection Free Tracking
DNN	Deep Neural Network
DetA	Detection Accuracy
fps	Frame(s) per Second
FN	False Negative
FP	False Positive
GPS	Global Positioning System
GT	Ground Truth
GPU	Graphics Processing Unit
HOG	Histograms of Oriented Gradients
HOTA	Higher Order Tracking Accuracy
IDF_1	Identification F_1
IDSW	Identity Switch, ID Switch
IMU	Inertial Measurement Unit
IoU	Intersection over Union
LocA	Localization Accuracy
mAP	mean Average Precision
mAR	mean Average Recall
MOT	Multiple Object Tracking, Multi-Object Tracking
MOTA	Multiple Object Tracking Accuracy
PET	Post-Encroachment Time
RD/DD ratio	Real Distance to Direct Distance Ratio
reID	Re-identification
SVM	Support Vector Machine

SMoS	Surrogate Measures of Safety
TCT	Traffic Conflict Technique
TI	Traffic Intelligence (Software)
TP	True Positive
TTC	Time to Collision
VRU	Vulnerable Road User

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Transportation systems are increasingly supported by data-driven approaches that enable the monitoring, analysis, and management of complex traffic infrastructures. Among these approaches, video-based analysis has emerged as a powerful and scalable solution for more complex transportation analysis, e.g., interaction and activities based studies in real world scenarios. In recent years, advances in deep learning, particularly in object detection and tracking, have significantly improved the ability to compute transportation variables, including traffic counts, speeds, queue lengths, safety indicators, time headway, and conflict frequency from traffic videos. These developments have made computer vision methods a fundamental tool for a wide range of transportation applications, such as safety assessment and behavior analysis. By automatically extracting trajectory data, these methods facilitate more efficient analysis of movement patterns and interactions based on trajectories.

Among these applications, safety analysis and behavior analysis are two domains that rely heavily on trajectory data. Traditional approaches to road safety primarily depend on historical crash records. However, since crashes are relatively rare and occur randomly, such approaches often require long observation periods before meaningful patterns can be identified. To overcome these limitations, proactive methods for safety analysis have been developed to assess potential safety risks based on road user interactions before collisions occur. Besides, automated proactive methods for safety analysis depend on accurate road user trajectories. The trajectories obtained from traditional detection and tracking pipelines often suffer from limited accuracy. Facing such challenges, researchers have increasingly leveraged advances in computer vision, applying deep learning-based object detection and tracking methods to safety analysis.

Since 2012, vision-based object detection and tracking have made significant progress, largely driven by the advances in deep learning and the availability of high-performance Graphics Processing Unit (GPU) computing. The breakthrough began with the ImageNet Large Scale Visual Recognition Challenge, where convolutional neural networks (CNNs) trained with GPU acceleration drastically reduced classification errors. Following these and subsequent advances in image classification, deep learning approaches were subsequently extended to more complex tasks, including object detection, which requires locating objects with bounding boxes, and object tracking, which adds the temporal dimension to follow objects across frames. These deep learningbased developments have greatly improved the performance and robustness of vision-based analysis in traffic scenarios. In particular, by applying deep

learningbased object detection and tracking algorithms to traffic videos, researchers can automatically extract trajectories of road users such as vehicles, pedestrians, and cyclists. As a result, deep learningbased detection and tracking algorithms have become a key component in recent studies in transportation applications, particularly those relying on trajectory-based analyses.

Although these methods have demonstrated strong performance on established datasets, their accuracy still falls short in complex real-world scenarios. This naturally raises a critical question: *How well do existing object detection and tracking algorithms perform in transportation analyses?* Since trajectory-based transportation analysis depends on the outputs of detection and tracking, errors in these methods can propagate to higher-level analyses, thereby compromising the reliability of derived transportation application indicators.

These limitations are likely to be further amplified when traffic scenes involve heterogeneous or atypical road users. In particular, pedestrians with assisting and walking devices, or pushing/pulling carts and strollers, abbreviated as pedestrians with AWDs, are often more vulnerable than typical VRUs. Yet, datasets and analyses focusing on these road users remain limited. Their visual appearance, motion patterns, and interaction behaviors deviate from those of pedestrians without AWDs, posing additional challenges for detection and tracking algorithms. This leads to a more specific question: *How effectively can current algorithms handle a broader range of pedestrian types that have similar or ambiguous appearance within real traffic scenes?*

In this thesis, we aim to address these two questions by conducting systematic road safety analyses, building a new dataset of pedestrian with AWDs, and carrying out a case study using data collected from a busy urban intersection in Montréal comparing the crossing behavior of pedestrians with and without AWDs.

This chapter introduces the research by first discussing the basic concepts underlying the proposed framework. It then describes the elements of the problem, followed by the research objectives and contributions. Finally, it outlines the structure of the thesis.

1.1 Definition and basic concepts

Vulnerable Road Users (VRUs) are road users who lack physical protection in traffic environments, including pedestrians, cyclists, users of mobility aids, etc., who are therefore at higher risk of severe injury in collisions. Among VRUs, **pedestrians with AWDs** refer to pedestrians with assisting and walking devices (AWDs) (e.g., canes, walkers, wheelchairs) or rolling devices(e.g. strollers, grocery carts), making them more vulnerable than pedestrians

and cyclists.

Object detection is a fundamental task in computer vision that aims to identify and localize instances of objects within an image or video frame, typically by providing a bounding box (or a segmentation) and a class label for each detected object. It extends beyond simple image classification by not only determining what objects are present, but also where they are located in the visual scene. Despite significant advances, object detection still faces several challenges. A major issue is **occlusion**, where objects are partially or fully blocked by other objects or scene elements, making it difficult for algorithms to correctly detect and classify them. Other challenges include variations in object scale, lighting conditions, motion blur, and background clutter.

Object tracking is the task of following one or more objects over time across consecutive video frames, maintaining their identities as they move through the scene. Tracking faces several challenges similar to detection. A major difficulty is **object association**, which involves correctly matching detected objects across frames to maintain their identities over time. Errors in association can lead to identity switches (IDSW), fragmented trajectories, or lost tracks. Occlusion also affects tracking, as partially or fully blocked objects make it difficult for the tracker to maintain continuity. Additional challenges include changes in object appearance, variations in scale, abrupt motion, and crowded or cluttered environments. These factors make robust tracking a complex task, whether using tracking-by-detection or jointly detection and tracking approaches.

1.2 Elements of the problem

The problem addressed in this research involves a set of interconnected challenges that arise from both the performance of computer vision methods and the methodological requirements of video-based transportation applications.

1.2.1 Challenges in object detection and tracking for transportation applications

Although deep learning has significantly advanced object detection and tracking in recent years, applying these methods to real-world traffic environments remains challenging. Busy urban intersections often contain heterogeneous road users, frequent occlusions, varying object scales, cluttered backgrounds, and complex interactions, all of which can degrade detection confidence and compromise tracking continuity. In particular, visually small objects, partially occluded pedestrians, and pedestrians with AWDs present additional difficulties for standard models trained predominantly on pedestrian without AWDs or vehicle categories.

These challenges directly affect the stability and accuracy of extracted trajectories, which are essential inputs for downstream behavioral and safety analyses. As a result, even small inconsistencies in detection and tracking can lead to fragmented or noisy trajectories, raising questions about the reliability of current computer vision pipelines when deployed in unconstrained outdoor settings.

1.2.2 Uncertain suitability of detection and tracking outputs for transportation analysis

When detection and tracking outputs are used as the foundation for transportation analysis, discrepancies from ground truth can propagate to higher-level analyses. Detection errors introduce spatial inaccuracies in object positions and/or categories, while tracking errors propagate these inaccuracies temporally through IDSW, and trajectory fragmentation. As trajectories are constructed by linking observations across consecutive frames, small localization errors are repeatedly integrated over time, propagating into higher-level indicators such as crossing speed, TTC, PET, and other safety indicators. Existing studies rarely quantify how these accumulated discrepancies influence the reliability of safety conclusions. Moreover, although computer vision methods have become increasingly prevalent in safety research, there is still no systematic investigation into whether current detection and tracking performance is accurate enough for robust transportation assessment, particularly in complex environments with heterogeneous road users. Consequently, a fundamental methodological gap persists between what computer vision methods provide and what transportation analysis requires.

1.2.3 Lack of pedestrians with AWDs for model training and transportation studies

Most publicly available traffic video datasets focus on a limited set of categories, such as cars, pedestrians, and cyclists, reflecting their common use in autonomous driving research. However, pedestrians with AWDs are rarely included or are represented only in small quantities. Without appropriate datasets, current models struggle to recognize these users accurately, and safety researchers lack the means to systematically analyze their behavior or quantify their exposure to risk. The scarcity of such data thus creates a major barrier to understanding and improving intersection safety for these vulnerable subgroups.

1.2.4 Challenge of handling visually similar but behaviorally distinct categories

Another fundamental challenge arises from the fact that pedestrians with AWDs share similar visual characteristics but represent behaviorally different user categories. For example, walker users, stroller users, grocery-cart users, and wheelchair users may be visually difficult to distinguish from one another by computer vision methods, yet the behaviors and vulnerabilities of the corresponding users differ substantially. Object detection models that are not explicitly trained on these fine-grained categories often misclassify such objects or confuse them with ordinary pedestrians. In addition, existing related transportation research rarely distinguishes these fine-grained categories when evaluating crossing behavior or exposure to motorized traffic. As a result, the unique challenges faced by pedestrians with AWDs remain insufficiently understood within transportation research and analysis. Therefore, addressing this gap is essential for developing more inclusive analytical frameworks, improving the accuracy of transportation assessment practices, and ensuring that vulnerable groups are appropriately represented in transportation application analysis.

1.3 Research objectives

The general objective of this research is to *evaluate the applicability and accuracy of object detection and tracking methods for video-based transportation applications.*

The specific objectives within this research are the following:

1. To evaluate how differences between predicted and ground-truth trajectories extracted by detection and tracking methods from vehicle-mounted or roadside video data affect safety analysis;

2. To evaluate the performance of recent object detection and tracking methods on pedestrians with AWDs; and

3. To conduct a video-based case study at an intersection to evaluate the ability of detection methods to identify pedestrians with AWDs, and to analyze and compare their crossing behaviors with those of pedestrians without AWDs.

These objectives address key research gaps and will guide the development of a robust framework for evaluating trajectory-based safety analysis (objective 1), essential datasets and baseline performance evaluations for pedestrians with AWDs (objective 2), and meaningful insights into the real-world behavior of pedestrians with and without AWDs (objective 3).

1.4 Contributions

The research presented in this dissertation was conducted in multiple phases between September 2020 and March 2026. The contributions of each article are described in relation to the corresponding research objectives. These contributions focus on the evaluation of object detection and tracking methods for video-based transportation applications, including both safety and behavior analysis.

1.4.1 Evaluation of detection and tracking methods on existing dataset

- **Vehicle-mounted dataset.** For this first research objective, the suitability of deep learningbased object detection and tracking methods for trajectory-based safety analysis is assessed using vehicle-mounted video data. The ability of these methods to generate reliable trajectories for the computation of safety indicator TTC is evaluated.
- **Roadside dataset.** In line with the first research objective, a dedicated evaluation framework is proposed to extend the safety analysis using a roadside video data at a signalized intersection. This framework enables comparing the safety indicators computed from object detection and tracking methods with those computed from the ground truth. Besides, a matching method is proposed to evaluate the performance of vision-based methods at the interaction-matched level.

1.4.2 Evaluation of detection and tracking on a dedicated dataset of mobility-aid users

For the second research objective, a dedicated dataset of pedestrians using AWDs (PMMA) is collected in an outdoor parking lot of Polytechnique Montréal in Montréal, Canada, to support their detection and tracking. The dataset provides fine-grained annotations for multiple pedestrian categories, including cane users, walker users, and wheelchair users, enabling the evaluation of detection and tracking performance for underrepresented road user groups. Baseline experiments with several recent methods highlight the challenges posed by visually similar categories, occlusions, and complex interactions.

1.4.3 A case study for comparing crossing behaviors at a signalized intersection

For the third research objective, a video-based analysis of pedestrian crossing behavior is conducted to compare pedestrians with and without AWDs using trajectories extracted from a real-world roadside video data. The analysis provides insights into differences in movement

patterns and interaction behaviors, highlighting the influence of AWDs on pedestrian crossing behaviors.

1.5 Thesis outline

This dissertation is organized into nine chapters. Chapter 2 provides a review of the literature on video-based transportation applications, visual perception, pedestrians with AWDs related datasets, and trajectory-based transportation analysis. Chapter 3 gives the overall research process and provides a summary of the links between the research objectives and the articles. Chapter 4 presents a study evaluating deep learningbased object detection and tracking methods for trajectory-based safety analysis using the vehicle-mounted KITTI dataset [7]. TTC is computed and analyzed by comparing the performance of multiple methods against the ground truth. Chapter 5 extends this analysis to a roadside video data LUMPI [8] using the *Traffic Intelligence* (TI) [9] framework, computing both TTC and PET. To further assess interaction-level performance, an interaction matching method is developed. Chapter 6 introduces a dedicated dataset of pedestrians with AWDs and benchmarks several detection and tracking methods on this dataset. Chapter 7 presents a case study analyzing crossing behavior at a signalized intersection, comparing pedestrians with and without AWDs. The analysis examines differences of crossing behaviors among these pedestrian groups. Chapter 8 provides a discussion on the effects of dataset type, coordinate projections methods, the use of interaction matching, and the representation of pedestrians with AWDs in both the collected dataset and real-world data used for case study. Finally, Chapter 9 summarizes the work, presents its limitations and potential areas for the future research.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews the literature related to video-based transportation studies, focusing on studies related to the methodological framework adopted in this dissertation. In particular, it summarizes prior research on sensors utilized for data collection in transportation applications, object detection and tracking methods for trajectory extraction, mobility aids related datasets for video-based traffic applications, safety indicators derived from trajectory data, and studies on pedestrian crossing behavior. The objective of this section is to position the proposed research within existing work and to identify the methodological foundations for video-based transportation application analysis.

2.1 Sensors for data collection in transportation applications

Several sensors have been used for collecting data in traffic data collection area, such as inductive loops, magnetic sensors, radars, LiDARs and cameras. Those sensors can be categorized as off-roadway, intrusive, and non-intrusive in the following sections. Off-roadway sensors, such as Global Positioning System (GPS)-based sensors, are installed outside roads. Intrusive sensors, such as inductive loops, are those sensors installed inside roads and non-intrusive sensors, such as radars, LiDARs and cameras, are those do not cause any damages to the roads.

Among intrusive sensors, an inductive loop is typically embedded under the road surface. When a vehicle passes over it, it captures the change of inductance and generates a time-varying signal [10]. Pavement-mounted magnetic sensors detect disturbances in the Earths magnetic field caused by passing vehicles, enabling the measurement of traffic variables such as vehicle presence, speed, length, and classification. Among non-intrusive sensors, radar exploits the reflections of radio signals on the vehicle body to perform classification by radio waves. It is less vulnerable to weather and light conditions than LiDAR but cannot detect the vehicles or other road users body with the same accuracy [10]. LiDAR measures distance by computing the time delay between emitted and reflected laser signals. It has been used in some applications such as precision agriculture, biodiversity assessment, and object detection and classification in robotics [11–13].

Cameras capture traffic scenes as image sequences. A single camera can cover multiple lanes while intrusive sensors can only cover one lane per sensor. Moreover, cameras can record more features, including colors and textures while 3D point clouds generating by LiDARs

cannot do this. Besides, cameras are much cheaper than LiDARs and can also offer spatial data. All the above strengths make cameras one of the most popular sensors for object detection and tracking. The most used sensors in transportation applications are reviewed and a summary of their strengths and limitations is given in Table 2.1.

Based on the advantages of cameras, the following sections focus on the applications of cameras (videos) in transportation applications analysis.

2.2 Deep learning-based perception

2.2.1 Object detection

Object detection is a fundamental component of video-based perception systems. Its objective is to identify objects of interest in an image and localize them using bounding boxes or segmentation masks while assigning semantic class labels.

Early object detection approaches relied on traditional computer vision techniques, including background subtraction [14,15], hand-crafted features [16] and sliding-window classifiers [17]. Methods such as Histogram of Oriented Gradients (HOG) [18] combined with Support Vector Machines (SVM) [19] were widely used for pedestrian detection. While these approaches achieved reasonable performance in relatively simple scenarios, their robustness was limited in complex traffic environments characterized by occlusions, varying illumination, and large appearance variations.

The emergence of deep learning has significantly improved object detection performance. **CNNbased** detectors can automatically learn hierarchical visual features from large-scale datasets, enabling more robust detection under diverse traffic scenarios. Modern deep learning detectors can be broadly categorized into **two-stage** and **one-stage** detectors. **Two-stage detectors** first generate candidate object regions and then classify them. Representative methods include the R-CNN [3], Fast R-CNN [20], and Faster R-CNN [21]. These methods typically achieve high detection accuracy but often involve relatively higher computational cost. In contrast, **one-stage detectors** perform object localization and classification simultaneously in a single network. This design significantly improves inference speed, making them suitable for real-time applications. Representative one-stage detectors include the You Only Look Once (YOLO) family [22–24] and SSD [25]. Recent YOLO-based models have been widely adopted in transportation research due to the favorable trade-off between accuracy and efficiency, as well as the ease of use and availability through frameworks such as Ultralytics [26, 27]. More recently, alternative detection paradigms have been proposed. Keypoint-based detectors such as CenterNet [28] represent objects using keypoints that corre-

spond to object centers, avoiding the need for anchor boxes. **Transformer-based** detectors such as DETR [29], Deformable DETR [30] employ attention mechanisms to directly predict object bounding boxes from global image representations. Several of these detection approaches have been used in this dissertation.

YOLO detectors formulate detection as a single regression problem that directly predicts object locations and class probabilities from an input image [22]. Instead of generating region proposals and then classifying them, YOLO-based models divide the image into a grid and allow each grid cell to predict a fixed number of bounding boxes together with confidence scores and class probabilities. Each predicted bounding box is typically parameterized by its center coordinates and spatial dimensions in the image plane, i.e., (x, y, w, h), where (x, y) represents the center coordinates of the box and (w, h) denote the width and height.

The YOLO architecture has evolved through several generations to improve both detection accuracy and computational efficiency. Recent versions such as YOLOv8 [26] introduce improvements including anchor-free detection heads and enhanced feature aggregation mechanisms. Subsequent developments such as YOLOv11 [27] further refine feature extraction and multi-scale detection capabilities.

CenterNet adopts a different detection paradigm by representing objects as points [28]. Instead of relying on anchor boxes or region proposals, the detector predicts object centers using a heatmap, where each pixel represents the probability of an object center at that location. Peaks in the heatmap correspond to detected object centers. In parallel with heatmap generation, regression heads densely predict object attributes across the image, and the corresponding values are retrieved at the detected center locations to estimate object size (w, h). A small offset is used to compensate for feature map discretizations, allowing the reconstruction of a 2D bounding box defined by (x, y, w, h).

DETR introduces a fundamentally different paradigm for object detection by leveraging the transformer architecture [31]. The model uses a convolutional backbone to extract image features, which are then processed by a transformer encoderdecoder architecture that models global relationships across the image. The transformer decoder predicts a fixed set of object queries, where each query corresponds to a potential object in the scene. For each query, the network directly predicts the class label and bounding box parameters (x, y, w, h). A bipartite match loss is used during training to establish a one-to-one correspondence between predicted objects and ground-truth annotations. This formulation enables the model to produce a unique prediction per object, therefore removing the need for heuristic post-processing steps such as non-max suppression (NMS), and avoiding the use of predefined anchor boxes. The global attention mechanism in DETR allows the model to capture long-range spatial de-

pendencies between objects in a scene, which can be beneficial in complex traffic environments where multiple road users interact. Although DETR typically requires longer training time compared with convolutional detectors, it offers a conceptually simpler detection framework by eliminating heuristic components. Subsequent work such as **Deformable DETR** [30] improves the original DETR architecture by introducing deformable attention mechanisms, which significantly accelerate convergence and improve the detection of small objects.

While many detection methods focus on two-dimensional (2D) object localization, certain transportation applications require three-dimensional (3D) spatial information. Extended from the YOLO detection framework, **YOLOStereo3D** addresses this need to perform stereo-based 3D object detection. The method utilizes stereo image pairs captured from synchronized cameras to estimate object depth through disparity information between the left and right images. YOLOStereo3D employs a convolutional backbone network, typically based on ResNet [32], to extract visual features from both stereo images. These features are combined to estimate the disparity map, which provides information about the relative depth of objects in the scene. Based on this information, the network predicts both 2D bounding boxes and additional parameters describing the 3D bounding box of each object, including its spatial location (x, y, z), dimensions (w, h, l), and orientation θ. In addition, the model integrates the Ghost module [33] to improve feature extraction efficiency while maintaining detection accuracy. By estimating 3D bounding boxes directly from stereo images, YOLOStereo3D enables more accurate reconstruction of object positions in real-world coordinates. This capability is particularly useful in transportation applications where the 3D position and orientation information of vehicles and pedestrians must be estimated for trajectory generation and safety analysis. **CenterNet3D** is a 3D extension of CenterNet that can predict additional parameters such as depth z, object orientation θ, and physical dimensions (w, h, l). In this case, objects can be represented by 3D bounding boxes $(x, y, z, w, h, l, \theta)$ in the camera coordinate system, enabling the estimation of spatial relationships between road users. **MonoDTR** is a transformer-based monocular 3D object detection method that leverages attention mechanisms to capture global context and directly predict 3D object attributes from single images.

Table 2.1 Strengths and weaknesses of mainstream sensor technologies, adapted from [1].

Technologies	Strengths	Weaknesses
Inductive Loops	• Flexible design to satisfy a large variety of applications • Mature, well-understood technology • Large experience base • Insensitive to inclement weather such as rain, fog, and snow	• Installation requires pavement cut • Maintenance requires lane-closure • Multiple detectors are usually required for monitoring one location
Magnetic sensors	• Indicate the presence of a metallic object by detecting the perturbation • High accuracy in vehicle detection, speed measurement, and classification [34].	• Installation requires pavement cut. • Not suitable for dense traffic • Cannot detect stopped vehicles or stopped pedestrians
Radars	• Insensitive to inclement weather at relatively short ranges • Can get speed data directly • Multiple lane operations are available.	• Occlusion of vehicles may affect the accuracy of detection
LiDARs	• Provide high-resolution data, extremely accurate. • Can cover a large area with 360 degrees view • Easy to install and maintain • Work day and night without the influence of light conditions	• Operation may be affected by colossal rain or snow. • Performance might be affected by fog. • Point clouds tend to be disordered and irregular. • Occlusion of objects may affect the accuracy of detection.
Cameras	• Monitors multiple lanes and multiple detection zones/lanes. • Easy to add and modify detection zones. • A rich array of data available. • Provides wide-area detection when information gathered at one camera location can be linked to another.	• Performance affected by in inclement weather such as fog, rain and snow • Privacy issues when recording images/videos in public areas • Day-to-night transitions may also affect the performance • Occlusion of vehicles may affect the accuracy of detection • Some models are susceptible to camera motion caused by strong winds or vibration of the camera mounting structure

2.2.2 Multi-object tracking

In complex traffic scenarios, trajectory extraction is essential for tasks that rely on modeling interactions and motion dynamics, such as the computation of safety indicators and behavioral analyses, including crossing speed and interactions with surrounding pedestrians and vehicles. In contrast, other applications, such as traffic counting and occupancy estimation, can be performed without trajectory information. Those trajectory-based tasks are addressed by multi-object tracking (MOT), which aims to associate detections belonging to the same object across consecutive frames in a video sequence. By maintaining consistent object identities over time, MOT methods enable the generation of trajectories that describe the motion of road users in traffic scenes.

Most modern MOT approaches follow a **tracking-by-detection** paradigm. In this framework, objects are first detected independently in each frame using an object detector, and a tracking algorithm then associates these detections across frames using motion models, appearance features, or spatial similarity measures.

SORT [35] is a lightweight tracking-by-detection algorithm that associates detections across frames using a Kalman filter [36] for motion prediction and the Hungarian algorithm [37] for data association. **DeepSORT** [38] extends the SORT tracking algorithm by incorporating deep appearance features to improve the robustness of data association. **ByteTrack** [39] improves tracking performance by considering both high-confidence and low-confidence detections during the association process, reducing missed associations in crowded scenes. More recent variants such as **BoT-SORT** [40] and **OC-SORT** [41] further improve tracking robustness by refining motion models and association strategies.

An alternative approach is **joint detection and tracking**, where both tasks are performed simultaneously within a single network architecture. Although such end-to-end approaches have recently attracted increasing attention, tracking-by-detection remains the dominant paradigm in many practical applications.

Built upon CenterNet by extending the keypoint-based detection framework to incorporate temporal information for tracking across consecutive frames. **CenterTrack** [42] is a joint detection and tracking method within a unified framework by predicting object centers and motion offsets simultaneously.

2.2.3 Trajectory generation for transportation analysis

Accurately obtaining road user trajectories under real-world conditions is a fundamental requirement for many transportation applications. In practice, such trajectories are inherently

3D, as road users move within a physical environment where their positions, orientations, and spatial extents must be represented in the real world. Depending on the level of detail, trajectories can be represented as point-based tracks, 2D ground-plane representations, such as bird's-eye view (BEV) projections, or full 3D bounding boxes that capture both motion and geometric properties.

To obtain such trajectories in the real world from video data, existing approaches can be broadly categorized into two types. The first category relies on stereo-camera [43, 44] or multiple-camera system [45, 46], which enable the recovery of 3D information through geometric correspondence across views. These approaches can provide direct depth estimation and object location in the real-world coordinate. The second category uses monocular views combined with camera calibration, distortion correction, and geometric transformations to project 2D image coordinates onto a real-world coordinate system [47, 48]. Besides, recent deep learning-based methods can directly deal with monocular images and predict 3D information, such as depth or object orientation [49–51]. Although these monocular approaches do not require explicit 3D sensors or multi-sensor setups, their accuracy is more sensitive to calibration quality and scene geometry.

In addition to positional information, many transportation applications require knowledge of road user size, which is essential for accurately characterizing spatial interactions, especially for safety analysis. For example, pedestrians with AWDs, cyclists, or pedestrians groups may occupy substantially different spatial extents. However, such differences cannot be captured by point trajectories alone and require either explicit geometric representations (e.g., bounding boxes) or additional assumptions about object dimension (e.g., applying average size to a specific road user type).

Overall, the quality of generated trajectories is directly influenced by the performance of the underlying detection and tracking algorithms and the errors can propagate to the downstream safety application analysis. As a result, the reliability of trajectory-based transportation applications, including road safety analysis and pedestrian crossing behavior analysis, depends strongly on the accuracy and temporal consistency of the perception and tracking processes used to generate these trajectories.

2.3 Pedestrians with AWDs related datasets

Pedestrians with AWDs constitute a diverse subgroup of VRUs, which is different from other road users in terms of walking behavior, speed and interactions with the surrounding environment. Understanding these differences is essential for transportation applications

such as pedestrian safety analysis, infrastructure design, and inclusive mobility planning. However, existing video-based analysis tools remain limited in their ability to accurately capture and represent pedestrians with AWDs in specific real-world case studies.

In order to develop and validate the accuracy of these tools, some pedestrian datasets have considered categories related to AWDs. For example, the dataset proposed by Dávila-Soberón et al. [52] focuses on wheelchair users, cane users, and mobility aids themselves. However, this dataset is not publicly available, which limits its accessibility for broader research purposes. Kollmitz et al. [53] introduced a dataset collected in an indoor hospital environment to study mobility aid users, providing valuable data for healthcare-related applications but lacking outdoor scene variability and typical urban traffic viewpoints. In addition, Zhang et al. [54] proposed a synthetic dataset generated in the X-World simulation environment that includes objects such as wheelchairs and canes. While useful for controlled experiments, simulation-based datasets often lack the visual complexity and behavioral diversity of real-world environments.

Overall, publicly available datasets that capture pedestrians with diverse AWDs in real-world outdoor environments remain extremely limited. In particular, there is currently a lack of datasets that simultaneously provide outdoor traffic scenes, detailed annotations of pedestrians with AWDs, and data suitable for tasks such as object detection and MOT. This gap poses a challenge for evaluating the performance of visionbased perception systems and for conducting inclusive transportation analysis involving pedestrians with AWDs.

2.4 Trajectory-based transportation analysis

2.4.1 Road safety analysis

Proactive methods for road safety analysis

Road safety analysis has traditionally relied on historical crash data to identify risk factors and inform interventions. This reactive approach requires crashes to occur before preventive measures can be implemented. This inherent limitation highlights the need for proactive road safety analysis methods that can identify and mitigate potential risks without relying on the occurrence of crash events [55]. Traffic conflict is defined as “an observable situation in which two or more road users approach each other in space and time to such an extent that there is a risk of collision if their movements remain unchanged” [56]. Early efforts to formalize traffic conflicts led to the development of **Traffic Conflict Techniques (TCTs)**, proactive and observational approaches that identify and analyze near-misses and conflicts between road users. Safety indicators are commonly used in TCT-based analyses to quantify

and evaluate the severity of traffic conflicts.

Among safety indicators, TTC and PET are two of the most commonly used metrics in trajectory-based safety analysis. TTC is the time remaining until a collision of two road users (or a road user with a stationary obstacle) assuming they continue traveling as initially planned [55]. PET is the duration between the instant the first road user leaves the crossing zone and the instant the second road user reaches the crossing zone if the trajectories of two road users cross [55]. These indicators provide interpretable measures of interaction severity and have been widely applied in safety studies [57–60]. Other safety indicators have been proposed in the literature, including measures based on deceleration requirements or time-to-brake [61]. Although these indicators provide complementary perspectives on interaction severity, TTC and PET remain among the most widely adopted measures due to their intuitive interpretation and their ability to be directly computed from trajectory data.

Computer vision-based road safety analysis

Automated video analysis systems have been used to extract trajectories of vehicles and VRUs from traffic videos and identify traffic conflicts at intersections using safety indicators [62,63]. Similar approaches have been applied to analyze large-scale traffic video datasets and detect near-miss events or hazardous interactions among road users [64, 65]. Other studies have further demonstrated the feasibility of automated trajectory extraction for large-scale traffic conflict analysis and safety assessment using video-derived trajectory data [59, 66]. These studies highlight the potential of computer vision-based pipelines to significantly reduce the need for manual trajectory annotation while enabling the analysis of extensive video datasets for safety research.

Despite the advance of computer vision-based methods, whether trajectories automatically extracted by automated video analysis frameworks are sufficiently reliable for transportation safety assessment remains unclear. Limited attention has been paid to their accuracy and robustness in the context of interaction-based analysis. Research on this gap is essential for understanding the extent to which such pipelines can support practical road safety studies.

2.4.2 Pedestrian crossing behavior analysis

Pedestrian crossing behavior has been extensively studied to better understand how crossing pedestrians interact with roadway environments, traffic control, and approaching vehicles [67–69]. Studies often relied and some continue to rely on manual observation or event-based coding to examine whether pedestrians complied with signals, accepted or rejected vehicle

gaps, changed speed while crossing [70–72]. More recently, the increasing availability of video-based trajectory data has enabled a shift toward trajectory-based pedestrian behavior analysis, where pedestrian movement can be examined continuously at a microscopic level rather than only through discrete crossing outcomes. This transition has made it possible to characterize not only whether a pedestrian crosses, but also how the crossing is performed in space and time, including trajectory shape, speed adaptation, waiting process, and interaction with nearby vehicles [57, 73–75].

Trajectory-based studies have been particularly useful for analyzing pedestrian crossing behavior such as crossing decisions and pedestrian-vehicle interactions at unsignalized and mid-block locations. For example, vision-based methods have been used to estimate pedestrian trajectories and derive crossing speed and accepted gap measures at mid-block crossings [73]. Other studies have used video-derived trajectories to investigate more complex interaction processes, such as secondary pedestrian-vehicle interactions at non-signalized intersections or conflict evolution during crossing events [58, 74]. These studies demonstrate that trajectory data provide a more precise quantitative measurement and finer temporal, spatial resolution than aggregated counts or manually coded observations since they preserve the temporal evolution of pedestrian motion and allow behavioral interpretation to be linked with interaction severity and exposure. As a result, trajectory-based approaches now form an important methodological basis for pedestrian crossing behavior analysis, especially when the goal is to connect behavior, operational performance, and safety assessment [75].

Another major theme in the literature is pedestrian heterogeneity. Numerous studies have shown that pedestrian crossing behavior varies across demographic and situational factors, including age, gender, group crossing, pedestrian platoon size, distraction, carried items, and traffic context [76–79]. For instance, observational studies have reported that age, gender, group size, and waiting time are associated with signal violations and dangerous crossings [77, 80,81], while other studies have found that accepted gaps and speed adaptation are influenced by age, vehicle speed, rolling behavior, and repeated crossing attempts [78, 82]. Distraction has also been shown to alter crossing behavior by increasing crossing time and reducing visual attention to traffic [79, 83]. These findings indicate that pedestrians should not be treated as a behaviorally homogeneous group when analyzing crossing dynamics.

However, pedestrian heterogeneity has still been addressed unevenly in the literature. Most existing crossing behavior studies focus on pedestrians without AWDs or on broad demographic categories such as age and gender, whereas pedestrians with AWDs remain underrepresented [84, 85]. Research involving pedestrians with AWDs has highlighted substantial differences in walking speed, spacing, and negotiation of the built environment, but such

work remains relatively limited and is often conducted in controlled environments rather than in naturalistic street-crossing contexts [84, 86]. A recent scoping review further noted that disability remains an understudied dimension in pedestrian road safety research, despite evidence of elevated collision and injury risks among disabled pedestrians [85]. Therefore, there is still a need for empirical crossing behavior studies that explicitly consider heterogeneous pedestrian populations, particularly those with AWDs, in real-world urban crossing environments.

Overall, pedestrian crossing behavior research has traditionally examined multiple aspects, such as compliance, responses to traffic lights, and crossing speed. With the availability of automated generating trajectory data, these behaviors can now be analyzed in a more detailed and automated manner based on continuous spatiotemporal information. Nevertheless, important gaps remain regarding how underrepresented pedestrian subgroups behave in real crossing environments, particularly pedestrians with AWDs and the behavioral differences across device types. Addressing these gaps is essential for developing a more inclusive understanding of pedestrian crossing behavior and for supporting the design and evaluation of safer pedestrian facilities.

CHAPTER 3 THE PROCESS OF THE RESEARCH

This dissertation follows an article-based thesis structure. The four main chapters (Chapters 4 to 7) of the dissertation correspond to published or submitted research articles, each addressing a specific aspect of the overall research problem. Together, these studies contribute to evaluate the reliability of vision-based object detection and tracking methods for transportation applications, including road safety analysis and crossing behavior analysis specifically.

3.1 Questioning the reliability of deep learning-based trajectory extraction for safety analysis

Deep learningbased object detection and tracking methods have recently been widely used to extract trajectories from video data for traffic safety analysis. These trajectories enable the computation of safety indicators, and allow large-scale analyses without manual trajectory annotation. However, most studies evaluate detection and tracking methods using conventional computer vision metrics, such as detection accuracy or tracking performance, rather than examining their impact on safety indicators. As a result, it remains unclear whether trajectories generated by these methods are sufficiently reliable for trajectory-based safety analysis.

This observation raised the first sub-question: *how do detection and tracking errors affect the safety indicators computed from predicted trajectories?* In Chapter 4, the first study of this dissertation evaluates the reliability of a safety indicator, TTC, derived from automated trajectories by comparing them with those obtained from ground-truth trajectories on a public vehicle-mounted dataset KITTI [7].

3.2 From statistics-based analysis to interaction-matched analysis

The first study evaluated the impact of detection and tracking errors on trajectory-based safety indicators by comparing TTC computed from predicted trajectories with those obtained from ground-truth trajectories. The results showed that automated trajectory extraction can influence the outcomes of safety analyses. However, several limitations were identified. First, the analysis relied on a vehicle-mounted dataset, whereas many traffic safety studies are based on roadside video recordings with different viewpoints and interaction patterns. Second, the comparison between predicted and ground-truth trajectories

was conducted mainly at an aggregate level and interactions between ground-truth and predicted trajectories were not explicitly matched, which does not fully capture differences at the interaction level.

These limitations lead to a refinement of the above sub-question: *How well do the interactions predicted by computer vision methods match the ground truth and how well do computer vision methods perform on a roadside dataset?* In Chapter 5, using roadside video data from the LUMPI dataset [8], an interaction-matching method is developed to associate interactions identified in predicted trajectories with those observed in ground truth, enabling more comprehensive comparisons of TTC and PET values.

3.3 Recognizing the importance of underrepresented vulnerable road users

The first two studies focused on evaluating the reliability of automated trajectory extraction for safety analysis. However, pedestrians with AWDs are underrepresented in existing computer vision datasets used to train and evaluate detection and tracking algorithms. As a result, the performance of deep learningbased methods for identifying and tracking these users remains largely unknown. To the best of the authors knowledge, publicly available datasets featuring pedestrians with AWDs remain scarce.

This limitation motivates the third research sub-question: *Do recent detection and tracking methods perform well enough on videos of pedestrians with AWDs?* Chapter 6 investigates the performance of some recent detection and tracking algorithms for pedestrians with AWDs through the creation and benchmarking on a dedicated dataset.

3.4 Behavioral analysis of pedestrians with AWDs in real-world intersections

The previous study examined whether deep learningbased detection and tracking methods can reliably identify pedestrians with AWDs by constructing a dedicated dataset and benchmarking several recent algorithms. Once this methodological foundation was established, the final sub-question is formulated as follows: *Are automated trajectory methods able to perform well when investigating the real-world behavior of pedestrians with AWDs*

In Chapter 7, the study therefore applies video-based trajectory analysis at a signalized intersection to compare the crossing behavior of pedestrians with and without AWDs. This case study extends the research from methodological evaluation to the practical application of automated video analysis for studying the behavior of vulnerable road users in real traffic environments.

3.5 Summary

As a summary, Figure 3.1 presents the **pipeline** for data processing and the relationship among **objectives** and **articles**. The research in this dissertation relies on a unified pipeline designed to extract trajectories from video data and compute safety or behavioral indicators.

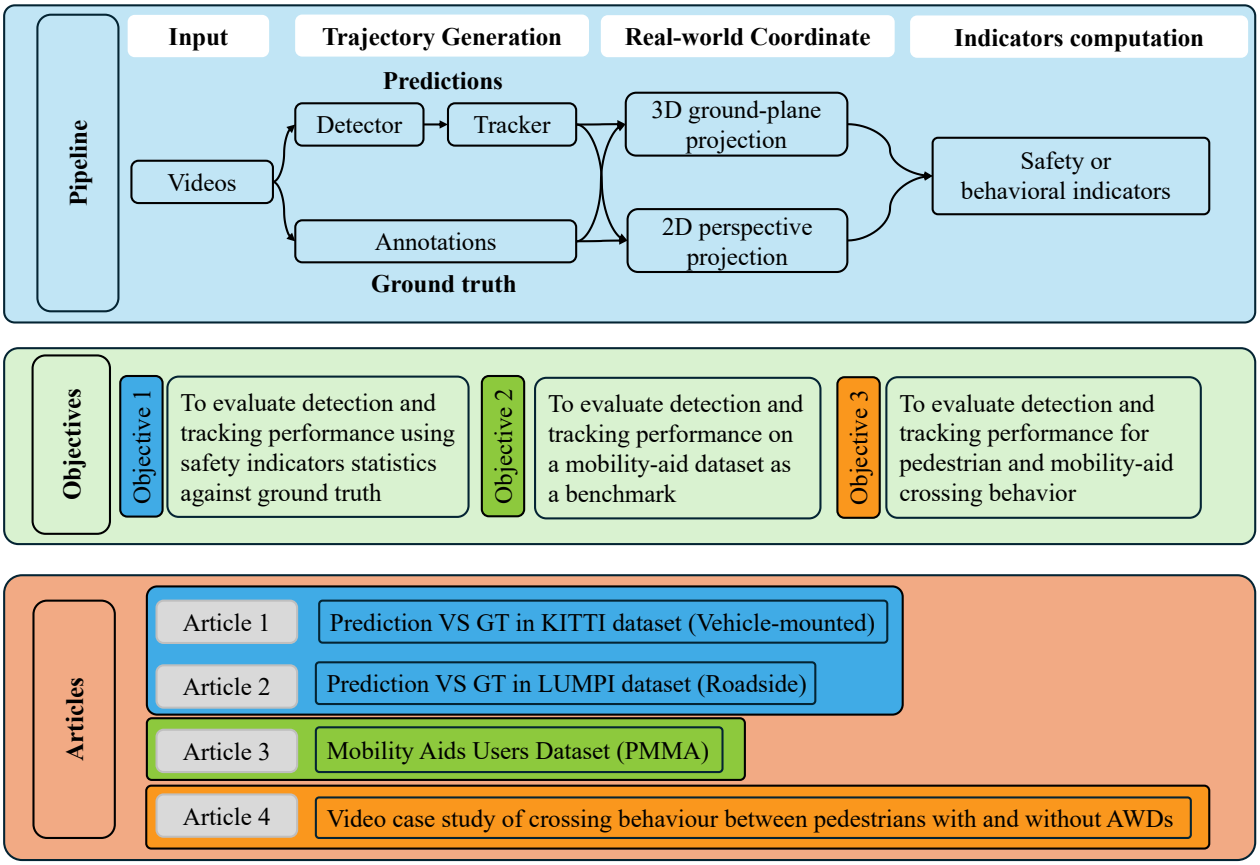


Figure 3.1 Overview of the research process and the relationship between the thesis objectives and the included articles. Matching colors between objectives and articles indicate that a given article addresses the corresponding objective. Articles 1 and 2 both contribute to Objective 1.

The pipeline has four main components:

1. **Input**, where video data serve as the primary input source;
2. **Trajectory generation**, where trajectories derived from predictions are obtained using object detection and tracking methods and ground truth trajectories are directly loaded from manual annotations;

3. **Real-world coordinate transformation**, where 3D trajectories can be projected to a ground-plane coordinate for further analysis, and 2D trajectories are transformed from image coordinate into real-world coordinates using perspective matrices; and

4. **Indicator computation**, where safety or behavioral indicators are calculated. These indicators include TTC and PET, as well as behavioral metrics, such as crossing speed, and speed-density fundamental diagrams.

Table 3.1 Summary of the data sources, locations, and data collection periods used in the thesis.

Dataset	**Article**	**Data source**	**Location**	**Data collection period**
KITTI	Article 1	Public vehicle-mounted dataset	Karlsruhe, Germany	Collected in 2011
LUMPI	Article 2	Public roadside dataset	Roadside traffic scenes in a large junction in Hanover, Germany	Collected in 2022
PMMA	Article 3	Public roadside dataset, collected for this thesis	Outdoor parking lot at Polytechnique Montréal, Montréal, Canada	Collected in April, 2025
Real-world intersection dataset	Article 4	Dataset collected for this thesis	Intersection of Côte-des-Neiges Road and Van Horne Avenue, Montréal, Canada	Collected in August, 2025

To clarify the data sources used in this thesis, Table 3.1 summarizes the datasets, data collection locations, data sources, and collection periods when available. The thesis relies on both public datasets and datasets collected specifically for this research. The KITTI and LUMPI datasets were used to evaluate detection and tracking methods for trajectory-based safety analysis, while the PMMA dataset and the real-world intersection dataset were used to investigate pedestrians with AWDs.

The ground-truth data used in this thesis were obtained in two different ways, depending on the dataset. For the KITTI and LUMPI datasets, the official manual annotations provided with the datasets are the ground truth. For the PMMA dataset and the real-world intersec-

tion dataset, the ground truth was generated for this research through manual annotation using the CVAT annotation tool [87].

CHAPTER 4 ARTICLE 1: HOW GOOD ARE DEEP LEARNING METHODS FOR AUTOMATED ROAD SAFETY ANALYSIS USING VIDEO DATA? AN EXPERIMENTAL STUDY

Qingwu Liu[1,2], **Nicolas Saunier**[1], **and Guillaume-Alexandre Bilodeau**[2]

[1] Civil, Geological and Mining Engineering Department, Polytechnique Montreal

[2] Computer Engineering and Software Engineering Department, Polytechnique Montreal

Presented in the conference *Transportation Research Board, 2024*

Submitted: 31 July 2023 / Accepted: 11 October 2023

DOI: https://doi.org/10.48550/arXiv.2503.09807

Abstract

Image-based multi-object detection (MOD) and multi-object tracking (MOT) are advancing at a fast pace. A variety of 2D and 3D MOD and MOT methods have been developed for monocular and stereo cameras. Road safety analysis can benefit from those advancements. As crashes are rare events, surrogate measures of safety (SMoS) have been developed for safety analyses. (Semi-)Automated safety analysis methods extract road user trajectories to compute safety indicators, for example, Time-to-Collision (TTC) and Post-encroachment Time (PET). Inspired by the success of deep learning in MOD and MOT, we investigate three MOT methods, including one based on a stereo-camera, using the annotated KITTI traffic video dataset. Two post-processing steps, IDsplit and SS, are developed to improve the tracking results and investigate the factors influencing the TTC. The experimental results show that, despite some advantages in terms of the numbers of interactions or similarity to the TTC distributions, all the tested methods systematically over-estimate the number of interactions and under-estimate the TTC: they report more interactions and more severe interactions, making the road user interactions appear less safe than they are. Further efforts will be directed towards testing more methods and more data, in particular from roadside sensors, to verify the results and improve the performance.

Keywords: monocular camera, stereo camera, road user detection and tracking, surrogate measures of safety

4.1 Introduction

According to the World Health Organization (WHO), road injury ranked 10^{th} among leading causes of death in both upper-middle-income and lower-middle-income countries and seventh in low-income countries in 2019 [88]. The impact of road crashes on society is considerable, from death and hospital treatment costs to traffic congestion and infrastructure repairing costs. The traditional approach to road safety relies on the analysis of historical crash data. Since crashes are relatively rare and random, they require long observation periods before any analysis can be carried out and counter-measures implemented [89]. This is why researchers developed proactive methods that do not require to wait for crashes to occur before road safety issues may be diagnosed. The first such kind of methods were the traffic conflict techniques (TCTs) initiated in the late 1950s by researchers at General Motors [90]. These techniques belong to the more general field of surrogate measures of safety (SMoS) that have drawn considerable interest with the development of more automated data collection methods that alleviate some of the shortcomings of the early manual methods [65, 91]. The most common SMoS are derived from the observations of user interactions and conflicts and the most frequent approach is to characterize the severity of conflicts and count the numbers of the most severe conflicts. The severity is assumed to reflect both the proximity of a conflict to a crash and the severity of the potential crash. Various objective and subjective indicators have been proposed to measure severity, such as Time-to-Collision (TTC) [90] and Post-encroachment Time (PET) [92]. These indicators require the trajectories of road users, which are most commonly extracted more or less automatically from video recordings.

Given their low cost and conveniences, cameras have been widely utilized in traffic data collection. Based on the camera configurations, camera setups can be categorized as monocular, stereo or multi-view. For stereo cameras, the disparities between the left and right images can be utilized to generate accurate depth information. Depth plays an important role on accurately measuring the position of the road users and their direction of motion. While monocular camera cannot provide depth information, neural networks can be trained to learn a representation that distills depth directly, which makes it possible to estimate depth from monocular camera without disparity information. For the object detection and tracking tasks, the depth information is implicit and encoded as the 3D coordinates of objects. For example, 3D CenterNet and 3D CenterTrack regard the depth as an output for generating 3D bounding boxes [28, 42]. In this study, we selected one stereo camera-based method and two monocular camera-based methods to investigate their performance and whether stereo information can help road user detection, tracking and safety analysis.

The main objective of this research is to investigate the usefulness of stereo information and

the performance of state-of-the-art deep learning detection and tracking methods to calculate the TTC. Our contributions can be summarized as follow:

- We propose a deep learning-based object detection, tracking and safety analysis framework;
- We compare three tracking methods, using monocular and stereo image data, to compute a safety indicator, namely the TTC.

4.2 Literature review

4.2.1 Object detection

Before the advent of deep neural networks (DNNs), the object detection algorithms followed a basic architecture: 1) an image as the input is passed into the model; 2) sliding windows with different sizes and ratios generate regions for analysis; 3) hand-crafted features are extracted for each interesting region, such as Histograms of Oriented Gradients (HOG) [18], Haar wavelets [93] and Scale-Invariant Feature Transform (SIFT) [94], 4) object categories are evaluated for each candidate region by a classification algorithm, such as Support Vector Machine (SVM) [19] and Adaboost [17].

With the increase of computing power and the occurrence of many large image datasets, methods based on hand-crafted features were gradually replaced by DNN-based object detection methods. Convolutional Neural Network (CNN)-based object detection methods can generally be classified into 1) multi-stage detectors, such as Region-based Convolutional Neural Network (R-CNN) [3], Fast R-CNN [95] and Faster R-CNN [21], and 2) single-stage detectors, such as Single Shot multiBox Detector (SSD) [25] and the You Only Look Once (YOLO) detector series [22, 24]. While these bounding box-based methods provided improvements, they have the drawback of using anchors that limits the number of objects that can be detected in close proximity. In CornerNet [96], the authors proposed a keypoints-based method as an alternative way to represent bounding boxes by keypoints for object representation, which showed a better performance and faster speeds. Two object corners were detected for object location. Following the idea of representing object by keypoints, CenterNet [28] represents each object by a single point, its center point, making CenterNet the state of the art (SOTA) when it was published.

4.2.2 Object tracking

According to the number of objects to be tracked, object tracking can be separated into two tasks: 1) Visual Object Tracking (VOT), which aims to locate only one object in the frames of a video given the ground-truth information in the first frame and 2) Multiple Object Tracking (MOT), which, instead of tracking a single object, detects all objects of interests and maintains their tracked identities [97] throughout a video. In the following part, we will mainly cover methods for MOT, since several road users are involved when calculating SMoS and no ground truth is available to initiate tracking in large traffic datasets.

Deep learning-based tracking by detection methods generally follow four basic steps: 1) object detection, 2) feature extraction/motion prediction, 3) affinity calculation and 4) data association [98]. For step 2), CNNs-based methods are widely used for the feature extraction step. As one of the first groups who utilized CNN for feature extraction, a pretrained CNN was applied for extracting features from the detections [99], ranking best when it was published. Following the same inspiration, a custom residual CNN was used in SORT [35] and DeepSORT [38] to extract visual information and a Fast R-CNN based on VGG-16 [100] was utilized for visual features extractions [101]. Siamese networks [102] are trained CNN with loss functions and remove the loss layer for extracting features. To predict the motions of objects, [103] employed a correlation filter to generate a response map that predicts the new positions of objects in next frame. For step 3), cosine distance between the extracted features is commonly used for the affinity calculation step [38, 104]. For step 4), apart from the Kalman filter [105] and its derived methods that treat association as an estimation and prediction problem, the association of objects among consequent frames can be regarded as a graph bipartite matching problem. Hungarian algorithm [37] is widely utilized to solve such a graph bipartite matching problem. Besides, apart from tracking by detection methods and detection free tracking (DFT), joint object and tracking methods train a single neural network model for both object detection and tracking task. For example, CenterTrack [42] was proposed based on the remarkable performance of CenterNet [28], achieved the simultaneous detection and tracking with a single CNN-based architecture and has inspired the following researches. In this study, we investigate the performances of both tracking methods on road safety analysis.

4.2.3 Surrogate measures of safety

Road safety analyses have generally relied and still rely on crash data. Crashes are rare, random, and there is a lack of information in crash reports on the several factors that lead to a crash such as the behaviors of the involved road users. Waiting for crashes to diagnose

safety is intrinsically reactive, there is a lot of interest and research in proactive methods, in particular to analyze interactions and near-miss traffic events. Instead of dealing with crashes data, traffic conflict techniques (TCTs) are based on the study of traffic events, whose types are related and represented by the famous safety pyramid, shown in Figure 4.1. Surrogate measures of safety such as the number of near misses can be derived from the direct observation of traffic events.

Compared with crash-based methods, traffic event-based methods, including SMoS, show many advantages, such as shorter periods of data collection, smaller collecting costs and richer data. Generally, the most common safety indicators include 1) speed and acceleration, 2) time-to-collision (TTC), which is defined as the time remaining until two road users collide if their movements remain unchanged [106], and 3) post-encroachment time (PET), which is defined as the duration between the leaving time of the first road user and the arriving time of the second road user in the crossing zone [2, 55].

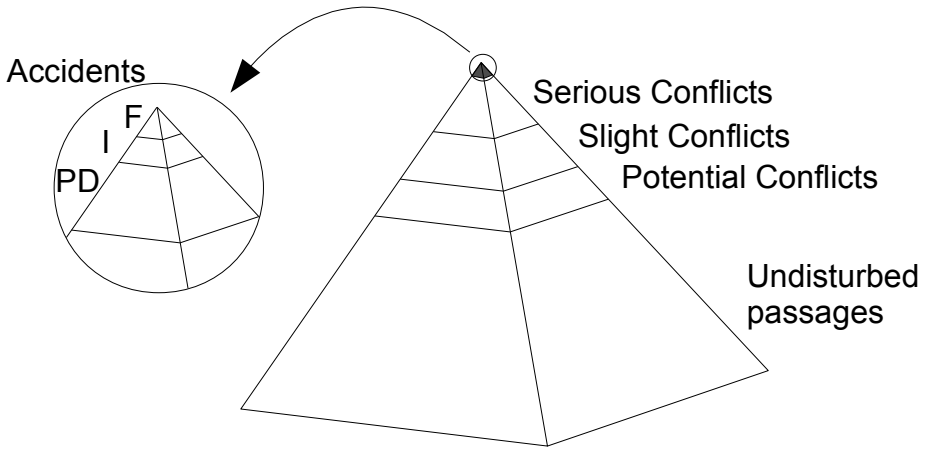


Figure 4.1 The safety pyramid, adapted from [2]. F refers to the fatal crashes, I to injury crashes and PD to property damage only crashes.

4.2.4 Data extraction methods for SMoS

At the early stage of traffic conflict techniques (TCTs), manual data collection methods have played the main role in traffic safety researches from the 1960s to the 1990s. Manual data collection works are still employed in some recent studies. For example, in order to validate SMoS, a manual identification method was adopted to extract the information of the encounters in six selected intersections [107]. Then the trajectories of each encounter were manually annotated in the T-Analyst software [108]. However, it is time-consuming to extract data manually, especially for large/long-time recording data. The demand for more

efficient trajectories generation methods pushed the developments of more or less automatic methods to generate trajectories.

In 2012, an open-source "Traffic Intelligence" (TI) project was released, which contains a feature-based tracker for road user detection and tracking, as well as other utilities for trajectory and road safety analysis [9]. Many projects have relied on TI to study the safety of various facilities such as cycle tracks, highway lane markings, roundabouts and automated shuttles. These case studies generally require to further improve the accuracy of the generated trajectories: for example, the results generated by the TI tracker [109] were manually verified and corrected.

In the 2010s, several smartphone GPS-based trajectory extracting methods were studied. Johnson et al. developed a smartphone-based system to identify aggressive driving behaviors of drivers in three vehicles [110]. GPS-enabled smartphones are utilized to collect the traffic data for calculating SMoS [111]. However, those smartphone based methods has a drawback that no enough data will be collected where the sites/locations are difficult to reach.

4.3 Methodology

4.3.1 Overview

In this study, we develop a framework for safety analysis using either monocular and stereo cameras. Three tracking methods are tested in the proposed framework. Trajectories are first generated from three deep learning-based object detection and tracking methods that differ in terms of 1) the type of input, monocular or stereo images, and 2) the type of tracking method, tracking by detection or joint detection and tracking. Secondly, as opposed to most existing methods, in which the objects are represented by their centroids, the real world coordinates of the road user volumes (3D boxes) are used thanks to the coordinate transferring parameters [7]. The safety indicators are then computed from each vehicle ground level bounding box, also called the bird-eye view (BEV). The safety analysis framework is shown in Figure 4.2. It is composed of four parts:

1. dataset and inputs,
2. detection and tracking,
3. coordinates projection and trajectories generation, and
4. safety indicator computation.

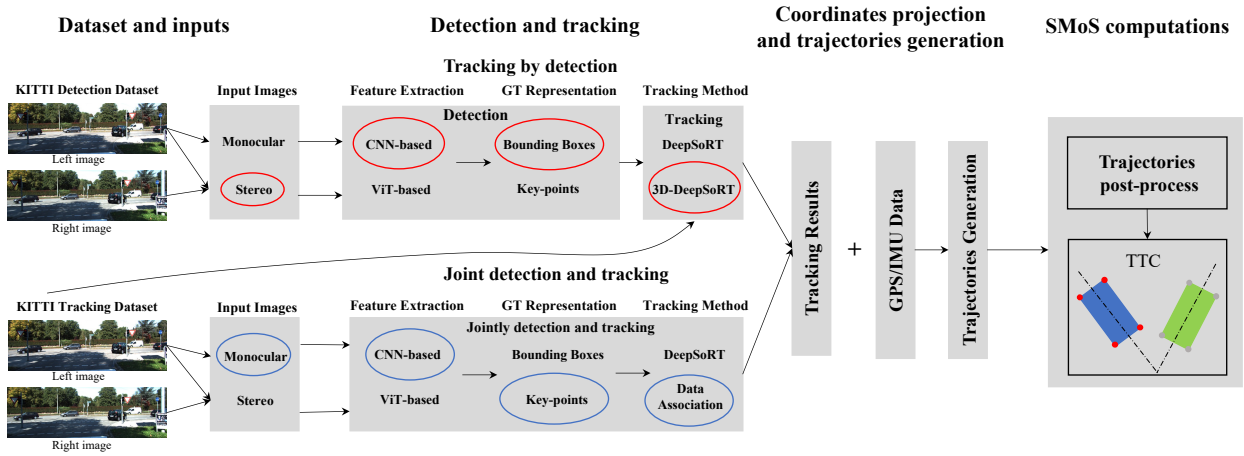


Figure 4.2 Processing pipeline for deep learning-based road safety analysis. GT represents ground truth, CNN represents convolutional neural network [3] and ViT represents vision transformer [4]. IMU represents inertial measurement unit. The examples for two object detection and tracking methods, that perform both steps either separately or jointly, are highlighted by the red ellipses and the blue ellipses, respectively. Details of the trajectory post-processing steps are illustrated in Figure 4.3.

4.3.2 Object detection and tracking

Among the three object detection and tracking methods, the first two belong to the category of tracking by detection, and the third performs joint detection and tracking. They are summarized in Table 4.1 and described below:

Table 4.1 Details of the three methods evaluated for generating 3D object detection and tracking results.

Detection	Tracking	Category	Input	Feature Extractor	Object Representation
MonoDTR	3D-DeepSORT	Tracking by detection	Monocular	ViT	Bounding boxes
YoloStereo3D	3D-DeepSORT	Tracking by detection	Stereo	ResNet	Bounding boxes
CenterTrack		Joint detection and tracking	Monocular	Stacked hourglass	Keypoints

1. **MonoDTR**: As the first method, we select MonoDTR because it is based on a monocular input and it utilizes a depth-aware transformer to achieve 3D object detection [112]. In MonoDTR, a novel depth positional encoding was implemented to deal with the problem of weak depth positional performance in transformers. 3D tracking is performed with a modified version of DeepSORT [38] by computing 3D intersection over union (IoU) instead of 2D IoU;

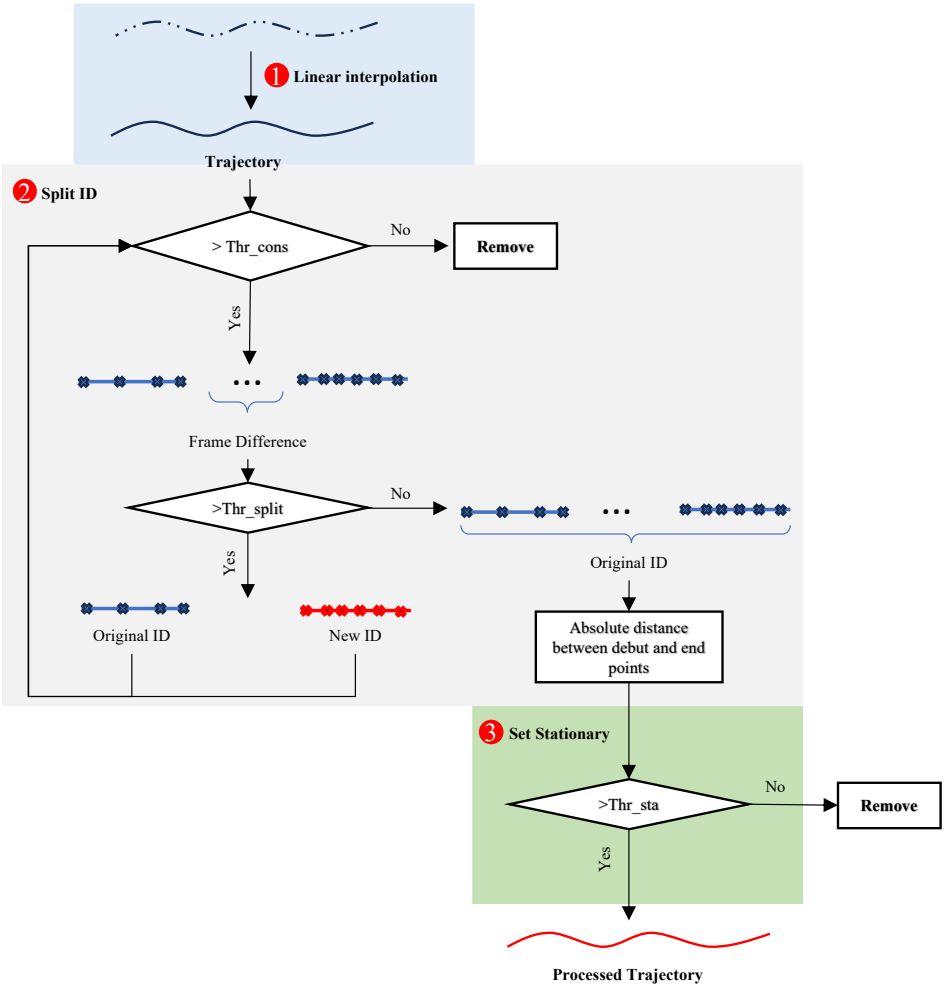


Figure 4.3 Post-processing steps for road safety analysis.

2. **YoloStereo3D**: As the second method, we choose YoloStereo3D [113] for detection as it is a stereo image-based 3D object detection method, which relies on ResNet [32] as the backbone and utilizes the ghost module [33] to improve the accuracy of 3D object detection. Tracking is also performed with the 3D version of DeepSORT;

3. **CenterTrack**: As a third method, we choose 3D CenterTrack [42] that conducts joint detection and tracking from monocular image inputs and generates 3D tracking results akin to the two previous methods.

The outputs of the tracking module follow the format of KITTI [7] and include the 3D center coordinates, 3D box size and the object rotation angles around the y-axis, which are employed for coordinate projection and trajectory generation.

4.3.3 Post-processing steps for safety analysis

In the videos recorded by fixed roadside cameras, a static zone is generally selected as the analysis zone where every pair of coexisting road users is analyzed as an interaction [91, 109, 114]. However, videos sequences in the KITTI dataset were recorded by a vehicle-mounted sensor system [7], which means that such a fixed analysis zone would contain few interactions. In this study, the same definition of interaction is used, where all pairs of simultaneously tracked road users are analyzed, including with the moving vehicle collecting the data. TTC is computed for each interaction at each instant.

Most video-based methods used for SMoS represent the road users by their centroids [109, 115] or polygons based on appearance features. This study takes advantage of the chosen methods where objects are represented by ground-level bounding boxes (BEV). TTC can be computed at each instant and depends on a motion prediction method. The simplest method is constant velocity, where the road users are assumed to continue moving in the same direction at constant speed. More realistic and robust methods have been proposed [114] and are used [109], but this study starts with the simple constant velocity method. TTC is the first time instant in the future at which the BEV bounding boxes of the two road users overlap, with a maximum value of 10 s. Although TTC cannot be generally computed at each instant, i.e., the road users may not be currently on a collision course at constant velocity, there can be several TTC measures at different instants for a given interaction. Various statistics can be derived to represent the interaction severity, with the minimum value TTC_{min} being one of the most commonly used [2].

To improve the performance of some detection and tracking methods and to investigate the factors influencing the TTC, we developed the following post-processing steps for the trajectories:

1. In some complex traffic scenarios, however, the data association step may connect different road users if the features of those road users are found to be similar, even when there are large temporal gaps between those tracked road users, e.g., over 40 frames. This kind of tracking error leads, after the next step if these objects are kept, to the generation of a large amount of interactions, which results in a dramatic increase of the amount of dangerous interactions. We therefore apply a trajectory splitting process to handle this problem and compare the results with and without the track splitting process. The track splitting process, called IDsplit in the results, is made of two steps: 1) given a splitting threshold Thr_{split}, trajectories with a temporal gap larger than Thr_{split} are split at each gap ($Thr_{split} = 10$ in this study). 2) Given a threshold Thr_{cons}

on the number of consecutive frames, the new trajectories are removed if they have fewer frames than Thr_{cons}, which is selected as 3 in this study. This situation occurs only for tracking-by-detection methods. Since CenterTrack does not connect objects with such long gaps, this step is not applied to CenterTrack.

2. In cases of missed detections, for example, caused by occlusions, the road user coordinates may be missing for several frames. For the analysis, they are imputed through linear interpolation.

3. The implemented tracking methods detect and track stationary road users as well. Yet, even in the ground truth, the coordinates of these road users will not be exactly constant, resulting in apparent slight random motions and more interactions with measurable TTCs. To explore the impact of these slight position fluctuations on the safety indicators, the last post-processing step, called SS in the results, consists in setting the positions and velocities to the mean position and zero, respectively, if the absolute distance between the debut and end of the x and y coordinates is smaller than Thr_{sta}, set to 2.0 m in this study. The interactions are put in two categories, whether the interaction involves a stationary road user, for further analysis.

4.4 Datasets and evaluation metrics

4.4.1 Datasets and evaluation metrics

The following datasets are used in the experiments:

KITTI detection: The KITTI detection benchmark consists of 7,481 training frames and 7,518 test frames. Those sequences are collected by a camera mounted on a car driving through different traffic scenarios. Following the splitting procedure in [116], the available training set is split into 3,712 training frames and 3,769 validation frames in order to fine-tune YoloStereo3D and MonoDTR to track cars, pedestrians and cyclists instead of only cars and pedestrians [117].

KITTI tracking: The KITTI tracking benchmark consists of 21 training sequences and 29 test sequences [7]. Those sequences are collected by a camera mounted on a car moving through traffic similarly to the KITTI detection dataset. The KITTI tracking ground truth consists of 2D information in pixel coordinates, 3D information in camera coordinates, including height, width, length and the coordinates of the bottom centers of the 3D bounding boxes for cars, pedestrians, and cyclists. Videos are captured with a rate of 10 frames per second (fps).

In addition to the analysis of safety indicators, we also compute their detection and tracking performance to better understand how it relates to road safety indicators. The object detection performance of YoloStereo3D and MonoDTR is measured in terms of mean Average Precision (mAP) on the validation dataset. The tracking performance is measured using the standard CLEAR MOT metrics [118]. The main metrics, MOTA and MOTP, are computed using the numbers of false negative FN, false positives FP, identity switches IDs and ground truth objects GT. They rely on the matching of the ground truth objects to the output of each tracking method, generally using the Hungarian algorithm on a cost matrix based on the distance between object centers in image space. We also use the mostly tracked MT, mostly lost ML, $IDF1$, $MODA$ and $MODP$ as metrics [118]. MOTA and MOTP are computed using the following equations:

$$MOTA = 1 - \frac{FN + FP + IDs}{GT} \tag{4.1}$$

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \tag{4.2}$$

where $d_{t,i}$ is the distance between the position of a ground-truth object i and its matched predicted object. c_t is the number of matches made between ground truth and the detection output in frame t. IDF_1, $MODA$ and $MODP$ are computed as:

$$IDF_1 = \frac{2TP}{2TP + FP + FN}, \tag{4.3}$$

$$MODA = \frac{|TP| - |FP|}{|TP| + |FN|} \tag{4.4}$$

$$MODP = \sum_{t=1}^{N} \frac{\sum_i^{c_t} IoU_i^t}{c_t} \tag{4.5}$$

where N represents the number of frames and IoU_i^t the intersection over union (IoU) of the ground truth object i with its matched predicted object at frame t.

4.5 Results and discussions

4.5.1 Detection and tracking performance

YoloStereo3D and MonoDTR are fine-tuned for three categories, car, pedestrian and cyclist. For training, we used a computer with a 32GB NVIDIA Tesla-V100 GPU and dual Intel(R) Xeon(R) Silver 4116@2.10GHz CPUs. CenterTrack was retrained using a computer with four

32GB V100 Volta GPUs and dual Intel Silver 4216 Cascade Lake@2.1GHz CPUs. The main training parameters, such as epoch and learning rate, have the same values as the original methods. The results are generated on the same computers as for the training process.

The object detection performance of YoloStereo3D and MonoDTR is presented in Table 4.2. YoloStereo3D seems superior to MonoDTR as it is close to MonoDTR for cars (except for the hard cases) and well above MonoDTR for the pedestrian and cyclist classes.

Table 4.2 Detection performance (mAP) on the training dataset for the three classes (easy, moderate and hard represent the difficulty of detection with respect to the severity of occlusions).

Classification	**Car**			**Pedestrian**			**Cyclist**		
Method	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MonoDTR	**99.3**	**90.84**	**80.9**	30.8	22.6	19.2	22.9	17.1	14.3
YoloStereo3D	97.1	86.7	69.2	**81.1**	**54.7**	**41.4**	**70.4**	**45.4**	**34.6**

For the two object detection methods, MonoDTR and YoloStereo3D, the tracking results are generated by 3D-DeepSORT with the detection results as inputs. For CenterTrack, the training dataset, containing 21 sequences, is split repeatedly into training and validation set using cross validation. Specifically, the training dataset is divided into eighteen sequences as training set and three sequences as validation set, which is repeated seven times. The tracking performance of three methods on the car, pedestrian and cyclist object categories are shown in Table 4.3. CenterTrack performs the best based on MOTA by a wide margin, and on most other metrics.

4.5.2 Safety analysis

Number of interactions

This section presents the results on the number of interactions where TTC can be computed. Two sequences are removed from the 21 training sequences of the KITTI tracking dataset: sequence 12, where no TTC_{min} are below 10 s for YoloStereo3D and the ground truth, and sequence 18, where no TTC_{min} are below 10 s for YoloStereo3D after the post-processing steps. We present detailed results for eight sequences (1, 3, 5, 7, 8, 17, 19 and 20) and the total of 19 sequences in Table 4.4. Sequences 1 and 7 were recorded in a scenario with many parked cars. Notably, there are five curved roads in sequence 7. Sequences 3 and 5 were recorded in simpler scenarios with few stationary road users. Sequence 8 was in a scenario with 368 frames, with both moving and stationary road users. Sequence 17 is a scenario with

Table 4.3 Tracking performance results for MonoDTR (MonoDTR+3D-DeepSORT), YoloStereo3D (YoloStereo3D+3D-DeepSORT) and CenterTrack (3D CenterTrack) on cars, pedestrians and cyclists.

(a) Tracking performance for cars.

	MOTA↑	MOTP↑	MODA↑	MODP↑	MT↑	ML↓	FP↓	FN↓	IDF1↑	IDSW↓	FRAG↓
MonoDTR	50.0	88.2	48.8	90.5	22.1	20.4	11.5	39.7	70.5	1042	1423
YoloStereo3D	45.7	**88.8**	49.3	**91.2**	22.6	27.5	6.96	43.7	69.1	963	1199
CenterTrack	**84.1**	88.4	**85.3**	90.4	**85.2**	**0.03**	**4.28**	**10.3**	**93.1**	**349**	**514**

(b) Tracking performance for pedestrians.

	MOTA↑	MOTP↑	MODA↑	MODP↑	MT↑	ML↓	FP↓	FN↓	IDF1↑	IDSW↓	FRAG↓
MonoDTR	22.0	68.9	32.4	92.7	14.4	27.0	17.2	50.4	59.6	1161	1505
YoloStereo3D	34.6	69.7	42.6	92.9	14.4	29.9	5.91	51.5	63.0	890	1229
CenterTrack	**66.2**	**77.3**	**68.2**	**93.4**	**53.9**	**7.8**	**3.9**	**27.8**	**82.1**	**215**	**530**

(c) Tracking performance for cyclists.

	MOTA↑	MOTP↑	MODA↑	MODP↑	MT↑	ML↓	FP↓	FN↓	IDF1↑	IDSW↓	FRAG↓
MonoDTR	7.69	82.4	11.0	98.2	13.5	43.2	30.8	58.1	48.5	63	99
YoloStereo3D	40.6	82.3	42.5	**98.4**	24.3	27.0	**7.2**	50.3	63.4	76	86
CenterTrack	**68.4**	**82.6**	**69.7**	97.4	**75.7**	**0**	17.1	**13.3**	**85.2**	**22**	**51**

pedestrians and cyclists crossing the road. Sequence 19 is a complex scenario where only pedestrians and cyclists are moving slowly in a crowded street. Sequence 20 was recorded with many vehicles moving slowly and stopping in a traffic congestion scenario.

YoloStereo3D performs best with the minimal numbers of interactions with TTC_{min} below 10 and 1.5 s among the three selected methods, although it still generates a considerable amount of interactions compared to the ground truth, especially the most severe with TTC_{min} below 1.5 s (2564 compared to 164 and 966 compared to 80 without and with the post-processing steps, respectively). With both post-processing steps, both YoloStereo3D and CenterTrack perform the best in three sequences (sequence 17, 19 and 20 for YoloStereo3D and sequence 1, 7 and 8 for CenterTrack) below 1.5 s, respectively. Both post-processing steps result in remarkable decreases compared to the initial methods, as presented in Table 4.5. Notably, the SS step has a huge influence on the numbers, even for the ground truth, resulting in reductions of the numbers of interactions with TTC_{min} below 10 and 1.5 s of 55.02 % and 51.21 %, respectively.

TTC values

The performance of each method should also be measured in terms of the TTC values, not just the number of interactions. The cumulative distribution functions (CDF) of TTC_{min} can be compared between each method and the ground truth through the D-statistic of the KolmogorovSmirnov test [119], which is computed as the maximum difference between the CDFs. The D-statistic is computed for each method and each sequence and presented as boxplots in Figure 4.4. Besides, the medians of the TTC_{min} distributions are compared between the output of each method and the ground truth for each sequence in a scatter plot (see Figure 4.5). The absolute differences between the median of each method and the ground truth for each sequence are depicted in a boxplot (see Figure 4.6). A shaded region ± 0.5 s around the $y = x$ line is added to visualize and quantify the similarities of the medians between the three methods and the ground truth. These results are generated after all the post-processing steps.

CenterTrack has the smallest median D-statistic and the smallest inter-quartile range (see Figure 4.4). However, all three methods have high D-statistics, which means that they all exhibit significant disparities with the ground truth. Looking at the boxplots of Figure 4.5, CenterTrack has the smallest quartiles and MonoDTR has the smallest inter-quartile range but these two methods also have one and two outliers, respectively. This further shows how large the TTC_{min} differences are for all methods. One can also look at the medians values themselves in Figure 4.6. First, it is clear that most methods under-estimate the TTC_{min} medians for most sequences, sometimes by very large margins. Only a few points lie close to the equality line ($y = x$): five for CenterTrack, four for YoloStereo3D and two for MonoDTR within 0.5 s from the ground truth. Second, the regression lines for each method are mostly horizontal, which means that all computer vision methods tend to generate similar median TTC_{min}, independently from the true severity of the data (which itself varies considerably).

Table 4.4 Comparisons of interactions with TTC_{min} below 10 and 1.5 s for each method and post-processing step.

Sequence_number	**0001**		**0003**		**0005**		**0007**		**0008**		**0017**		**0019**		**0020**		**Total**	
Method	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s	10s	1.5s
MonoDTR	1022	756	36	32	123	87	911	638	100	48	137	104	983	439	1638	1188	6633	4087
YoloStereo3D	750	588	34	38	128	96	486	288	82	38	63	50	372	124	718	546	4099	2564
CenterTrack	894	377	20	16	170	85	476	143	68	24	71	19	516	214	1140	668	6860	3279
Ground Truth	157	30	10	4	26	8	114	10	14	0	1	0	88	9	138	0	1296	164
MonoDTR+IDsplit	538	322	16	12	133	92	655	339	98	40	90	60	744	213	1269	712	4927	2306
YoloStereo3D+IDsplit	470	276	16	12	137	107	342	124	60	30	46	33	353	67	450	218	3243	1446
MonoDTR+IDsplit+SS	206	182	10	4	26	16	260	128	94	40	59	50	443	115	970	592	2684	1430
YoloStereo3D+IDsplit+SS	192	170	8	8	32	26	112	62	58	30	29	23	205	33	356	200	1685	966
CenterTrack+SS	212	126	10	8	68	26	179	36	42	16	105	79	331	108	834	460	3155	1496
GroundTruth+SS	9	2	4	2	34	10	39	8	16	0	2	0	116	16	142	0	583	80

Table 4.5 Summary of the reductions in the number of interactions with TTC_{min} below 10 and 1.5 s for each method and post-processing step (✗ indicates a post-processing step is not applied).

Post-processing	**IDsplit**		**IDsplit+SS**	
Method	10s	1.5s	10s	1.5s
MonoDTR	25.72%	43.58%	59.53%	65.67%
YoloStereo3D	20.88%	43.60%	58.89%	62.32%
CenterTrack	✗	✗	54.00%	54.38%
Ground Truth	✗	✗	55.02%	51.21%

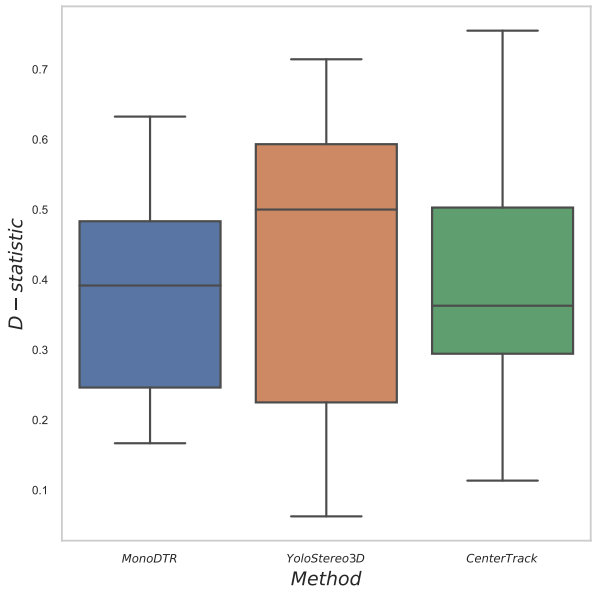


Figure 4.4 Boxplots of the D-statistics for each method for all the sequences.

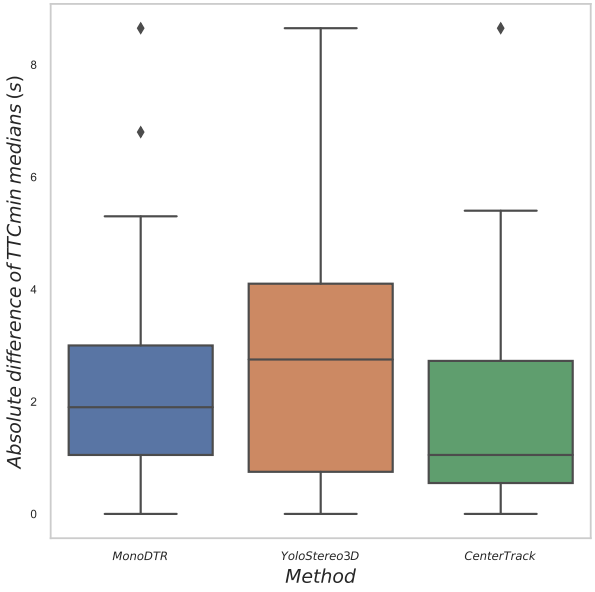


Figure 4.5 Boxplots of the absolute differences of the TTC_{min} medians between each method and the ground truth for all the sequences.

Finally, let us look at the CDFs of TTC_{min} in detail for sequences 1, 7, 17 and 20 in Figures 4.7 to 4.10 respectively. For all the four sequences, the IDsplit step reduces the numbers of interactions for MonoDTR, YoloStereo3D and the ground truth. The probabilities of the

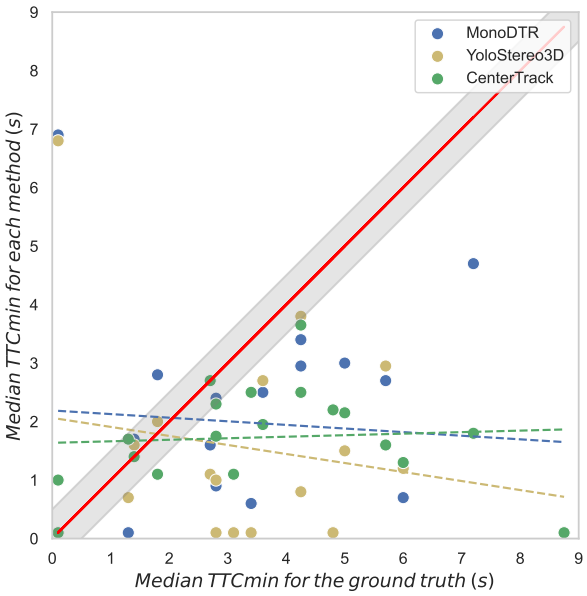


Figure 4.6 Scatter plot of the median of TTC_{min} for each method as a function of the median of TTC_{min} for the ground truth for all the sequences; the red line is the $y = x$ curve and the gray shaded area represents a range of $\pm$ 0.5 s around the red line.

numbers of interactions below 1.5 s meet considerable reductions for sequences 1, 7 and 20, which indicates that many road users involved in interactions with TTC_{min} below 1.5 s have been incorrectly labeled with the same IDs , compared with those involved in interactions with TTC_{min} between 1.5 and 10 s. However, the proportion of those interactions in sequence 17 only meets a slight reduction, which indicates that the IDsplit step does remove some interactions but its impact is not as pronounced as for sequences 1, 7 and 20. With the SS step, the numbers of interactions of all three methods and even the ground truth meet large reductions on all four sequences, which indicates that incorrectly measuring the motion of stationary road users has a large influence on the computing of TTC. The slight position fluctuations can lead to numerous inaccurately computed dangerous interactions.

For sequences 1, 7 and 17, the CDFs of the interactions involving a stationary road user ($Interactions_2$) are more similar to the CDFs for all interactions after the IDsplit and SS steps, which indicates that the contribution of the interactions involving a stationary road user ($Interactions_2$) overweigh the interactions between two moving road users ($interactions_1$). For sequence 20, the CDFs of both $Interactions_1$ and $Interactions_2$ are similar to the CDFs for all the interactions after the two post-processing steps, which indicates that both types of interactions affect the computing of $TTCmin$ in this traffic congestion scenario.

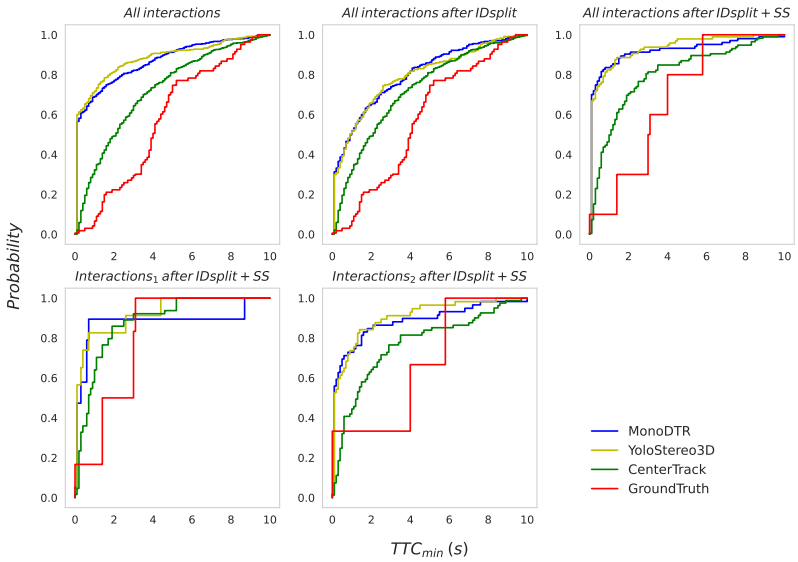


Figure 4.7 CDFs of TTC_{min} for three methods and ground truth for sequence 1. The top three figures show the CDFs for all the interactions without post-processing, after IDsplit, and after IDsplit+SS. The bottom two figures show the CDFs after IDsplit+SS for the interactions between two moving road users ($Interaction_1$), and the interactions involving a stationary road user ($Interaction_2$), respectively.

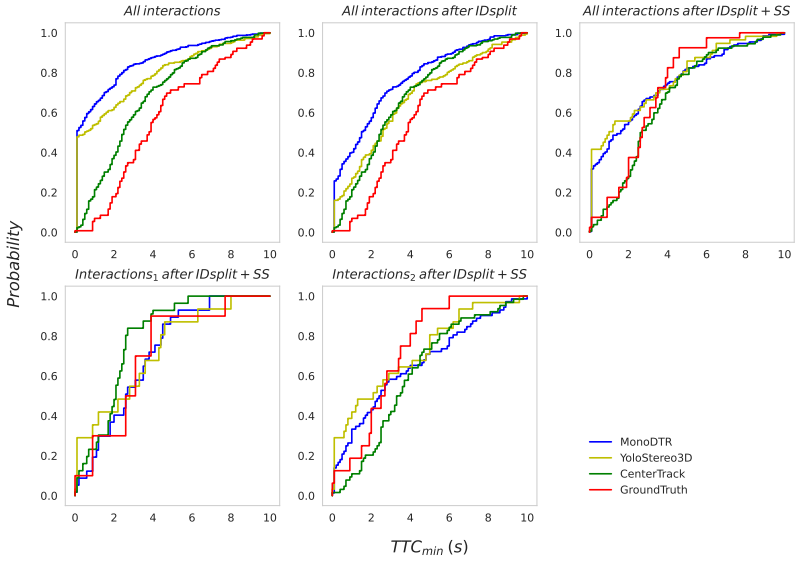


Figure 4.8 CDFs of TTC_{min} for three methods and ground truth for sequence 7 (see description of Figure 4.7).

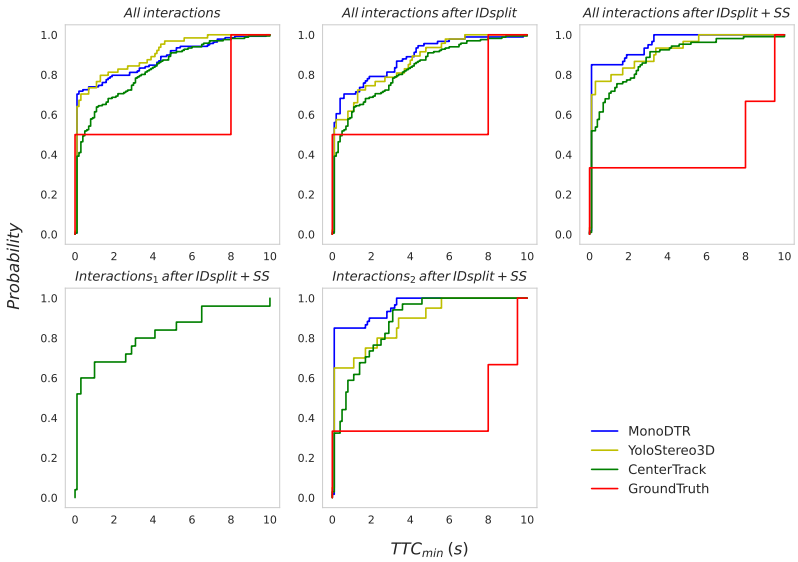


Figure 4.9 CDFs of TTC_{min} for three methods and ground truth for sequence 17 (see description of Figure 4.7). Note there are no TTC_{min} below 10 s in the fourth plot for MonoDTR, YoloStereo3D and GroundTruth.

.

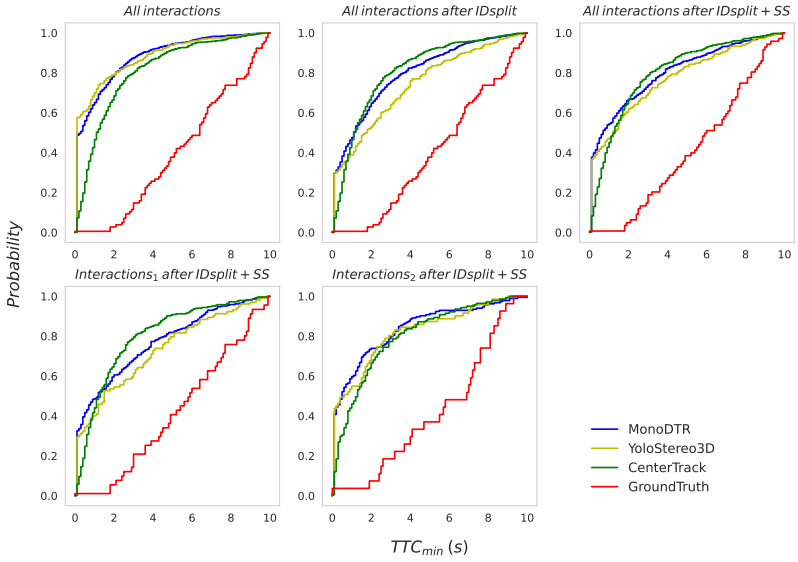


Figure 4.10 CDFs of TTC_{min} for three methods and ground truth for sequence 20 (see description of Figure 4.7).

4.5.3 Discussion

The results are definitely mixed for the three tested methods and the different sensor types. There is no clear winner. Worse than that, one could see all tested methods as losers given how poorly they reflect the safety as measured by TTC.

The first part of the results section reported the performance of the methods for the detection and tracking tasks. Comparing these to the safety performance, we see some relationships, as the best tracking method, CenterTrack, is also the slightly better method for TTC. The second best (and superior for detection), YoloStereo3D, is the best in terms of the numbers of interactions. Yet good tracking performance does not translate in accurate TTC values.

4.6 Conclusion

In this study, we investigated how deep-learning based tracking methods perform for automated road safety analysis. A framework is developed and three different tracking methods, MonoDTR, YoloStereo3D and CenterTrack were selected to compute the road safety indicator TTC on the KITTI tracking dataset. Two post-processing steps are utilized to improve the tracking results and explore the factors that influence the deep-learning based tracking results on TTC. Results show that YoloStereo3D outperforms the others with respect to the numbers of interactions with TTC_{min} below 10 and 1.5 s. However, CenterTrack is slightly better in terms of TTC_{min} values as shown by the D-statistic with the ground truth and the TTC_{min} median. Despite these differences and the improvements brought by the proposed post-processing steps, all methods generate many more interactions with much lower TTC_{min} values than in reality. The comparisons of CDFs show that interactions involving a stationary road user have a larger impact than interactions with two moving road users on the TTC_{min} distribution, except in a traffic jam scenario where both types of interactions contribute equally.

In conclusion, it is difficult to recommend any of the tested methods. The method based on a stereo-camera shows a slight advantage for some sequences in terms of the number of interactions, but has lower performance than a method based on a monocular camera in terms of the TTC values, and all over-estimate the severity of road user interactions.

Future research will focus on computing and reporting more road safety indicators, such as PET, and on further evaluating the vehicle-mounted dataset to better understand why TTC appears to be underestimated. Efforts will be made to explore and understand the reasons behind the substantial disparities between deep learning methods and the ground truth to develop improved road user detection and tracking methods for safety analysis. Stereo and

monocular cameras will be also tested on videos from fixed roadside sensors to verify the current findings.

Acknowledgements

The authors gratefully acknowledge financial support from the China Scholarship Council. The authors would also like to thank the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for the computing servers.

Author Contributions

Qingwu Liu, Nicolas Saunier, and Guillaume-Alexandre Bilodeau jointly conceptualized the research idea. Qingwu Liu implemented the algorithms, conducted the experiments, and wrote the initial draft of the manuscript. Nicolas Saunier provided the Traffic Intelligence codebase and contributed to the interpretation of the results. Nicolas Saunier and Guillaume-Alexandre Bilodeau participated in the discussion and analysis of the experimental results, and reviewed and revised the manuscript. All authors read and approved the final manuscript.

CHAPTER 5 ARTICLE 2: VIDEO ANALYSIS FOR SURROGATE MEASURES OF SAFETY: IS DEEP LEARNING GOOD ENOUGH?

Qingwu Liu[1,2], **Nicolas Saunier**[1], **and Guillaume-Alexandre Bilodeau**[2]

[1] Civil, Geological and Mining Engineering Department, Polytechnique Montreal

[2] Lab. LITIV, Computer Engineering and Software Engineering Department, Polytechnique Montreal

Published to the journal *Transportation Research Part C: Emerging Technologies*

Submitted: 19 May 2025 / Accepted: 20 February 2026

DOI: https://doi.org/10.1016/j.trc.2026.105596

This chapter has been edited based on comments from the PhD jury, with additional clarification of the results where appropriate.

Abstract

Road crashes are one of the leading cause of death and injuries in the world and the situation is still deteriorating in many countries. Surrogate measures of safety (SMoS) are based on the observation of conflicts between road users, which are characterized using indicators such as, Time-to-Collision (TTC) and Post-encroachment Time (PET). These indicators are estimated by human observers or calculated from road user trajectories. Although computer vision methods are used for automated road safety studies, to the best of the authors' knowledge, there has been no research evaluating their performance on safety analyses. Despite the continuous improvement in computer vision methods, most recently with deep learning, automatically understanding some traffic scenarios, such as crowded mixed intersections are still open problems. In this study, we assess the performance of three deep learning-based object detection methods, CenterNet2D, YOLOv8 and CenterNet3D, and one object tracking method, ByteTrack, to derive safety indicators on a crowded real-world dataset by comparing with the indicators obtained from the ground truth trajectories. Results reveal that CenterNet2D outperforms the other two methods. Deep learning methods perform better on PET than TTC. However, there is still a big gap regarding the values of safety indicators compared with the ground truth. Tracking errors result in large differences on the safety indicators evaluation performance in the crowd scenarios. The performance by applying a

pretrained model on another dataset drops dramatically.

Keywords: roadside dataset, video analysis, object detection, object tracking, surrogate measures of safety

5.1 Introduction

As the World Health Organization (WHO) reported in 2023 [120], road crashes cause approximately 1.19 million of deaths each year. Over half of all road deaths are among vulnerable road users, including pedestrians, cyclists and motorcyclists. Before the 1950s, to explore the factors causing road crashes and to reduce road injury rates, safety analyses relied on historical crash data. However, crashes are relatively rare and difficult to capture as they are happening. Even when crash data are collected, they often suffer from quality issues. Many low and middle income countries lack consistent national crash data registers. From the late 1950s, proactive methods, such as traffic conflict techniques (TCTs), have been developed to record conflicts and measure safety [121]. The most common way is to identify the most severe conflicts, near-misses or near-crashes, and derive the expected number of crashes from the number of severe conflicts. Conflict severity is measured through several safety indicators, the most widely used being Time-to-Collision (TTC) and Post-encroachment Time (PET). The computation of both indicators requires the trajectories of road users, which can be extracted by various methods, e.g., with feature-based methods [109, 122], or with computer vision methods [123, 124].

While computer vision methods have seen considerable improvements, most recently with deep learning, they may not be able to handle the most complicated scenarios, such as crowded mixed traffic intersections, due to the difficulty of detecting and tracking objects when they are occluded, densely packed, very small and their trajectories overlap. Improving methods to extract trajectories is still ongoing. For example, several object detectors have been proposed, such as Fast Region-based Convolutional Neural Network (Fast R-CNN) [95], You Only Look Once (YOLO) [22], Single Shot MultiBox Detector (SSD) [25], CenterNet [28] and DEtection TRansformer (DETR) [30] and several trackers as well, such as Simple Online and Realtime Tracking (SORT) [35], deep SORT [38] and ByteTrack [39]. This raises the question, *are these methods good enough now for safety analysis in typical traffic scenarios?*

In this research, we aim to answer this question by applying recent computer vision methods for object detection and tracking to road safety analysis. To the best of our knowledge, there have been no studies that evaluate how computer vision methods perform to derive safety indicators. We apply three deep learning object detection methods, CenterNet2D, YOLOv8

and CenterNet3D, and one object tracking method, ByteTrack, to assess their performance by comparing the safety indicators computed from the automatically generated trajectories with those obtained from the ground truth trajectories.

The main contributions of this study are summarized as follows:

1. We formulate a novel evaluation method based on object matching and interaction matching to compare safety indicators generated from ground truth trajectories and those obtained from computer vision methods;
2. We apply three different deep learning object detection methods and one object tracking method on a crowded traffic dataset recorded from a fixed roadside camera, and evaluate their performance in safety indicator computation tasks; and
3. We compare the performance of 2D and 3D object detections to make recommendations for utilizing computer vision methods for safety analysis.

The remainder of this paper is organized as follows. Section 5.2 presents the related works. Section 5.3 explains the pipeline and formulates the object matching and interaction matching method. Section 5.4 introduces the selected dataset and provides the experiment's information. Section 5.5 presents the results and the discussion, and in the end Section 7.5 concludes the paper.

5.2 Related work

In this section, we firstly summarize the development of surrogate measures of safety (SMoS). Then the existing works about pipelines for computing safety indicators are reviewed. Finally, we review the studies of the existing evaluation metrics.

5.2.1 SMoS and safety indicators

In general, SMoS are derived from the observation of conflicts between road users. A common way is to count the most severe conflicts, and then to estimate the expected number of crashes from this count. The severity of conflicts aims to reflect the proximity or probability of a crash and its potential severity. Conflict severity is measured through various safety indicators, the most common being TTC [125] and PET [92]. TTC is defined for two road users at each instant as the time remaining to a potential collision if their movements remain unchanged [125]. PET is defined as the time between the departure of one road user and the arrival of another at the same point or area of a potential collision [92].

5.2.2 Computing safety indicators from video frames

Various types of sensors, such as inductive loop detectors [126–128], Light Detection and Ranging (LiDAR) [129, 130] and cameras [109, 122, 131], are used to collect traffic data. However, installing inductive loop detectors can potentially damage road pavements and is restricted to detecting vehicles only at the specific locations where the detectors are placed. LiDAR provides rich and accurate 3D position data of all road users around it, but comes at a high financial cost. As a result, methods based on video frames and computer vision are still the most popular due to their affordability, extensive coverage, and relative ease of installation [132]. Most video-based pipelines depend on the extraction of trajectories as an intermediary output [109, 133], from which the safety indicators can be computed. Trajectory-based pipelines generally follow four steps: 1) data collection, 2) object detection and tracking, 3) trajectory generation, and 4) safety indicator computation. These trajectory-based pipelines may rely on any object tracking methods, such as feature-based tracking and tracking by detection [134, 135], the former detecting distinctive features and the latter associating the objects detected by deep learning models.

Based on feature-based tracking, [133] relied on a trajectory-based pipeline to track road users and conducted a surrogate safety analysis to investigate the safety of limited-access highway facilities. The analysis studied the effect of a lane-change ban treatment at some on-ramps and off-ramps scenarios. Using a similar pipeline, [60,109] utilized the feature-based tracker available in the open-source Traffic Intelligence project [136] to extract trajectories and compute the TTC and PET of interactions involving automated shuttles. However, feature-based tracking methods often require extensive manual quality control to deal with crowded scenarios [109, 133].

Tracking by detection is becoming the most popular approach to obtain trajectories, as it benefits from the performance of recent object detectors [97]. On a self-collected dataset in Chula Vista, California, USA, [137] compared the performance of Fast R-CNN and SSD with OpenCV optical flow using the LucasKanade estimation method [138] for detecting pedestrians and skateboarders and compute TTC and PET, with the eventual goal for the traffic center to manage traffic based on the detected dangerous situations. In another example, [139] utilized PWC-Net [140] and MonoDepth [141] to estimate the speed of the ego vehicle and identify the leading vehicle in manually selected interactions where the leader stops, and then compute TTC.

A deep learning tracking by detection method in a commercial computer vision software, named as Brisk Lumina [142], to compute TTC and PET and the effectiveness of temporary low-cost countermeasures at risky selected pedestrian crossings utilized [131]. However, to

the best of our knowledge, no studies have yet evaluated the performance of detection and tracking methods on safety indicator computation in any trajectory-based pipeline.

More recently, researchers have tested other pipelines that are able to generate TTC directly by detecting pixels that are predicted to collide with the camera plane within a given time [143, 144]. To avoid ambiguity, they all deal with car-based video sensors to compute TTC between the vehicle equipped with the sensors and the vehicles in the image on each pixel, similar to the idea of camera-based depth estimation [145,146]. [144] estimated TTC by a series of simpler, binary classifications as whether the objects will reach the camera plane or not with the scale factor computed by two initial video frames. Following this idea, [147] utilized an attention-based scale encoder called FP-TTC, including a transformer-based feature extraction backbone, the scale encoder, and the time-series decoder to predict TTC pixel by pixel, avoiding the computational delay caused by the system time step interval. Yet, these methods are limited to car-based sensors and the interactions of that car (the "ego-vehicle") with other road users. No end-to-end methods in the literature can estimate TTC and PET at the same time. End-to-end methods are therefore limited and cannot be applied to roadside video data collection with multiple simultaneous conflicts, or when one needs more than one safety indicator.

5.2.3 Performance metrics for object detection, tracking and safety indicators

Multi-object tracking accuracy (MOTA) [148] and higher order tracking accuracy (HOTA) [149] are the two most popular evaluation metrics for multiple object tracking (MOT). MOTA measures the overall number of errors in the tracking results compared to the ground truth (GT). It takes into account the number of false positive (FP), false negative (FN) and identity switches (IDSW), which occur if a ground truth trajectory g that was previously matched to a tracker hypothesis h_i at time $t-1$, gets matched to a different hypothesis h_j at time t, where $i \neq j$, and then normalizes the sum of these three counts by the total number of GT tracks. However, MOTA overemphasizes the accuracy of detection while under-emphasizing the accuracy of association because it assigns a higher weight to detection errors (false positives and false negatives) compared to identity switches, which represent association errors [149]. HOTA has been more recently proposed to address these issues [149]: it considers the detection accuracy (DetA), which refers to how accurately objects are detected, association accuracy (AssA), which refers to how well objects are consistently tracked across frames and localization accuracy (LocA), which refers to how closely the predicted bounding box corresponds to the ground-truth annotation of an object, separately and provides a more balanced overall assessment, which has led to HOTA becoming nowadays the most popular

MOT metrics.

However, as detection and tracking are only inputs for safety analysis, object detection and tracking metrics cannot be the sole metrics for evaluating performance in safety analysis. If automated methods have perfect results, identical to the ground truth, the trajectories, conflicts and safety indicators will be the same as the ground truth. Yet, even recent tracking by detection methods are not perfect and do not replicate perfectly the ground truth, especially in complex crowded scenarios. However, to the best of the authors' knowledge, there are no specific performance metrics for safety indicators used for SMoS, and the impact of tracking performance on the accuracy of safety indicators has not yet been investigated. This paper aims to fill this gap by comparing safety indicators derived from the road user trajectories, either extracted from automated video analysis or from the ground truth.

5.3 Methodology

5.3.1 Pipeline

The pipeline developed in this paper for safety analysis is based on the following components:

1. Object detection;
2. Trajectory generation and smoothing; and
3. Safety indicator computation.

Three object detection methods are applied. They differ in terms of architecture and 2D or 3D detection outputs. Then, the same tracking method, ByteTrack, is applied to the detections to build the road user trajectories before computing the safety indicators. The pipeline, illustrated in Fig. 7.1, is capable of processing both 2D and 3D detections.

Object detection

We used three object detection methods: YOLOv8, CenterNet and CenterNet3D. YOLOv8 is a recent state-of-the-art one-stage detector with strong accuracy and speed, CenterNet is a widely used anchor-free keypoint-based method for 2D detection, and CenterNet3D extends this framework to 3D. Together, they provide a representative and complementary set of baselines for our study. They are applied to input video frames, and they output the detected objects in the form of 2D bounding boxes (BB) in the image space or 3D BBs in the world space.

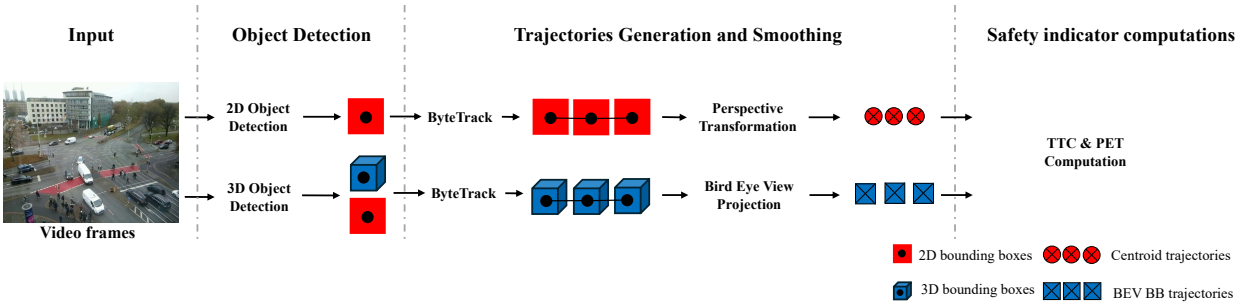


Figure 5.1 Processing pipeline for safety analysis. ByteTrack is the tracker applied in this study. The red shapes represent the 2D processing pipeline, while the blue shapes represent the 3D processing pipeline (BEV BB denotes bird's-eye-view bounding box).

In detail, CenterNet [28] regresses object by their center point with coordinates $[x_c, y_c]$, and then the BB width w and height h in image coordinates for the 2D task (CenterNet2D). It can additionally regress the depth z, the 3D BB dimensions $[w, h, l]$, with width w, height h and length l, and rotation angle θ in world coordinates for the 3D task (CenterNet3D). The YOLOv8 object detector [26] improves over previous YOLO versions [22, 24]. It implements the C2F module to improve the bottleneck, and it includes a better decoupled output head for the BB regression, $[x, y, w, h]$, and the object classification, cls. The C2F module is a variant of the Cross Stage Partial (CSP) [150] bottleneck structure. It employs a split-and-merge strategy with two convolutional layers and multiple shortcut connections to enhance gradient flow while reducing computational cost.

CenterNet2D and YOLOv8 are used in the 2D processing pipeline, while CenterNet3D is used in the 3D processing pipeline.

Trajectory generation and smoothing

Detections need to be associated to track each individual road user and generate their trajectories. The chosen tracking method is *ByteTrack* [39]. It is a simple, effective and generic object association method for tracking detected objects. High score detection bounding boxes (BBs) are associated first, while the low score detections are recovered by a second round of association for each frame. ByteTrack is designed for tracking 2D BBs $[x, y, w, h]$, therefore, it is applied on the 2D object detection results in the 3D processing pipeline.

After data association with ByteTrack, we apply these three steps to generate the final trajectories:

1. Coordinate transformation;

2. Linear interpolation; and

3. Trajectory smoothing.

The *coordinate transformation* is different for the 2D and 3D processing pipelines. In the 2D pipeline, image-space positions are transformed into world coordinates using a perspective transformation matrix, which is estimated by matching at least four pairs of corresponding points between the image and world coordinate systems. For the 3D pipeline, the 3D information, including 3D location, dimension and rotation angle, $[x_{3d}, y_{3d}, z_{3d}, w, h, l, \theta]$, is directly provided in the world space. Therefore, the road user bird's-eye-view (BEV) BB is the rotated base of the 3D cuboid.

Trajectory gaps caused by missing detections are filled in through *linear interpolation*. Trajectories are then *smoothed* to reduce the noise on the center point location or object BEV BBs. A simple moving average filter is applied for this purpose. Object velocity in each frame is derived from the smoothed trajectories. In the end, 2D object tracking provides centroid trajectories, and 3D object tracking provides BEV BB trajectories.

Safety indicator computation

The safety indicators, namely TTC and PET, are computed with the library from the *Traffic Intelligence* project [136]. The computations are applied to interactions, which are precursor situations to conflicts: an interaction is constituted by two road users existing in the same time period. For centroid-based trajectories (2D pipeline), a fixed-size circle is used to represent the outline of road users, while, for the BEV BB trajectories (3D pipeline), the BBs are utilized. A collision consists in an overlap of the road user outlines.

TTC requires predicting each road user's motion at every instant to evaluate if it may collide in the future. While more sophisticated and realistic methods exist, a simple motion prediction model at constant velocity is used to compute the TTC at each instant of each interaction [114]. There is generally no TTC if two road users are moving away from each other. Otherwise, it results in a time series, and the minimum value, TTC_{min}, is used as the representative for the interaction. PET is computed by first finding if and where the trajectories overlap, then by taking the difference of the instant the first road user leaves that position, and the instant of the second reaches that position. If it exists, PET is generally a unique value as trajectories intersect at most once in an area of interest, except for car-following situations.

5.3.2 Object matching and interaction matching

Measuring the performance of the computer vision methods is not trivial. It requires comparing the output trajectories of computer vision methods with the ground truth trajectories. The ground-truth trajectories are provided in the dataset annotations. Because of object detection or tracking errors, the generated trajectories will differ from those from the ground truth in space and time. Trajectories may be fragmented, some trajectories may not have been extracted, and predicted trajectories may be false alarms. Therefore, we need to match the trajectories predicted by computer vision methods with those from the ground truth by comparing their positions in time and space.

A second step is needed to compare interactions. Studies based on SMoS require both accurate counts of conflicts and accurate predictions of their safety indicators. A two-step matching method is therefore developed to perform: 1) object matching between the ground truth trajectories and predicted trajectories by the computer vision method, and 2) interaction matching between the ground truth interactions and the interactions obtained from the predicted trajectories. This enables in particular to measure errors in the safety indicator calculations. The object matching results are the base for interaction matching.

Similarly to precision and recall [151] for classification and object detection problems [152–154], the object matching and interaction matching results are reported by considering first the set of the ground truth trajectories (resp. interactions) and how they are matched to the predicted trajectories (resp. interactions), called "GT matching", then the set of predicted trajectories (resp. interactions) and how they are matched to the ground truth trajectories (resp. interactions), called "Prediction matching". Only the first set will thus contain misses, and only the second will contain false alarms.

Object matching

Fig. 5.2 illustrates how the proposed novel object matching method works. All the trajectories of the ground truth $\mathbb{G}$ and the predictions $\mathbb{P}$ are taken as the inputs for object matching. For each object i, a ground truth trajectory G_i is represented as a series of points $[G_i^b, G_i^{b+1}, \cdots, G_i^e]$, where G_i^t is the position of object i at frame t between the first and last frames b and e. Similarly, for a computer vision method, the trajectory of each tracked object j can be represented similarly as P_j: $[P_j^b, P_j^{b+1}, \cdots, P_j^e]$.

The Hungarian Algorithm [37] with a distance threshold α is applied frame by frame for object matching. For each frame t, the sets of the w_t ground truth objects and of the v_t predicted object detections are taken as the inputs to the Hungarian Algorithm that performs

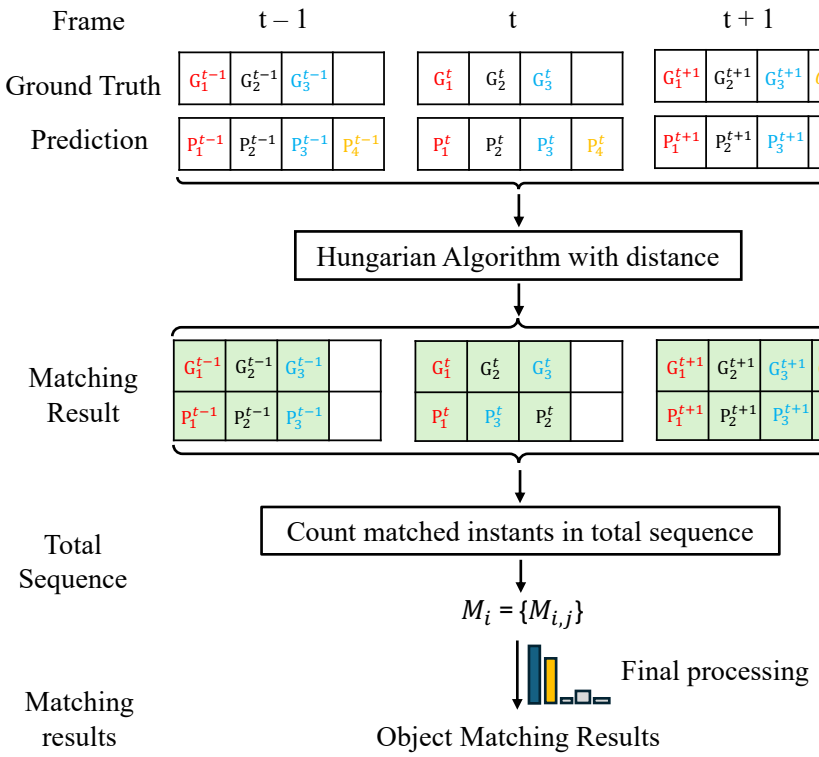


Figure 5.2 Illustration of object matching. The green background color means that the ground truth object has a match, while no background color means no match.

a bipartite matching using a $w_t \times v_t$ cost matrix C, where each element is defined as:

$$C_{ij} = \begin{cases} d_{i,j} & \text{if } d_{i,j} \leq \alpha \\ \infty & \text{if } d_{i,j} > \alpha \end{cases}, \tag{5.1}$$

where $d_{i,j}$ denotes the Euclidean distance between the coordinates of the i-th ground truth object and of the j-th prediction at t.

The matching is performed on all the frames and $M_{i,j}$ represents the number of instant matches between the i-th ground truth object and the j-th prediction. A decision rule is employed to determine the final set of matched results. Let $\bar{M}_i$ be the mean and $\hat{s}_i$ the standard deviation of the non-zero numbers of matches $M_{i,j}$ for the ground truth object i.

We use $\hat{s}_i$ to distinguish two cases of low and high variability using a threshold β. The necessity of these two cases lies in the balance between robustness and completeness. In high-variability cases, additional filtering is needed to avoid noisy matches and ensure reliability, while in low-variability cases, keeping all the matches prevents the loss of valid associations. Thus, the two conditions serve as an adaptive mechanism to achieve a trade-off between

robustness and completeness.

1. $\hat{s}_i \geq \beta$ (high variability): we keep the matching predicted object j such that $M_{i,j} \geq \gamma \bar{M}_i$ where $\gamma \geq 1$ is a variability threshold,
2. $\hat{s}_i < \beta$ (low variability): we keep all the matching predicted objects with at least one instant match.

As a result, there are six types of object matches:

1. 1-to-1, where exactly one predicted trajectory matches one and only one ground truth trajectory;
2. 1-to-n, where more than one predicted trajectories match one ground truth trajectory;
3. n-to-1, where one predicted trajectory matches multiple ground truth trajectories;
4. 1-to-0, where the ground truth object is missed;
5. 0-to-1, where the predicted object is a false alarm; and
6. m-to-n, where multiple predicted trajectories are mapped to multiple ground truth trajectories.

The algorithm proposed in [155] is applied to decouple the m-to-n object matching into the other five matching types. In detail, it firstly builds a ground truth and a prediction graph. Then all the matched objects are decoupled by removing some connections sequentially. As a result, the m-to-n matching is eventually decoupled to the other five matching types. A condition of temporal overlap is added to eliminate pairs with short predicted trajectories and minimal temporal overlap. Let $ti(o)$ denote the time interval of an object o and δ the threshold. A ground truth object o_g matches a predicted object o_p if $|ti(o_g) \cap ti(o_p)| \geq \delta$.

Interaction matching

The interactions are generated separately from the ground truth objects and the predicted objects. They are then matched based on the object matches, as shown in Table 5.1. An interaction is denoted by the set of involved objects, respectively $\{o_{g_1}, o_{g_2}\}$ for ground truth and $\{o_{p_1}, o_{p_2}\}$ for predicted trajectories. The ground truth and predicted interactions must meet two conditions to be matched:

1. *Object correspondence.* Denoting the 1-to-1, 1-to-n or n-to-1 matching of objects by $\sim$, the involved object trajectories should match:

$$(o_{g_1} \sim o_{p_1} \text{ and } o_{g_2} \sim o_{p_2}) \text{ or } (o_{g_1} \sim o_{p_2} \text{ and } o_{g_2} \sim o_{p_1}) \tag{5.2}$$

2. *Temporal overlap.* The time intervals during which the objects are present in the traffic scene must overlap. Specifically, the intersection of the time intervals of the interacting objects in the ground truth and predicted interactions should have a longer duration than a threshold ε:

$$|ti(o_{g_1}) \cap ti(o_{g_2}) \cap ti(o_{p_1}) \cap ti(o_{p_2})| \geq \epsilon \tag{5.3}$$

These two conditions mean that misses (in interaction GT matching) and false alarms (in interaction prediction matching) may be caused by either a missed (or resp. a falsely detected) object (condition 1) or, if there are matching objects, by a lack of temporal overlap (condition 2). These are referred at type 1 and 2 misses or false alarms respectively.

There are two cases of interaction matching, shown in Table 5.1:

1. *Single interaction matching.* There are neither 1-to-n nor n-to-1 in the involved object matching types. The interaction matching will result in only one possible type, shown as table cells without any symbol. For example, the combination of object matching 1-to-0 and object matching 1-to-0 only results in interaction matching 1-to-0.

2. *Multiple interaction matching.* There is at least a 1-to-n or n-to-1 in the involved object matching types, shown as table cells with the symbols $\square$, $\triangle$ or $\square\triangle$. The interaction matching will not only result in the object matching type, but may also generate extra misses (1-to-0) or false alarms (0-to-1). For example, the object matching combination of 1-to-n and 1-to-1 may not only generate an 1-to-n interaction matching, but also false alarms.

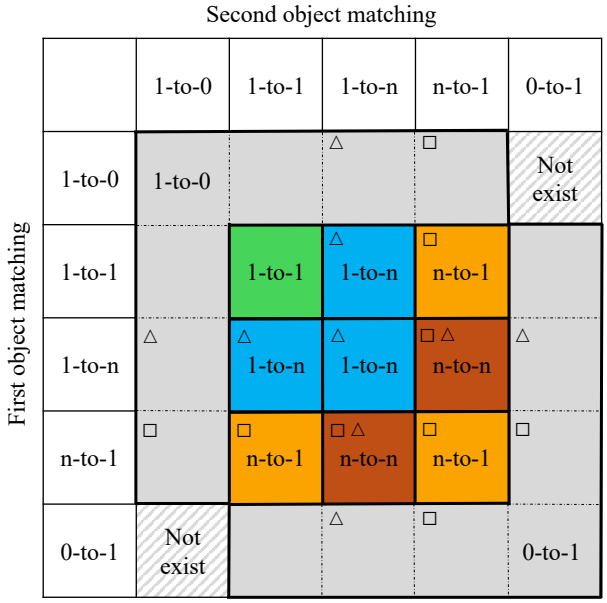
5.3.3 Evaluation metrics

Two types of metrics are used to compare the performance of the road user detection and tracking methods in terms of safety analysis:

1. *General performance metrics*:

   (a) the numbers of interactions with a safety indicator value for either TTC_{min} or PET, and the numbers of dangerous interactions with the value below 1.5 s, disaggregated by road user types;

Table 5.1 Mapping of object matching types to interaction matching types. The colors (green, blue, yellow, brown and gray) in this table indicate the areas that belong to the same interaction matching category. □ indicates possible extra misses, while △ indicates possible extra false alarms.



(b) the Cumulative Distribution Functions (CDF) and distributions of the safety indicators, disaggregated by road user types; and

(c) the HOTA [149] metrics for object detection and tracking.

2. *Detailed performance metrics based on object and interaction matching*:

(a) the distributions of object and interaction matching types;

(b) the scatter plots of the safety indicators of the matched interactions; and

(c) the distributions of the safety indicators for the unmatched interactions (misses and false alarms).

5.4 Experimental methodology

5.4.1 Dataset

The **L**eibniz **U**niversity **M**ulti-**P**erspective **I**ntersection Dataset (LUMPI) [8] was selected as the dataset for our safety analyzes as it was recorded at an intersection with high traffic volume, with over 50 road users in each frame. It is a multi-view, multi-sensor dataset with five LiDAR and two camera sensors, recorded at Königsworther Platz in Hanover,

Germany. This is a complex intersection of four streets, with mixed traffic of motorized vehicles, pedestrians and cyclists. The recording frequencies of the cameras are 15 or 30 fps. Each road user in LUMPI is manually annotated with a tracking ID and both 2D and 3D information, including object dimensions, positions, and orientations, obtained respectively from the camera and LiDAR sensors. Four pairs of annotated points at different image locations were selected to compute the perspective transformation matrix, ensuring good spatial coverage.

The GT trajectories are constituted of the BEV BBs of the 3D annotations, from which the TTC and PET are computed. For object matching, the projected BB centers and the BEV BB centers are employed to compute the distances.

The video recorded by one camera in the 4^{th} LUMPI experiment is chosen and denoted as view 1 in our experiments and shown in Fig. 5.3a. The video recorded by another camera in the 5^{th} LUMPI experiment is denoted as view 2 and shown in Fig. 5.3c. For the safety analysis, the areas of interest (AoI) are defined by excluding the background buildings and regions where road users are too small (top portion of the frames). The AoI of view 1 and view 2 are given in Fig. 5.3b and Fig. 5.3d, respectively.

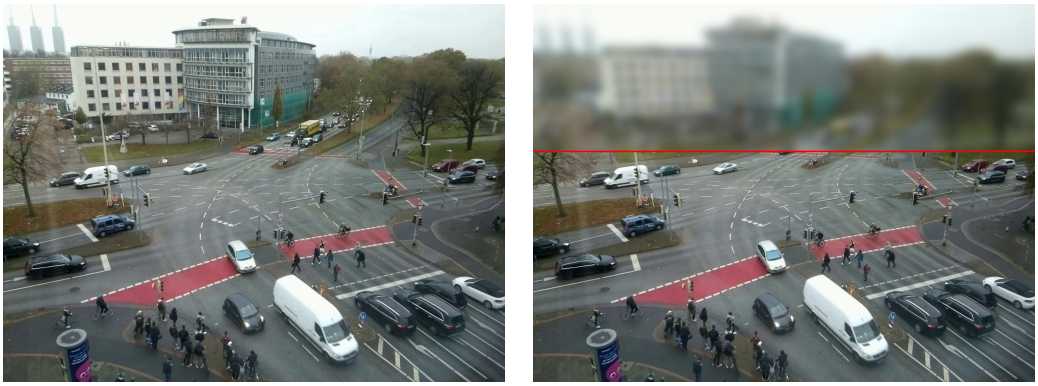
Table 5.2 Summary of the data splitting (in number of frames) for view 1 and view 2.

Set	**View 1**	**View 2**	**Total**
Train	7,286	–	7,286
Validation	1,822	–	1,822
Test	15,922	4,500	20,422

For training and evaluation, we take 9,108 frames (about five minutes) from the view 1 as the training-validation set, and 15,922 frames (about five minutes and 26 seconds) as the test set. In order to test the performance on another view, we take 4,500 frames (two minutes and 30 seconds) from the view 2 as the second unseen test set. As applying a 3D method to the other view requires retraining to accurately predict 3D information, we only applied CenterNet2D and YOLOv8 to view 2 (CenterNet3D is excluded). The information about the training, validation and test sets is summarized in Table 5.2.

5.4.2 Model training and parameters

CenterNet2D and CenterNet3D are pretrained on the KITTI dataset [7]. YOLOv8 is pretrained on the COCO dataset [156]. All three detection methods are fine-tuned for three categories: car, pedestrian and cyclist.



(a) View 1 (b) View 1 (AoI)

(c) View 2 (d) View 2 (AoI)

Figure 5.3 Example frames of two views from LUMPI: blurred areas are not analyzed.

For the training of the deep learning methods, we used a workstation equipped with two 32 GB NVIDIA Tesla V100 GPUs and dual Intelő Xeonő Silver 4116 CPUs. For object detection, the main training parameters are the same for CenterNet2D and CenterNet3D, that is, the two models were trained for 30 epochs with a batch size of 8 and a learning rate of $1e^{-4}$. YOLOv8 was trained for 50 epochs with a batch size of 8 and learning rate of $1e^{-4}$.

For object tracking, ByteTrack is used with its default parameters. In detail, the object confident threshold is set as 0.25, which determines whether the confidence of a detection is sufficient to be tracked. The tracking buffer is set as 30 frames, determining how many frames a track can persist without a detection. The matching threshold is set as 0.8, for matching tracks to detections based on IoU.

The half-width of the smoothing window in the moving average filter is chosen as 2 frames. For object matching, the distance threshold in the Hungarian Algorithm α is set as 3 m, the determining threshold β is set as 5 frames, and the variability threshold γ is set as 2, determined empirically by testing various values and selecting those that yielded optimal performance. The temporal overlap threshold δ is set as 0.1. For interaction matching,

the temporal overlap of interaction matching ϵ is set as 5 frames. These thresholds were determined empirically through experiments.

5.5 Results

In this section, we first present the general performance results obtained using the proposed pipeline described in section 5.3.1. Next, we evaluate the performance of each method employing the proposed matching approach outlined in section 5.3.2. Finally, we present a discussion of the results.

5.5.1 General performance results

Number of interactions

The considered interactions can be split into three sub-categories based on the types of road users involved: a car with a car (car-car), a car with a cyclist (car-cyc) and a car with a pedestrian (car-ped). Table 5.3 presents the total and road-user-wise numbers of interactions with a value for either TTC_{min} or PET ("all"), and the subset of "dangerous" interactions with either indicator smaller than 1.5 s.

In view 1 for TTC_{min}, YOLOv8 generates the highest numbers of *all* and *dangerous* interactions, significantly higher than in the ground truth, while CenterNet2D generates the number of interactions most similar to the ground truth. For PET, the three methods generate more similar numbers of interactions compared with the ground truth. For *dangerous* interactions, CenterNet2D and YOLOv8 generate fewer interactions than the ground truth, while CenterNet3D generates more. YOLOv8 is the closest to the ground truth.

In view 2 for TTC_{min}, CenterNet2D generates more interactions than the ground truth, while YOLOv8 generates fewer. For *dangerous* interactions, both CenterNet2D and YOLOv8 are closer to the ground truth. For PET, both methods generate much fewer interactions compared to the ground truth. The numbers of *dangerous* interactions for both methods are closer to the ground truth.

For both PET and TTC, the car-car interactions are the most numerous among all interactions and the dangerous ones for both views, compared with the other two categories. Notably, there are very few dangerous car-cyclist and car-pedestrian interactions based on PET. For TTC, in both views, CenterNet2D achieves the best performance for all interaction types except for carcar interactions, for which YOLOv8 performs the best. For PET, in both views, YOLOv8 performs the best for the *dangerous* interactions, while CenterNet3D and

Table 5.3 Comparison of the number of interactions (with a value for either TTC_{min} or PET) and dangerous interactions (with either TTC_{min} or PET $\leq 1,5$ s). car-car, car-cyc, and car-ped denote interactions between a car and another car, a cyclist, and a pedestrian, respectively. **Bold** indicates the closest to ground truth.

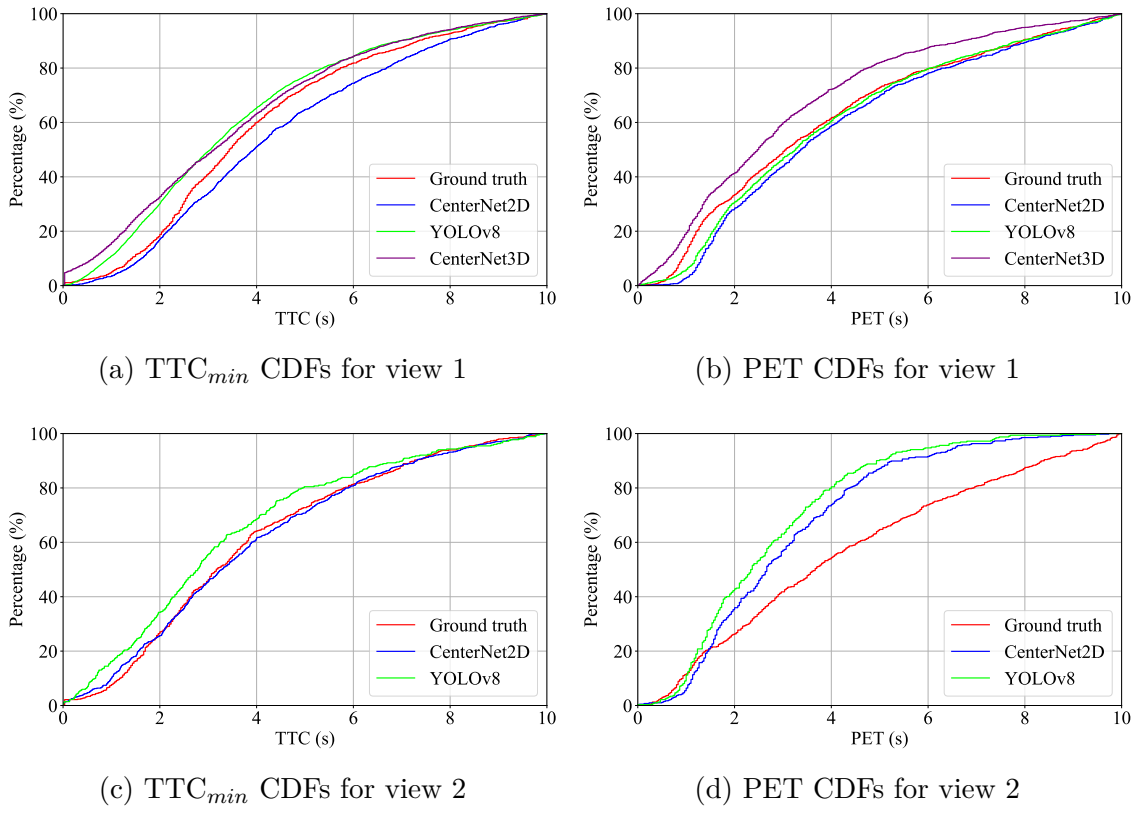
Method	**All**				**Dangerous**			
	total	**car-car**	**car-cyc**	**car-ped**	**total**	**car-car**	**car-cyc**	**car-ped**
TTC for view 1								
Ground truth	1,008	613	227	168	115	74	22	19
CenterNet2D	**1,522**	**868**	**331**	**323**	**126**	**80**	**22**	**24**
YOLOv8	3,124	1,856	721	547	629	414	129	86
CenterNet3D	2,269	1,505	369	395	578	430	84	64
PET for view 1								
Ground truth	1,061	895	99	67	283	277	5	1
CenterNet2D	**967**	800	84	83	128	127	0	1
YOLOv8	1,132	**933**	114	85	**214**	**211**	**2**	1
CenterNet3D	1,471	1,301	**108**	**62**	499	482	16	1
TTC for view 2								
Ground truth	647	232	199	216	104	35	27	42
CenterNet2D	**697**	335	**108**	**254**	**122**	**59**	**26**	**37**
YOLOv8	352	**285**	66	1	83	66	17	0
PET for view 2								
Ground truth	626	463	118	45	132	129	3	0
CenterNet2D	**405**	**376**	**10**	**19**	87	86	0	1
YOLOv8	321	319	2	0	**91**	**91**	0	0

CenterNet2D achieve the best results for *all* interactions in view 1 and view 2, respectively.

CDFs and distributions of the safety indicators

In view 1 in Fig. 5.4a, all three methods have broadly similar TTC_{min} CDF shapes compared with the ground truth. CenterNet2D has the smallest maximum vertical difference with the ground truth while CenterNet3D has the largest. Besides, CenterNet3D has a significant proportion of interactions with a TTC_{min} value lower than 0.5 s, called "extremely dangerous interactions".

In Fig. 5.5a, all three methods generate more interactions with TTC, especially in the range [0, 5] s. Among them, YOLOv8 generates the highest number of interactions across all values, while CenterNet2D is the closest to the ground truth. Looking specifically at the worst



(a) TTC_{min} CDFs for view 1

(b) PET CDFs for view 1

(c) TTC_{min} CDFs for view 2

(d) PET CDFs for view 2

Figure 5.4 CDFs of TTC_{min} and PET.

Table 5.4 Proportions of interactions with either safety indicator smaller than 1.5 s. The numbers in parentheses indicate the difference between the predicted and ground truth values. **Bold** indicates the closest to ground truth.

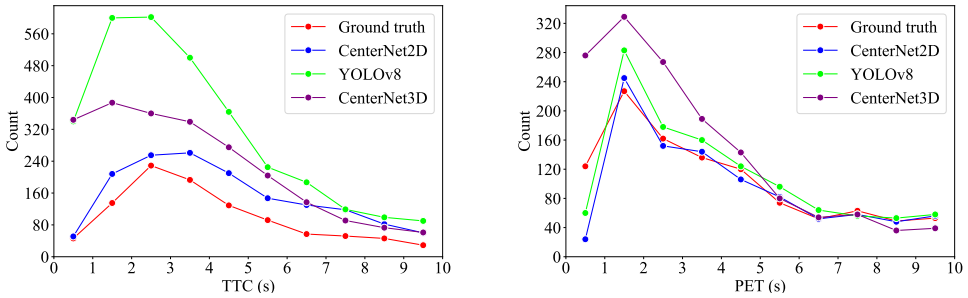
View 1

Method	**TTC**	**PET**
Ground truth	11.49 %	26.74 %
CenterNet2D	**8.33 % (-3.16 %)**	15.37 % (-11.37 %)
YOLOv8	20.14 % (8.65 %)	18.96 % (-7.78 %
CenterNet3D	25.03 % (13.54 %)	**33.94 % (7.2 %)**

View 2

Method	**TTC**	**PET**
Ground truth	16.36 %	21.18 %
CenterNet2D	**18.10 % (1.74 %)**	**21.62 % (0.44 %)**
YOLOv8	23.73 % (7.37 %)	28.48 % (7.3 %)

performer, YOLOv8, we find it generates more than twice the number of interactions over all intervals, specifically, over seven times more interactions within [0, 1] s and approximately



(a) TTC counts for view 1

(b) PET counts for view 1

(c) TTC counts for view 2

(d) PET counts for view 2

Figure 5.5 Distributions for TTC_{min} and PET.

4.5 times more within [1, 2] s. The higher counts in the dangerous values are related to the higher number of false alarms generated by YOLOv8 than CenterNet2D and the higher noise in the trajectories of YOLOv8.

On the other hand, Fig. 5.4b shows YOLOv8 has the most similar PET CDF shape to the ground truth, and CenterNet2D performs closely to YOLOv8. Meanwhile, CenterNet3D has the largest difference compared to the ground truth. Fig. 5.5b shows that both CenterNet2D and YOLOv8 have similar numbers of interactions to the ground truth, whereas CenterNet3D generates more interactions with PET in the range [0, 5] s. In general, compared to TTC_{min}, the distributions of PET values for the automated methods are more similar to the ground truth.

In view 2 in Fig. 5.4c, CenterNet2D has a more similar TTC_{min} CDF shape to the ground truth, exhibiting a smaller probability difference compared to YOLOv8. In Fig. 5.5c, CenterNet2D has slightly higher numbers of interactions at most TTC_{min} values compared to the ground truth. Fig. 5.4d shows that CenterNet2D and YOLOv8 have higher proportions of interactions than the ground truth for PETs higher than 1.5 s. Fig. 5.5d indeed shows Cen-

terNet2D and YOLOv8 produce more interactions with PET in the range [1, 2] s, but fewer for all other ranges. The PET distributions in view 2 for the automated methods are similar to view 1, but different from the ground truth, suggesting that the automated methods do not reflect the actual PETs of road user interactions.

Looking at the proportion of interactions below 1.5 s in Table 5.4, CenterNet2D performs best for TTC_{min} in both views. For PET, CenterNet3D performs best, slightly better than YOLOv8, in view 1 while CenterNet2D performs best in view 2.

The CDFs and indicator distributions per road user categories are presented in A.1 and A.2. CenterNet2D outperforms other methods across most views, indicators and road user categories, even with smaller road users. In view 2, YOLOv8 performs dismally, while even CenterNet2D cannot accurately measure TTC_{min} and PET for car-cyclist interactions.

Object detection and tracking performance

Table 5.5 Object detection and tracking performance in views 1 and 2. "Predictions" and "GT" refer to the number of objects detected and tracked by each automated method and present in the ground truth (GT), respectively. **Bold** indicates the best performance.

Category	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
View 1									
Car	CenterNet2D	**79.64**	**75.05**	**85.67**	**87.90**	139,870	150,458	**489**	484
	YOLOv8	68.00	62.62	75.96	83.05	**145,823**		517	
	CenterNet3D	49.38	41.18	60.86	80.12	91,099		529	
Cyclist	CenterNet2D	**54.23**	**51.50**	**57.62**	**83.88**	**2,9455**	33,321	**89**	69
	YOLOv8	46.21	45.95	47.25	77.57	40,256		134	
	CenterNet3D	22.38	18.35	27.95	73.42	11,817		104	
Pedestrian	CenterNet2D	**48.64**	**48.13**	**49.68**	**82.77**	38,079	42,194	**126**	130
	YOLOv8	34.26	33.00	35.92	74.00	**43,786**		214	
	CenterNet3D	11.12	14.05	16.32	75.28	13,736		158	
View 2									
Car	CenterNet2D	**40.96**	**34.46**	**50.36**	74.10	**31,996**	54,798	**264**	204
	YOLOv8	29.52	22.05	40.12	**75.55**	20,184		284	
Cyclist	CenterNet2D	4.51	**2.63**	7.83	68.29	1,612	29,808	77	27
	YOLOv8	**6.35**	1.88	**22.89**	**70.27**	**1,032**		**28**	
Pedestrian	CenterNet2D	**4.14**	**3.79**	**4.59**	62.55	**4,474**	23,315	132	53
	YOLOv8	0.12	0.01	1.05	**70.00**	8		3	

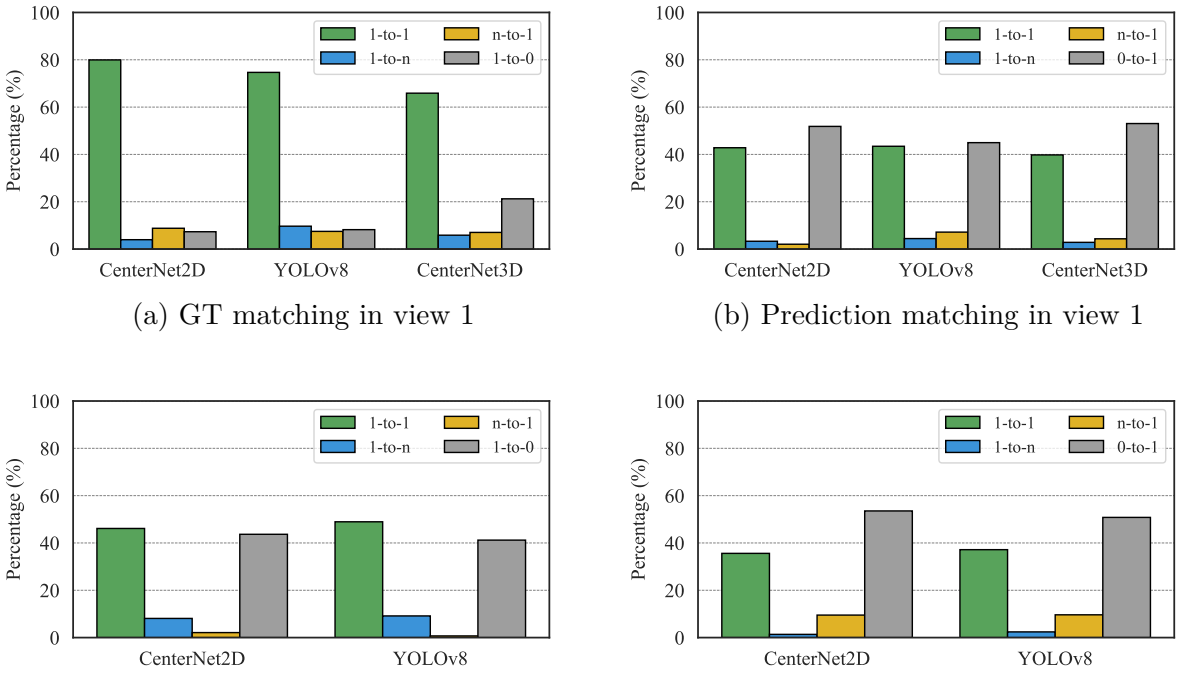
Table 5.5 presents the object detection and tracking performance of the selected methods for views 1 and 2. For both views, the car category is the easiest to be detected (DetA)

and tracked (AssA) for all three methods, followed by the cyclist category. The pedestrian category remains the most challenging to be detected and tracked.

In view 1, CenterNet2D outperforms YOLOv8 and CenterNet3D on all three categories. CenterNet2D generates the closest number of detected and tracked objects to the ground truth, except for the number of detected cars and pedestrians, where YOLOv8 performs best. However, for these two categories, CenterNet2D remains very close to YOLOv8, while CenterNet3D shows the poorest performance. In view 2, since CenterNet2D and YOLOv8 are trained only on the data from view 1, the performance of both methods drops dramatically for all object categories. Furthermore, many objects are missed, and no method shows similar performance to the ground truth in terms of the number of tracked objects.

5.5.2 Detailed performance results based on object and interaction matching

Object matching



(a) GT matching in view 1

(b) Prediction matching in view 1

(c) GT matching in view 2

(d) Prediction matching in view 2

Figure 5.6 Distributions of object matching types.

Fig. 5.6 illustrates the performance of each method in views 1 and 2 based on object matching: In view 1 for GT matching, CenterNet2D outperforms YOLOv8 and CenterNet3D, with CenterNet2D having the largest 1-to-1 matching percentage. CenterNet3D has the poor-

est performance, with the smallest percentage of 1-to-1 matching and largest percentage of missed object matching. For prediction matching, the three methods have similar 1-to-1 performance, but YOLOv8 has a smaller percentage of false alarms.

In view 2, consistent with their performance in object detection and tracking, CenterNet2D and YOLOv8 exhibit a similarly large drop in the object matching performance compared to view 1.

Interaction matching

Fig. 5.7 illustrates the interaction matching results, where a higher proportion of 1-to-1 matches reflects more accurate interactions. One should note that only interactions with either a TTC_{min} or PET value are considered for both GT and prediction matching, but they may be matched to all predicted or ground truth object respectively, whether these have a safety indicator value or not. In both views, all the methods perform better for interactions with a PET value than interactions with a TTC_{min} value.

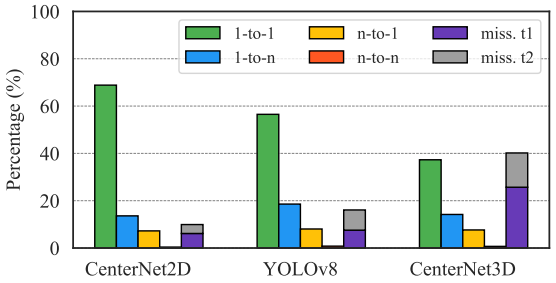
In view 1, CenterNet2D outperforms YOLOv8 and CenterNet3D for both GT and prediction matching in both indicators. The percentage of 1-to-1 matching of both CenterNet2D and YOLOv8 is lower in prediction matching than in GT matching for TTC_{min} while the differences are smaller for PET.

In view 2, consistent with their performances in the previous metrics, for GT matching, the performance of CenterNet2D and YOLOv8 drops dramatically, with around 80 % of interactions missed for TTC_{min}. The results are relatively better for PET, for which the missed-interaction rate decreases by about 20 percentage points, to around 60 %. The performance for prediction matching is better, but still far worse than for view 1.

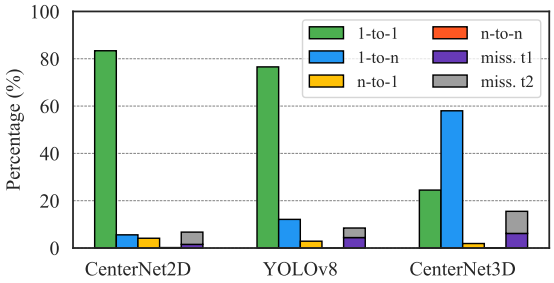
Safety indicator comparison

To compare the similarity of indicator values between matched ground truth and predicted interactions, we first present scatter plots of these values for the matched interactions for each method. Closer alignment of points with the line $y = x$ indicates higher prediction accuracy. However, these scatter plots cannot illustrate the misses or false alarms. The distributions of indicator values for the misses and false alarms are thus also presented.

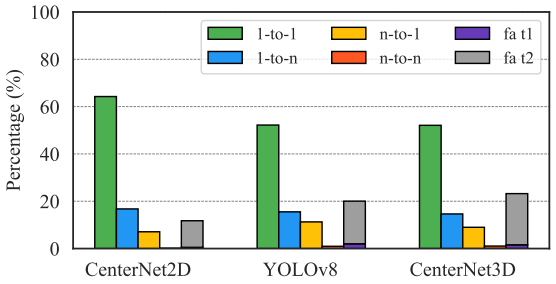
Scatter plots for matched interactions For view 1 (Fig. 5.8), most methods tend to underestimate TTC_{min} as most interactions lie above the line $y = x$, most clearly for CenterNet3D. The TTC_{min} points are widely distributed, with many interactions predicted to



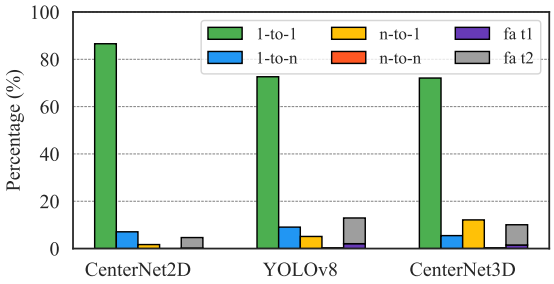
(a) GT matching for TTC_{min} in view 1



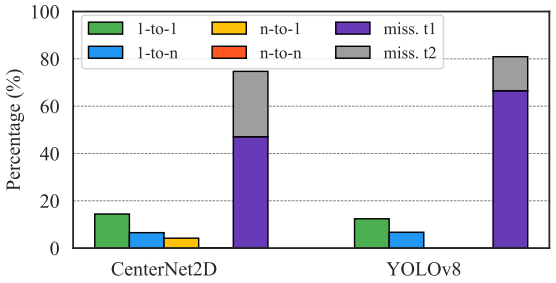
(b) GT matching for PET in view 1



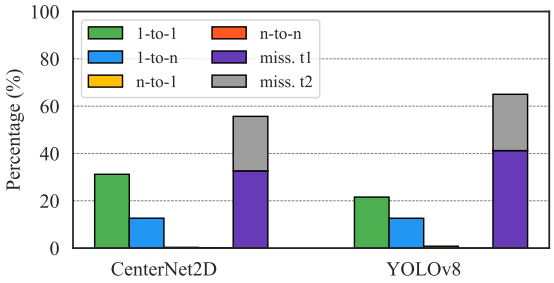
(c) Prediction matching for TTC_{min} in view 1



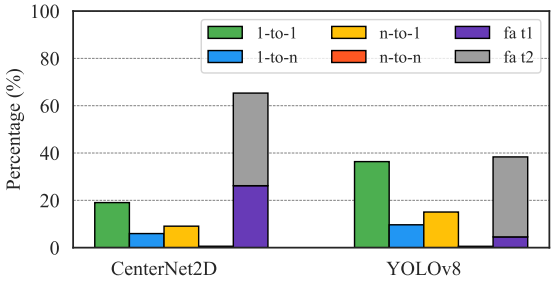
(d) Prediction matching for PET in view 1



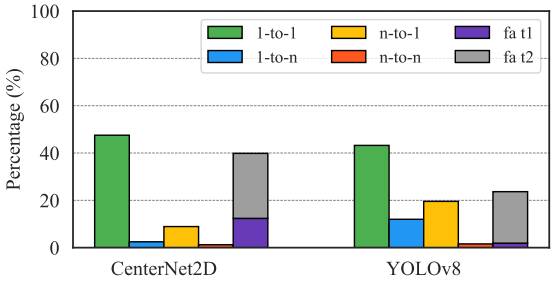
(e) GT matching for TTC_{min} in view 2



(f) GT matching for PET in view 2

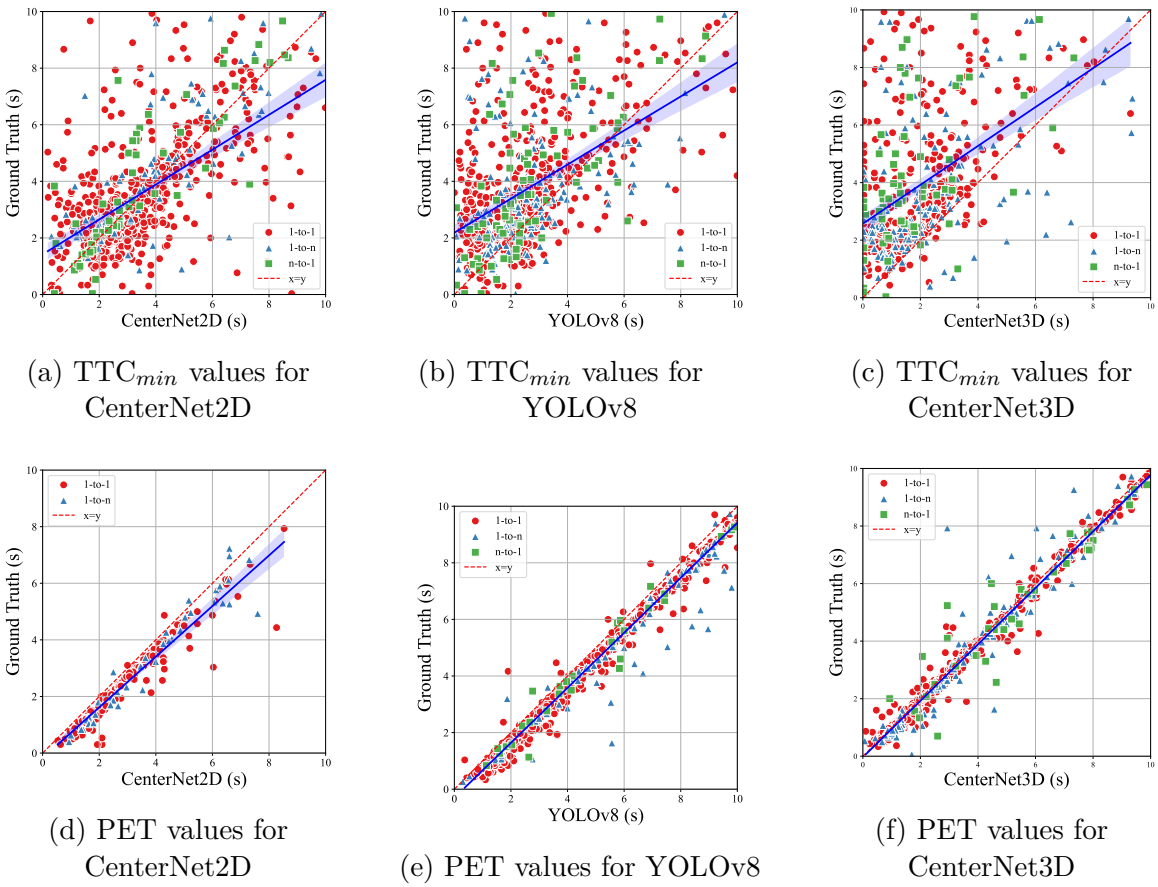


(g) Prediction matching for TTC_{min} in view 2



(h) Prediction matching for PET in view 2

Figure 5.7 Distributions of interaction matching types. The types "miss. t1" and "miss. t2" represent misses and interactions with no temporal overlap in the matching process, respectively. Similarly, "fa t1" and "fa t2" represent false alarms and interactions with no temporal overlap.



(a) TTC_{min} values for CenterNet2D

(b) TTC_{min} values for YOLOv8

(c) TTC_{min} values for CenterNet3D

(d) PET values for CenterNet2D

(e) PET values for YOLOv8

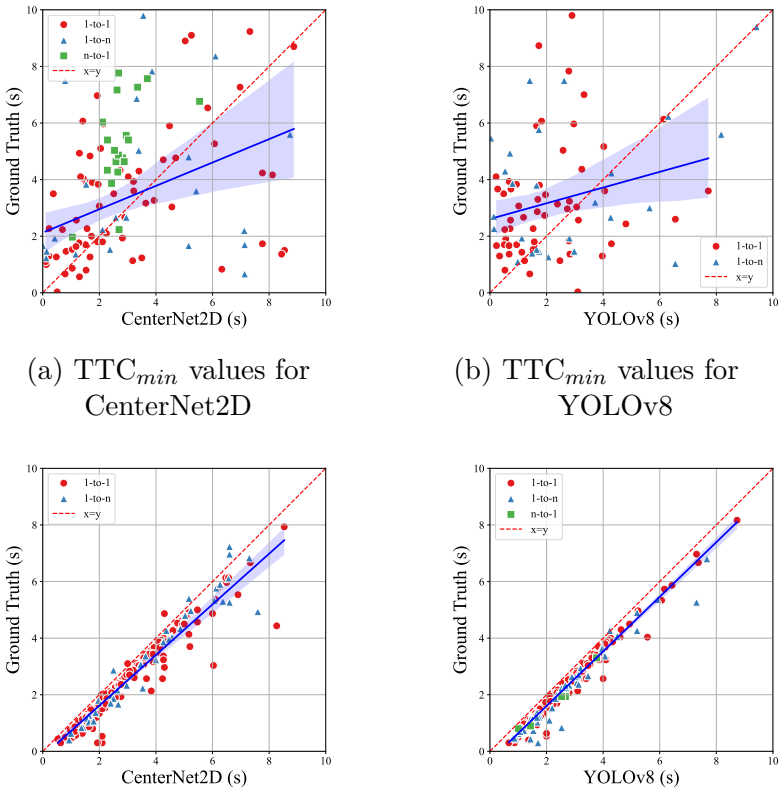
(f) PET values for CenterNet3D

Figure 5.8 Scatter plots of the safety indicators of the matched interactions for the three methods compared with the ground truth in view 1.

have small TTC_{min} by YOLOv8 and CenterNet3D despite varying true TTC_{min} values. While CenterNet2D seems to have the most interactions close to the line $y = x$, its RMSE is slightly better than YOLOv8, but in fact much higher compared to CenterNet3D (Table 5.6). CenterNet3D is helped by a much smaller number of 1-to-1 matched interactions, for which it performs better. One must be careful with the interpretation of RMSE as it is not computed over the same set of interactions for each method. For view 2, CenterNet2D and YOLOv8 generate much fewer matched interactions with TTC_{min} values, with similar spreads (Fig. 5.9).

For PET in views 1 (Fig. 5.8) and 2 (Fig. 5.9), most interactions for all three methods are concentrated near and slightly below the line $y = x$, indicating that PET is much easier to predict accurately than TTC_{min} and all three methods tend to produce slightly less dangerous PET values compared to the ground truth. Regarding RMSE in Table 5.6, CenterNet3D and YOLOv8 have the best RMSE for PET in views 1 and 2, respectively, but CenterNet2D is

close for both views with the highest number of matched interactions.



(a) TTC_{min} values for CenterNet2D

(b) TTC_{min} values for YOLOv8

(c) PET values for CenterNet2D

(d) PET values for YOLOv8

Figure 5.9 Scatter plots of the safety indicators of the matched interactions for CenterNet2D and YOLOv8 compared with the ground truth in view 2.

Distributions for the misses and false alarms Fig. 5.10 illustrates the distributions of the percentage of misses (including interactions without values in 1-to-1 matching in GT matching) and false alarms as a proportion of the ground truth or predicted interactions over each 1-s interval between 0 and 10 s.

In view 1, for missed interactions, CenterNet2D achieves the best or the second best performance for both TTC_{min} and PET, whereas CenterNet3D performs the worst.

The high proportion of missed interactions with the lowest TTC_{min} value is noticeable for all the methods. The proportions are lower for all the methods for PET. Furthermore, the proportion of missed interactions for all three methods with a PET smaller than 1 s is the lowest (or close to the lowest) overall.

Table 5.6 RMSE for TTC_{min} and PET for matched 1-to-1 interactions for the different methods ("Counts" denotes the number of matched 1-to-1 interactions). **Bold** indicates the best performance and ***bold italic*** indicates the biggest number of interactions.

View 1

Method	**TTC_{min}**		**PET**	
	RMSE	Counts	RMSE	Counts
CenterNet2D	5.33	***600***	0.71	***821***
YOLOv8	5.80	509	2.65	777
CenterNet3D	**2.83**	345	**0.30**	537

View 2

Method	**TTC_{min}**		**PET**	
	RMSE	Counts	RMSE	Counts
CenterNet2D	**2.48**	***65***	0.66	***171***
YOLOv8	2.52	56	**0.49**	118

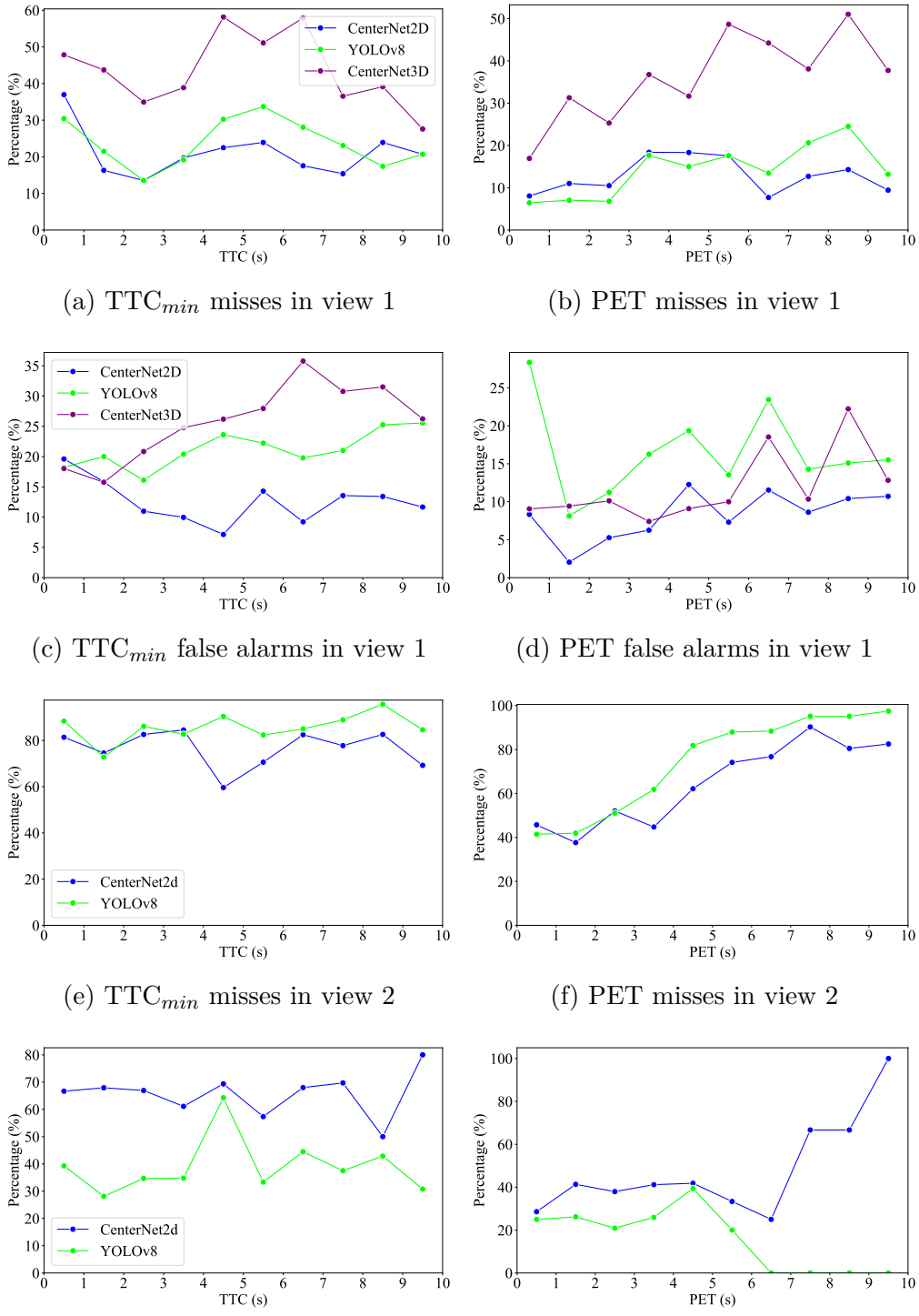
Regarding false alarms, CenterNet2D performs the best for both TTC_{min} and PET, whereas CenterNet3D performs the poorest for TTC_{min} and YOLOv8 performs the poorest for PET. All three methods have lower percentage of false alarms compared with misses, but the numbers are still high. 10 to 20% of the most dangerous predicted interactions (with values smaller than 1 s) are false alarms depending on the method and the indicator.

In View 2, both CenterNet2D and YOLOv8 miss more than 60 % of interactions for TTC_{min}, while the missed-interaction rate is lower for PET, at more than 35 %. It reaches almost 100% for high values of PET. Both methods also generate a high share of false alarms for TTC_{min} and PET. As the number of interactions with PET value in the range [6, 10] s is small (see Fig. 5.5d), the false alarms percentage of CenterNet2D is very high. It is surprising that the proportion remains very low for YOLOv8.

Yet, despite the sometime high values, there is no clear trend in the distributions, whether one method or another systematically misses or over-detects very dangerous interactions for example.

5.5.3 Discussion

In view 1, 2D methods outperform the 3D method in most categories, CenterNet2D performs best in most categories, except for the number of interactions with a PET and the number of *dangerous* interactions, as well as two HOTA metrics, where YOLOv8 outperforms Cen-



(a) TTC_{min} misses in view 1

(b) PET misses in view 1

(c) TTC_{min} false alarms in view 1

(d) PET false alarms in view 1

(e) TTC_{min} misses in view 2

(f) PET misses in view 2

(g) TTC_{min} false alarms in view 2

(h) PET false alarms in view 2

Figure 5.10 Distributions of the indicator values for missed interactions (including interactions without values in 1-to-1 matching in GT matching) and false alarms, expressed respectively as a percentage of the number of ground truth interactions (for missed interactions) and predicted interactions (for false alarms) within the same interval of indicator values.

terNet2D by a small margin. It is consistent with their performance measured by object and interaction matching, and indicator accuracy. Therefore, better performance of object detection and tracking generate better object trajectories as expected, which directly influence the performance for the safety indicators as illustrated by the object matching results.

CenterNet3D perform poorest in most categories. The reason why CenterNet3D performs poorest can be explained from two perspectives. Firstly, apart from 2D information, CenterNet3D needs to regress three variables for the 3D information, resulting in lower object detection and tracking accuracies compared with 2D methods. Its safety indicator performance is consistent with its object detection and tracking performance. Besides, it has a small but significant proportion of extremely dangerous interactions based on TTC_{min} that is mainly caused by inaccurate rotation angles, resulting in objects overlapping with each other, i.e. collisions. The mean absolute angular change $\bar{\sigma_\theta}$ is used to reflect the inaccuracy and inconsistency of the rotation angle predicted by CenterNet3D and is defined as

$$\bar{\sigma_\theta} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i - 1}\sum_{j=2}^{N_i}|(\theta_{i,j} - \theta_{i,j-1})| \tag{5.4}$$

where M, N_i, $\theta_{i,j}$ represent the number of trajectories, the temporal length of each trajectory i, and the j-th rotation angle of trajectory i, respectively.

For CenterNet3D, $\bar{\sigma_\theta}$ is 0.971 rad (55.65°), whereas for the ground truth, it is 0.01 rad (0.57°). The inaccuracy and inconsistency of the rotation angle lead to overlaps between bounding boxes when they are sufficiently close to each other, thus generating very small TTC_{min} values.

It is also worth investigating the relation between the object and interaction matching results. Regarding the object matching evaluation in view 1, all the methods generate over 40 % of false alarms in prediction matching. However, very few become interaction false alarms. It can be explained by three reasons:

1. Most of the object false alarms have no temporal overlaps, which generate few interactions;

2. In the interactions generated by CenterNet2D, object false alarms are involved in few interactions overall, while object false alarms for YOLOv8 and CenterNet3D cause more interactions, but few with values for TTC_{min} or PET, as shown in Table 5.7; and

3. Prediction matching is based on the interactions with values, which are not counted in interaction matching.

Table 5.7 Number of interactions involving object false alarms in view 1, with or without a value for TTC_{min} or PET ("with values" or "w/o values").

(a) TTC_{min}

Method	**w/o values**	**with values**
CenterNet2D	67	8
YOLOv8	646	63
CenterNet3D	363	36

(b) PET

Method	**w/o values**	**with values**
CenterNet2D	90	2
YOLOv8	725	23
CenterNet3D	370	22

Regarding the accuracy for the safety indicators, we observe that all three methods achieve better results on PET compared to TTC_{min}. This is intrinsically related to the complexity of computing them. PET depends only on detecting if two trajectories overlap and when each road user leaves and reaches the crossing point. This duration can be computed even without real-world coordinates or speed computation. PET is robust to small position errors or consistent spatial shifts of both trajectories. On the other hand, TTC_{min} requires accurate positions and velocities over whole time intervals, from which the minimum is derived. Such calculations are more sensitive to missed detections, inaccurate and noisy positions, which generate in turn inaccurate velocities.

Besides, for the missed interactions, we noticed that all three methods have a high percentage of missed interactions with TTC_{min} in the range [0, 1] s in view 1 (Fig. 5.10a). In addition to the unmatched interactions, this issue may also be caused by automated methods generating short trajectories that match only a small portion of the ground truth trajectories, resulting in missing TTC_{min} values in the predictions. The decrease of the missed proportion with TTC_{min} values in the [1, 3] s range is related to the peak in the ground truth distribution. Therefore, it is rare to see results like those in the [0, 1] s range, where all three methods exhibit relatively high proportions.

Some observations in view 1 are also reflected in view 2: better object and tracking performance leads to more accurate safety indicators and PET is easier to be predicted than TTC_{min}. Indeed, in view 2, the performance of object detection and tracking of both CenterNet2D and YOLOv8 is much lower compared to view 1, which is consistent with the performance on other indicators. This shows that even within the same intersection, changes in camera views significantly impact the performance of deep learning-based methods. This may be attributed to the occlusion caused by trees in this view, and the smaller size of objects in view 2 compared to view 1.

This study has obvious limitations, first and foremost related to the dataset used for the experimental results. It is unclear how the results may change at sites with different characteristics like road geometry, traffic volume, traffic composition, and signal control, in different weather and visibility conditions, or with different camera views, e.g. if the objects are closer. There are also methodological limitations. The computation of TTC relies on the commonly used but simple method of motion prediction at constant velocity [114]. The impact of more realistic motion prediction methods on the accuracy of the safety indicators derived from the trajectories generated by the different detection and tracking methods is difficult to anticipate and should be investigated in future work. Besides, the proposed object and interaction matching methods involve multiple experimental thresholds. While effective in our setting, such threshold-dependent approaches may limit generalization across datasets, suggesting the need for more robust and adaptive matching strategies.

5.6 Conclusion

In this study, we developed a pipeline for exploring the performance of object detection and tracking methods on road safety analyses. To the best of our knowledge, this is the first study on the evaluation of automated video-based methods for road safety analysis and the computation of the two most common safety indicators, TTC_{min} and PET. Three object detection methods, CenterNet2D, YOLOv8 and CenterNet3D and one object tracking method, ByteTrack, were applied on a crowded roadside dataset. We developed a novel object and interaction matching method to characterize the performance of their predictions compared with the ground truth.

Results show that CenterNet2D performed the best on most metrics compared to YOLOv8 and CenterNet3D. The gap between CenterNet2D and YOLOv8 is small while CenterNet3D performs significantly worse. For different views of the dataset, both 2D methods perform better in view 1 than in view 2. PET is easier to compute than TTC_{min}.

The observed gap between computer vision-based approaches and the ground truth underscores the difficulty of accurately conducting safety analyses in real traffic environments. While object detection and multi-object tracking (MOT) are well-studied in computer vision, leveraging these methods for safety analyses, including safety indicator computation, remains a challenging task that may not be accurate enough for many applications.

Future work should therefore be conducted along two directions. The first is to improve computer vision methods, and methods for other sensors like LiDAR, that generate road user trajectories to ensure better performance in safety analysis, as measured by the method

proposed in this paper. Efforts should be made to handle imperfect data with issues such as missing detections, noise, small objects, and occlusions. Research is needed to ensure pretrained models can be transferred to new sites and camera views, with minimal to no retraining. The second direction deals with the proposed method to evaluate the safety performance of automated trajectory extraction methods. The most urgent task is to replicate this study on other datasets with different characteristics and using other trajectory extraction methods, including from other data sources. Benchmarks are needed to establish the best methods and conditions with acceptable performance. Finally the proposed evaluation method should be refined to robustly support these benchmarks.

Acknowledgement

The authors gratefully acknowledge financial support from the China Scholarship Council, as well as the financial support of NSERC through the Discovery Grant 2023-05626. The authors would also like to thank the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for the computing servers and to NVIDIA Corporation for providing two GPUs. This research was also enabled in part by support provided by Digital Research Alliance of Canada, using CPU and GPU resources on the Cedar and Béluga clusters.

Author Contributions

Qingwu Liu, Nicolas Saunier, and Guillaume-Alexandre Bilodeau jointly conceptualized the research idea. Qingwu Liu implemented the algorithms, conducted data preprocessing and experiments, performed visualization of the results, and wrote the initial draft of the manuscript. Nicolas Saunier provided the Traffic Intelligence codebase and contributed to the interpretation of the results. Nicolas Saunier and Guillaume-Alexandre Bilodeau participated in the discussion and analysis of the experimental results, and reviewed and revised the manuscript. All authors read and approved the final manuscript.

CHAPTER 6 ARTICLE 3: PMMA: THE POLYTECHNIQUE MONTRÉAL MOBILITY AIDS DATASET

Qingwu Liu[1,2], **Nicolas Saunier**[1], **and Guillaume-Alexandre Bilodeau**[2]

[1] Civil, Geological and Mining Engineering Department, Polytechnique Montreal

[2] Lab. LITIV, Computer Engineering and Software Engineering Department, Polytechnique Montreal

Submit to the journal *IEEE Open Journal of Intelligent Transportation Systems*

Submitted: 31 March 2026, under review

Dataset DOI: https://doi.org/10.5683/SP3/XJPQUG

Abstract

This study introduces a new object detection dataset of pedestrians using mobility aids, named PMMA. The dataset was collected in an outdoor environment, where volunteers used wheelchairs, canes, and walkers, resulting in nine categories of pedestrians: pedestrians, cane users, two types of walker users, whether walking or resting, five types of wheelchair users, including wheelchair users, people pushing empty wheelchairs, and three types of users pushing occupied wheelchairs, including the entire pushing group, the pusher and the person seated on the wheelchair. To establish a benchmark, seven object detection models (Faster R-CNN, CenterNet, YOLOX, DETR, Deformable DETR, DINO, and RT-DETR) and three tracking algorithms (ByteTrack, BOT-SORT, and OC-SORT) were implemented under the MMDetection framework. Experimental results show that YOLOX, Deformable DETR, and Faster R-CNN achieve the best detection performance, while the differences among the three trackers are relatively small. The PMMA dataset is publicly available at: `https://doi.org/10.5683/SP3/XJPQUG` and the video processing and model training code is available at: `https://github.com/DatasetPMMA/PMMA`.

Keywords: computer vision, image processing, pedestrian flows and crowds, mobility aids, object detection and tracking dataset

Figure 6.1 Example frames of our Mobility Aids Dataset. There are nine categories, with detailed descriptions in Table 6.3 and icons adapted from [5, 6]

6.1 Introduction

Cameras have become ubiquitous in traffic environments, serving as essential tools for capturing and analyzing road users to better understand their behaviors and anticipate their movements. In both computer vision and intelligent transportation systems, object detection and tracking are fundamental tasks. Existing publicly available object detection and tracking datasets classify road users into broad categories such as pedestrians, cyclists, cars and trucks. However, few datasets provide finer-grained distinctions among pedestrians, particularly pedestrians who use mobility aids such as wheelchairs, walkers, or canes. Although individuals using mobility aids represent a small fraction of road users, they are more vulnerable than other groups. This underscores the need for dedicated datasets that

specifically identify mobility aid users and enable a thorough evaluation of state-of-the-art (SOTA) detection and tracking algorithms. Such efforts are crucial for advancing inclusive and safety-focused intelligent transportation systems.

Image-based object detection and tracking methods have seen significant progress, driven not only by the advancement of computational power, the increase of GPU resources, but also by the evolution of deep learning models, such as convolutional neural networks (CNN) [157] and Transformers [31], the availability of large-scale annotated datasets, and improved training techniques. These advancements have enabled SOTA models to achieve strong performance on public datasets such as COCO [156], KITTI [7], and Cityscapes [158]. However, due to the limited category granularity in existing datasets, how these methods perform when faced with multiple visually similar categories remains an open question.

Therefore, we introduce in this paper our mobility aids dataset, which contains over 28,000 annotated images with a resolution of 2208 $\times$ 1242 pixels and a maximum frame rate of 15 frames per second (fps). We collected the dataset using a ZED 2 stereo camera in an outdoor parking lot of Polytechnique Montréal, located in Montréal, Québec, Canada. Three videos were annotated from two points of view with nine categories of pedestrians using computer vision annotation tools (CVAT) [87], as visualized in Figure 6.1. We compare our dataset with existing object detection datasets in Table 6.1.

To evaluate the performance of detectors and trackers in our dataset, we benchmark seven detectors and three trackers on our dataset. We further discuss their performance and the shortcomings. Finally, we summarize the most challenging categories and the limitations of those methods on our dataset.

The main contributions of this study are summarized as follows:

1. We constructed a dataset containing nine categories of pedestrians with mobility aids, with annotations provided for all images and occlusion information recorded for each annotated instance.

2. We applied seven object detection methods and three tracking methods on the dataset; and

3. We analyzed the experimental results and highlighted the main challenges associated with the detection and tracking of pedestrians using mobility aids.

The remainder of this paper is organized as follows. Section 6.2 presents the related work. Section 6.3 introduces the details of our dataset. Section 6.4 presents the experimental

protocol and evaluation metrics. Section 6.5 presents the results and the discussion, and section 9 concludes the paper.

6.2 Related work

In this section, prior datasets and data collection paradigms relevant to our work are reviewed. We begin by discussing autonomous-driving datasets collected from vehicle-mounted sensors, which provide rich annotations but remain limited by car-centric visibility and short sequence durations. We then review fixed-camera and multi-camera surveillance datasets that offer high-density pedestrian tracking from elevated viewpoints. Finally, we highlight datasets that, similar to ours, focus on fine-grained pedestrian sub-categories, particularly those involving mobility aids.

Several widely used datasets in the autonomous driving domain have been collected using vehicle-mounted sensors. Notable examples include KITTI [7], Waymo Open Dataset [159], SemanticKITTI [160], JAAD [161], nuScenes [162], and Cityscapes [158]. These datasets provide high-quality sensor data, such as LiDAR and RGB images, collected as vehicles drive through urban environments. The low camera position and occlusions by other vehicles limit visibility, and most of the data is captured in multiple short-duration segments, typically lasting only a few seconds. Even with longer sequences, detailed traffic analysis remains challenging due to the inherent limitations of the vehicle perspective. In addition, although these datasets are useful for tasks, such as object detection and semantic segmentation in road scenes, they mainly focus on general object categories, such as pedestrians, cars, and cyclists, without distinguishing between pedestrians with or without mobility aids.

In crowded public spaces, several datasets focus on fixed-camera surveillance or multi-camera scenarios. MOT20 [163] offers high-density pedestrian tracking sequences from viewpoints at lamppost or building height. The A9-Dataset [164] provides high-resolution multi-modal data collected from roadside sensor infrastructure along a three kilometers stretch of the A9 autobahn near Munich, Germany. It includes precision-timestamped camera and LiDAR frames captured from overhead gantry bridges, with objects annotated using 3D bounding boxes. WILDTRACK [165] and CityFlow [166] are multi-view datasets that allow for 3D localization and multi-camera tracking. LUMPI [8] is another multi-perspective dataset designed for analyzing pedestrian behavior across viewpoints. While these datasets provide multi-angle coverage and dense pedestrian annotations, they do not include detailed pedestrian categories either.

Similar to our dataset, a few datasets categorize pedestrians into multiple sub-classes to

enable fine-grained analysis. SynPoses [167] proposes a framework for generating high-quality synthetic human data with diverse and complex poses, addressing the limited pose variability in existing real-world datasets. RAP [168] is a large-scale pedestrian dataset designed for person retrieval in real surveillance environments, featuring rich attribute annotations and identity labels that enable both attribute-based and image-based person re-identification. The dataset mainly focuses on appearance-related attributes, such as clothing styles and hair characteristics, while mobility-aid users are not explicitly considered. Some datasets explicitly consider categories related to mobility aids. For instance, the dataset proposed by Sonia Dávila-Soberón et al. [52] specifically focuses on wheelchair users, cane users, and the mobility aids themselves. However, this dataset is not publicly available, which limits its accessibility for broader research purposes. Another dataset [53] targets mobility aid users in an indoor hospital environment, providing valuable data for healthcare-related applications but lacking outdoor scene variability and typical viewpoints. In addition, Zhang et al. [54] introduced a synthetic dataset in a simulation environment called X-World, which includes objects such as wheelchairs and canes. While useful for controlled experiments, simulation-based data often lacks the complexity and diversity of real-world environments.

To the best of our knowledge, there is currently no publicly available dataset that captures in real-world outdoor conditions people using diverse mobility aids, such as wheelchairs, walkers and canes, while also supporting tasks like object detection and multi-object tracking (MOT). This highlights a gap in existing datasets for developing and evaluating vision-based systems that aim to understand and assist mobility-impaired individuals in dynamic, open outdoor environments.

6.3 The PMMA dataset

6.3.1 Data Collection

Acquisition and ethics

The data collection for this dataset was approved by the ethics committee ("Comité d'éthique à la recherche avec des êtres humains") of Polytechnique Montréal. All participants were volunteers who consented to be filmed and have their data included in the dataset. None of the participants are actual users of mobility aids; instead, they are graduate students performing scenarios to simulate behaviors with mobility aids. Volunteers change and use different mobility aids within each session.

Location

The dataset was recorded from two camera positions in an outdoor parking lot of Polytechnique Montréal, in Montréal, Québec, Canada. The recording zone captured from Google Maps is shown in Figure 6.2. We focus on the area delineated by traffic cones, which marks the boundary of the region of interest in the scene, highlighted in Figure 6.1.

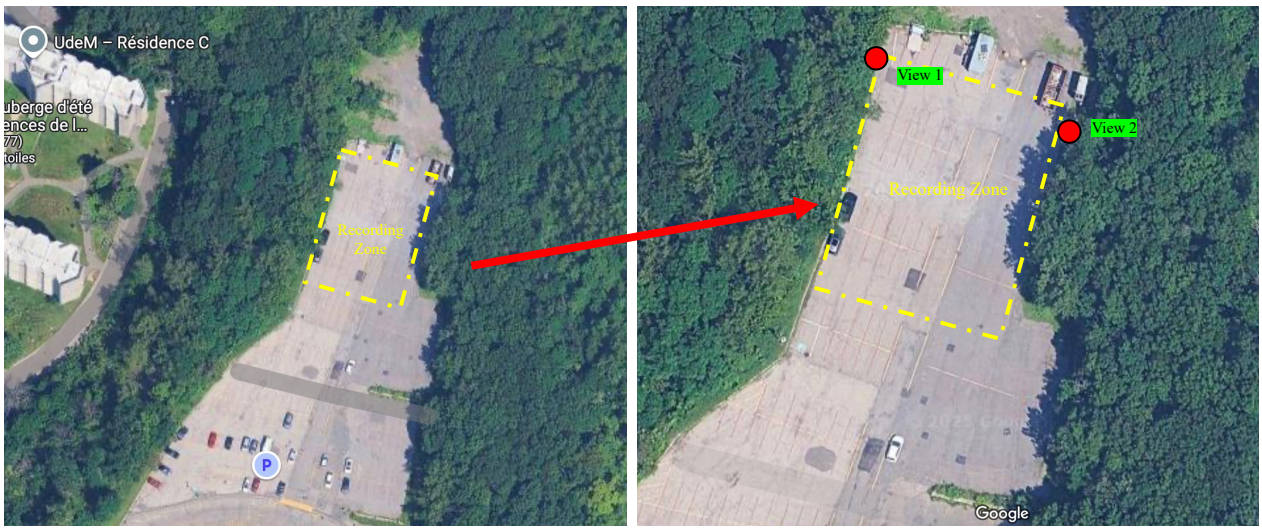


Figure 6.2 Data collection area in the parking lot of Polytechnique Montréal

Camera

The dataset was recorded utilizing a stereo camera ZED 2 [169]. It provides dual streams in a side-by-side configuration with resolution 2208×1242 at 15 frame per second (fps) and has a maximum field of view (FOV) of horizontal $\times$ vertical $\times$ diagonal angles of $110° \times 70° \times 120°$. The camera was mounted on a 4-m-high pole, which was mounted on two distinct streetlight poles giving two separate viewpoints: one of the installations is shown in Figure 6.3. A Linux system laptop was used to record the video stream from the camera via the VLC media player [170].

Recording sessions

Three recording sessions were conducted on the same day. The sunlight varies throughout the recording period due to partially cloudy weather conditions. The recording sessions are summarized in Table 6.2.

Figure 6.3 Illustration of the camera and the pole

Object categories

The mobility aids utilized in this data collection includes three wheelchairs, two walkers and two canes. With these mobility aids, the dataset includes nine pedestrian categories, representing a variety of mobility aids and users, shown in Table 6.3. Going into details, the walker category is further refined into two subcategories based on the activity status, walking or resting. These subcategories allow for more fine-grained behavior analysis, though they can be easily merged into a single walker category if needed. For wheelchair-related instances, we introduce a more structured annotation strategy. Specifically, we consider a wheelchair and the associated person(s) as a single unit and classify them into five distinct subcategories: a person alone in a wheelchair, a person pushing an empty wheelchair, a person pushing a person in a wheelchair as a group and its components separately, the person pushing and the seated person.

This hierarchical categorization allows us to model complex pedestrian interactions and mobility aid usage more accurately, while maintaining the flexibility to merge classes as needed for specific tasks or evaluation protocols.

Table 6.1 Comparison of the PMMA dataset statistics with existing benchmarks

Dataset	View	Images	Categories with persons	Outdoor	Real world	Mobility Aids
Vehicle-perspective						
KITTI [7]	ego	15k	3	✓	✓	×
Waymo [159]	ego	1M	2	✓	✓	×
Cityscapes [158]	ego	25k	2	✓	✓	×
JAAD [161]	ego	346k	1	✓	✓	×
nuScenes [162]	ego	1.4M	2	✓	✓	×
Multi-perspective						
WILDTRACK [165]	surv+ego	215k	1	✓	✓	×
CityFlow [166]	surv+surv	118k	2	✓	✓	×
A9-Dataset [164]	surv	5.4k	3	✓	✓	×
LUMPI [8]	surv+ego	200k	3	✓	✓	×
Detailed categories						
Sonia Dávila-Soberón et al. [52]	surv	2.8k	3	✓	✓	✓
Indoor hospital [53]	ego	17k	5	×	✓	✓
X-world [54]	surv	72k	6	✓	×	✓
PMMA (Ours)	surv	28k	9	✓	✓	✓

Notes: surv and ego denote surveillance and vehiclemounted views, respectively.

Table 6.2 Details of the collected video recordings

Recording session	Duration	FPS	Position	Number of Frames
Video 1	10 min	15	1	9,000
Video 2	10 min	15	1	5,000
Video 3	15 min	13	2	14,637

Table 6.3 Description of pedestrian-related categories

Icon	Category	Description
	Ped	Pedestrian without mobility aid
	Cane	Pedestrian using a cane
	Wheelchair	Person in a self-propelled wheelchair
	WalkerWalking	Person walking with a walker
	WalkerResting	Person resting with a walker
	PushEmptyWheelchair	Person pushing an empty wheelchair
	WheelchairGroup	Person pushing a person in a wheelchair as a group
	WheelchairPusher	Person pushing only (from WheelchairGroup)
	WheelchairPushedUser	Person in a wheelchair only (from WheelchairGroup)

Table 6.4 Frame distribution across training, validation, and test sets from different video sources

	Train		**Validation**		**Test**	
Video	Video 1	Video 3	Video 2	Video 3	Video 2	Video 3
Number	9,000	13,248	1,700	534	4,357	799
Total	22,248		2,234		5,156	

6.3.2 Annotation

Video pre-processing

Each stereo video is saved to a sequence of stereo images firstly. Each stereo image is then separated into left and right images, and the left images are annotated.

Annotation tool

We use the computer vision annotation tool (CVAT) [87], which is an online platform for annotating object detection and tracking. The annotations are in the COCO format to facilitate model training and evaluation.

Annotation details

The annotations were separated into three tasks with respect to the three capture sessions. The annotations were initially generated by labeling key frames at every 100 frames using the interpolation-based tracking tool. Linear interpolation was applied between key frames to estimate intermediate object locations, followed by manual frame-by-frame corrections to ensure accuracy.

We also include occlusion attributes in the annotations. Following common occlusion conventions, such as those used in KITTI [7], we assign labels 0, 1, and 2 to represent no occlusion, partial occlusion, and full occlusion, respectively. Additionally, we introduce an extra label, 3, to indicate objects in the tree shadows, which is not defined in the original KITTI annotation scheme.

6.3.3 Statistics

Around 30,000 images were annotated in our PMMA dataset. Figure 6.4 illustrates the annotation counts for each category per video. The frame distributions for the training, validation and test sets are shown in Table 6.4. For the tracking task, the two video clips in the test set are used to evaluate the tracker performance.

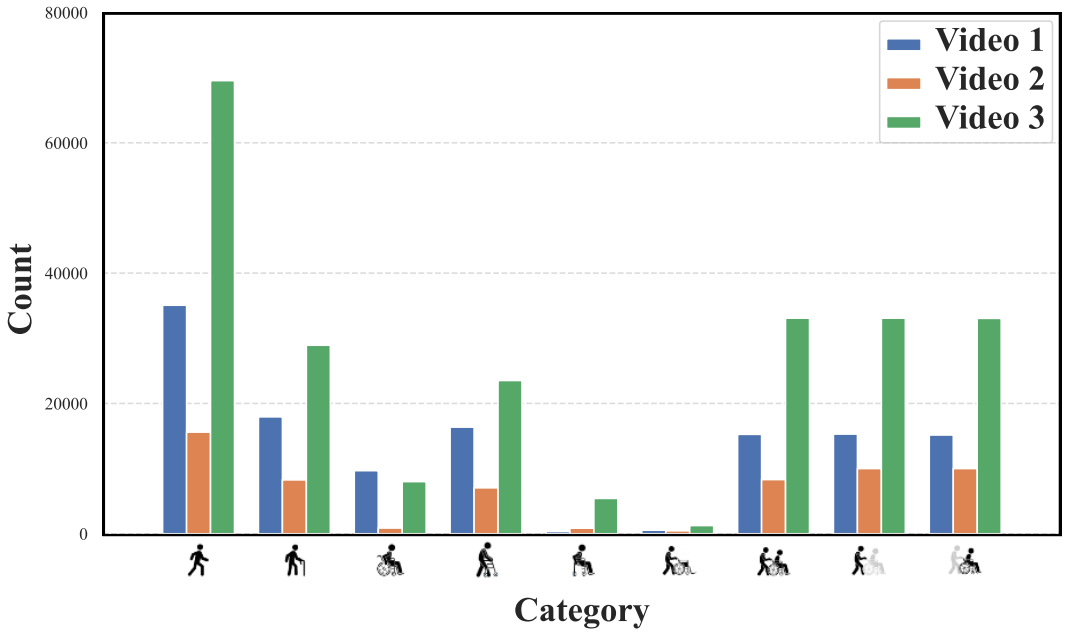


Figure 6.4 Category-wise annotation counts for each video

Table 6.5 Comparison of object detection models

Model	Type	Stage	Anchor-free	Backbone
Faster R-CNN [21]	CNN	Two-stage	No	ResNet-50
CenterNet [28]	CNN	One-stage	Yes	Hourglass
YOLOX [171]	CNN	One-stage	Yes	CSPDarknet
DETR [29]	Transformer	One-stage	Yes	ResNet-50
Deformable DETR [30]	Transformer	One-stage	Yes	ResNet-50
DINO [172]	Transformer	One-stage	Yes	ResNet-50
RT-DETR [173]	Transformer	One-stage	Yes	ResNet-50

Table 6.6 Comparison of tracking methods (Re-ID refers to the use of appearance-based features to associate detections across frames, enabling identity recovery after occlusion or long-term target loss.)

Model	Re-ID	Detection-based Tracking
ByteTrack [39]	No	Yes
BOT-SORT [40]	Yes	Yes
OC-SORT [41]	No	Yes

6.4 Experimental methodology

6.4.1 Introduction

To establish baseline performance on our dataset, we conducted a series of experiments comparing the performance of object detection and MOT methods. For object detection, we used the MMDetection [174] framework and selected seven representative methods: Faster R-CNN [21], CenterNet [28], YOLOX [171], DETR [29], Deformable DETR [30], DINO [172] and RT-DETR [173], as summarized in Table 6.5. All models were pretrained on the COCO dataset and then trained on our dataset to evaluate their detection performance across the nine pedestrian-related categories.

For MOT, we applied three trackers, including ByteTrack [39], BOT-SORT [40] and OC-SORT [41], on the detection results, as summarized in Table 6.6. This experimental setup allows us to establish baseline detection and tracking performance on the dataset and provides a reference for future research using our data.

For object detection, we trained for 50 epochs and we use early-stopping. We used an AdamW [175] optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}.

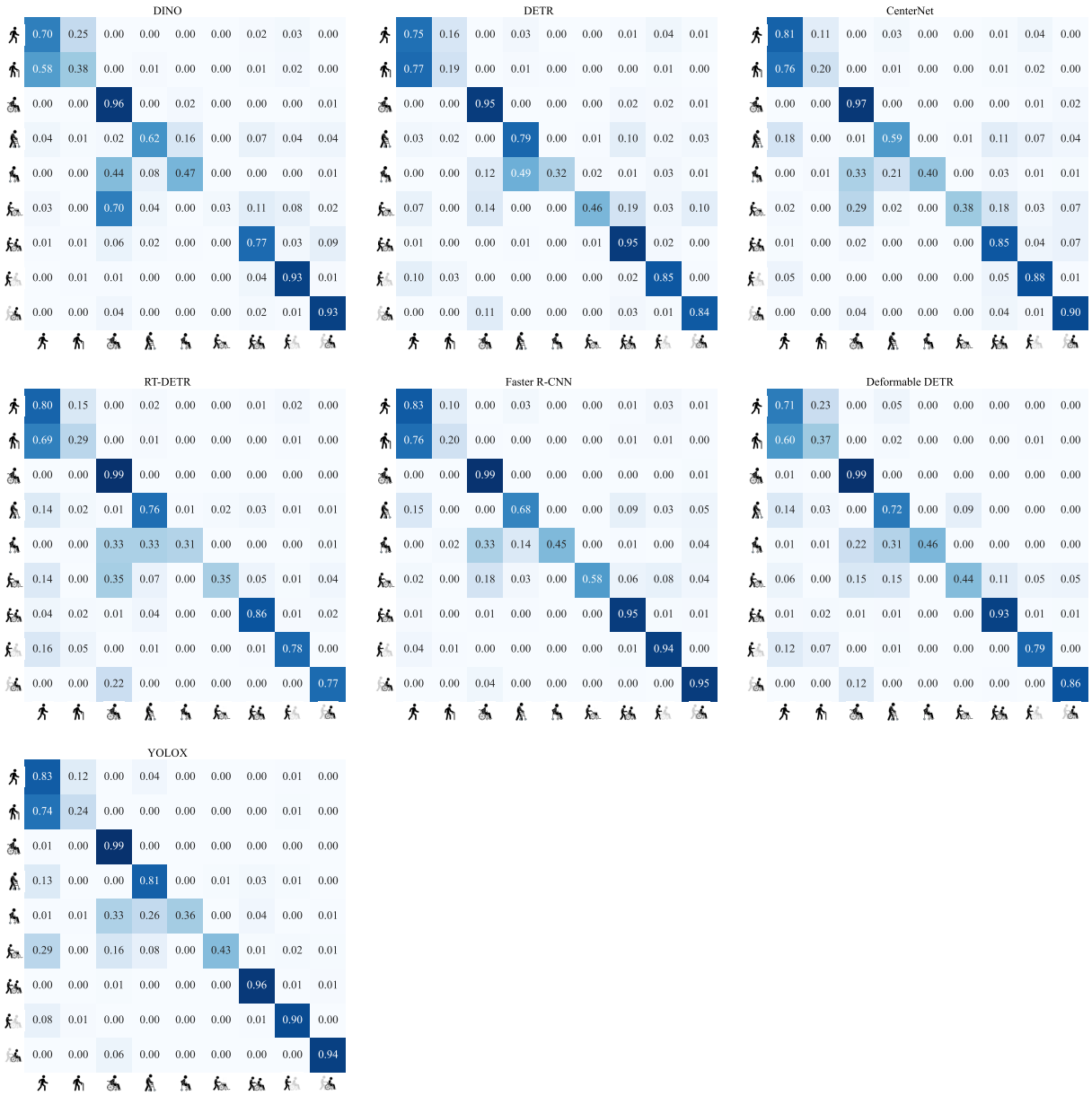


Figure 6.5 Row-normalized confusion matrices of all detection methods on the test set. Each row and column represent the ground truth (GT) and predicted classes, respectively

Table 6.7 Detection performance evaluated on test set across all categories

Method	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**AP**$_S$(%)	**AP**$_M$(%)	**AP**$_L$(%)	**mAR**(%)	**AR**$_S$(%)	**AR**$_M$(%)	**AR**$_L$(%)
DINO	27.8	55.6	24.1	7.2	24.9	42.2	57.5	9.9	53.2	**73.1**
DETR	30.4	60.7	26.7	0.0	29.7	38.6	45.4	0.3	44.8	52.7
RT-DETR	40.0	59.3	48.6	19.3	38.1	43.0	47.3	21.3	44.8	51.1
CenterNet	44.4	67.7	52.5	**46.9**	48.9	<u>45.0</u>	<u>60.7</u>	*47.0*	*62.6*	61.4
Deformable DETR	<u>47.4</u>	<u>68.4</u>	<u>58.4</u>	*22.2*	*50.4*	42.6	**64.7**	**54.4**	**65.5**	*64.5*
Faster R-CNN	*49.4*	**73.9**	*60.5*	<u>19.8</u>	**51.5**	**49.5**	<u>58.8</u>	21.9	<u>59.8</u>	<u>59.5</u>
YOLOX	**49.5**	*69.5*	**61.9**	14.8	<u>49.9</u>	*49.0*	57.4	<u>32.8</u>	58.8	56.0

Notes: Bold, *italic*, and <u>underline</u> represent the first, second, and third best performances, respectively.

6.4.2 Evaluation metrics

Object detection

We evaluated the performance of the detection methods using the standard COCO metrics [156]:

- **mAP:** The mean of average precision (AP), where AP is computed by integrating the precision–recall curve at a fixed IoU threshold, measuring overall detection performance.
- **AP**$_{50}$**:** Average precision over multiple recall levels at an Intersection over Union (IoU) threshold of 0.5, measuring detection accuracy when predicted boxes overlap with ground truth by at least 50%.
- **AP**$_{75}$**:** Average precision over multiple recall levels at an IoU threshold of 0.75, reflecting stricter overlap criteria for more precise detections.
- **AP**$_S$, **AP**$_M$, **AP**$_L$: Average precision scores over multiple recall levels for small (area $\leq 32^2$ pixels), medium ($32^2 <$ area $< 96^2$ pixels), and large (area $\geq 96^2$ pixels) objects, respectively, assessing performance across different object scales.
- **mAR:** The mean average recall over multiple IoU thresholds, indicating the model's ability to find all relevant objects.
- **AR**$_S$, **AR**$_M$, **AR**$_L$: Average recall for small, medium, and large objects, respectively, where the object size definitions are the same as those used for AP$_S$, AP$_M$, and AP$_L$.
- **mAP** and **mAR** are also reported across different occlusion levels, including no occlusion, partial occlusion, and full occlusion. Objects in shadows are included in the Full occlusion category.

All evaluation metrics are reported for the overall performance across the nine categories, whereas for the category-wise analysis, only mAP, AP_{50}, and AP_{75} are presented.

Object tracking

We adopted the Higher Order Tracking Accuracy (HOTA) metric [149] for each test set. HOTA is a comprehensive metric that jointly considers detection accuracy (DetA), association accuracy (AssA), and localization accuracy (LocA). HOTA is defined as follows:

$$\mathrm{HOTA}_\alpha = \sqrt{\frac{\sum_{c\in\{\mathrm{TP}\}} \mathcal{A}(c)}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}} \tag{6.1}$$

where |TP|, |FN|, and |FP| denote the number of true positives, false negatives, and false positives, respectively. Each c represents a true positive match between a ground truth ID and a predicted ID. $\mathcal{A}(c)$ denotes the *association accuracy* of match c, which is defined as:

$$\mathcal{A}(c) = \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(c)|} \tag{6.2}$$

where $|\mathrm{TPA}(c)|$, $|\mathrm{FNA}(c)|$, and $|\mathrm{FPA}(c)|$ refer to the number of true positive, false negative, and false positive associations related to match c, respectively.

Specifically, we report HOTA along with DetA, AssA, and LocA, presenting both the overall performance across all nine categories and the category-wise performance. We also include the total number of detected objects and tracked identities.

6.5 Results and discussion

6.5.1 Object detection

The object detection performance across all categories is given in Table 6.7. YOLOX, Deformable DETR, and Faster R-CNN achieve the top three performances across most metrics. Specifically, YOLOX has the best AP while Deformable DETR has the best AR. However, the performance differences among CenterNet, RT-DETR, Faster R-CNN, Deformable DETR, and YOLOX are relatively small, whereas DINO and DETR lag significantly behind the other five.

Moreover, the object detection performance for each category is given in Table 6.8. YOLOX, Deformable DETR, and Faster R-CNN achieve the best performance. CenterNet performs comparably well to these three methods. DETR and DINO consistently exhibit the lowest

performance.

Table 6.8 Detection performance for each of the nine categories on the test set

Method	Ped			Cane			Wheelchair		
	mAP(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)
DETR	23.1	51.7	16.4	4.7	11.2	3.1	51.1	76.8	63.6
DINO	27.8	58.1	22.6	9.0	**26.5**	5.3	34.9	66.0	20.0
RT-DETR	37.2	60.6	41.1	9.2	16.3	*9.8*	59.0	78.7	73.7
Faster R-CNN	36.5	61.3	38.9	8.7	16.6	7.8	62.2	86.4	79.4
CenterNet	38.5	*63.1*	42.0	*10.2*	18.9	9.5	46.0	62.7	61.1
Deformable DETR	*39.7*	62.3	*47.3*	**14.0**	*26.0*	**13.9**	*69.5*	*87.7*	*86.9*
YOLOX	**44.1**	**66.2**	**53.7**	8.8	16.2	9.0	**72.2**	**88.9**	**88.4**
Method	WalkerWalking			WalkerResting			PushEmptyWheelchair		
	mAP(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)
DINO	8.4	12.4	11.1	39.1	68.6	42.4	12.7	38.0	2.3
DETR	24.1	43.7	23.7	37.6	71.0	38.1	24.3	44.7	27.8
RT-DETR	25.7	35.8	34.0	48.0	73.3	56.1	21.6	35.2	25.8
CenterNet	*41.4*	*55.0*	*54.2*	47.9	70.7	56.4	35.4	*60.9*	38.8
Deformable DETR	23.8	33.1	30.3	45.6	73.1	50.9	*40.5*	55.8	*52.4*
YOLOX	34.4	46.9	44.3	**53.9**	**79.5**	**62.6**	27.3	39.5	38.9
Faster R-CNN	**52.2**	**71.6**	**70.3**	*48.6*	*77.7*	*56.9*	**44.2**	**61.3**	**58.6**
Method	PushWheelchair1			PushWheelchair2			PushWheelchair3		
	mAP(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)	**mAP**(%)	**AP**$_{50}$(%)	**AP**$_{75}$(%)
DETR	42.2	91.5	25.4	26.3	73.5	11.3	40.7	82.4	30.8
DINO	51.9	80.4	64.8	30.7	68.9	23.9	35.9	81.6	24.0
RT-DETR	60.8	79.9	77.4	47.3	75.5	55.6	50.7	78.1	63.9
CenterNet	68.7	92.2	85.0	56.5	*92.5*	64.2	55.1	93.0	61.2
Deformable DETR	72.0	94.8	91.7	57.6	89.8	*69.6*	*64.3*	93.3	*82.3*
Faster R-CNN	*74.0*	*97.1*	*93.3*	*58.5*	**95.9**	68.0	59.7	*97.3*	71.7
YOLOX	**74.4**	**98.3**	**95.0**	**59.9**	92.3	**74.0**	**70.8**	**97.6**	**90.9**

Regarding the categories, cane users are the most challenging to detect. This could be attributed to the fact that the cane, being a thin object, is difficult to detect and cane users are therefore confused with pedestrians without any mobility aid. Wheelchair-related categories achieve higher mAP and mAR; however, PushEmptyWheelchair shows the lowest accuracy among them, possibly due to its visual similarity to WheelchairGroup. Figure 6.5 shows that the confusion matrices on the test set are similar for all the methods. There is substantial confusion (1) between pedestrians without any mobility aid and cane users, and (2) between wheelchair users and walker users.

Table 6.9 presents the detection results with respect to different occlusion levels. DINO has the best mAP in case of no occlusion and poor performance on the partial and full occlusion cases. Generally, Faster R-CNN, Deformable DETR and YOLOX have the top performance consistent with the overall detection results.

6.5.2 Object tracking

Based on the detection performance, we applied three tracking algorithms to the top five methods, excluding DINO and DETR. We report the object tracking results on overall performance and per-category performance separately on videos 2 and 3 from the test set in

Table 6.9 Detection results on the test set with respect to different occlusion levels

Methods	**mAP**(%)			**mAR**(%)		
	No	Partial	Full	No	Partial	Full
DETR	45.20	10.34	8.33	70.56	72.63	63.38
CenterNet	50.99	12.37	10.04	76.79	84.42	75.04
RT-DETR	52.50	11.28	**11.19**	71.87	69.35	73.82
Deformable DETR	53.83	11.91	10.89	70.32	69.36	68.25
Faster R-CNN	54.10	*13.03*	10.79	**81.85**	**88.36**	**81.02**
Faster R-CNN	54.10	*13.03*	10.79	**81.85**	**88.36**	**81.02**
YOLOX	*55.98*	**13.38**	*11.05*	*80.22*	*85.28*	*78.11*
DINO	**57.06**	10.56	5.72	57.56	48.73	29.48

Table 6.10.

For overall performance, among the detectors, YOLOX achieves the best HOTA on video 2, while Deformable DETR performs best on video 3. Among the trackers, ByteTrack consistently outperforms the others on both videos, though the differences between the three trackers are relatively small and OC-SORT with Faster R-CNN on video 2 and BOT-SORT with Deformable DETR on video 3 show good performance. All trackers show strong DetA and LocA. The AssA for video 2 is relatively low, primarily because the tracker generates more IDs than the ground truth objects, resulting in a reduced overall HOTA score.

The category-wise performances are summarized in Tables B.1 to B.6. On video 2, the performance of all methods on the categories Cane, WalkerResting, and PushEmptyWheelchair is lower than 10 %; The performance on the categories Ped and WalkerWalking is between 10 % and 40 %; the performance on Wheelchair, WheelchairGroup, WheelchairPusher and WheelchairPushedUser is the highest, with the best performance over 40 %. YOLOX and Deformable DETR have quite close performance. On video 3, all the methods perform better on Ped, Cane, Wheelchair, WheelchairGroup, WheelchairPusher and WheelchairPushedUser, with the best performance over 60 %. Deformable DETR and YOLOX demonstrate the strongest overall performance across most mobility-related categories. On Video 2, Deformable DETR ranks within the top three for 6 of the 9 categories, while YOLOX achieves top-three performance in all 9 categories. On Video 3, Deformable DETR attains top-three results in 8 of the 8 evaluated categories, and YOLOX performs similarly with 7 top-three finishes. Overall, the performance differences among the trackers are relatively small; detection accuracy remains the key factor influencing tracking performance.

Table 6.10 Tracking performance across different detectors and trackers for all categories on videos 2 and 3 from the test set, sorted by HOTA

Video 2									
Detector	**Method**	**Tracking Metrics (%)**				**Detection Counts**		**Track ID Counts**	
		HOTA	**DetA**	**AssA**	**LocA**	**Predictions**	**GT**	**Predictions**	**GT**
Deformable DETR	OC-SORT	32.41	62.63	16.92	84.82	53,970		453	
CenterNet	OC-SORT	33.38	63.16	17.80	83.69	63,446		617	
CenterNet	BOT-SORT	34.10	60.35	19.43	83.54	74,418		1,543	
RT-DETR	OC-SORT	34.14	63.60	18.50	82.87	60,730		468	
RT-DETR	BOT-SORT	35.62	62.82	20.36	82.6	67,962		1,075	
RT-DETR	ByteTrack	35.87	63.74	20.36	82.83	66,914		584	
CenterNet	ByteTrack	36.27	61.29	21.61	83.75	72,990		668	
Deformable DETR	BOT-SORT	37.88	67.68	21.36	84.43	64,415	64,678	1,125	35
Deformable DETR	ByteTrack	39.00	68.03	22.55	84.7	63,348		499	
Faster R-CNN	BOT-SORT	44.54	60.85	32.86	83.27	76,882		1,053	
Faster R-CNN	ByteTrack	45.34	61.76	33.59	83.27	75,869		527	
Faster R-CNN	OC-SORT	46.39	63.70	34.12	83.39	70,059		466	
YOLOX	BOT-SORT	52.62	67.93	*41.02*	85.02	71,718		910	
YOLOX	OC-SORT	*53.02*	**70.07**	40.38	*85.14*	65,796		407	
YOLOX	ByteTrack	**54.91**	*68.59*	**44.23**	**85.21**	70,864		451	
Video 3									
Detector	**Method**	**Tracking Metrics (%)**				**Detection Counts**		**Track ID Counts**	
		HOTA	**DetA**	**AssA**	**LocA**	**Predictions**	**GT**	**Predictions**	**GT**
CenterNet	OC-SORT	55.41	69.85	44.13	**94.94**	13,634		101	
CenterNet	BOT-SORT	62.50	65.34	59.97	84.89	15,374		290	
CenterNet	ByteTrack	65.70	66.19	65.47	84.7	15,129		108	
Faster R-CNN	OC-SORT	66.89	74.43	60.38	86.02	13,490		71	
RT-DETR	OC-SORT	69.08	76.16	63.12	85.54	12,941		47	
Faster R-CNN	BOT-SORT	70.88	72.66	69.43	86.06	14,232		187	
Faster R-CNN	ByteTrack	72.45	73.37	71.80	86.05	14,072		94	
YOLOX	ByteTrack	72.97	77.20	69.32	85.89	13,142	12,784	84	17
YOLOX	BOT-SORT	73.30	76.54	70.58	85.93	13,284		168	
Deformable DETR	OC-SORT	73.48	79.73	68.00	*87.06*	12,373		36	
YOLOX	OC-SORT	74.08	78.07	70.68	85.95	12,756		70	
RT-DETR	BOT-SORT	77.56	75.82	**79.85**	85.47	13,431		115	
RT-DETR	ByteTrack	77.71	76.23	*79.70*	85.43	13,333		52	
Deformable DETR	BOT-SORT	*79.30*	*80.50*	78.38	87	12,747		71	
Deformable DETR	ByteTrack	**79.42**	**80.53**	78.57	86.98	12,691		34	

Note: Methods are sorted based on HOTA in this table and the following ones.

6.6 Conclusion

In this study, we collected a mobility aids dataset, named PMMA, and used it to benchmark SOTA detection and tracking methods. PMMA contains high-resolution videos with annotations for nine mobility-aid-related categories, including pedestrians, cane users, two types of walker users (walking and resting), five types of wheelchair users, including wheelchair users, people pushing empty wheelchairs, and three types from wheelchair groups (Wheelchair group, wheelchair pusher and wheelchair user together with wheelchair). The dataset was divided into training, validation, and test sets. We evaluated seven object detectors, including Faster R-CNN, CenterNet, YOLOX, DETR, Deformable DETR, DINO, and RT-DETR,

and applied three MOT trackers, namely ByteTrack, BOT-SORT, and OC-SORT, on the top five detection models. We reported both overall and category-wise performance. The results show that YOLOX, Deformable DETR, and Faster R-CNN achieved the best detection performance, while the differences among the three trackers were small. In terms of categories, Wheelchair, WheelchairGroup, WheelchairPusher, and WheelchairPushedUser were the easiest to detect and track, whereas performance for the remaining categories was less stable and strongly influenced by the specific scenarios.

Overall, YOLOX, Deformable DETR, and Faster R-CNN achieved the best performance among the seven evaluated methods. They performed well on wheelchair-related categories. However, performance on the other categories varies significantly across different scenarios. The differences among the three trackers were minimal, as their results largely depended on the quality of the detections. Nevertheless, no single method performed well across all nine categories, highlighting the ongoing challenge of automatic data collection of mobility aid users. Transportation studies focusing on these users still require improving detection and tracking accuracy.

Acknowledgments

The authors gratefully acknowledge financial support from the China Scholarship Council, as well as the financial support of NSERC through the Discovery Grant 2023-05626. The authors would also like to thank the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for the computing servers and to NVIDIA Corporation for providing GPUs. This research was also enabled in part by support the computing resources provided by Digital Research Alliance of Canada.

Author contributions

Qingwu Liu, Nicolas Saunier, and Guillaume-Alexandre Bilodeau jointly conceptualized the research idea and contributed to the ethics approval process, as well as the preparation and execution of the data collection. Qingwu Liu implemented the algorithms, conducted data preprocessing and experiments, and wrote the initial draft of the manuscript. Nicolas Saunier and Guillaume-Alexandre Bilodeau participated in the discussion and analysis of the experimental results, and reviewed and revised the manuscript. All authors read and approved the final manuscript.

CHAPTER 7 ARTICLE 4: VIDEO-BASED ANALYSIS OF CROSSING BEHAVIOR OF PEDESTRIANS WITH ASSISTING AND WALKING DEVICES AT A SIGNALIZED INTERSECTION

Qingwu Liu[1,2], **Nicolas Saunier**[1], **and Guillaume-Alexandre Bilodeau**[2]

[1] Civil, Geological and Mining Engineering Department, Polytechnique Montreal

[2] Lab. LITIV, Computer Engineering and Software Engineering Department, Polytechnique Montreal

Submit to *Transportation Research Record: Journal of the Transportation Research Board*

Submitted: 1 April 2026, under review

Abstract

Accurately characterizing the crossing behavior of pedestrians with Assisting and Walking Devices (AWDs) is important for representing pedestrian heterogeneity, which is often not adequately captured in existing transportation analyses. To address this need, this study collected video data at an urban signalized intersection and conducted a video-based behavior analysis of five pedestrian categories: pedestrians with no AWD, and users of canes, walkers, strollers and grocery-carts. Three trajectory-based indicators were examined: average crossing speed, the ratio between the real traveled distance and the direct (straight line) distance (RD/DD), and the relationship between the average instantaneous speed and the number of pedestrians in the crossing zones. The results reveal behavioral differences across pedestrian categories for crossing speed, although the RD/DD ratios are similar across all categories, demonstrating comparable path straightness during crossing. The speeddensity analysis shows that the mean pedestrian speeds are not strongly affected by the number of pedestrians or the presence of pedestrians using AWDs, although the higher speeds decrease as pedestrian numbers increase. Overall, the findings highlight the importance of distinguishing pedestrians and their AWDs when assessing crossing behavior and point to the need for larger datasets to further validate behavioral differences among pedestrians with and without AWDs.

Keywords: Video analysis, Pedestrians, Crossing behavior, Mobility Aids, Walking Devices

7.1 Introduction

While walking is promoted among transport modes to improve the sustainability of transportation systems [176], pedestrians are non-protected road users that are very vulnerable to injury and death in collisions with motor vehicles. Pedestrians are most vulnerable when crossing the road, while roads constitute important physical barriers to convenient walking trips. It is thus important to understand how pedestrian cross roads in various environments and conditions to improve their safety and comfort. Pedestrian crossing behavior includes walking speed, path choice and adaptive response to traffic control devices like traffic lights. Among these behavioral aspects, walking speed plays a central role because it directly determines the crossing duration and the amount of time pedestrians are exposed to motorized traffic and the risk of injury.

There is considerable heterogeneity in capabilities and behaviors among pedestrians. In particular, pedestrians using Assisting and Walking Devices (AWDs), i.e., mobility aids like canes and walkers, and other devices like strollers and grocery carts, may exhibit distinct crossing patterns due to their mobility-related constraints or the manipulation of their devices. While walking speed and crossing behavior have been investigated with respect to attributes such as age and gender [177] and to a limited extent for pedestrians using mobility aids such as canes and walkers [178], research on pedestrians using AWDs and on a broader range of AWDs remains limited.

Recent advances in video-based trajectory extraction enable the collection of fine-grained spatial and temporal pedestrian movement data in real-world environments. Such data support detailed analyses of crossing behavior beyond aggregate measures, such as instantaneous speed and path deviation [179, 180], and allow examination of behavioral variations across different contextual conditions and pedestrian categories.

Despite growing interest in pedestrian behavior analysis, few studies have systematically compared crossing behavior among pedestrians with and without AWDs using video data. To address this gap, this study attempts to detect and track several pedestrian categories automatically and analyzes road user trajectories extracted from real-world video recordings. A detailed analysis of crossing behavior is conducted, focusing on average crossing speed, the ratio of the actual travelled distance over the direct (straight line) distance (RD/DD), and pedestrian speed-density diagrams. The findings aim to provide empirical evidence for inclusive pedestrian infrastructure design and safety evaluation. Our contributions are summarized as follows:

1. We find that deep learning-based object detection and tracking methods are not good

enough for pedestrians with AWDs and other devices; and

2. We compare the crossing behavior of pedestrians with and without AWDs, finding significant differences in average crossing speed, but not path deviation (RD/DD ratio).

The remainder of this paper is organized as follows. Section 7.2 presents the related work. Section 7.3 presents the 3-step methodology: data acquisition, pre-processing and analysis. Section 7.4 contains the results and the discussion, with the conclusion in Section 7.5.

7.2 Related work

Pedestrian crossing behavior has been studied from multiple perspectives, including data acquisition, trajectory extraction, and the analysis of pedestrian movements at intersections. Existing studies differ in terms of types of data and indicators applied for crossing behavior analysis. In line with the typical workflow of pedestrian behavior studies, this section first reviews data sources for crossing behavior studies, with an emphasis on video-based approaches for trajectory extraction. It is followed by a review of crossing behavior studies.

7.2.1 Data sources for crossing behavior studies

Pedestrian crossing behavior can be investigated using several types of data, including field observations [77, 177, 178], and instrumented measurements (e.g., radar, infrared, and video sensors) [181–183] that can be processed with varying levels of automation. Among these, cameras in the visible spectrum have become the most widely used source for pedestrian data collection, as they enable continuous naturalistic observation of pedestrians (without interfering with their natural behavior). Besides, video recordings provide detailed spatial-temporal information, generally as series of positions over time or trajectories, that allows researchers to examine how pedestrians cross intersections, adapt their speeds according to their surroundings and interact with other pedestrians and road users.

7.2.2 Video-based methods for pedestrian trajectory extraction

The increasing use of video data has been enabled the widespread use of computer vision methods for object detection and tracking in urban environments. Modern video-based road user trajectory extraction pipelines combine object detection, multi-object tracking (MOT), and trajectory reconstruction. A recent survey highlights the adoptions of deep learning-based methods for object detection and tracking [184, 185]. [184] reviewed various techniques for detecting vulnerable road users (VRU), including pedestrians, cyclists, and micro-mobility

users. They concluded that VRU detection performance is dominated by deep learning-based vision models, particularly convolutional neural networks (CNNs) and transformer architectures, which significantly outperform rule-based or background subtraction methods due to their superior robustness in complex urban scenes. These advances have enabled large-scale trajectory extraction for generic pedestrians and other vulnerable road users, including cyclists and micro-mobility users. However, they also highlight persistent challenges in real-world deployments, including sensitivity to occlusions, adverse weather, illumination changes, and the limited representation of micro-mobility users in existing benchmarks.

In addition to trajectory extraction, prior work has also examined the automatic estimation of pedestrian and micro-mobility user attributes from video data, such as pedestrian age and gender [186, 187]. These attributes are typically inferred at the detection or tracking stage and subsequently used to support more detailed behavior and safety analyses. However, while these visually observable characteristics have received some attention, to the best of our knowledge, pedestrians with AWDs have not been explicitly considered as a distinct pedestrian subcategory in existing video-based trajectory extraction studies.

A critical but often implicit assumption underlying these pipelines is the availability of representative training data. The performance of deep learning-based detection and tracking models strongly depends on the diversity and annotation quality of the dataset used for training. While existing benchmarks provide extensive coverage of pedestrians and other common object categories, they rarely distinguish pedestrians with AWDs and the AWD types. This limitation becomes particularly pronounced as these pedestrians often exhibit subtle visual differences from ordinary pedestrians and are subject to partial occlusions by their mobility aids or surrounding pedestrians. Only a few datasets explicitly include pedestrians with AWDs, e.g., PMMA [188], and most of them are either not publicly available [52], restricted to indoor environments [53], or designed only for object detection without any video [189].

Like other micro-mobility users, such as cyclists and users of e-scooters, pedestrians with AWDs have challenges related to detection and classification, particularly for categories that are underrepresented in training datasets. These challenges highlight a fundamental gap between current video-based trajectory extraction pipelines and the need for behavior analysis that explicitly accounts for pedestrian heterogeneity, motivating further investigation into both data and methodological limitations. However, existing video-based trajectory extraction studies generally treat pedestrians as a homogeneous category and do not explicitly distinguish sub-categories during detection and tracking. As a result, existing pipelines provide limited support for reliably identifying and extracting trajectories of pedestrians with AWDs.

7.2.3 Pedestrian crossing behavior analysis

Studies on pedestrian crossing speed

Several studies have investigated pedestrian crossing behavior using manual observation methods, such as stopwatches. For example, [178] conducted a field study in Winnipeg, Canada, where two observers measured pedestrian walking times using stopwatches and found that pedestrian walking speed decreased with age and varied across crossing locations. In São Paulo, Brazil, a cross-sectional study directly measured walking speed using a standardized walking test under controlled conditions among 1,191 older adults and reported that a substantial proportion of elderly pedestrians walked at speeds below commonly assumed design values [190]. In Québec, Canada, a naturalistic field observation study manually recorded the crossing behavior of 2,073 pedestrians at 135 signalized intersections [191]. The study found that hesitation before stepping off the curb, longer crossing durations, and the use of walking aids were associated with greater crossing difficulty, particularly a higher likelihood of not finishing the crossing within the allotted pedestrian signal time. In Chongqing, China, [192] observed 658 elderly pedestrians and 1,176 adults at three signalized crosswalks, two at intersections and the last at mid-block, and found that significant differences in crossing speed and gap acceptance behavior between elderly pedestrians and younger adults. In Belgrade, Serbia, elderly pedestrians were timed while crossing different types of intersections, and questionnaires were used to collect additional information, leading to the conclusion that intersection geometry and traffic conditions significantly influenced the crossing speed and perceived safety of elderly pedestrians [193].

A few studies rely on video-based crossing speed computation methods. Early work by Ismail et al. The feasibility of automatically extracting pedestrian trajectories from video data and deriving basic kinematic measures were demonstrated, such as walking speed, in urban environments [181]. Pedestrian behavior using fixed video recordings at an urban crossing was assessed and a computer vision-based pedestrian detection and tracking framework was applied, in which pedestrians were identified in video frames, tracked across time, and pedestrian movements were reconstructed into trajectories through camera calibration [194]. The study concluded that video-based trajectory analysis enables non-intrusive and quantitative assessment of pedestrian crossing behavior under real traffic conditions. A video-based method that uses unmanned aerial vehicle (UAV) footage to extract pedestrian crossing speeds at the street scale, that is over several roadway segments rather than at a single crossing point, was proposed in [195]. In this approach, pedestrians are detected and tracked using YOLOv3 [23] and DeepSORT [38], and their speeds relative to the ground are calculated through a combination of SIFT-RANSAC [16, 196] and geometric correction

algorithms. The results demonstrated that UAV-based video analysis can capture pedestrian walking speeds over large spatial areas, supporting scalable pedestrian behavior measurement in urban environments. A CycleGAN-based [145] loop-learning framework was proposed to enhance pedestrian detection performance under various environmental conditions, such as daytime, nighttime, and rainy scenes [197]. The model iteratively generates and labels synthetic images to augment limited real datasets. Their results indicate that synthetic-to-real domain adaptation can effectively improve detection robustness under challenging visual conditions, particularly when annotated real-world data are limited. Pedestrian gap acceptance and crossing speed at mid-block crossings using video-based image processing was studied in Izmir, Turkey [73]. Trajectories of 498 pedestrians were extracted from synchronized multi-camera recordings, and pedestrian as well as item detection, where items refer to whether pedestrians were carrying objects during crossing, were performed using YOLOv3 [23] and YOLACT [198] models. The study concluded that pedestrian crossing speed is strongly influenced by traffic conditions and pedestrian characteristics, such as gender and group size, and that vision-based trajectory extraction enables quantitative analysis of pedestrian crossing behavior in real traffic environments.

Although numerous studies have investigated pedestrian behavior, existing work has primarily examined crossing speed differences with respect to age, location, or traffic conditions. Some studies have conducted comparative analyses of crossing-related behaviors between pedestrians and other road users, such as cyclists and micro-mobility users including e-bike and e-scooters using trajectories extracted from video data [199,200]. However, comparatively little attention has been given to differences in crossing speeds among heterogeneous pedestrian categories themselves, particularly pedestrians with AWDs in real-world intersections remains insufficiently understood.

Speed-density fundamental diagram

Understanding how pedestrian speed changes with surrounding density has been a central topic in pedestrian flow studies and facility design. In an early empirical work for example, multiple speed-density models in [201] were evaluated and, similarly to other modes, mean pedestrian walking speed was found to decrease as density increases. Later studies refined these relationships with laboratory experiments. It was demonstrated in [202] that unidirectional flows and bidirectional flows yield different speed-density patterns, with bidirectional streams may lead to a tendency toward a plateau in speed at certain density ranges due to the interplay between head-on conflicts and lane formation.

Previous studies have consistently shown that pedestrian walking speed is strongly related

to local crowd density. Empirical and probabilistic analyses report that higher densities lead to lower mean speeds and increased variability in individual walking behavior [203, 204]. These observations reflect the transition from free-flow to constrained and congested movement regimes as pedestrian interactions intensify. Comprehensive reviews further confirm that walking speed is influenced by density as well as individual characteristics such as age, physical ability, and carried load [205]. Therefore, crossing speed is not only an indicator of individual walking performance but also reflects the interaction conditions within the crossing environment.

Despite extensive research on speeddensity relationships, existing studies overwhelmingly do not distinguish among pedestrians. Little is known about how pedestrians with AWDs adapt their speed under varying levels of pedestrian density and how they may impact pedestrian traffic, particularly in real-world intersections. The limitation underscores the need for detailed, trajectory-based analyses that incorporate density effects for diverse pedestrian categories at signalized intersections. To address this gap, this study compares crossing speeds derived from trajectory data across different pedestrian categories, including pedestrians with and without AWDs.

7.3 Methodology

This video-based crossing behavior study consists of the following steps, illustrated in Fig. 7.1:

1. Video acquisition;
2. Trajectory generation; and
3. Crossing behavior analysis.

7.3.1 Video acquisition

The video data was recorded using a GoPro camera mounted on a fixed pole adjacent to the intersection of Chemin de la Côte-des-Neiges and Avenue Van Horne in Montréal, Canada. An illustration of this intersection and its surrounding is shown in Fig. 7.2. The nearby surrounding includes four bus stops that result in high pedestrian volume and a diverse population, including pedestrians with and without AWDs in the camera field of view.

The dataset was recorded using a fixed camera between 10:00 and 15:00, with a resolution of 1920 $\times$ 1080 pixels and a frame rate of 30 frames per second (fps). A complete 17-min video file from 10:34 to 10:51 is used for the trajectory extraction of pedestrians without

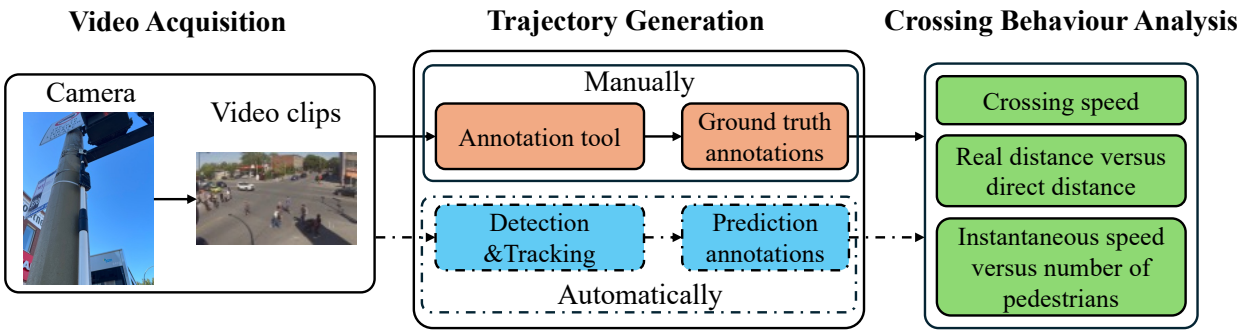


Figure 7.1 Overview of the steps for pedestrian trajectory extraction and crossing behavior analysis. The dashed arrows and frames indicate that fully automatic detection and tracking were attempted, but not used for the crossing behavior analysis, as they were not sufficiently accurate.

AWDs. The whole video dataset is used for generating video clips containing pedestrians with AWDs. Each video clip ranges from 18 to 30 s in duration, and a total of 74 clips were extracted. The crossing distances d_{cross}, as measured from Google Maps, are 14.0 m in the westeast direction and 16.5 m in the southnorth direction.

7.3.2 Trajectory generation

Trajectories can be obtained either through manual or semi-automated annotation using dedicated tools (e.g., CVAT [87]), which produce ground-truth trajectories, or through automated detection and tracking pipelines, e.g. using a detector like YOLOv11 [27] and a tracker like BoT-SoRT [40], as illustrated by the dashed frames in Fig. 7.1. The publicly available YOLOv11 model pre-trained on the COCO dataset and BoT-SORT with its default configuration integrated with YOLOv11 are used for semi-automated annotation.

In this step, both trajectory generation methods were considered in this study. The initial objective was to generate pedestrian trajectories automatically from video data. A detection and tracking pipeline based on a YOLOv11 model fine-tuned with pedestrians with AWDs categories and BoT-SORT was implemented and evaluated. YOLOv11 was trained on the annotated video clips for 30 epochs, using weights pretrained on COCO, with a batch size of 4 and a learning rate of $1e^{-4}$. The dataset was divided at the video-clip level: half of the clips were used for the training and validation sets and the remaining clips for the test set so that no temporally adjacent frames from the training and validation sets appeared in the test set. Frames for the training and validation sets were sampled at 30 fps, while the test set was sampled at 2 fps to reduce redundancy. This resulted in 20,444 training images and 5,111 validation images and 1,882 test images. The mean average precision (mAP) is used to

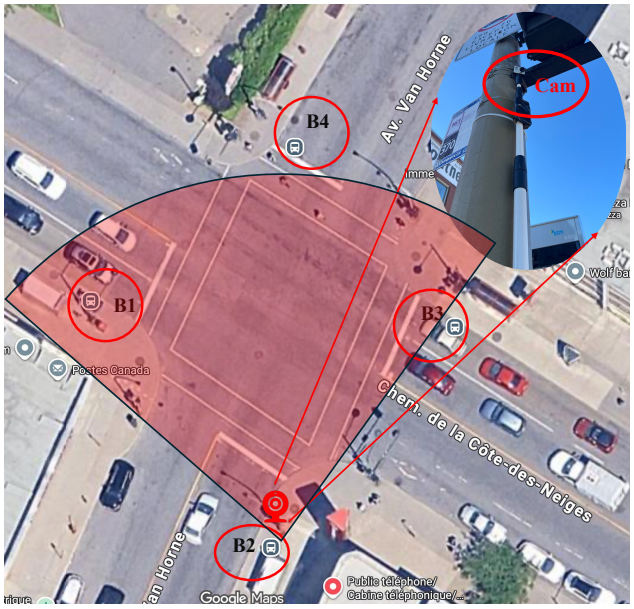


Figure 7.2 Illustration of the intersection and its surroundings, showing the GoPro camera installation point (marked as Cam), bus stops (marked as B1, B2, B3, and B4), and the approximate camera field of view, represented by the red sector-shaped area. This illustration was created using Google Maps.

evaluate the detection performance. It corresponds to the mean average precision at various overlap level (50% to 95%) between the detected bounding boxes and ground-truth bounding boxes. For example, average precision at 50% (AP_{50}) considers a detection as correct if it has a 50% overlap with the corresponding ground-truth, while an average precision at 95% (AP_{95}) would require a 95% overlap that is more difficult to achieve.

7.3.3 Crossing behavior analysis

Computation of average crossing speed

Fig. 7.3 illustrates the crossing and waiting zones defined within the camera field of view used to determine the entry and exit instants for the computation of crossing speeds. The crossing zones extend beyond the physical crosswalk boundaries to account for localization noise, perspective distortion, and natural pedestrian behavior (e.g., entering slightly before or leaving slightly after the marked crosswalk). The average crossing speed v_{cross} is obtained by dividing the known crosswalk width d_{cross} by the crossing duration t_{cross} measured in the video sequence. In each frame, the pedestrian location is represented by the midpoint of the bottom edge of the detected bounding box, which approximates the pedestrian's ground contact point and is used to construct the trajectory. A valid crossing event occurs when

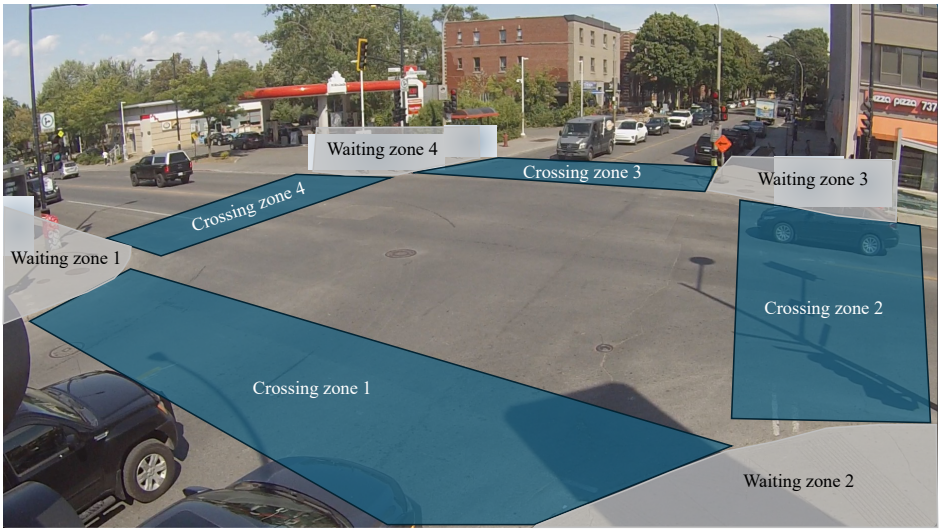


Figure 7.3 Illustration of the crossing zones. Pedestrians are hidden using white rectangles to respect their privacy.

a pedestrian trajectory intersects the two opposite boundary segments of the polygon of the crossing zone. The entry and exit times are determined by the intersection between the trajectory segments and the corresponding polygon boundary, with the crossing instant linearly interpolated between consecutive frames. Trajectories are excluded if the pedestrian first appears inside the crosswalk polygon without intersecting a boundary segment, exits through lateral boundaries, or the trajectory is truncated by occlusion or tracking loss before intersecting the opposite boundary.

Computation of the RD/DD Ratio, instantaneous speed and number of pedestrians

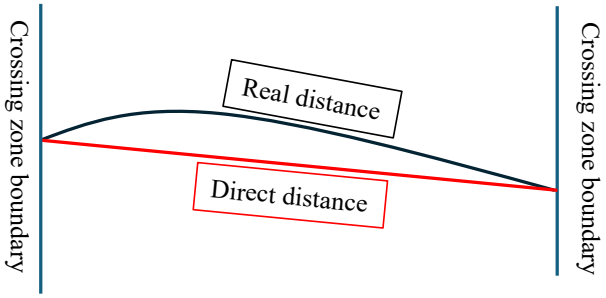


Figure 7.4 Illustration of real distance and direct distance.

Fig. 7.4 illustrates the definition of the real distance RD and the direct distance DD. The real distance corresponds to the length of the pedestrian trajectory between the entry and

exit points, whereas the direct distance is the straight-line distance between the entry and exit points. The RD/DD ratio represents the degree of deviation of the pedestrian trajectory from a straight crossing path.

The distances and instantaneous speeds are computed in the real-world coordinate system. Only crossing zones 1 and 2, which are located closer to the camera and experience minimal perspective effects, are considered. For each crossing zone, a specific perspective matrix is computed to transform trajectories from pixel coordinates to real-world coordinates. The mean of the instantaneous speeds of all the pedestrians in a given zone at a given frame is defined as

$$\bar{v}_{\text{inst}}(t) = \frac{1}{n(t)} \sum_{i=1}^{n(t)} v_i(t), \tag{7.1}$$

where $n(t)$ is the number of pedestrians in crossing zone CZ, with $CZ \in \{1, 2, 3, 4\}$, at frame t and $v_i(t)$ is the instantaneous speed of the i-th pedestrian in CZ at t. The plot of $\bar{v}_{\text{inst}}(t)$ as a function of $n(t)$ every 15 frames constitutes the pedestrian fundamental diagram. This is analyzed separately per crossing zone since their size is different.

7.4 Results and discussion

7.4.1 Object detection performance

Object detection performance is evaluated by comparing the bounding boxes of ground-truth objects with those that are detected by YOLOv11 using intersection over union (IoU) to calculate the overlap. In Tab. 8.2, we report the average precision at IoU thresholds of 50% (AP_{50}), and 75% (AP_{75}), and mean average precision from 50% to 95% IoU ($mAP_{0.5:0.95}$) for five categories: pedestrians without AWDs, cane users, walker users, stroller pushers and grocery-cart users. Among the four categories of pedestrians with AWDs, stroller pushers have the highest mAP owing to their more distinctive appearance. The mAP values for cane, walker and grocery-cart users are all under 10 % primarily due to their visual similarity to the pedestrians without AWDs and frequent occlusions.

The confusion matrix shown in Fig. 7.5 provides further insight into the relatively low mAP values. All four categories of pedestrians with AWDs are frequently misclassified as pedestrians without AWDs. Among them, stroller pushers have the highest recall, indicating that the model correctly identified a larger proportion of true stroller pushers than for the other categories.

Because the detection performance remains insufficient, ground-truth annotations are used

Table 7.1 Object detection performance of YOLOv11 on the test set measured by $mAP_{0.5:0.95}$, AP_{50} and AP_{75} (%).

Category	$mAP_{0.5:0.95}$	AP_{50}	AP_{75}
Pedestrian without AWDs	61.0	80.3	72.3
Cane	4.4	8.1	4.0
Walker	6.8	8.4	8.4
Stroller	30.5	43.3	37.9
Grocery cart	8.5	12.0	10.5

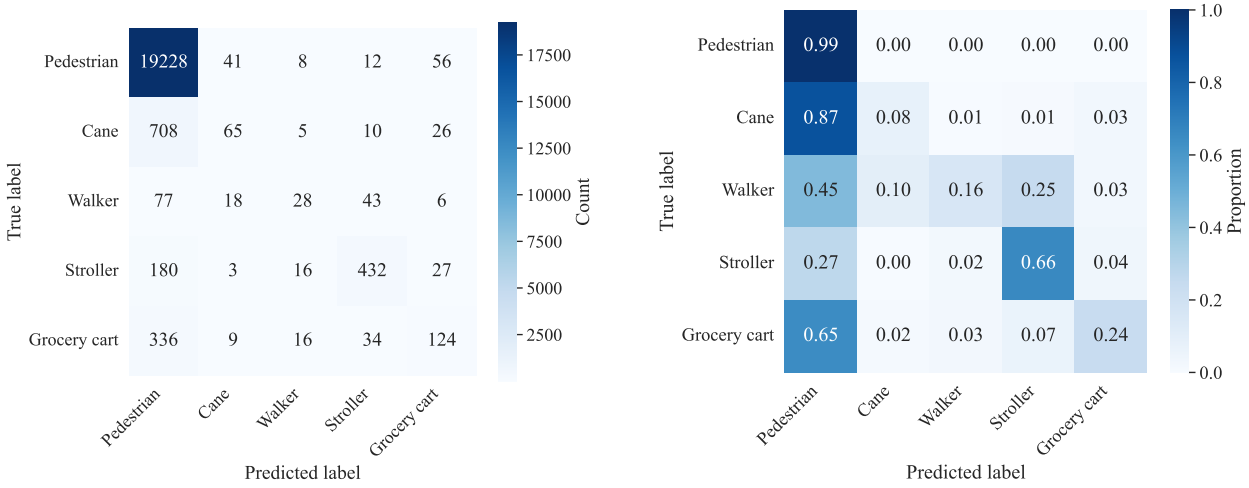


Figure 7.5 Confusion matrix of YOLOv11 for pedestrian categories. The left figure presents the confusion matrix as raw counts, while the right shows the row-wise normalized matrix; in both cases, rows correspond to ground-truth labels and columns to predicted labels. Pedestrian refers to the pedestrians without AWDs.

to compute pedestrian crossing speeds, the RD/DD ratio, and the speed-density fundamental diagram in crossing zones.

7.4.2 Crossing behavior analysis

Average crossing speed

Fig. 7.6 presents the distributions of average crossing speeds across pedestrian categories. A KruskalWallis test indicates a significant overall difference among groups ($H = 54.03$, $p < 0.001$), confirming that crossing speeds vary across pedestrian types. Pairwise Dunn tests further show that cane users ($p < 0.001; median = 0.84\ m/s$) and grocery-cart users ($p < 0.001; median = 1.04\ m/s$) walk significantly more slowly than pedestrians without AWDs (median=1.22 m/s), whereas stroller pushers ($p = 0.74; median = 1.22\ m/s$) and

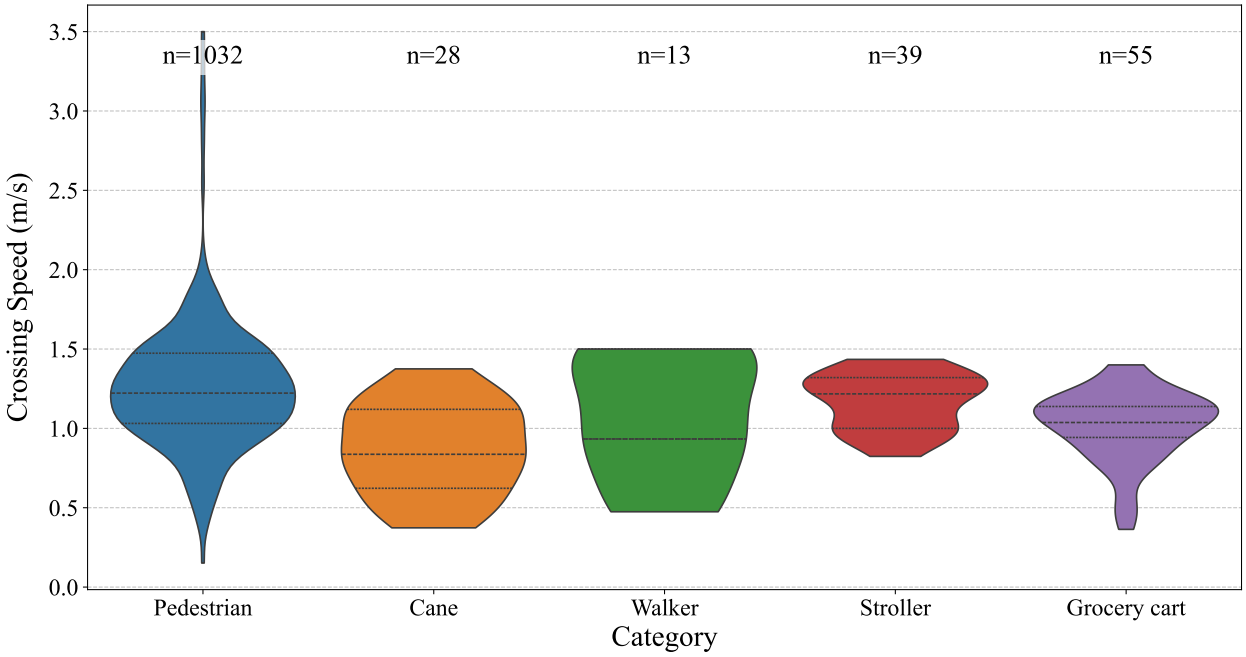


Figure 7.6 Average crossing speed distributions across the pedestrian categories (zones 1 to 4). The numbers over each violin indicate the number of samples in each violin plot.

walker users ($p = 0.74; median = 0.93\ m/s$) do not exhibit statistically significant differences. The lack of significance for walker users despite the much lower median may be attributed to the small sample size. On the other hand, the average crossing speed of stroller pushers is very close to that of pedestrians without AWDs, likely reflecting the younger age profile of individuals pushing strollers (e.g., caregivers with infants).

Pedestrians without AWDs display moderate variability (IQR=0.44 m/s), with the highest observed value (about 3 m/s) corresponding to a running pedestrian. In comparison, pedestrians with AWDs generally show greater dispersion, with IQR=0.50 m/s for cane users and IQR=0.57 m/s for walker users, indicating a wider spread of crossing speed within these groups. Pedestrians using wheeled devices demonstrate lower dispersion: grocery-cart users exhibit a relatively narrow distribution (IQR = 0.20 m/s), indicating more consistent walking behavior, while stroller pushers show moderate variability (IQR = 0.32 m/s). Overall, these results show that mobility-related assistive devices are associated with greater variability in walking speeds, whereas other wheeled devices tend to promote more consistent movement patterns.

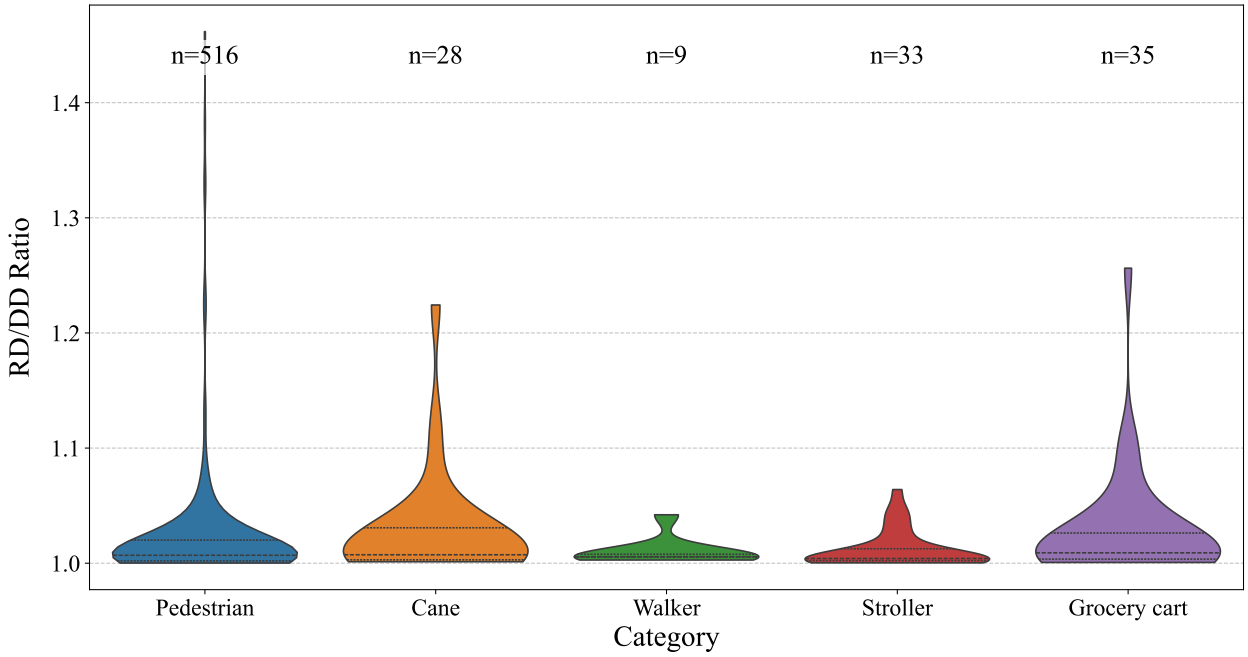


Figure 7.7 RD/DD ratio distributions across the pedestrian categories (zones 1 and 2).

RD/DD ratios

Fig. 7.7 presents the distributions of the RD/DD ratios across the five pedestrian categories. Overall, the results show that this metric remains highly consistent regardless of pedestrian type, as shown by the H-statistic of 4.90 and p-value of 0.297 for the KruskalWallis test. All categories exhibit median RD/DD ratios slightly above 1.00, with values ranging only from 1.004 to 1.009 across the categories. This indicates that the relative difference between the RD and DD is generally minimal, suggesting a stable and category-invariant crossing behavior. This small deviation largely reflects that most pedestrians traverse the crossing zones along nearly straight paths. The variability of the RD/DD ratio further reinforces this observation. All categories demonstrate very small IQRs, spanning approximately 0.002 to 0.028. Walker users show the smallest variability (IQR = 0.002), reflecting highly stable alignment between RD and DD estimates, followed by people pushing strollers. Cane and grocery-cart users exhibit larger IQRs, but still within a very narrow range.

Speed-density fundamental diagram

The mean walking speeds of all pedestrians is compared in two conditions: (i) No AWD, where no pedestrian with AWD is present, and (ii) Mixed, where at least one pedestrian with AWD is present. As for RD/DD ratios, the diagrams are plotted only for zones 1 and 2.

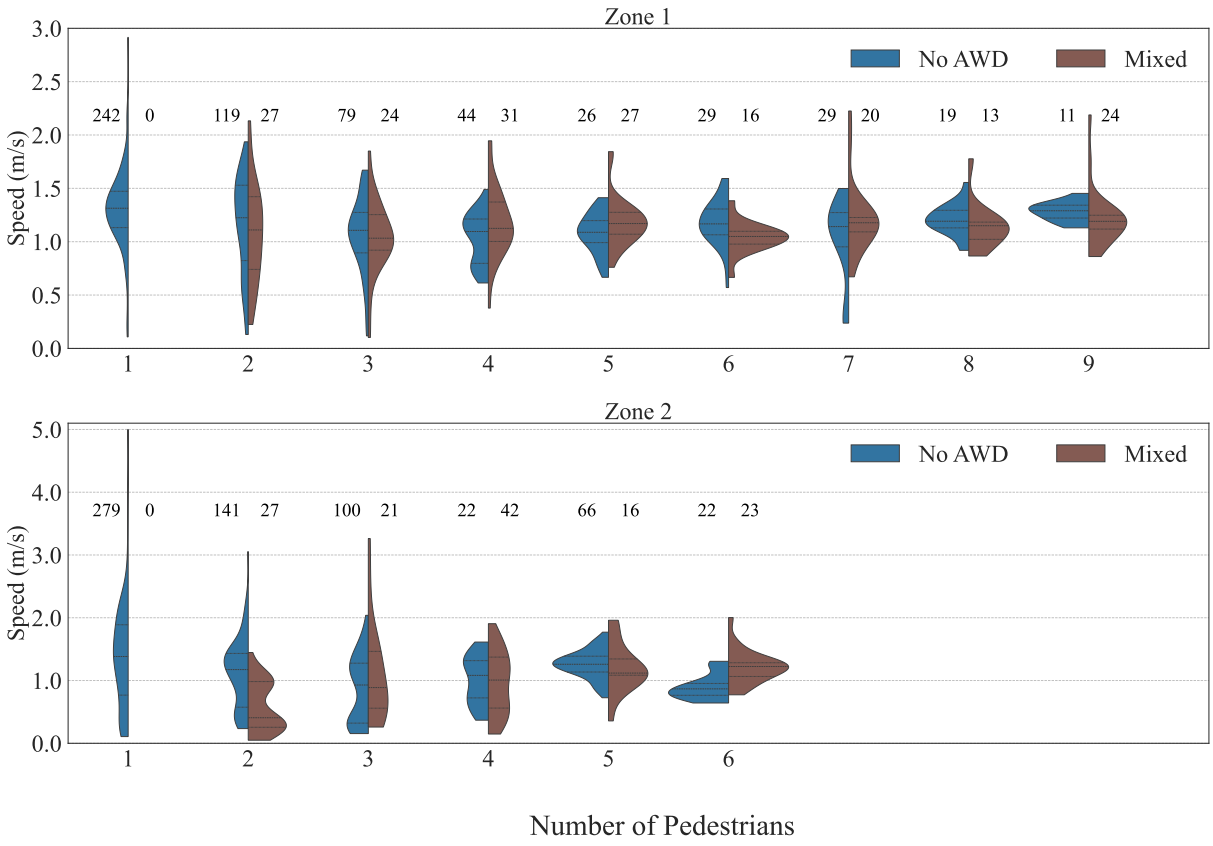


Figure 7.8 Mean instantaneous speed versus the number of pedestrians in zones 1 and 2. No AWD and Mixed represent instants when only pedestrians without AWDs are moving in a crossing zone, and when at least one pedestrian with AWD is present, respectively. The number above each pair of violins indicate the number of samples in each violin plot.

Fig. 7.8 presents the relationship between mean instantaneous walking speed and the number of pedestrians, while Fig. 7.9 shows the corresponding median and 90th-centile of the mean instantaneous walking speeds derived from the same observations.

In crossing zone 1, the mean walking speeds remain broadly stable as the number of pedestrians increases under both conditions. The median speeds show close values between the two conditions, indicating that walking speed is only weakly influenced by the presence of pedestrians with AWDs. The 90th-percentile speeds decrease with increasing pedestrian numbers for both conditions, demonstrating that higher pedestrian densities mainly reduce the speeds of faster walkers while the typical walking speed remains largely unchanged, which is expected and has been reported in previous studies [201–203].

In crossing zone 2, similar trends are observed in the absence of pedestrians with AWDs ("No AWD"), i.e., stable medians independently of pedestrian density and decreasing 90-th centile

with increasing density. In the presence of at least a pedestrian with AWD ("Mixed"), both the median and 90-th centile show increasing trends as the number of pedestrians rises. This is mostly caused by the specific sample of instantaneous speeds for two pedestrians which were observed at the beginning of the signal time interval when pedestrians start crossing. The resulting lower median speed creates the impression of increasing speeds between two and five pedestrians.

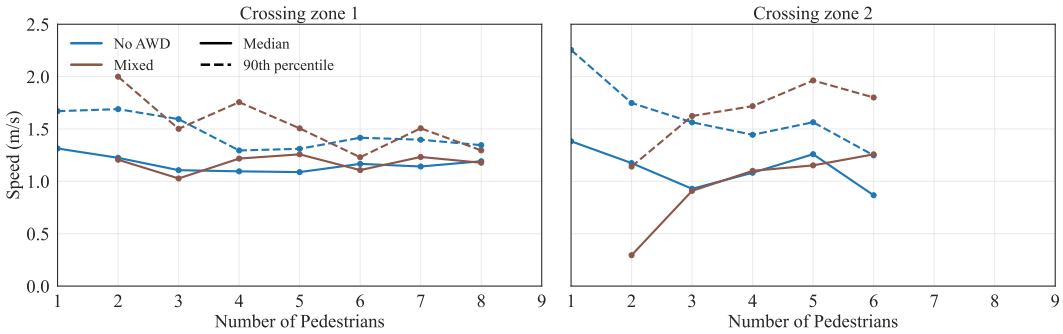


Figure 7.9 Median and 90 % centiles of the mean instantaneous speeds versus the number of pedestrians in crossing zones 1 and 2.

7.5 Conclusion

This paper investigates pedestrian crossing behavior using video-based analysis, with a particular focus on pedestrians with and without AWDs. Three trajectory-based indicators were examined: average crossing speed, the RD/DD ratio representing path deviation, and the relationship between mean instantaneous speed and the number of pedestrians in a crossing zone.

The results indicate that pedestrians using mobility-related AWDs generally move more slowly than pedestrians without AWDs, while also exhibiting greater variability in their crossing speeds. In contrast, the RD/DD ratio remains highly consistent across all pedestrian categories, indicating that crossing trajectories are largely similar regardless of AWD usage. The speeddensity analysis further shows that pedestrian speeds remain relatively stable under most density conditions, although higher pedestrian densities tend to constrain faster walkers. Overall, these findings demonstrate that the presence of mobility aids primarily affects walking pace rather than trajectory during road crossing, with limited influence of AWD users over other pedestrians.

These results highlight the importance of explicitly distinguishing pedestrians with AWDs in pedestrian behavior studies. Treating pedestrians as a single category overlooks important

behavioral differences associated with mobility limitations and AWD use.

The main limitation of this study is that the amount of data available is not large enough to support an extensive comparison between pedestrians with and without AWDs across all conditions. In addition, the data was collected on a single day, and potential influences of seasonal variation and weather conditions were not captured. Moreover, the study did not consider traffic signal timing, which may significantly influence pedestrian crossing behavior. Future work will focus on collecting more data involving pedestrians with AWDs. Additional factors, such as seasonal variation, weather condition, traffic signal timing, will also be considered in the following studies.

Acknowledgment

The authors gratefully acknowledge the financial support of NSERC through the Discovery Grant 2023-05626. The authors would like to thank the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for the computing servers, and NVIDIA Corporation for their donation of GPUs. This research was also enabled in part by support provided by the Digital Research Alliance of Canada, using CPU and GPU resources on rorqual clusters.

Author Contributions

Qingwu Liu, Nicolas Saunier, and Guillaume-Alexandre Bilodeau jointly conceptualized the research idea, the preparation of the data collection. Nicolas Saunier provided instructions on the use of data collection equipment and assisted Qingwu Liu in learning data collection procedures. Qingwu Liu collected the video data, implemented the algorithms, conducted data preprocessing and experiments, and wrote the initial draft of the manuscript. Nicolas Saunier and Guillaume-Alexandre Bilodeau participated in the discussion and analysis of the experimental results, and reviewed and revised the manuscript. All authors read and approved the final manuscript.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

Funding

Financial support by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant 2023-05626.

Data availability statements

The video data collection for this study was approved by the ethics committee (Comité d'éthique à la recherche avec des êtres humains) of Polytechnique Montréal and the authorities of the city of Montréal (Ville de Montréal). As it is not desirable for a naturalistic study to obtain consent from all pedestrians appearing in the footage, the data cannot be publicly released as an open dataset.

CHAPTER 8 GENERAL DISCUSSION

In this section, various topics related to vision-based object detection and tracking methods applied to transportation applications are discussed, including comparisons between vehicle-mounted and roadside datasets, and pedestrians with AWDs in both the collected dataset in Chapter 6 and real-world data used for the case study in Chapter 7. To clarify how the research objectives were addressed, Table 8.1 summarizes the correspondence between the research objectives, the articles included in the thesis, the datasets used, the tested methods, and the main evaluation focus.

8.1 Video data collected from vehicle-mounted and roadside cameras

The video data collected from vehicle-mounted and roadside cameras present fundamentally different characteristics, which have important implications for detection, tracking, and trajectory-based transportation applications.

In vehicle-mounted datasets, the cameras are attached to a moving vehicle, making them suitable for analyzing local interactions and safety around the ego-vehicle. However, the motion of the ego-vehicle leads to continuously changing viewpoints. As a result, the apparent size and position of surrounding road users vary not only with their distance to the ego-vehicle but also with the ego-vehicle's motion. In contrast, roadside datasets are captured from fixed cameras, where the scene remains stationary, the distance between the camera and objects is usually larger than that in vehicle-mounted datasets. Roadside datasets enable the analysis of overall traffic conditions within a fixed area, although limited to that location.

Moreover, trajectory estimation in vehicle-mounted datasets is further affected by the accuracy of ego-localization systems, (e.g., GPS and IMU), as errors in these measurements will propagate through coordinate transformations and introduce uncertainty into the reconstructed trajectories. In addition, the low camera viewpoint and the vehicle's positioning relative to the scene further increase sensitivity to projection errors.

Besides, the visual appearance of road users differs significantly between the two types of datasets due to camera placement and viewpoint. In general, in vehicle-mounted cameras settings, the low viewpoint places the camera close to the scene, resulting in larger apparent object sizes and richer visual details.

Table 8.1 Summary of the research objectives, corresponding articles, datasets, tested methods, and evaluation focus.

Research objective	**Article**	**Dataset**	**Methods tested**	**Evaluation focus**
Objective 1: To evaluate how differences between predicted and ground-truth trajectories extracted by detection and tracking methods from vehicle-mounted or roadside video data affect safety analysis	Article 1	KITTI	MonoDTR + DeepSORT; YOLOStereo3D + DeepSORT; CenterTrack	Trajectory quality and its impact on TTC computation
	Article 2	LUMPI	CenterNet2D, YOLOv8, and CenterNet3D with ByteTrack	Interaction-level matching and surrogate safety indicators
Objective 2: To evaluate the performance of recent object detection and tracking methods on pedestrians with AWDs	Article 3	PMMA	YOLOv8/YOLOv11 and ByteTrack	Detection and tracking performance for pedestrians with and without AWDs
Objective 3: To conduct a video-based case study at an intersection to evaluate the ability of detection methods to identify pedestrians with AWDs, and to analyze and compare their crossing behaviors with those of pedestrians without AWDs	Article 4	Real-world intersection dataset	YOLOv11 and BoT-SORT	Pedestrian crossing behavior analysis, focusing on pedestrians with AWDs

In contrast, roadside cameras are typically installed at higher positions, providing a more global view of the scene and capturing more complete objects shapes. This higher viewpoint reduces occlusions and allows more road users to be observed simultaneously in roadside datasets. However, objects appear smaller and may lack fine-grained visual details. Moreover, occlusions can still occur in group situations, where interactions between closely spaced road users lead to partial overlaps, although they are generally less severe than in low-viewpoint settings.

In a summary, both types of datasets are affected by occlusions. In vehicle-mounted datasets, trajectory extraction accuracy is additionally influenced by the performance of ego-localization systems, which is an issue that does not arise in roadside datasets. As a result, trajectory extraction is generally more sensitive to such error sources in vehicle-mounted dataset than in roadside datasets.

8.2 3D versus 2D perception methods

3D-based perception methods, including stereo vision and learning-based monocular depth estimation, operate directly in real-world coordinate systems, therefore can provide improved geometric consistency, particularly in representing object size and spatial relationships. However, their performance is sensitive to the accuracy of depth estimation, which may degrade in some complex scenes or under challenging conditions.

In contrast, 2D perception methods rely on monocular images combined with camera calibration and geometric transformations to project observations onto the ground plane. While these approaches are simpler to deploy and widely applicable in roadside settings, they depend on assumptions such as planar ground surfaces and are sensitive to calibration errors, which can affect the accuracy of the reconstructed trajectories. When the road surface is uneven or contains multiple elevation changes, the scene often needs to be segmented into several sub-planes, each with its own perspective transformation. For example, in Chapter 7, multiple perspective transformations are used to maintain acceptable accuracy.

Overall, 2D projection offers a more practical and cost-effective solution but is sensitive to calibration and environment assumptions, whereas 3D projection provides more accurate spatial representation at the expense of higher system complexity and data requirements.

8.3 Comparisons of pedestrians with AWDs in the PMMA and case study dataset

There are substantial differences between the PMMA dataset in Chapter 6 and the real-world case study dataset in Chapter 7 in terms of the representation of pedestrians with AWDs. In PMMA, the proportion of pedestrians with AWDs is relatively high and remains fixed across the dataset. In addition, the types and appearances of AWDs are limited and exhibited little variation, due to the constrained number of volunteers and the limited variability of AWD equipment. Furthermore, the dataset is constructed in a controlled environment, where only pedestrians with and without AWDs are present, and no other types of road users are included. Consequently, it does not reflect the complexity of real traffic scenarios.

In contrast, the case study dataset is collected in a real-world intersection, where pedestrians with AWDs are observed alongside other vulnerable road users and vehicles such as cars, buses, and trucks. The overall proportion of pedestrians with AWDs is much lower, and their appearances are more diverse, with variations in the type of AWDs and individual characteristics. Moreover, the occurrence of pedestrians with AWDs is not consistent over time, whereas in the PMMA dataset such users appear in every frame.

Table 8.2 Object detection performance of YOLOv11 on the test set measured by $mAP_{0.5:0.95}$ under different percentages of the PMMA dataset (Chapter 7). The percentage categories indicate the proportion of the PMMA dataset incorporated into the training set.

Category	**0%**	**10%**	**20%**	**100%**
Pedestrian without AWDs	**61.4**	47.1	<u>47.5</u>	*47.6*
Cane	**4.4**	<u>0.2</u>	*0.9*	*0.9*
Walker	**6.8**	<u>0.2</u>	<u>0.2</u>	*0.3*
Stroller	*30.5*	<u>29.0</u>	26.8	**32.2**
Grocery cart	**8.5**	2.7	*4.6*	<u>3.7</u>

Notes: Bold, *italic*, and <u>underline</u> indicate the first, second, and third best performances, respectively.

These differences have direct implications for deep learning-based model transferability. Although the PMMA dataset can provide useful samples for training, its lack of appearance variability and fixed background limit its effectiveness when applied to real-world scenarios. In the case study of Chapter 7, the PMMA dataset was used to fine-tune YOLOv11 by proportionally augmenting the collected intersection dataset. The training data consisted of the intersection dataset combined with varying proportions of the PMMA dataset. The results in Table. 8.2 show that incorporating the PMMA dataset does not improve the detection performance for both pedestrians with and without AWDs. When only comparing the configurations training with PMMA, the performance generally exhibits a slight increasing trend as the proportion of the PMMA dataset increases. However, when compared to the baseline without PMMA, all configurations with PMMA consistently achieve lower performance, indicating that the inclusion of PMMA does not enhance detection performance. In addition, these trends are not strictly monotonic. An exception is observed as the pedestrians with grocery cart achieves higher performance at 20 % than at 100 %, demonstrating that the inclusion of PMMA does not necessarily lead to further improvements. These findings indicate that the PMMA benchmark may have limited relevance to real-world scenarios, as improvements observed on the PMMA dataset do not translate into better detection performance when applied to the case study.

As a result, models fine-tuned on the PMMA dataset exhibit limited generalization when

applied to real-world scenarios, indicating that the controlled nature and limited diversity of the PMMA dataset restrict its ability to capture the variability of real-world conditions, thereby limiting its effectiveness for robust deployment in transportation applications.

CHAPTER 9 CONCLUSION

This final chapter presents a synthesis of the work carried out in this doctoral research. It discusses the limitations of the proposed approaches. Recommendations and directions for future research are presented in the final part of the chapter.

9.1 Summary of works

This research investigates the applicability of vision-based object detection and tracking methods for transportation applications analysis. The main objective of the dissertation is to assess whether object detection and tracking methods are good enough for the analysis of transportation applications, including safety analysis and crossing behavior analysis. To address this objective, the research is structured into three main components: 1) a systematic comparison between detection and tracking methods and ground-truth trajectories for safety analysis on vehicle-mounted and roadside dataset, 2) the development and evaluation of a dedicated dataset focusing on pedestrians with AWDs, and 3) an in-depth case study on video-based crossing behavior analysis in a real-world scenario.

The first component of this research focused on evaluating the safety indicators predicted by object detection and tracking methods, comparing with those computed from ground truth within a unified analysis framework. A complete processing pipeline was established, covering video input, object detection and tracking, trajectory generation, projection to real-world coordinates, and the computation of safety indicators. This framework ensured methodological consistency across all experiments and enabled a quantitative analysis of the impact of perception errors on downstream safety metrics. Experiments were conducted using both vehicle-mounted and roadside video data, several object detection and methods were applied, and safety indicators, including TTC and PET were computed based on both predicted and ground-truth trajectories. The results demonstrated that even when detection and tracking methods achieve reasonable performance according to conventional perception metrics, substantial discrepancies could still arise in the derived safety indicators. These findings indicate that standard detection and tracking benchmarks alone are insufficient to assess the suitability of perception methods for safety analysis applications.

The second component addressed a critical gap in existing video-based transportation research by focusing on pedestrians with AWDs, which are largely under-represented in current datasets. A dedicated dataset for pedestrians with AWDs, named as PMMA, was developed

to capture a wide range of pedestrian categories and movement characteristics. Using the proposed PMMA dataset, seven object detectors and three MOT methods were systematically evaluated across pedestrian with AWDs categories. The results showed that some detectors achieved the best overall detection performance. Moreover, no single detection and tracking method consistently performed well across all categories, highlighting the challenges in achieving robust perception for diverse pedestrian types.

The third component of the dissertation consisted of a detailed case study on video-based crossing behavior analysis at a signalized intersection. Using roadside video data, crossing behavior indicators, including crossing speed, the ratio of road distance to direct distance (RD/DD), and the speed-density fundamental diagram, were analyzed for different pedestrian categories. By comparing pedestrians with and without AWDs, the case study provided empirical insights into behavioral differences and adaptation strategies under varying traffic conditions. The results showed that pedestrians with AWDs generally exhibited lower crossing speeds and distinct movement patterns, illustrating the need for analyzing heterogeneous pedestrian groups when interpreting video-based behavioral data.

Overall, the works presented in this dissertation provide a systematic and empirical analysis of the suitability of object detection and tracking methods for video-based road safety and crossing behavior analysis. By combining a unified evaluation framework, multiple datasets, and a real-world case study, this research demonstrates that detection and tracking methods cannot be treated as neutral preprocessing steps. Instead, their limitations and population-dependent performance must be explicitly considered when interpreting safety and behavioral analysis results, particularly for pedestrians with AWDs.

9.2 Limitations

The work presented in this dissertation is subject to several limitations, which can be broadly classified into two main categories. The first category relates to limitations associated with the data and experimental settings adopted in this research. The second category concerns limitations arising from the use of existing object detection and tracking methods and their integration into video-based safety and behavioral analysis pipelines.

The first group of limitations is related to the datasets and traffic scenarios considered in this dissertation. Although multiple datasets were employed, including vehicle-mounted datasets, roadside intersection videos, and the proposed PMMA dataset, the data remain restricted to a limited number of locations, viewpoints, and traffic conditions. Consequently, certain factors commonly encountered in real-world traffic environments, such as extreme weather

conditions, or highly heterogeneous urban layouts, were not explicitly addressed. In addition, the PMMA dataset was collected with the participation of volunteers and under controlled and semi-closed experimental settings, which may not fully capture the full variability and spontaneity of mobility-aid usage observed in naturalistic traffic environments. While the PMMA dataset provides a detailed categorization of pedestrian types with AWDs, it does not cover all possible assistive devices or behavioral variations that may exist across different cultural, infrastructural, or situational contexts. Furthermore, the case study on crossing behavior analysis was conducted on a single intersection and a limited number of observation periods, and additional data collected across a broader range of locations and temporal conditions would be required to further validate and generalize the observed behavioral patterns. As a result, the findings of this dissertation should be interpreted within the scope of the evaluated datasets, scenarios, and case studies.

The second group of limitations is associated with the object detection and tracking methods used to generate trajectories for safety and behavioral analysis. This research relied on recent detection and tracking algorithms without introducing methodological modifications or task-specific adaptations. While this choice enabled a fair and systematic comparison of existing methods, it also implies that the observed trajectory inaccuracies and performance variations are inherently constrained by the capabilities of current perception models. Moreover, the majority of experiments were conducted using monocular video data, which limits the accuracy of spatial localization and depth estimation and may lead to trajectory instability, particularly in crowded or complex traffic scenes.

Another limitation concerns the computation and interpretation of safety and behavioral indicators derived from predicted trajectories. Indicators such as TTC and PET, are sensitive to the quality and temporal continuity of trajectories. Small errors in object detection or tracking can propagate and accumulate throughout the processing pipeline and result in amplified deviations in the computed indicators, especially at the interaction or individual level. More studies are needed to investigate how errors in object detection and tracking directly or indirectly influence transportation application analysis.

9.3 Future research

To further advance deep learning-based video-based road safety and crossing behavior analysis, several directions for future research can be considered in light of the limitations discussed above.

With respect to data and experimental settings, future studies will focus on expanding larger-

scale datasets with more diverse traffic scenarios and pedestrians categories. Collecting video data across a wider range of locations, viewpoints, and traffic conditions will enable a more comprehensive assessment of detection, tracking, and transportation application analysis pipelines. In particular, extending data collection to include adverse weather, and more heterogeneous urban environments would improve the generalizability of the findings. For pedestrians with AWDs datasets, future efforts should aim to collect data in naturalistic and open traffic environments, involving a broader range of assistive devices and participant profiles beyond volunteer-based and controlled settings. Such extensions would allow for a more realistic characterization of usage of AWDs and pedestrian behavior in real-world scenarios.

From a methodological perspective, future research needs to explore advances in object detection and tracking models that are better suited for downstream safety and behavioral analysis. While this dissertation focused on evaluating existing deep learning-based methods, developing task-aware or safety-oriented perception models should help improve trajectory quality for critical pedestrian categories and complex interactions. In addition, incorporating depth information through stereo vision, multi-view camera systems, or sensor fusion should enhance spatial localization accuracy and trajectory stability, particularly in crowded or occluded scenes. These improvements will contribute to more robust and reliable trajectory generation for safety indicator computation.

Beyond perception models, future work will also aim to further refine interaction matching methodologies. Moreover, systematically investigating the independent impact of different tracking metrics on the accuracy of safety and behavioral indicators would help clarify how tracking performance translates into downstream analysis quality, and would inform the selection or adaptation of evaluation metrics that are more directly aligned with safety analysis objectives.

Finally, future research should extend the proposed analysis framework to support larger-scale, multi-site analysis and cross-site comparisons. By deploying networks of cameras across corridors or neighborhoods, it will examine interactions and behavioral changes at a larger spatial scale. Combining video-based analysis with complementary data sources, such as signal timing information, weather variation, environmental data, or mobility-related sensor data, will further enhance the understanding of factors influencing pedestrian safety and crossing behavior. These directions collectively provide a solid foundation for building more comprehensive, inclusive, and trustworthy video-based transportation analysis systems.

REFERENCES

[1] J. Chen *et al.*, "Architecture of vehicle trajectories extraction with roadside lidar serving connected vehicles," *Ieee Access*, vol. 7, pp. 100 406–100 415, 2019.

[2] A. Laureshyn *et al.*, "Review of current study methods for vru safety. appendix 6–scoping review: surrogate measures of safety in site-based road traffic observations: Deliverable 2.1–part 4." 2016.

[3] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[4] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] Karyative, "Physiotherapy icons," https://www.flaticon.com/free-icons/physiotherapy, 2025, accessed: 2025-04-02.

[6] Leremy, "Wheelchair icons," Flaticon, 2025. [Online]. Available: https://www.flaticon.com/free-icons/wheelchair

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[8] S. Busch *et al.*, "Lumpi: The leibniz university multi-perspective intersection dataset," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1127–1134.

[9] N. Saunier, "Traffic intelligence software," https://bitbucket.org/Nicolas/trafficintelligence/wiki/Home, 2012, accessed July 26, 2012.

[10] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[11] H. Yu *et al.*, "The fertilizing role of African dust in the Amazon rainforest: A first multiyear assessment based on data from Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations," *Geophysical Research Letters*, vol. 42, no. 6, pp. 1984–1991, mar 2015.

[12] J. L. Hernández-Stefanoni *et al.*, "Improving species diversity and biomass estimates of tropical dry forests using airborne LiDAR," *Remote Sensing*, vol. 6, no. 6, pp. 4741–4763, 2014.

[13] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[15] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 28–31.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[21] S. Ren *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/abs/1506.01497

[22] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] ——, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[25] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[26] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics yolo," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] Ultralytics, "Yolov11: Ultralytics real-time object detection," https://github.com/ultralytics/ultralytics, 2024, accessed: 2026-01-29.

[28] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[29] N. Carion *et al.*, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[30] X. Zhu *et al.*, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[31] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[32] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[33] K. Han *et al.*, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[34] S. Taghvaeeyan and R. Rajamani, "Portable roadside sensors for vehicle counting, classification, and speed measurement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 73–83, 2013.

[35] A. Bewley *et al.*, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[36] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[37] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[38] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[39] Y. Zhang *et al.*, "Bytetrack: Multi-object tracking by associating every detection box," Springer, pp. 1–21, 2022.

[40] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," 2022. [Online]. Available: https://arxiv.org/abs/2206.14651

[41] J. Cao *et al.*, "Observation-centric sort: Rethinking sort for robust multi-object tracking," 2023. [Online]. Available: https://arxiv.org/abs/2203.14360

[42] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*. Springer, 2020, pp. 474–490.

[43] H. Laga *et al.*, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 1738–1764, 2020.

[44] H. Xu *et al.*, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 941–13 958, 2023.

[45] Y. Wei *et al.*, "Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation," in *Conference on robot learning*. PMLR, 2023, pp. 539–549.

[46] Y. Shi *et al.*, "Ega-depth: Efficient guided attention for self-supervised multi-camera depth estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 119–129.

[47] A. Masoumian *et al.*, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, 2022.

[48] D. Wofk *et al.*, "Fastdepth: Fast monocular depth estimation on embedded systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.

[49] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methodsa state-of-the-art review," *Sensors*, vol. 20, no. 8, p. 2272, 2020.

[50] R. T. Rodrigues *et al.*, "Active depth estimation: Stability analysis and its applications," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2002–2008.

[51] A. Masoumian *et al.*, "Absolute distance prediction based on deep learning object detection and monocular depth estimation models." in *CCIA*, 2021, pp. 325–334.

[52] S. Dávila-Soberón, A. Morales-Díaz, and M. Castelán, "A novel image dataset for detecting and classifying mobility aid users," *Expert Systems with Applications*, vol. 293, p. 128697, 2025.

[53] M. Kollmitz *et al.*, "Deep 3d perception of people and their mobility aids," *Robotics and Autonomous Systems*, vol. 114, pp. 29–40, 2019.

[54] J. Zhang *et al.*, "X-world: Accessibility, vision, and autonomy meet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9762–9771.

[55] N. Saunier and A. Laureshyn, "Surrogate measures of safety," in *International Encyclopedia of Transportation*, R. Vickerman, Ed. Elsevier, 2021, pp. 662–667.

[56] F. Amundsen and C. Hyden, "Proceedings of first workshop on traffic conflicts," *Oslo, TTI, Oslo, Norway and LTH Lund, Sweden*, vol. 78, 1977.

[57] T. Fu, L. Miranda-Moreno, and N. Saunier, "A novel framework to evaluate pedestrian safety at non-signalized locations," *Accident Analysis & Prevention*, vol. 111, pp. 23–33, 2018.

[58] T. Fu *et al.*, "Investigating secondary pedestrian-vehicle interactions at non-signalized intersections using vision-based trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 222–240, 2019.

[59] Q. Zhang, Z. Yang, and D. Sun, "Automated data collection and safety analysis at intersections based on a novel video processing system," *Transportation research record*, vol. 2673, no. 4, pp. 136–144, 2019.

[60] Q. Liu, N. Saunier, and G.-A. Bilodeau, "How good are deep learning methods for automated road safety analysis using video data? an experimental study," *arXiv preprint arXiv:2503.09807*, 2025.

[61] A. Laureshyn, Å. Svensson, and C. Hydén, "Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation," *Accident Analysis & Prevention*, vol. 42, no. 6, pp. 1637–1646, 2010.

[62] R. Chandra *et al.*, "Robusttp: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," *arXiv preprint arXiv:1907.08752*, 2019.

[63] X. Huang *et al.*, "Intelligent intersection: Two-stream convolutional networks for real-time near accident detection in traffic video," *arXiv preprint arXiv:1901.01138*, 2019.

[64] K. Ismail, T. Sayed, and N. Saunier, "Automated analysis of pedestrian–vehicle conflicts: Context for before-and-after studies," *Transportation research record*, vol. 2198, no. 1, pp. 52–64, 2010.

[65] N. Saunier, T. Sayed, and K. Ismail, "Large-scale automated analysis of vehicle interactions and collisions," *Transportation Research Record*, vol. 2147, no. 1, pp. 42–50, 2010.

[66] P. St-Aubin, N. Saunier, and L. Miranda-Moreno, "Large-scale automated pedestrian and cyclist conflict analysis using video data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 363–379, 2015.

[67] B. J. Schroeder, *A behavior-based methodology for evaluating pedestrian-vehicle interaction at crosswalks*. North Carolina State University, 2008.

[68] G. Asaithambi, M. O. Kuttan, and S. Chandra, "Pedestrian road crossing behavior under mixed traffic conditions: a comparative study of an intersection before and after implementing control measures," *Transportation in developing economies*, vol. 2, no. 2, p. 14, 2016.

[69] M.-S. Cloutier *et al.*, "outta my way! individual and environmental correlates of interactions between pedestrians and vehicles during street crossings," *Accident Analysis & Prevention*, vol. 104, pp. 36–45, 2017.

[70] M. Iryo-Asano and W. K. Alhajyaseen, "Modeling pedestrian crossing speed profiles considering speed change behavior for the safety assessment of signalized intersections," *Accident Analysis & Prevention*, vol. 108, pp. 332–342, 2017.

[71] P. Onelcin and Y. Alver, "Illegal crossing behavior of pedestrians at signalized intersections: factors affecting the gap acceptance," *Transportation research part F: traffic psychology and behaviour*, vol. 31, pp. 124–132, 2015.

[72] M. R. Virkler, "Pedestrian compliance effects on signal delay," *Transportation research record*, vol. 1636, no. 1, pp. 88–91, 1998.

[73] Y. Alver *et al.*, "Evaluation of pedestrian critical gap and crossing speed at midblock crossing using image processing," *Accident Analysis & Prevention*, vol. 156, p. 106127, 2021.

[74] Z. Yang *et al.*, "Analysis of pedestrian-related crossing behavior at intersections: A latent dirichlet allocation approach," *International journal of transportation science and technology*, vol. 12, no. 4, pp. 1052–1063, 2023.

[75] P. Olszewski *et al.*, "Surrogate safety indicator for unsignalised pedestrian crossings," *Accident Analysis & Prevention*, vol. 150, p. 105900, 2020.

[76] K. Aghabayk *et al.*, "Observational-based study to explore pedestrian crossing behaviors at signalized and unsignalized crosswalks," *Accident Analysis & Prevention*, vol. 151, p. 105990, 2021.

[77] M. Brosseau *et al.*, "The impact of waiting time and other factors on dangerous pedestrian crossings and violations at signalized intersections: A case study in montreal," *Transportation research part F: traffic psychology and behaviour*, vol. 21, pp. 159–172, 2013.

[78] B. R. Kadali and P. Vedagiri, "Models for pedestrian gap acceptance behaviour analysis at unprotected mid-block crosswalks under mixed traffic conditions," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 32, pp. 114–126, 2015.

[79] L. L. Thompson *et al.*, "Impact of social and technological distraction on pedestrian crossing behaviour: an observational study," *Injury Prevention*, vol. 19, no. 4, pp. 232–237, 2013.

[80] H. Guo *et al.*, "Modeling pedestrian violation behavior at signalized crosswalks in china: A hazards-based duration approach," *Traffic injury prevention*, vol. 12, no. 1, pp. 96–103, 2011.

[81] V. Perumal *et al.*, "Study on pedestrian crossing behavior at signalized intersections," *Journal of traffic and transportation engineering (English edition)*, vol. 1, no. 2, pp. 103–110, 2014.

[82] B. R. Kadali and V. Perumal, "Evaluation of pedestrian crossing speed change patterns at unprotected mid-block crosswalks in india," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 7, no. 2, pp. 241–252, 2020.

[83] A. O'Dell *et al.*, "The impact of pedestrian distraction on safety behaviours at controlled and uncontrolled crossings," *Safety*, vol. 9, no. 3, p. 65, 2023.

[84] M. S. Sharifi *et al.*, "A large-scale controlled experiment on pedestrian walking behaviour involving individuals with disabilities," *Transportation Research Procedia*, vol. 25, pp. 14–25, 2017.

[85] N. Schwartz *et al.*, "Disability and pedestrian road traffic injury: A scoping review," *Health & place*, vol. 77, p. 102896, 2022.

[86] M. S. Sharifi *et al.*, "A large-scale controlled experiment on pedestrian walking behavior involving individuals with disabilities," *Travel behaviour and society*, vol. 8, pp. 14–25, 2017.

[87] CVAT.ai Corporation, "Computer vision annotation tool (cvat)," 2023, accessed: 2025-10-30. [Online]. Available: https://github.com/cvat-ai/cvat

[88] WHO, "https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death," 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[89] R. Elvik, "Some difficulties in defining populations of entities for estimating the expected number of accidents," *Accident Analysis & Prevention*, vol. 20, no. 4, pp. 261–275, 1988.

[90] J. Hayward, *Near misses as a measure of safety at urban intersections.* Pennsylvania Transportation and Traffic Safety Center, 1971.

[91] M. G. Mohamed and N. Saunier, "Behavior analysis using a multilevel motion pattern learning framework," *Transportation Research Record*, vol. 2528, no. 1, pp. 116–127, 2015.

[92] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," Tech. Rep., 1978.

[93] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings. international conference on image processing*, vol. 1. IEEE, 2002, pp. I–I.

[94] T. Lindeberg, "Scale invariant feature transform," 2012.

[95] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

[96] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[97] W. Luo *et al.*, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[98] G. Ciaparrone *et al.*, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[99] C. Kim *et al.*, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4696–4704.

[100] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[101] H. Hu *et al.*, "An automatic tracking method for multiple cells based on multi-feature fusion," *IEEE Access*, vol. 6, pp. 69 782–69 793, 2018.

[102] M. Kim, S. Alletto, and L. Rigazio, "Similarity mapping with enhanced siamese network for multi-object tracking," *arXiv preprint arXiv:1609.09156*, 2016.

[103] L. Wang *et al.*, "Online multiple object tracking via flow and convolutional features," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3630–3634.

[104] Y. Lu, C. Lu, and C.-K. Tang, "Online video object detection using association lstm," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2344–2352.

[105] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[106] J. C. Hayward, "Near miss determination through use of a scale of danger," 1972.

[107] C. Johnsson, A. Laureshyn, and C. Dágostino, "A relative approach to the validation of surrogate measures of safety," *Accident Analysis & Prevention*, vol. 161, p. 106350, 2021.

[108] C. Johnsson *et al.*, "T-analyst-semi-automated tool for traffic conflict analysis," *InDeV, Horizon*, 2020.

[109] É. Beauchamp, N. Saunier, and M.-S. Cloutier, "Study of automated shuttle interactions in city traffic using surrogate measures of safety," *Transportation research part C: emerging technologies*, vol. 135, p. 103465, 2022.

[110] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Ieee, 2011, pp. 1609–1615.

[111] J. Stipancic, L. Miranda-Moreno, and N. Saunier, "Vehicle manoeuvers as surrogate safety measures: Extracting data from the gps-enabled smartphones of regular drivers," *Accident Analysis & Prevention*, vol. 115, pp. 160–169, 2018.

[112] K.-C. Huang *et al.*, "Monodtr: Monocular 3d object detection with depth-aware transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4012–4021.

[113] Y. Liu, L. Wang, and M. Liu, "Yolostereo3d: A step back to 2d for efficient stereo 3d detection," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 018–13 024.

[114] M. G. Mohamed and N. Saunier, "Motion prediction methods for surrogate safety analysis," *Transportation research record*, vol. 2386, no. 1, pp. 168–178, 2013.

[115] N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," in *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE, 2006, pp. 59–59.

[116] Y. Chen *et al.*, "Dsgn: Deep stereo geometry network for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 536–12 545.

[117] X. Chen *et al.*, "3d object proposals for accurate object class detection," *Advances in neural information processing systems*, vol. 28, 2015.

[118] E. Ristani *et al.*, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*. Springer, 2016, pp. 17–35.

[119] I. Chakravarti, R. Laha, and J. Roy, *Handbook of Methods of Applied Statistics, Volume I.* John Wiley and Sons, 1967.

[120] World Health Organization, "Road traffic injuries," https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2023, accessed: 20251030.

[121] S. R. Perkins and J. I. Harris, *Criteria for traffic conflict characteristics, signalized intersections.* Research Laboratories, General Motors Corporation, 1967.

[122] B. Navarro *et al.*, "Do stop-signs improve the safety for all road users? a before-after study of stop-controlled intersections using video-based trajectories and surrogate measures of safety," *Accident Analysis & Prevention*, vol. 167, p. 106563, 2022.

[123] X. Shi *et al.*, "Video-based trajectory extraction with deep learning for high-granularity highway simulation (high-sim)," *Communications in transportation research*, vol. 1, p. 100014, 2021.

[124] N. Anwari *et al.*, "Investigating surrogate safety measures at midblock pedestrian crossings using multivariate models with roadside camera data," *Accident Analysis & Prevention*, vol. 192, p. 107233, 2023.

[125] F. Amundsen and C. Hyden, "Proceedings: First workshop on traffic conflicts," in *First Workshop on Traffic Conflicts*, F. Amundsen and C. Hyden, Eds., Oslo, Norway, 1977, p. 1.

[126] C. Oh, S. Park, and S. G. Ritchie, "A method for identifying rear-end collision risks using inductive loop detectors," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 295–301, 2006.

[127] Z. Li *et al.*, "Surrogate safety measure for evaluating rear-end collision risk related to kinematic waves near freeway recurrent bottlenecks," *Accident Analysis & Prevention*, vol. 64, pp. 52–61, 2014.

[128] A. R. Mamdoohi *et al.*, "Comparative analysis of safety performance indicators based on inductive loop detector data," *PROMET-Traffic&Transportation*, vol. 26, no. 2, pp. 139–149, 2014.

[129] J. Wu *et al.*, "An improved vehicle-pedestrian near-crash identification method with a roadside lidar sensor," *Journal of safety research*, vol. 73, pp. 211–224, 2020.

[130] N. Bhattarai *et al.*, "Proactive safety analysis using roadside lidar based vehicle trajectory data: A study of rear-end crashes," *Transportation research record*, vol. 2678, no. 3, pp. 772–785, 2024.

[131] L. Scholl *et al.*, "A surrogate video-based safety methodology for diagnosis and evaluation of low-cost pedestrian-safety countermeasures: The case of cochabamba, bolivia," *Sustainability*, vol. 11, no. 17, p. 4737, 2019.

[132] E. Marti *et al.*, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[133] P. St-Aubin, L. Miranda-Moreno, and N. Saunier, "An automated surrogate safety analysis at protected highway ramps using cross-sectional and before–after video data," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 284–295, 2013.

[134] M. Özuysal *et al.*, "Feature harvesting for tracking-by-detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III 9*. Springer, 2006, pp. 592–605.

[135] N. I. Husna Fauzi, Z. Musa, and F. Hujainah, "Feature-based object detection and tracking: A systematic literature review," *International Journal of Image and Graphics*, vol. 24, no. 03, p. 2450037, 2024.

[136] N. Saunier, "Traffic intelligence software," https://trafficintelligence.confins.net, 2024, accessed July 26, 2024.

[137] C. E. Shourov *et al.*, "Deep learning architectures for skateboarder–pedestrian surrogate safety measures," *Future transportation*, vol. 1, no. 2, pp. 387–413, 2021.

[138] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[139] R.-A. Rill and K. B. Faragó, "Collision avoidance using deep learning-based monocular vision," *SN Computer Science*, vol. 2, no. 5, p. 375, 2021.

[140] D. Sun *et al.*, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.

[141] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[142] Brisk Synergies, "Brisklumina - on demand safety analysis," https://brisksynergies.com/brisklumina/, 2019, accessed: 2019-07-31.

[143] A. Badki *et al.*, "Bi3d: Stereo depth estimation via binary classifications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1600–1608.

[144] ——, "Binary ttc: A temporal geofence for autonomous navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 946–12 955.

[145] J.-Y. Zhu *et al.*, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[146] C. Zhao *et al.*, "Monocular depth estimation based on deep learning: An overview," *Science China Technological Sciences*, vol. 63, no. 9, pp. 1612–1627, 2020.

[147] C. Li *et al.*, "Fp-ttc: Fast prediction of time-to-collision using monocular images," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[148] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, p. 246309, 2008.

[149] J. Luiten *et al.*, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[150] C.-Y. Wang *et al.*, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[151] C. W. Cleverdon, "The cranfield tests on index language devices," *Aslib Proceedings*, vol. 19, no. 6, pp. 173–194, 1967.

[152] R. M. Cormack, "A review of classification," *Journal of the Royal Statistical Society: Series A (General)*, vol. 134, no. 3, pp. 321–353, 1971.

[153] S. B. Kotsiantis *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[154] Z.-Q. Zhao *et al.*, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[155] N. Saunier, T. Sayed, and K. Ismail, "An object assignment algorithm for tracking performance evaluation," in *11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2009), Miami, Fl, USA, June*, vol. 25. Citeseer, 2009, pp. 9–17.

[156] T. Lin *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[157] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 8490, may 2017. [Online]. Available: https://doi.org/10.1145/3065386

[158] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 3213–3223.

[159] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[160] J. Behley *et al.*, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[161] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (jaad)," *arXiv preprint arXiv:1609.04741*, 2016.

[162] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," 2020. [Online]. Available: https://arxiv.org/abs/1903.11027

[163] P. Dendorfer *et al.*, "Mot20: A benchmark for multi object tracking in crowded scenes," 2020. [Online]. Available: https://arxiv.org/abs/2003.09003

[164] C. Creß *et al.*, "A9-dataset: Multi-sensor infrastructure-based dataset for mobility research," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 965–970.

[165] T. Chavdarova *et al.*, "Wildtrack: A multicamera hd dataset for dense unscripted pedestrian detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 5030–5039.

[166] Z. Tang *et al.*, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019, pp. 8797–8806.

[167] Y. Nie *et al.*, "Synposes: Generating virtual dataset for pedestrian detection in corner cases," *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 801–804, 2022.

[168] D. Li *et al.*, "A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios," *IEEE transactions on image processing*, vol. 28, no. 4, pp. 1575–1590, 2018.

[169] Stereolabs, "Zed 2 stereo camera," https://www.stereolabs.com/zed-2/, accessed: 2025-08-04.

[170] VideoLAN, "Vlc media player," 2025, accessed: July 29, 2025. [Online]. Available: https://www.videolan.org/vlc/

[171] Z. Ge *et al.*, "Yolox: Exceeding yolo series in 2021," 2021. [Online]. Available: https://arxiv.org/abs/2107.08430

[172] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[173] W. Lv *et al.*, "Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer," 2024. [Online]. Available: https://arxiv.org/abs/2407.17140

[174] MMDetection Contributors, "OpenMMLab Detection Toolbox and Benchmark," Aug. 2018. [Online]. Available: https://github.com/open-mmlab/mmdetection

[175] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[176] Gouvernement du Québec and Ministère des Transports, de la Mobilité durable et de lÉlectrification des transports, "Cadre dintervention en transport actif," Gouvernement du Québec, Tech. Rep., 2018. [Online]. Available: https://www.transports.gouv.qc.ca/fr/ministere/role_ministere/DocumentsPMD/PMD-09-cadre-intervention.pdf

[177] J. Montufar *et al.*, "Pedestrians' normal walking speed and speed when crossing a street," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2002, pp. 90–97, 2007.

[178] J. Arango and J. Montufar, "Walking speed of older pedestrians who use canes or walkers for mobility," *Transportation Research Record*, vol. 2073, no. 1, pp. 79–85, 2008.

[179] X. Chen and Others, "Analysis of path distribution characteristics and pedestrian path deviation based on trajectory data," *Scientific World Journal*, vol. 2022, p. Article ID 2924008, 2022.

[180] G. Filomena *et al.*, "Empirical characterisation of agents spatial behaviour in pedestrian movement," *Journal of Environmental Psychology*, vol. 82, p. 101807, 2022.

[181] K. Ismail, T. Sayed, and N. Saunier, "Automated collection of pedestrian data using computer vision techniques," in *Transportation Research Board Annual Meeting Compendium of Papers, Washington, DC*, 2009, pp. 09–1122.

[182] P. Nimac *et al.*, "Pedestrian traffic light control with crosswalk fmcw radar and group tracking algorithm," *Sensors*, vol. 22, no. 5, p. 1754, 2022.

[183] A. Muralidharan *et al.*, "Management of intersections with multi-modal high-resolution data," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 101–112, 2016.

[184] Z. Zhang *et al.*, "Vulnerable road user detection for roadside-assisted safety protection: A comprehensive survey." *Applied Sciences (2076-3417)*, vol. 15, no. 7, 2025.

[185] S. Zhang *et al.*, "A comprehensive review on artificial intelligence empowered solutions for enhancing pedestrian and cyclist safety," *arXiv preprint arXiv:2510.03314*, 2025.

[186] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 34–42.

[187] J. Abidi and F. Filali, "Real-time ai-based inference of people gender and age in highly crowded environments," *Machine Learning with Applications*, vol. 14, p. 100500, 2023.

[188] Q. Liu, N. Saunier, and G.-A. Bilodeau, "Pmma: The polytechnique montreal mobility aids dataset," 2026. [Online]. Available: https://arxiv.org/abs/2602.10259

[189] D. Sharma, T. Hade, and Q. Tian, "Comparison of deep object detectors on a new vulnerable pedestrian dataset," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2024, pp. 278–283.

[190] E. Duim, M. L. Lebrão, and J. L. F. Antunes, "Walking speed of older people and pedestrian crossing time," *Journal of Transport & Health*, vol. 5, pp. 70–76, 2017.

[191] U. Lachapelle and M.-S. Cloutier, "On the complexity of finishing a crossing on time: Elderly pedestrians, timing and cycling infrastructure," *Transportation Research Part A: Policy and Practice*, vol. 96, pp. 54–63, 2017.

[192] H. Zhang, "Threshold research on the elderly pedestrian ratio in pedestrian crossing speed setting on signalized crosswalks in china," *Promet-Traffic&Transportation*, vol. 32, no. 1, pp. 55–63, 2020.

[193] A. Trpković *et al.*, "The crossing speed of elderly pedestrians," *Promet-Traffic&Transportation*, vol. 29, no. 2, pp. 175–183, 2017.

[194] L. H. Barrero-Solano *et al.*, "Video-based assessment of pedestrian behavior: Development and testing of methods," *Revista de Salud Pública*, vol. 19, pp. 182–187, 2017.

[195] D. Jiao and T. Fei, "Pedestrian walking speed monitoring at street scale by an in-flight drone," *PeerJ Computer Science*, vol. 9, 2023.

[196] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[197] K.-C. Wang *et al.*, "Pedestrian-crossing detection enhanced by cyclicgan-based loop learning and automatic labeling," *Applied Sciences*, vol. 15, no. 12, p. 6459, 2025.

[198] D. Bolya *et al.*, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[199] M. Che *et al.*, "Impact of keep left measure on pedestrians, cyclists and e-scooter riders at a crossing of a signalised junction," *Transportation Research Part A: Policy and Practice*, vol. 179, p. 103942, 2024.

[200] C. Zhang *et al.*, "Modeling bidirectional mixed flow of pedestrians, bicycles and e-mopeds at signalized crosswalks," *Reliability Engineering & System Safety*, vol. 266, p. 111654, 2026.

[201] M. R. Virkler and S. Elayadath, "Pedestrian speed-flow-density relationships," *Transportation Research Record*, no. 1438, pp. 51–58, 1994.

[202] J. Zhang and A. Seyfried, "Empirical characteristics of different types of pedestrian streams," *Procedia engineering*, vol. 62, pp. 655–662, 2013.

[203] M. Nikoli, M. Bierlaire, and B. Farooq, "Probabilistic speed-density relationship for pedestrians based on data driven space and time representation," in *Swiss Transport Research Conference (STRC)*, 2014.

[204] Y. Wang and X. Shen, "A decay model for the fundamental diagram of pedestrian movement," *Physica A: Statistical Mechanics and its Applications*, vol. 531, p. 121731, 2019.

[205] E. Giannoulaki, L. Dimitriou, and H. N. Koutsopoulos, "Pedestrian walking speed analysis: A systematic review," *Sustainability*, vol. 16, no. 11, p. 4813, 2024.

APPENDIX A ARTICLE 2: SUPPLEMENTARY RESULTS

A.1 Road-user-wise CDFs

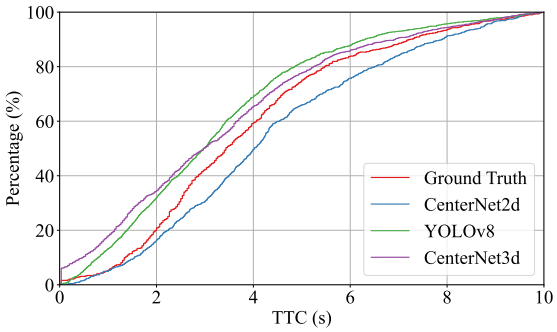
We report the road-user-wise CDFs for both views, shown in Fig. A.1 and Fig. A.2. In view 1, for TTC_{min}, the three interaction categories have similar shapes to the overall CDFs. CenterNet2D outperforms YOLOv8 and CenterNet3D in carcar and carcyclist interactions, with particularly strong performance in the carcyclist category, where its results are very close to the ground truth. For carpedestrian interactions, CenterNet3D achieves the best performance among the three methods.

For both views, the PET CDFs for car-car interactions are similar to the overall CDFs, which is expected as car-car interactions represent more than 80 % of all interactions with a PET, while the PET CDFs for car-cyclist and car-pedestrian interactions show different shapes, with fewer interactions in the range [0, 2] s and a higher proportion in the range [2, 10] s. In view 1, YOLOv8 demonstrates the best performance compared to the other two methods across all three interaction types, whereas CenterNet2D is a close second in carcyclist and carpedestrian interactions.

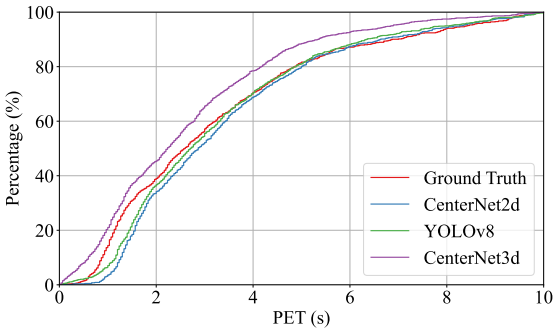
In view 2, there are fewer automatically detected car-cyclist and car-pedestrian interactions, especially for YOLOv8 as seen in Table 5.3. CenterNet2D shows trends similar to the ground truth across all three categories and outperforms YOLOv8 in terms of TTC_{min}. However, for PET, both methods still exhibit considerable discrepancies with the ground truth across all interaction types, although CenterNet2D still slightly outperforms YOLOv8.

A.2 Road-user-wise indicator distributions

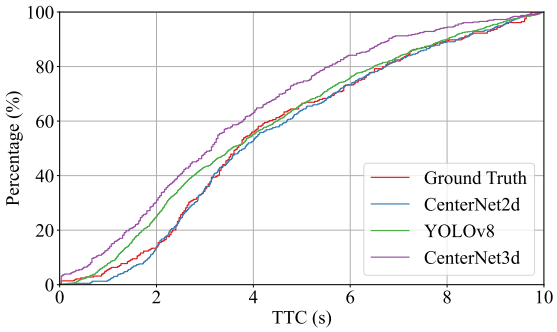
Fig. A.3 and Fig. A.4 illustrate the road-user-wise indicator distributions in view 1 and view 2 respectively. Compared with GT, in view 1, CenterNet2D shows broadly similar distribution patterns to GT for both indicators across all three categories. YOLOv8 performs the worst on TTC with a much higher number of interactions over the whole range of values, whereas for PET, all three methods produce distributions comparable to GT , except for the inflation of car-car and car-cyclist interactions with very low values by CenterNet3D. In view 2, both methods demonstrate strong similarity to the GT for the carcar category, while substantial differences are observed in the carcyclist and carpedestrian categories, where CenterNet2D outperforms YOLOv8. The car-cyclist interactions are the hardest to reliably detect across



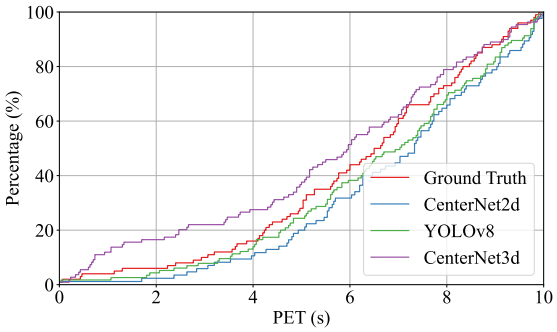
(a) TTC_{min} CDFs for car-car interactions



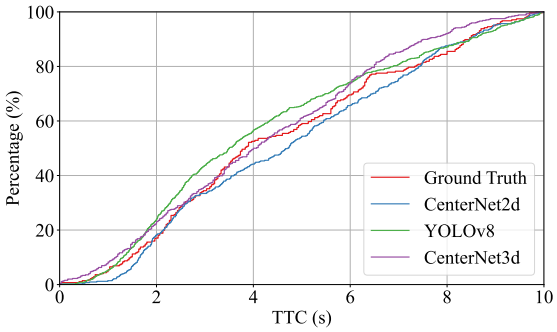
(b) PET CDFs for car-car interactions



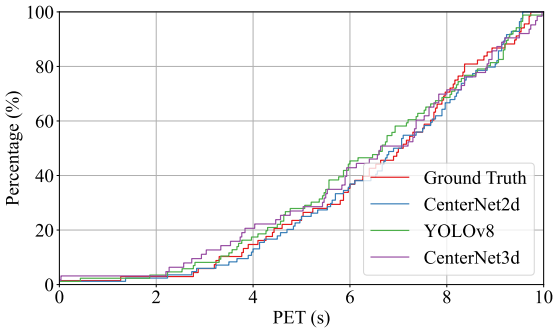
(c) TTC_{min} CDFs for car-cyc interactions



(d) PET CDFs for car-cyc interactions

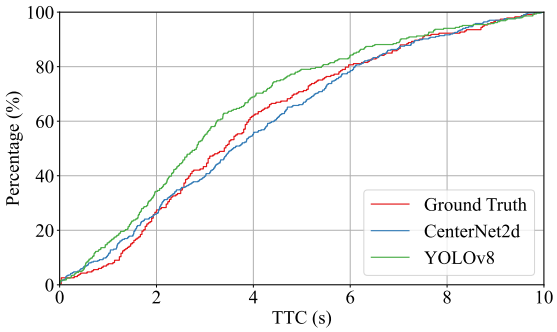


(e) TTC_{min} CDFs for car-ped interactions

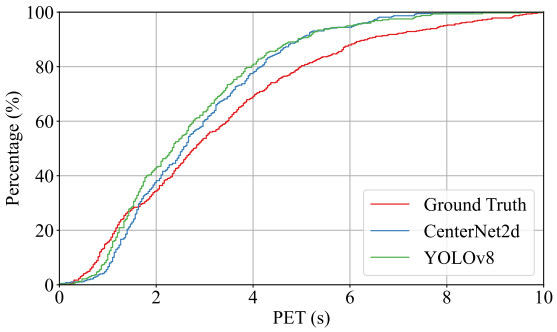


(f) PET CDFs for car-ped interactions

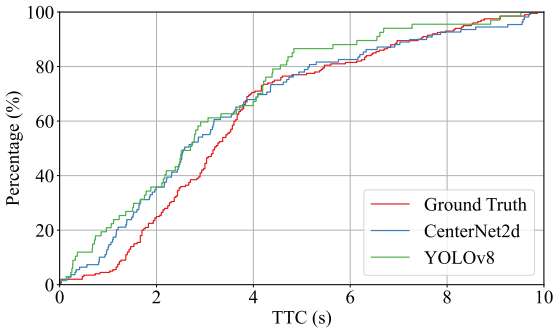
Figure A.1 Road-user-wise CDFs of TTC_{min} and PET for view 1.



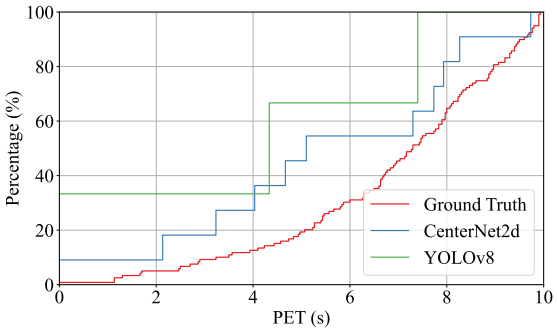
(a) TTC_{min} CDFs for car-car interactions



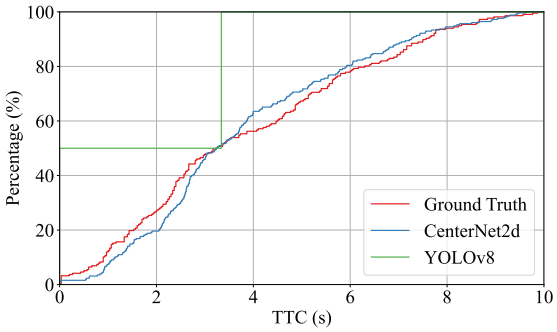
(b) PET CDFs for car-car interactions



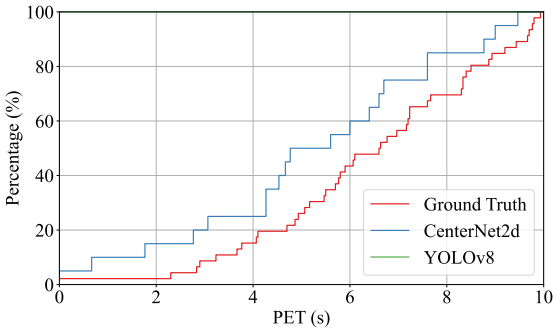
(c) TTC_{min} CDFs for car-cyc interactions



(d) PET CDFs for car-cyc interactions



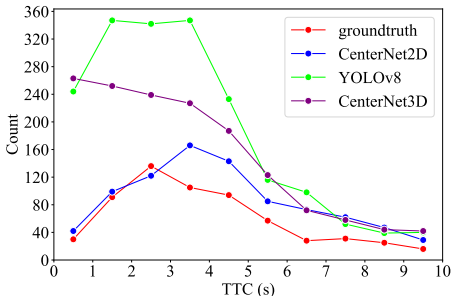
(e) TTC_{min} CDFs for car-ped interactions



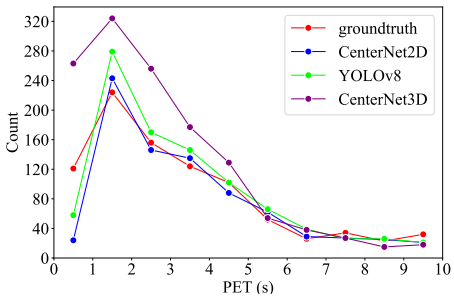
(f) PET CDF for car-ped interactions

Figure A.2 Road-user-wise CDFs of TTC_{min} and PET for view 2.

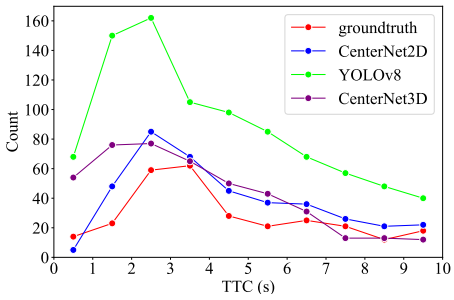
large ranges.



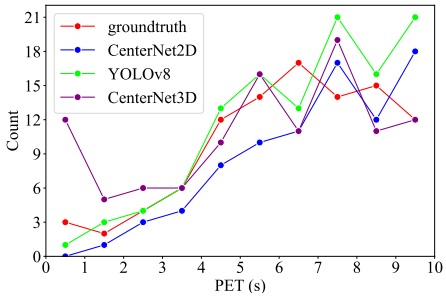
(a) TTC_{min} counts for car-car



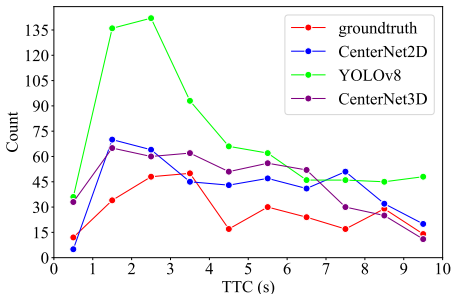
(b) PET counts for car-car



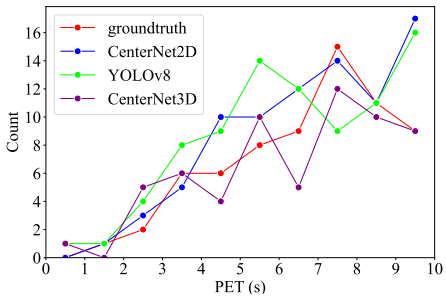
(c) TTC_{min} counts for car-cyc



(d) PET counts for car-cyc

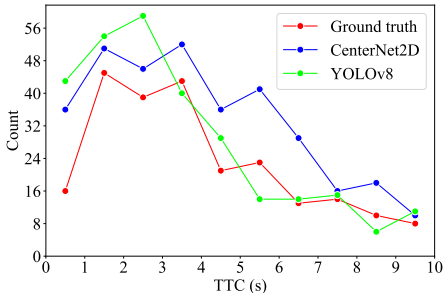


(e) TTC_{min} counts for car-ped

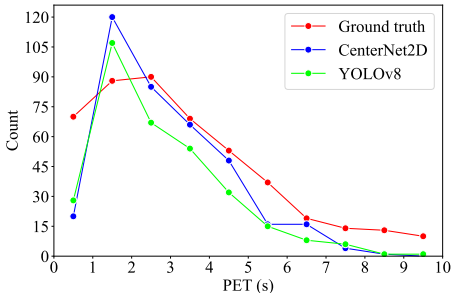


(f) PET counts for car-ped

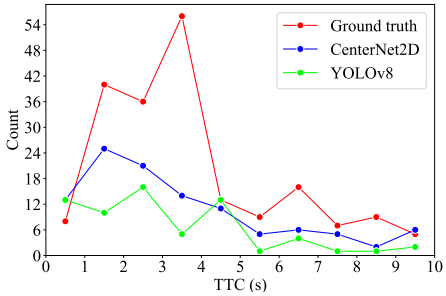
Figure A.3 Road-user-wise distributions for TTC_{min} and PET for view 1.



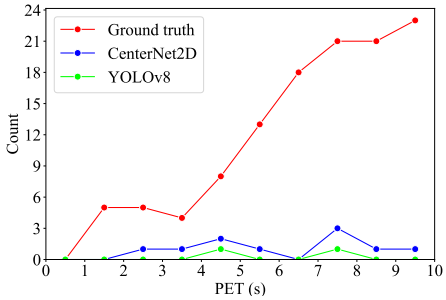
(a) TTC_{min} values for car-car



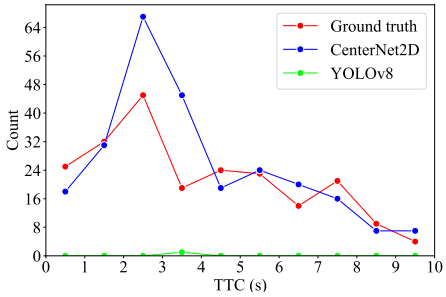
(b) PET counts for car-car



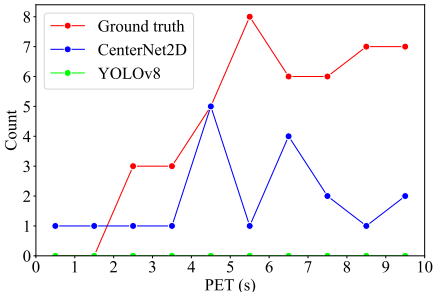
(c) TTC_{min} counts for car-cyc



(d) PET counts for car-cyc



(e) TTC_{min} counts for car-ped



(f) PET counts for car-ped

Figure A.4 Road-user-wise distributions for TTC_{min} and PET for view 2.

APPENDIX B ARTICLE 3: SUPPLEMENTARY RESULTS

B.1 Tracking performance across different detectors and trackers for video 2 test set

Table B.1 Tracking performance across different detectors and trackers for selected categories on video 2 test set (Part 1)

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
Ped									
Deformable DETR	OC-SORT	17.06	26.90	10.86	83.31	16,829		254	
CenterNet	ByteTrack	17.67	28.57	10.97	81.76	32,880		587	
CenterNet	BOT-SORT	17.75	27.93	11.33	81.73	34,088		1,239	
Faster R-CNN	OC-SORT	18.27	31.22	10.74	81.54	29,371		419	
Deformable DETR	BOT-SORT	18.32	28.36	11.89	82.77	22,584		713	
Deformable DETR	ByteTrack	18.40	28.86	11.81	82.98	21,903		392	
RT-DETR	OC-SORT	18.66	31.60	11.07	81.15	23,898		342	
Faster R-CNN	ByteTrack	18.98	29.71	12.17	81.41	34,389	17,383	482	9
Faster R-CNN	BOT-SORT	19.40	29.12	12.98	81.36	35,277		909	
CenterNet	OC-SORT	19.73	31.64	12.35	82.08	25,376		496	
RT-DETR	BOT-SORT	20.71	29.81	14.43	80.61	29,154		835	
YOLOX	OC-SORT	20.97	**34.93**	12.61	**84.15**	26,140		327	
RT-DETR	ByteTrack	21.45	30.49	15.13	80.74	28,332		485	
YOLOX	BOT-SORT	*23.24*	33.40	**16.18**	83.98	31,045		768	
YOLOX	ByteTrack	**23.34**	*33.89*	*16.09*	*84.07*	30,304		401	
Cane									
Faster R-CNN	OC-SORT	2.56	2.39	2.82	73.87	12,205		407	
Faster R-CNN	BOT-SORT	2.71	3.17	2.42	73.47	17,642		896	
Faster R-CNN	ByteTrack	2.75	2.98	2.65	73.88	16,717		477	
CenterNet	BOT-SORT	3.27	4.07	2.79	75.01	17,156		1,207	
CenterNet	ByteTrack	3.34	4.08	2.89	75.37	15,812		570	
CenterNet	OC-SORT	3.55	3.85	3.40	75.67	9,499		498	
YOLOX	BOT-SORT	5.09	4.32	6.10	78.57	12,966		758	
YOLOX	ByteTrack	5.16	4.26	6.34	78.86	12,147	7,770	399	3
Deformable DETR	OC-SORT	5.25	7.47	3.72	*79.25*	7,783		286	
RT-DETR	ByteTrack	5.40	5.77	5.25	77.39	12,648		463	
YOLOX	OC-SORT	5.52	4.02	7.63	**79.78**	8,536		333	
RT-DETR	OC-SORT	6.01	5.54	*6.85*	77.59	8,785		369	
RT-DETR	BOT-SORT	6.90	5.75	**8.94**	76.91	13,583		822	
Deformable DETR	BOT-SORT	*7.18*	*8.32*	6.21	78.45	12,964		820	
Deformable DETR	ByteTrack	**7.31**	**8.33**	6.45	78.43	12,196		407	
Wheelchair									
CenterNet	BOT-SORT	17.80	7.55	41.96	87.75	17,765		1,160	
CenterNet	ByteTrack	18.52	8.13	42.20	87.84	16,498		557	
RT-DETR	OC-SORT	23.22	13.47	40.12	85.00	9,699		314	
CenterNet	OC-SORT	23.34	13.01	41.89	87.98	10,139		471	
Faster R-CNN	BOT-SORT	24.46	7.54	79.51	87.22	18,034		868	
Faster R-CNN	ByteTrack	25.16	7.94	79.79	87.47	17,172		468	
RT-DETR	BOT-SORT	25.97	9.41	71.74	84.92	13,950		768	
RT-DETR	ByteTrack	26.75	9.97	71.84	85.12	13,170	1,569	442	1
Faster R-CNN	OC-SORT	28.86	10.49	79.53	87.26	12,889		390	
YOLOX	BOT-SORT	30.68	10.88	86.56	89.64	12,921		730	
YOLOX	ByteTrack	31.96	11.72	**87.27**	**90.04**	12,081		392	
Deformable DETR	BOT-SORT	35.48	15.08	83.55	88.86	9,167		640	
Deformable DETR	ByteTrack	36.91	*16.23*	84.03	89.13	8,539		360	
YOLOX	OC-SORT	*37.19*	15.96	*86.76*	*89.73*	8,785		313	
Deformable DETR	OC-SORT	**43.74**	**22.87**	83.68	88.90	5,998		201	

Table B.2 Tracking performance across different detectors and trackers for selected categories on video 2 test set (part 2)

WalkerWalking

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	BOT-SORT	20.84	23.22	18.87	81.35	19,178		1,182	
CenterNet	ByteTrack	21.49	24.52	18.99	81.44	17,947		570	
Deformable DETR	OC-SORT	24.84	*38.82*	15.96	80.66	8,357		223	
Deformable DETR	BOT-SORT	26.09	32.97	20.71	81.83	12,291		760	
CenterNet	OC-SORT	26.17	31.53	21.93	**85.02**	11,263		480	
Faster R-CNN	BOT-SORT	26.27	24.40	28.76	80.10	20,728		876	
Faster R-CNN	ByteTrack	26.97	25.39	29.20	83.80	19,858		470	
RT-DETR	BOT-SORT	28.66	31.03	26.73	83.25	15,267	7,390	763	5
Faster R-CNN	OC-SORT	29.27	30.08	28.99	82.06	15,450		395	
RT-DETR	ByteTrack	30.06	31.78	28.70	*84.86*	14,455		455	
YOLOX	BOT-SORT	32.02	31.24	33.00	81.56	16,678		765	
RT-DETR	OC-SORT	32.35	38.13	27.75	82.39	10,653		324	
Deformable DETR	ByteTrack	32.80	34.47	31.31	82.44	11,617		379	
YOLOX	ByteTrack	*32.84*	32.67	*33.20*	83.66	15,908		411	
YOLOX	OC-SORT	**37.71**	**38.69**	**36.88**	81.51	12,206		333	

WalkerResting

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
RT-DETR	OC-SORT	0.00	0.00	83.29	0.00	4,620		305	
CenterNet	OC-SORT	0.00	0.00	83.68	0.00	6,797		459	
RT-DETR	ByteTrack	0.06	0.02	84.80	0.22	8,061		434	
RT-DETR	BOT-SORT	0.07	0.03	84.53	0.20	8,836		732	
CenterNet	ByteTrack	0.09	0.03	84.08	0.28	13,172		551	
CenterNet	BOT-SORT	0.13	0.05	82.32	0.35	14,399		1,150	
Faster R-CNN	OC-SORT	0.25	0.13	82.61	0.49	9,987		389	
YOLOX	OC-SORT	0.41	0.18	82.71	0.93	5,705	749	307	1
Faster R-CNN	BOT-SORT	0.48	0.27	82.60	0.84	15,096		857	
Faster R-CNN	ByteTrack	0.48	0.25	82.51	0.92	14,238		470	
Deformable DETR	OC-SORT	0.49	0.31	84.60	0.79	2,595		184	
YOLOX	BOT-SORT	0.57	0.24	82.84	1.36	9,784		715	
YOLOX	ByteTrack	0.60	0.26	*84.82*	1.41	9,034		387	
Deformable DETR	BOT-SORT	*1.52*	*0.89*	**84.98**	*2.60*	5,592		629	
Deformable DETR	ByteTrack	**1.58**	**0.90**	84.49	**2.77**	5,008		357	

PushEmptyWheelchair

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
RT-DETR	BOT-SORT	9.86	3.79	81.35	26.03	9,194		731	
CenterNet	OC-SORT	4.69	3.30	81.44	6.73	6,919		460	
Deformable DETR	OC-SORT	**16.45**	**9.58**	80.66	28.68	3,438		184	
Deformable DETR	ByteTrack	*14.04*	*6.78*	81.83	29.64	5,917		354	
Faster R-CNN	ByteTrack	13.69	3.96	**85.02**	47.49	14,698		466	
Deformable DETR	BOT-SORT	13.09	6.15	80.10	28.43	6,503		624	
Faster R-CNN	BOT-SORT	13.05	3.69	83.80	**46.34**	15,567		854	
Faster R-CNN	OC-SORT	12.33	4.90	83.25	31.16	10,419	801	388	1
RT-DETR	OC-SORT	11.16	5.27	82.06	23.96	4,976		306	
CenterNet	ByteTrack	10.92	3.79	*84.86*	*31.64*	13,469		551	
YOLOX	OC-SORT	10.88	4.96	81.56	24.17	6,066		307	
YOLOX	ByteTrack	10.70	4.19	82.39	27.78	9,451		388	
RT-DETR	ByteTrack	10.51	4.12	82.44	27.22	8,386		434	
CenterNet	BOT-SORT	10.23	3.42	83.66	30.70	14,664		1,150	
YOLOX	BOT-SORT	10.06	3.89	81.51	26.34	10,196		717	

Table B.3 Tracking performance across different detectors and trackers for selected categories on video 2 test set (part 3)

WheelchairGroup

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	ByteTrack	26.50	33.94	20.69	*88.28*	23,035		564	
CenterNet	BOT-SORT	26.64	32.20	22.04	87.80	24,156		1,224	
RT-DETR	BOT-SORT	27.95	36.58	21.37	87.02	16,572		756	
RT-DETR	ByteTrack	28.76	38.32	21.61	87.55	15,754		452	
CenterNet	OC-SORT	28.88	44.25	18.87	87.84	16,363		482	
RT-DETR	OC-SORT	32.38	44.27	23.69	87.32	11,964		329	
Deformable DETR	OC-SORT	33.22	*51.20*	21.59	87.76	9,871		195	
Deformable DETR	BOT-SORT	38.80	44.61	33.77	87.29	13,365	9,672	646	5
Faster R-CNN	BOT-SORT	40.13	32.84	49.04	87.92	25,414		874	
Deformable DETR	ByteTrack	40.39	46.60	35.02	87.93	12,768		366	
Faster R-CNN	ByteTrack	40.99	34.15	49.19	**88.37**	24,582		475	
Faster R-CNN	OC-SORT	50.91	40.98	63.25	87.95	20,121		391	
YOLOX	BOT-SORT	51.89	41.76	*64.53*	87.57	19,493		732	
YOLOX	OC-SORT	*57.67*	**52.02**	**64.01**	87.64	15,239		314	
YOLOX	ByteTrack	**60.65**	43.87	**83.85**	88.14	18,700		391	

WheelchairPusher

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	OC-SORT	26.41	41.01	17.01	83.21	16,259		475	
RT-DETR	BOT-SORT	26.46	32.89	21.31	82.65	15,978		767	
RT-DETR	ByteTrack	27.04	34.09	21.47	82.76	15,207		449	
CenterNet	BOT-SORT	27.16	29.59	24.94	83.17	24,449		1,239	
Deformable DETR	OC-SORT	27.20	*46.06*	16.08	*84.19*	9,338		190	
RT-DETR	OC-SORT	28.29	39.19	20.44	82.96	11,422		312	
Faster R-CNN	BOT-SORT	28.48	30.25	26.83	83.04	24,948		888	
CenterNet	ByteTrack	29.50	31.15	27.94	83.27	23,161	9,672	570	5
Faster R-CNN	OC-SORT	30.46	37.28	24.89	83.06	19,678		398	
Faster R-CNN	ByteTrack	31.17	31.41	30.93	83.27	24,070		477	
Deformable DETR	BOT-SORT	33.88	41.22	27.86	83.69	12,671		636	
Deformable DETR	ByteTrack	34.58	42.63	28.06	84.01	12,093		359	
YOLOX	BOT-SORT	38.34	39.08	**37.65**	84.16	18,167		727	
YOLOX	ByteTrack	*38.73*	40.66	*36.92*	84.16	17,440		391	
YOLOX	OC-SORT	**41.44**	**47.40**	36.27	**84.33**	14,100		318	

WheelchairPushedUser

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	BOT-SORT	24.87	28.97	21.49	82.16	23,723		1,184	
CenterNet	OC-SORT	27.85	38.88	20.11	82.02	15,183		468	
RT-DETR	BOT-SORT	27.88	34.09	23.25	82.46	15,772		741	
RT-DETR	ByteTrack	28.50	35.43	23.43	82.61	14,989		434	
RT-DETR	OC-SORT	30.58	41.48	23.01	82.76	11,393		307	
CenterNet	ByteTrack	32.65	30.63	34.89	82.26	22,376		556	
Deformable DETR	OC-SORT	33.88	**53.49**	21.48	86.09	10,169		184	
Deformable DETR	BOT-SORT	35.40	45.64	27.47	86.05	13,454	9,672	633	5
Deformable DETR	ByteTrack	36.07	47.25	27.55	86.23	12,907		357	
Faster R-CNN	BOT-SORT	42.47	30.87	58.45	82.80	24,776		863	
Faster R-CNN	ByteTrack	43.28	31.99	58.55	82.86	23,905		470	
Faster R-CNN	OC-SORT	45.54	38.25	54.21	82.80	19,635		393	
YOLOX	BOT-SORT	59.05	42.74	*81.59*	87.06	18,740		718	
YOLOX	ByteTrack	*60.16*	44.29	**81.72**	*87.13*	18,071		387	
YOLOX	OC-SORT	**65.52**	*53.06*	80.90	**87.14**	14,659		311	

B.2 Tracking performance across different detectors and trackers for video 3 test set

Table B.4 Tracking performance across different detectors and trackers for selected categories on video 3 test set (Part 1)

Ped

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	OC-SORT	52.11	59.83	45.42	85.83	5,198		80	
Faster R-CNN	OC-SORT	52.47	62.94	43.78	85.33	4,975		53	
CenterNet	BOT-SORT	61.33	49.43	76.22	86.02	6,686		237	
RT-DETR	OC-SORT	61.36	71.75	52.66	85.99	4,501		36	
CenterNet	ByteTrack	61.91	51.32	74.82	86.05	6,436		103	
Faster R-CNN	BOT-SORT	63.66	58.37	69.48	85.38	5,589		164	
Faster R-CNN	ByteTrack	68.18	59.93	77.65	85.48	5,444		83	
YOLOX	BOT-SORT	73.25	67.84	79.35	85.24	4,735	3,995	130	5
YOLOX	ByteTrack	73.41	69.96	77.35	85.29	4,591		71	
Deformable DETR	OC-SORT	74.47	**79.63**	69.79	86.49	4,047		17	
YOLOX	OC-SORT	75.04	73.27	77.17	85.23	4,315		53	
RT-DETR	BOT-SORT	75.91	68.16	85.12	86.01	4,796		82	
RT-DETR	ByteTrack	76.58	69.43	85.01	86.04	4,706		40	
Deformable DETR	BOT-SORT	*81.99*	78.79	*85.48*	*86.56*	4,201		47	
Deformable DETR	ByteTrack	**82.44**	*79.46*	**85.70**	**86.70**	4,166		23	

Cane

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	BOT-SORT	29.90	27.51	32.54	82.02	4,222		242	
Faster R-CNN	BOT-SORT	34.51	37.54	31.91	82.28	3,188		172	
CenterNet	ByteTrack	35.25	28.95	43.10	81.87	3,993		105	
Faster R-CNN	ByteTrack	35.41	39.37	32.02	82.28	3,028		81	
CenterNet	OC-SORT	40.17	36.36	44.53	81.99	2,864		84	
YOLOX	ByteTrack	44.17	50.85	38.58	81.67	2,193		66	
YOLOX	BOT-SORT	45.04	48.47	41.98	81.66	2,313		133	
YOLOX	OC-SORT	45.99	55.91	38.32	81.71	1,916	1,598	53	2
Faster R-CNN	OC-SORT	50.60	42.67	60.05	81.88	2,587		56	
RT-DETR	OC-SORT	53.44	54.64	52.30	81.67	2,065		34	
Deformable DETR	OC-SORT	56.33	*64.44*	49.39	**82.94**	1,532		21	
Deformable DETR	BOT-SORT	58.15	63.62	53.23	82.60	1,726		48	
Deformable DETR	ByteTrack	58.66	**64.58**	53.36	*82.63*	1,695		20	
RT-DETR	BOT-SORT	*60.08*	49.76	*72.56*	81.69	2,373		79	
RT-DETR	ByteTrack	**61.37**	51.77	**72.74**	81.71	2,278		38	

Wheelchair

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	BOT-SORT	38.59	18.46	80.71	85.66	3,578		231	
CenterNet	ByteTrack	40.13	19.88	81.06	85.93	3,312		102	
Faster R-CNN	BOT-SORT	47.57	27.28	83.00	86.15	2,435		161	
Faster R-CNN	ByteTrack	49.09	29.06	82.97	86.07	2,283		80	
CenterNet	OC-SORT	49.12	29.64	81.42	85.76	2,184		78	
Faster R-CNN	OC-SORT	53.99	35.14	82.99	85.89	1,884		52	
YOLOX	BOT-SORT	62.45	44.71	87.25	90.63	1,537		121	
RT-DETR	BOT-SORT	62.97	44.50	89.11	90.21	1,616	799	73	1
RT-DETR	ByteTrack	64.63	46.95	88.99	90.18	1,529		36	
YOLOX	ByteTrack	65.26	48.78	87.32	90.66	1,406		67	
RT-DETR	OC-SORT	68.84	53.22	89.05	90.22	1,347		31	
YOLOX	OC-SORT	71.59	58.73	87.26	90.63	1,160		50	
Deformable DETR	BOT-SORT	78.31	68.79	89.15	91.45	1,023		42	
Deformable DETR	ByteTrack	*79.76*	*71.02*	**89.59**	**91.66**	992		20	
Deformable DETR	OC-SORT	**82.75**	**76.64**	*89.34*	*91.51*	913		15	

Table B.5 Tracking performance across different detectors and trackers for selected categories on video 3 test set (part 2)

WalkerWalking									
Detector	**Method**	**Tracking Metrics (%)**				**Detection Counts**		**Track ID Counts**	
		HOTA	**DetA**	**AssA**	**LocA**	**Predictions**	**GT**	**Predictions**	**GT**
CenterNet	ByteTrack	1.17	0.38	3.58	82.30	3,127		102	
CenterNet	BOT-SORT	1.17	0.36	3.78	84.41	3,443		234	
CenterNet	OC-SORT	1.30	0.61	2.80	84.77	2,092		80	
Deformable DETR	BOT-SORT	1.51	1.28	1.78	86.32	737		43	
Deformable DETR	ByteTrack	1.57	1.36	1.81	*87.38*	707		20	
Faster R-CNN	BOT-SORT	1.71	0.74	3.98	80.50	1,696		162	
YOLOX	ByteTrack	1.76	1.11	2.78	83.71	1,106		69	
Faster R-CNN	ByteTrack	1.80	0.81	4.00	80.76	1,541	16	80	1
RT-DETR	OC-SORT	1.82	0.98	3.45	83.99	1,366		31	
Faster R-CNN	OC-SORT	2.08	1.09	3.96	80.02	1,135		54	
YOLOX	OC-SORT	2.12	*1.52*	2.97	**87.40**	870		50	
YOLOX	BOT-SORT	2.24	1.06	4.73	86.70	1,229		123	
Deformable DETR	OC-SORT	4.56	**1.53**	13.58	86.62	617		16	
RT-DETR	BOT-SORT	*4.82*	0.81	*29.11*	83.98	1,641		79	
RT-DETR	ByteTrack	**5.08**	0.89	**29.54**	83.98	1,503		41	
WalkerResting									
Detector	**Method**	**Tracking Metrics (%)**				**Detection Counts**		**Track ID Counts**	
		HOTA	**DetA**	**AssA**	**LocA**	**Predictions**	**GT**	**Predictions**	**GT**
RT-DETR	OC-SORT	26.96	36.39	20.19	80.69	1,106		32	
CenterNet	BOT-SORT	44.62	30.32	65.96	80.56	3,641		231	
CenterNet	ByteTrack	46.04	32.05	66.40	80.73	3,420		104	
Deformable DETR	OC-SORT	46.43	**53.39**	40.63	**88.00**	1,110		16	
RT-DETR	ByteTrack	48.67	37.11	64.17	80.51	1,446		36	
RT-DETR	BOT-SORT	48.79	36.95	64.76	80.38	1,485		80	
YOLOX	BOT-SORT	49.90	39.16	63.71	83.11	1,600		122	
YOLOX	ByteTrack	51.16	40.94	64.02	83.24	1,473	1,582	66	2
Faster R-CNN	BOT-SORT	51.77	38.86	69.33	85.18	3,003		161	
CenterNet	OC-SORT	52.76	47.50	59.34	80.60	2,202		79	
Faster R-CNN	ByteTrack	52.99	40.70	69.37	85.25	2,853		80	
YOLOX	OC-SORT	53.65	43.41	66.38	83.50	1,208		50	
Faster R-CNN	OC-SORT	56.48	46.13	69.48	85.36	2,441		52	
Deformable DETR	BOT-SORT	*60.17*	51.14	**71.12**	*87.85*	1,264		44	
Deformable DETR	ByteTrack	**60.67**	**52.05**	*71.04*	**88.00**	1,222		20	

Table B.6 Tracking performance across different detectors and trackers for selected categories on video 3 test set (part 3)

WheelchairGroup

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	ByteTrack	35.42	29.73	42.34	85.70	4,351		102	
CenterNet	OC-SORT	36.22	40.31	32.66	86.92	3,196		84	
CenterNet	BOT-SORT	38.02	28.52	50.70	86.72	4,660		247	
Faster R-CNN	BOT-SORT	62.82	44.08	*89.53*	90.32	3,258		163	
Faster R-CNN	ByteTrack	63.77	45.82	88.75	89.91	3,108		82	
Faster R-CNN	OC-SORT	68.27	52.30	89.11	90.26	2,720		55	
YOLOX	BOT-SORT	71.12	57.05	88.66	89.32	2,480		132	
YOLOX	OC-SORT	72.44	66.14	79.40	89.27	2,076	1,598	52	2
RT-DETR	BOT-SORT	72.52	58.91	89.29	89.80	2,422		75	
YOLOX	ByteTrack	72.58	60.02	87.78	88.83	2,338		72	
RT-DETR	ByteTrack	73.44	60.75	88.78	89.50	2,335		38	
RT-DETR	OC-SORT	76.72	66.01	89.17	89.81	2,151		33	
Deformable DETR	OC-SORT	77.77	**81.83**	73.98	*90.41*	1,709		17	
Deformable DETR	BOT-SORT	*83.38*	77.58	**89.61**	**90.56**	1,842		46	
Deformable DETR	ByteTrack	**83.49**	*78.39*	88.93	90.02	1,808		21	

WheelchairPusher

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	BOT-SORT	31.90	29.48	34.53	85.47	4,466		243	
CenterNet	OC-SORT	34.66	42.94	28.00	85.47	2,920		80	
Faster R-CNN	OC-SORT	35.08	48.03	25.69	**86.33**	2,687		53	
YOLOX	BOT-SORT	40.05	54.77	29.33	85.80	2,372		126	
YOLOX	ByteTrack	41.23	57.88	29.40	85.80	2,240		68	
Faster R-CNN	BOT-SORT	48.50	40.85	57.60	*86.30*	3,287		163	
Faster R-CNN	ByteTrack	49.72	42.84	57.74	**86.33**	3,129		81	
CenterNet	ByteTrack	49.92	30.96	**80.53**	85.46	4,242	1,598	102	2
RT-DETR	BOT-SORT	55.86	54.77	57.04	85.47	2,405		77	
RT-DETR	ByteTrack	56.87	56.69	57.11	85.49	2,323		37	
RT-DETR	OC-SORT	58.15	60.22	56.21	85.69	2,109		34	
YOLOX	OC-SORT	67.54	64.43	*70.85*	85.81	1,955		53	
Deformable DETR	BOT-SORT	69.24	71.19	67.41	86.26	1,825		45	
Deformable DETR	OC-SORT	*70.12*	**74.09**	66.46	**86.33**	1,664		17	
Deformable DETR	ByteTrack	**70.54**	*72.27*	68.93	86.08	1,786		22	

WheelchairPushedUser

Detector	Method	Tracking Metrics (%)				Detection Counts		Track ID Counts	
		HOTA	DetA	AssA	LocA	Predictions	GT	Predictions	GT
CenterNet	OC-SORT	40.23	42.93	37.78	85.85	2,925		82	
CenterNet	BOT-SORT	44.14	30.28	64.36	85.74	4,376		235	
CenterNet	ByteTrack	45.10	32.15	63.28	85.52	4,119		102	
Faster R-CNN	BOT-SORT	59.53	41.22	86.00	87.48	3,354		161	
Faster R-CNN	ByteTrack	60.93	43.18	85.98	87.49	3,200		80	
Faster R-CNN	OC-SORT	64.21	48.87	84.39	87.45	2,782		53	
YOLOX	OC-SORT	67.35	68.75	66.02	87.86	2,007		52	
RT-DETR	BOT-SORT	69.49	56.19	85.96	87.31	2,440	1,598	74	2
RT-DETR	ByteTrack	70.80	58.34	85.95	87.26	2,351		38	
YOLOX	BOT-SORT	71.14	58.00	**87.29**	87.86	2,401		121	
YOLOX	ByteTrack	73.00	61.23	*87.04*	87.77	2,268		67	
RT-DETR	OC-SORT	73.71	63.19	86.00	87.30	2,167		33	
Deformable DETR	BOT-SORT	80.53	74.96	86.52	**88.27**	1,837		43	
Deformable DETR	ByteTrack	*81.06*	*76.11*	86.34	*88.23*	1,806		21	
Deformable DETR	OC-SORT	**82.48**	**79.17**	85.94	88.11	1,719		15	