



Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users: Phase-I

Ted L. Morris, MS

Varadaraya G. Shenoy

Ben Shepard, MS

Vassilios Morellas, PhD



Center for
Transportation
Studies

[1]



**CENTER FOR CONNECTED
AND AUTOMATED
TRANSPORTATION**

Report No. CTS 26-08
Project Start Date:02/2024
Project End Date:02/2026

July 2026

Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users: Phase-I

by

Ted Morris

Research Engineer

**Dept. of Computer Science and Engineering, University of
Minnesota**



Northwestern



DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3551747105 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation: Morris, T.L., Shenoy, V.G., Shepard, B., Morellas, V. Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users: Phase I. (2026). University of Minnesota.

Contacts

For more information:

Vassilios Morellas
Department of Electrical and Computer
Engineering, University of Minnesota
4-174 Keller Hall, 200 Union St. SE
Minneapolis, MN 55455
Phone: 612-624-4822
Email: morellas@umn.edu

CCAT
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
umtri-ccat@umich.edu
(734) 763-2498



Technical Report Documentation Page

1. Report No. CTS 26-08	2. Government Accession No. Leave blank – not used	3. Recipient's Catalog No. Leave blank - not used
4. Title and Subtitle Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users: Phase-I		5. Report Date July 2026
		6. Performing Organization Code Enter any/all unique numbers assigned to the performing organization, if applicable.
7. Author(s) Ted L. Morris, M.S., https://orcid.org/0009-0007-7933-4263 Varadaraya. G. Shenoy, https://orcid.org/0009-0002-0393-3829 Ben Shepard, M.S., https://orcid.org/0009-0005-0964-4309 Vassilios Morellas, Ph.D., https://orcid.org/0000-0002-9198-8442		8. Performing Organization Report No. Enter any/all unique alphanumeric report numbers assigned by the performing organization, if applicable.
		10. Work Unit No.
9. Performing Organization Name and Address Minnesota Robotics Institute (MnRI) Gemini-Huntley Robotics Research Laboratory Shepherd Laboratories 100 Union St SE Minneapolis, MN 55455		11. Contract or Grant No. Contract No. 69A3551747105
		13. Type of Report and Period Covered Final Report (Feb 2024 – Feb 2026)
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE Washington, DC 20590		14. Sponsoring Agency Code OST-R
		15. Supplementary Notes Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.



16. Abstract

Autonomous and connected vehicle perception capabilities can be enhanced with roadside sensing, by increasing the spatial range of traffic events for all road users and revealing ‘blind spots’ where onboard vehicle sensors cannot ‘see’. A multi-lane roundabout intersection can manifest such scenarios, that introduce perception and control challenges to the autonomous vehicle. This project developed and assessed a roadside sensing and detection system that can be integrated with connected and autonomous vehicle control and information sharing research studies. The project was proposed as the first of two phases, whereby the intention of the first phase was to develop the roadside traffic detection system architecture for all road users. The system consists of a single high resolution fisheye camera sensor, located above the center of an urban arterial 2-lane roundabout intersection, a wireless 5G communication link, and edge computing hardware. During the time of the project, 3 State-of-the-art Deep Learning one-shot YOLO (You Only Look Once) models were trained and evaluated for fisheye optics. Robust detection was achieved for all models, with mAP50-95 scores 0.743, 0.634, and 0.607 for multi-image, and direct single fisheye image object detection methods, across diverse environment conditions. Vehicle tracking evaluations indicated that over 70% of all vehicle trajectories are fully tracked across all directions and maneuvers; the track errors primarily occurred before vehicles entered the roundabout. Improvements to pedestrian identification and detection should be pursued specifically to better address vulnerable road user movement states within the roundabout intersection and improve overall system performance. This will benefit the planned Phase-II objective of developing and testing a probabilistic self-learning, adaptive approach to control a connected, autonomous entering vehicle through Active Inference.

17. Key Words

Detection and Identification, vehicle control, traffic and pedestrian safety

18. Distribution Statement

No restrictions.

19. Security Classif. (of this report)

Unclassified

20. Security Classif. (of this page)

Unclassified

21. No. of Pages

43

22. Price

Leave blank –
not used

Form DOT F 1700.7 (8-72)

Reproduction of completed page authorized



Northwestern



TABLE OF CONTENTS

1	Introduction.....	1
2	Background	2
3	Urban Roundabout Test Site	4
4	Detection and Tracking Methodology.....	5
4.1	Camera Calibration.....	5
4.1.1	Single image formation and calibration	6
4.1.2	Fisheye multi-image field calibration	7
4.2	Deep Learning object detection models	8
4.2.1	YOLOv8	9
4.2.2	YOLOv11	10
5	Object Detection and Tracking Results	12
5.1	YOLO Omniview Multi-Image Detection	12
5.1.1	YOLOv8 Multi-Image Model Training	12
5.1.2	YOLOv8 Multi-Image Performance	15
5.2	YOLO Fisheye Image Detection	18
5.2.1	YOLO Fisheye Model Training	18
5.2.2	YOLO Fisheye Model Performance	20
5.3	Computational Performance	24
5.4	Object Tracking.....	25
5.4.1	Object tracking performance.....	29
6	Conclusions and Recommendations.....	32
7	REFERENCES.....	33

TABLE OF FIGURES

Fig. 3.1: Urban arterial roundabout camera sensor deployment, Richfield MN	4
Fig. 3.2: 5G Router set up for sensor and edge computer communication.....	5
Fig. 4.1: Pin-hole (top) and fisheye rectilinear (bottom) camera image projection models.	6
Fig. 4.2: Divided images directionally aligned. Note the rotation increments were irregular due to the offset of the camera center from the center of the intersection.....	7
Fig. 4.3: Input image (left) and the rectified image calibration result.(right).....	8
Fig. 4.4: YOLOv8 architecture featuring C2f backbone, PAN-FPN hybrid neck, and anchor-free detection head. Adapted from [37]	9
Fig. 5.1: Example of mosaic data augmentation using patches from four of the roundabout training data images.	14
Fig. 5.2: YOLOv8 training results.....	15
Fig. 5.3: YOLOv8 multi-image detection class confusion matrix.....	16
Fig. 5.4: Detection occlusion cases for YOLOv8s multi-image detection. TOP: vehicle queuing, foreground occlusions, BOTTOM: vehicle edge detection.....	17
Fig. 5.5: Pedestrian crosswalk detection. TOP: Pedestrian crossing sequence with fail to yield by roundabout exiting vehicles; MIDDLE-Left/BOTTOM: Pedestrian occlusion-based <i>FN</i> cases during group crossing sequence from initial merging of pedestrians and roadside sign in refuge island, and BOTTOM-Center/Right: from background confusion.	18
Fig. 5.6: YOLOv11n-OBB training results	20
Fig. 5.7: F1-Confidence curves test results.	20
Fig. 5.8: YOLOv11n-OBB test results normalized confusion matrix	21
Fig. 5.9: YOLOv8n-OBB test results normalized confusion matrix	21
Fig. 5.10: Unrectified (top row) and rectified (bottom row) fisheye images during fair daytime conditions.	22
Fig. 5.11: Unrectified (top row) and rectified (bottom row) fisheye images during daytime snowy conditions.	23
Fig. 5.12: Unrectified (top row) and rectified (bottom row) fisheye images during nighttime conditions	24
Fig. 5.13: Google Earth view with latitude, longitude calibration points.....	26
Fig. 5.14: Westbound vehicle tracked recovered the lost track ID.....	27
Fig. 5.16: West-Southbound inside lane vehicle tracked	28
Fig. 5.15: Southbound outer-lane vehicle tracked recovered the lost track ID	28
Fig. 5.17: Eastbound stopped vehicle tracked; time points B and C overlap on LEFT intersection diagram	29
Fig. 5.18: LEFT: Samples of detected track IDs, RIGHT: Real-time object detection track ID observations.	30
Fig. 5.19: Ground truth object trajectory time lengths	30
Fig. 5.20: Object tracking association <i>TP/FN/FP</i> accuracy--all classes.....	31

TABLE OF TABLES

Table 4.1 Comparison of YOLOv8 and YOLOv11 architectures and their relevance to fisheye object	11
Table 5.1 Class-wise test multi-image performance of YOLOv8	15
Table 5.2: Class-wise test performance for YOLOv8n-OBB.....	21
Table 5.3: Class-wise test performance for YOLOv11n-OBB.....	22
Table 5.4. Computation performance comparison of models and detection approach.....	24

1 Introduction

Roundabout intersections have proliferated in the United States as an established innovation to reduce congestion and improve traffic safety. Several studies confirm a key design principle of reducing dangerous crashes that occur within traditional signalized and unsignalized intersections [1], [2]. Local municipalities are replacing their entirety signalized intersection assets with uncontrolled roundabout designs, or have adopted instituted policies which favor roundabouts as a first alternative when upgrading or replacing traditional intersections, due to evidence of significant reduction of serious injury and fatal crashes and vehicle speeds [3], [4]. Note however, these and other observational studies also revealed that vulnerable road users perceive they are less safe than traditional intersections. Past research determined that vehicles had low rate of yielding compliance for pedestrians, particularly with 2-lane roundabouts [5], [6]. Several studies, particularly in Europe, found significant increases in bicycle-vehicle related crashes under different contexts related to intersection design, and operational and perceptual characteristics [7].

Accordingly, human drivers entering and navigating within these intersections must make a multitude of perceptually nuanced decisions weighing gap selection and simultaneous interactions with other vehicles and vulnerable road users – such as crossing pedestrians, bicyclists, and public transit users (appendix B in [1]). Considering vehicle interactions, predicting driver entering driver critical gap acceptance and follow up headways has proven to be a challenging problem due to a variety of factors, such as the intersection capacity, the number of circulating and entrance lanes, a driver’s familiarity, and perceptual assumptions of another vehicles maneuver intent. and regional driving behavior differences [8], [9], [10]. As the transition to more connected and autonomy in vehicles progress, the vehicles will be required to embody such nuanced perceptual decision making and actions in uncertain conditions to improve driver safety and autonomous vehicle performance. Roadside sensing can play a critical role in enhancing these capabilities. Roadside sensing enhances ego vehicle perception to ‘see’ past blind spots, and areas that are beyond or outside the on-vehicle sensing envelope, as well as providing real-time location-based information from non-connected road users.

This study is cast as a two-phase research project. The objective of the first phase is to develop and test a fixed, roadside sensor perception system to track the movement of all road users. The realtime output can be integrated with wireless communication standards to broadcast real time traffic motion data to connected vehicles that are approaching, or within, the roundabout intersection. This report summarizes results of the implementation of the roadside sensing system for monitoring and detecting all road user traffic data. A planned second phase will develop and test an embodied AI approach for connected/autonomous vehicle control through an Active Inference framework that will use the in-situ traffic states perceived from the roundabout roadside sensing system and an experimental CAV. This will be achieved through the Infrastructure to Vehicle (I2V) Sensor Data Shared Message (SDSM) SAE J2735 wireless communication protocols [11]. In this first project phase, the sensor observations provide required relative location, heading, and speed and object type fields, which can be encoded into the SDSM messages broadcasted out to receiving autonomous or connected vehicles [12], [13]. Although the work has focused on Roundabouts because of the perceptual and control challenges they introduce, the work in this project could be exploited to benefit traffic safety of all users within traditional intersections, or to address other perceptual ‘blind-spots’ of vehicle mounted sensors that arise during driving tasks.

The report is organized as follows. Chapter 2 will provide background of past methods used for roadway and intersection vehicle and pedestrian detection. Chapter 3 will describe the testing site and edge hardware used for implementing real-time sensing and detection. Chapter 4 introduces the methodologies used for single-camera fisheye detection and image rectification. Model training, data preparation, and performance results are discussed in chapter 5, with concluding lessons learned and future research directions prvided in Chapter 6.

2 Background

Autonomous vehicle sensor perception and control implementations have been driven in part by the rapid evolution of Artificial Intelligence AI and Generative Artificial (GenAI) deep learning modeling. Data fusion from onboard sensor technologies provides situational awareness with the immediate surroundings of the vehicle, allowing new paradigms to be explored for intelligent decision making and action control policies such as collision avoidance of pedestrians [14], vehicles and other obstacles with model predictive control [15], Reinforcement Learning [16], Learning by Cheating (LBC) [17], and pedestrian collision avoidance through Active Inference (AIF) [18]. These advances have important implications for improving traffic safety, the work implies ‘extended’ perception beyond their sensing envelope will improve the vehicle anticipatory action policies, such as traffic scenarios where there are ‘blind spots’—e.g, beyond the range or observational capabilities of the sensor systems [18]. Vehicle to vehicle (V2V), and Vehicle to Infrastructure (V2I) Basic Safety Messages (BSM) are one mechanism to address these cases, since other vehicles themselves because the extended ‘eyes’ to enhance vehicle perception that can be applied to human and autonomous vehicle cooperative driving tasks (up to 300 meters dedicated short range communication (DSRC) distance). For instance, V2V ameliorate a significant number of serious crashes by applying the concept to ‘unseen’ speeding vehicles, or ‘see through’ vehicles in front of the ego vehicle to anticipate rapid upstream breaking, or unsafe ‘blind’ passing conditions , or importantly, intersection movement assist applications (IMA) to reduce intersection related crashes [19]. The disadvantage of V2V is that all road user traffic entities must be equipped with V2V transmitting and receiving devices to directly benefit from this approach.

Roadside camera sensors can also extend perceptual boundaries of vehicles, removing the requirement that all road users must be connected through a communication protocol. Enan et. al [20] proposed and assessed a Cellular Vehicle to Everything (C-V2X) using camera based object detection to generate BSM parcels to create a data package that follows the SAE J2735 formatting and encoding standard. In their system, a camera pointed straight-on to front facing vehicles. An image-axis aligned bounding box encapsulated the front bottom of the vehicle, while a planar homography transformed the pixel coordinates into latitude-longitude geolocations. Their approach indicated significant vehicle geolocation accuracy improvements over a commercially available vehicle on-board C-V2X unit (OBU). Other researchers proposed a realtime vehicle detection and tracking approach using multiplicity of cameras surrounding roundabout intersection. Their approach suggested tracking accuracies from 0.1 to over 1 meters compared with RTK differential correction GPS (DGPS) trajectory traces [21].

Particularly at intersections, a very large extended perceptual boundary for roadside sensing affords more complete coverage for all traffic participants to improve anticipatory actions. Zhang et. al [22] demonstrated a multi-camera fusion approach to broaden intersection extents of a 4-way 2-lane intersection, using world coordinate transformed coordinates of the traffic participants, with Kalman filtering to join tracks between disparate cameras facing different viewpoints, illustrating position accuracies within the 1.5m Basic Safety Message (BSM) standard. Zou et. al (2022) used a similar approach for a 2-lane roundabout intersection. The geodetic location and heading was estimated by training a one-shot object detector, MobileNetv2, to regress on 3D oriented bounding box (OBB) projections using vehicle traffic ‘ground truth’ labeled camera view images synthesized in CARLA [21]. Rather than a multiplicity of cameras, Morteza et. al. (2025) [23] proposed a single fisheye camera to also provide a wide-area roadside perception for a 4-way intersection. Vehicle geodetic location and heading was estimated by fitting the 2D projection of class type defined 3D boxes, onto the 2D image detection bounding boxes (re-trained YOLOv5) and iteratively aligned for maximum overlap, achieving an average accuracy of .31 and .76 meters for pedestrians and vehicles.

Researchers have also investigated the use of drones to achieve wide area birds eye view (BEV) detection to examine roadway and intersection operations safety, while nearly eliminating object occlusions that arise with height constrained terrestrial sensors. As part of an urban arterial wide digital twin database, CitySim, Zheng, et

al. (2024) [24] created a high accuracy, fully annotated extracted vehicle trajectories and oriented bounding box data set and from over 1,000 minutes of drone data using the Mask R-CNN . Islam, et al. (2022) used real-time detection and tracking from the CitySim drone high resolution video to estimate Post-Encroachment Time (PET) between road users at a 9-lane signalized intersection, demonstrating the value of oriented bounding boxes for such calculations [25]. Relevant to this project, Kehagia & Anagostopoulos (2022) constructed PET heat maps within roundabout intersections, using stabilized UAV high resolution video as input [26]. Work by Köpeczi-Bócz et. al used UAV data to instead train a YOLO detection architecture to regress on the homography projection of BEV processed OBB physical location on the road surface, onto a roadside fixed camera image, that encompasses the the same area as the overhead drone within the intersection, but from a terrestrial vantage point. After training, the object detection locates a predicted 3D oriented bounding box projected within the camera, which can then be expressed in world coordinates [27]. Li and Wu (2022) released a comprehensive BEV drone database of 7,133 traffic images, that contains varied environmental conditions, and a proposed image registration, preprocessing and detection framework using YOLOv5 with oriented bounding box (OBB) detection, and an enhanced object tracking algorithm Although accurate methodologies have been proposed to characterize traffic road user movement behaviors, the approaches still can be constrained by weather conditions, battery life, and other possible flying restrictions, making its broader use as a reliable, around-the-clock ‘roadside’ detector still impractical.

Like Morteza et. al. (2025), we employ a single fisheye camera sensor for an unsignalized arterial roundabout intersection through center island sensor placement.

3 Urban Roundabout Test Site

A study site of an urban arterial 2-lane roundabout intersection was selected in the city of Richfield, a first-tier suburb located on the southern boarder of Minneapolis, MN. The intersection, at 66th and Portland Ave., is served by 2 metro transit operated bus lines. A regional Bus Rapid Transit line (the D Line) runs north/south-bound along Portland Ave. and through downtown Minneapolis, and the Route 515 runs east/west-bound and West along 66th street, both terminating at Mall of America—a popular shopping, entertainment, and tourist destination,. A recreational park, activity center, and water park are located immediately North and East of the intersection. The roadside sensor deployment consists of a 35 foot crank-down traffic camera pole (Millerbernd, Winsted, MN), with a secured equipment cabinet, supplied with 110/120AC dedicated circuit. At the request of the city, the pole was painted black to match surrounding infrastructure to further reduce conspicuity.

A Panasonic outdoor 2992x2992 pixel fisheye camera was deployed at the top of the pole, and offset four feet horizontally from the pole itself to reduce occlusion effects of the pole. The camera is capable of capturing up to 30 frames per second at the full resolution. Care was taken to position the camera such that the pole occlusion does not interfere with the viewpoints for any of the 4 pedestrian crosswalks and roadway legs. The design choice to deploy a single fisheye camera was motivated by several factors. First, potentially a single fisheye camera can monitor a broad area of the intersection, reducing the number of more traditional narrower field of view cameras that would need to be deployed [28], as well as reducing deployment complexities and cost. For near-field monitoring, such as pedestrian detection in crowded urban spaces and traffic areas, fisheye cameras provide more context than narrow FOV cameras [29]



Fig. 3.1: Urban arterial roundabout camera sensor deployment, Richfield MN

The pole cabinet houses a managed switch with selectable PoE 802.at/af, and a Jetson Orin AGX edge computer for processing and detection. The edge computer and camera are powered and controlled through an addressable network A/C power-reboot device also secured within the cabinet. To provide video and data access, a 5G router with secured 802.11 WiFi Access Point (MAX BR, Peplink, Mountain View, CA) (Fig. 3.2) was installed within an integrated antenna enclosure at the top of the pole.

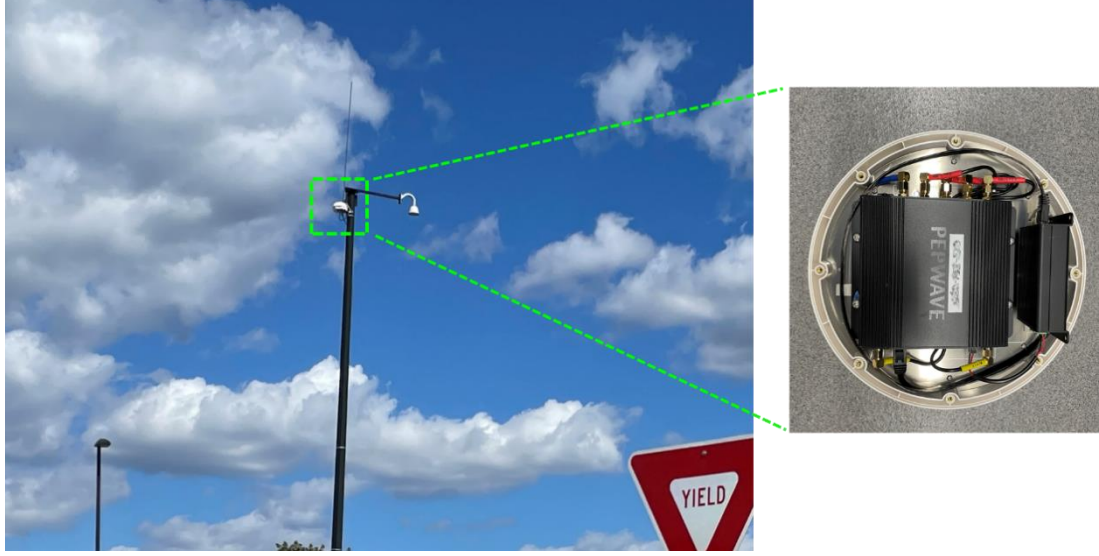


Fig. 3.2: 5G Router set up for sensor and edge computer communication

Full resolution (2992x2992) video at 15 frames per second was recorded during morning from 6:00AM – 10:00AM, mid-day 12:00-1:30PM, and afternoon 3:00PM – 6:00PM peak periods, in 20 to 30 minute segments, capturing different traffic and environmental conditions.

4 Detection and Tracking Methodology

As summarized in the previous chapter, the roadside uses a single camera sensor for estimating road user traffic movements instead of fusing observations from a multiplicity of deployed camera sensors. Although high spatial accuracy has been shown by fusing data together from multiple sensors stationed in different locations, the deployment is more complex to manage and implement. Besides creating additional points of failure, precise corrections for data synchronization, extrinsic and intrinsic uncertainties between the camera sensors must be considered. However, our single camera sensor approach introduces other sensing and detection challenges. Camera image reprojection and calibration, and road user detection and tracking methods are interdependent. Machine learning based models have been trained on images that preserve real-world proportions using images with small or negligible distortion. An overview of calibration and the Deep Learning (DL) detection and tracking models, and the image representations that were explored and tested to address these challenges are summarized below.

4.1 Camera Calibration

Fisheye cameras offer the capability to enhance full scene situational awareness. However, they introduce significant projective image distortion that can affect object detection accuracy and reliability. DL convolutional neural network (CNN) detection model architectures assume object appearance translation invariance within the image for robust performance. This condition is preserved with low distortion, narrow field of view images, where straight lines are preserved. However high distortion images generated by fisheye cameras do not conform to this principle, and the model must therefore learn multiple appearance representations of the object that is dependent on its location within the image [30]. This can, in part, be mitigated by removing the barrel distortion artifact. Referring to Fig. 4.1, calibration methods based on flat-surface pinhole projection model, $r = f \cdot \tan(\theta)$ are invalid due to the projection divergence singularity as $\theta \rightarrow \pm \pi/2$.

4.1.1 Single image formation and calibration

A common fisheye ray reprojection model is the rectilinear model (Fig. 4.1). Under this model assumption the projection is proportional to a spherical surface point location relative to the optical center. The model, summarized in the equations (4.1) - (4.3), is based on pure radial distortion corrections that were shown to compensate for a majority of the distortion artifact from fisheye cameras [31].

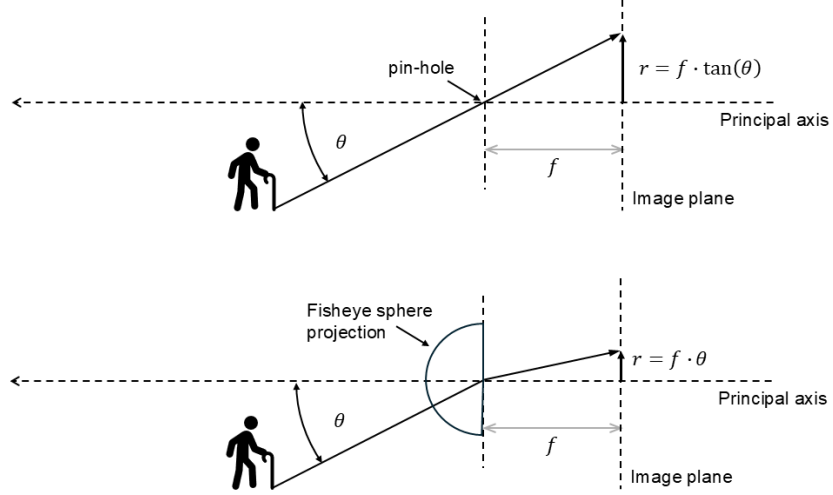


Fig. 4.1: Pin-hole (top) and fisheye rectilinear (bottom) camera image projection models.

$$r' = \theta(1 + d_1\theta^2 + d_2\theta^4 + d_3\theta^6 + d_4\theta^8) \quad (4.1)$$

The radial correction, r' yields the corrected (u, v) camera coordinates lying on the image, normalized within the range $[-1, +1]$ by:

$$\begin{aligned} u &= r' \cos(\varphi) \\ v &= r' \sin(\varphi) \end{aligned} \quad (4.2)$$

where φ is the projected image plane azimuth angle.

In addition to solving for the radial distortion parameters, $d_i, i = 1..4$, the affine transform 3×3 intrinsic camera matrix, $\mathbf{K}$, is also estimated in order to transform the normalized $[-1, 1]$ (u, v) coordinates back into (x, y) $W \times H$ image pixel coordinates ($W = H$). In accordance with the upper left corner of the image as the origin, and the sensor physical size $= w \times h$, ($w = h$), with an estimated focal length, $\|f\|$ and principal axis coordinates (c_x, c_y) in units of pixels, the transformation of (u, v) into (x, y) with $\mathbf{K}$, is defined as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x \cdot \frac{W}{w} & 0 & c_x \\ 0 & f_y \cdot \frac{H}{h} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4.3)$$

The image rectification calibration procedure for the model was performed in the laboratory, using the OpenCV sub-pixel level corner point extraction of a 2x2 inch² 8x8 calibration grid object from a set of full

2992x2992 resolution captured video images to create an aggregated set of image pixel-3D object point correspondences. The fisheye model parameter estimation was then carried out using the OpenCV library implementation [32]. The non-linear intrinsic and rigid body extrinsic camera parameter estimation resulted in an RMS reprojection error of 1.2 pixels.

4.1.2 Fisheye multi-image field calibration

The multi-image rectification approach divided the fisheye image into $j = [1, 2, \dots, 8]$ overlapping ‘viewpoint’ images. The goal was to leverage pre-trained YOLOv8 models to facilitate effective transfer learning over the complete operational field of view of the intersection. We surmised the object detection performance would benefit from this approach because each image could be positioned to cover smaller, more detailed regions with more consistent appearance characteristics of objects across the images. The images are constructed by first transforming the fisheye image through a set of prescribed affine rotations and then extracting $W_j \times H_j$, $W_j > H_j$, rectangular regions of interest (ROI) images. A 4-point homography was then applied to scale each $W_j \times H_j$ image into 640x480 sized image regions to match the input image size of the YOLOv8 pre-trained models. The OpenCV warpAffine and perspectiveTransform functions were used for this step (Fig. 4.3).

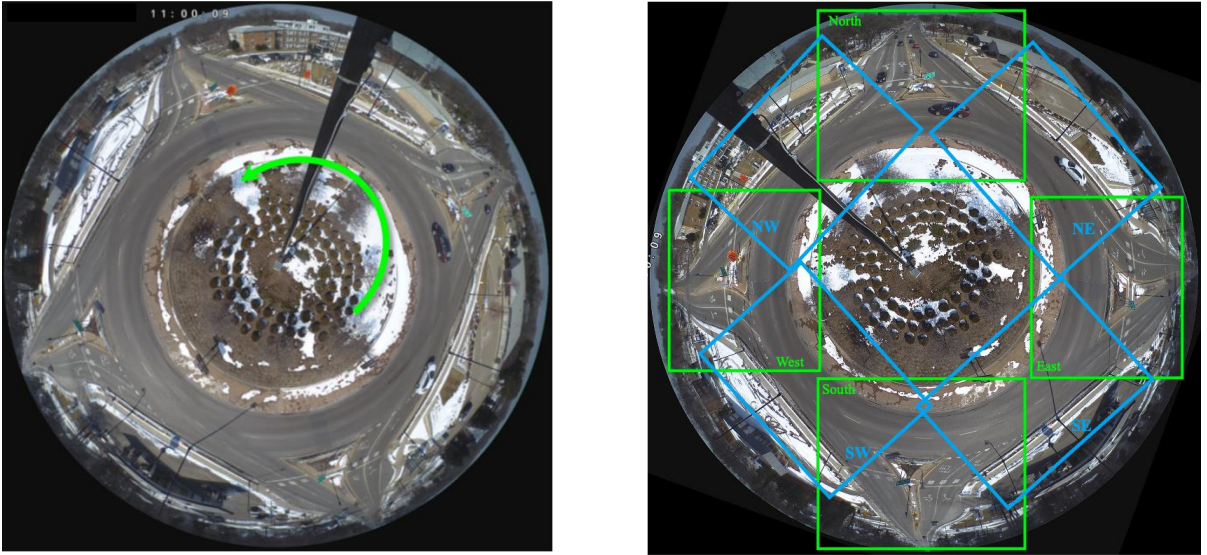


Fig. 4.2: Divided images directionally aligned. Note the rotation increments were irregular due to the offset of the camera center from the center of the intersection.

To further rectify the image to remove fisheye distortion artifacts, first-order radial intrinsic distortion calibration parameters were initially estimated from Tsai’s technique using 11 Google Earth ($X_w, Y_w, Z_w \approx 0$) road mark point and image coordinates correspondences in each of the 8 images [33]. The radial distortion coefficients, $k_1 = -0.07, k_2 = 0.005$, for the images were then obtained through adjustment with the OpenCV library (Fig. 4.3).



Fig. 4.3: Input image (left) and the rectified image calibration result.(right)

4.2 Deep Learning object detection models

We focused on fine tuning recent YOLO (You Only Look Once) single shot detection models since they have demonstrated robust real-time detection performance for a myriad of applications and are available in multiple distilled versions to address accuracy-inference computational performance trade-offs [34]. The models investigated herein can be run in real-time (suitable for roadside BSM broadcasting at 10 Hz) on widely available GPU low-cost edge processors, such as the one deployed for this study.

The YOLO architecture implicitly captures global features for localized detection. The first YOLO model architectures divided an RGB image into $S \times S$ grid of square cells that contain trained feature map locations, and assign ‘responsibilities’ (e.g. contain an object center bounding box location) for the detection and classification of objects within each cell. Introduced in 2016, early versions of YOLO models passed the grid cell channels into 24 convolutional layers followed by 2 fully connected layers to regress on class assignment and axis aligned bounding box locations and sizes [35]. The trained output of YOLO is then a $S \times S \times M$ tensor where the elements of M contains a list of bounding box location and sizes (relative to each cell), and their associated class and object probabilities.

Although the accuracy and computational performance were promising for ‘typical’ camera perspective images, the initial architectures with fixed grid-cell based detection were unsuitable for fisheye perspective imagery. The shallow CNN architecture cannot capture global spatial dependencies and nonlinear, high distortions that characterize fisheye cameras, and performed poorly on varying object aspect ratios—which is further exacerbated with the fisheye camera. These constraints lead to poor detection accuracy, especially for small appearance or peripheral objects within the fisheye camera image.

Several significant architectural changes to YOLO architecture have in theory, addressed the aforementioned shortcomings. An early version enhancement to improve object detection robustness, added a feature pyramid network (FPN) to perform detection simultaneously on geometrically decreasing cell sizes that double in size, and picking the best pyramid level having the strongest detection (Fig. 4.4). Before version 5 each cell determined if it contained a potential object’s center, based on the learned features within it and then tested a set predefined anchor box shapes to bound the object [36]. However, due to using pre-defined anchor box shapes to estimate cell center locations, the performance still suffered when true objects varied significantly in scale and shape within the image. To better address real-world object scale and shape variability, as well as handle RGB image sizes that are more akin to camera image aspect ratios, the YOLO architecture regresses separately on bounding box centers and shapes.

4.2.1 YOLOv8

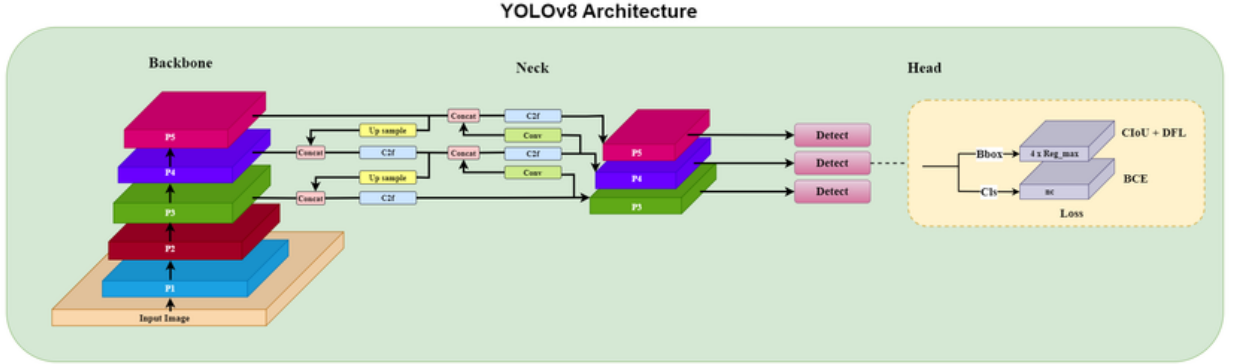


Fig. 4.4: YOLOv8 architecture featuring C2f backbone, PAN-FPN hybrid neck, and anchor-free detection head. Adapted from [37]

From YOLOv4 and on, the architecture was changed to follow a more standard practice of dividing the DL models into a backbone, neck and head. Specifically the backbone of YOLOv5+ is Darknet53 which was a new network architecture aimed at feature extraction with small convolution filter windows and residual connections [38].

YOLOv8 is built upon YOLOv5 with multiple architectural changes and improvements to the past version [37]. Due to the smaller size, it became instantly one of the most accessible and popular single-shot detectors. YOLOv8 represents a major design shift toward simplicity, speed, and flexibility [39]. It employs a lightweight C2f backbone, where the C2f (Cross-Stage Partial with two fusion layers) module splits features into two branches, applies bottleneck transformations to one branch, and then concatenates them to reuse intermediate representations efficiently. This design improves feature flow and representational capacity while keeping the parameter count low. A hybrid Path Aggregation Network–Feature Pyramid Network (PAN–FPN) neck merges multi-scale features more effectively than earlier architectures, enhancing the detection of small and peripheral objects. The key innovation is its anchor-free decoupled detection head, which independently predicts object probability, classification, and object centers and box dimensions instead of relying on predefined anchor boxes. This makes YOLOv8 particularly robust under fisheye distortions, where object shapes and sizes vary nonlinearly across the image plane. The network also incorporates Task-Aligned Assignment (TAL) for better balancing of classification and localization during training.

In the output layer of YOLOv8, the sigmoid function is used as the activation function for the object probability score, representing the probability that the bounding box contains an object. It uses the SoftMax function for the class probabilities, representing the objects' probabilities belonging to each possible class [39].

The YOLO8 loss function in equation (4.4), expresses the necessity to weigh the number of actual objects in an image sample, with a bounding box prediction loss, a class label loss, and a loss function meant to compensate for class imbalances.

$$\mathcal{L}(\theta) = \frac{\lambda_{\text{box}}}{N_{\text{pos}}} \mathcal{L}_{\text{box}}(\theta) + \frac{\lambda_{\text{cls}}}{N_{\text{pos}}} \mathcal{L}_{\text{cls}}(\theta) + \frac{\lambda_{\text{dfl}}}{N_{\text{pos}}} \mathcal{L}_{\text{dfl}}(\theta) + \phi \|\theta\|_2^2 \quad (4.4)$$

In (4.4) N_{pos} is the total number of grid cells containing an object, while the hyperparameters λ_{box} , λ_{cls} , and λ_{dfl} are the bounding box loss, multiclass loss, and focal loss weighting factors. This study set them to experimentally derived values of $\lambda_{\text{box}} = 7.5$ for bounding box loss, $\lambda_{\text{cls}} = 0.5$ for class loss, and the *focal loss*

function $\lambda_{\text{dfl}} = 1.5$ that were used for the trained, frozen models. The focal loss function was added to handle class imbalances, which was an issue with our datasets due to significantly more road vehicle traffic than pedestrian traffic events within the videos, as well as the occurrence of object occlusion and smaller sizes, as their physical distance increases from the camera sensor [40]. The full training loss function (4.5), shows YOLO8 loss functions that reflect class imbalances and object sizes that could bias training results. More specifically for the bounding box loss function, the complete IoU between ground truth and predicted bounding boxes are incorporated. Multi-class cross entropy loss is used for classification errors.

$$\begin{aligned} \mathcal{L} = & \frac{\lambda_{\text{box}}}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{c_{x,y}^*} \left[1 - q_{x,y} + \frac{\|b_{x,y} - \hat{b}_{x,y}\|_2^2}{\rho^2} + \alpha_{x,y} v_{x,y} \right] \\ & + \frac{\lambda_{\text{cls}}}{N_{\text{pos}}} \sum_{x,y} \sum_{c \in \text{classes}} y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c) \\ & + \frac{\lambda_{\text{dfl}}}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{c_{x,y}^*} [-(q_{(x,y)+1} - q_{x,y}) \log(\hat{q}_{x,y}) + (q_{x,y} - q_{(x,y)-1}) \log(\hat{q}_{(x,y)+1})] \end{aligned} \quad (4.5)$$

Where $v_{x,y} = \frac{4}{\pi^2} \left(\arctan\left(\frac{w_{x,y}}{h_{x,y}}\right) - \arctan\left(\frac{\hat{w}_{x,y}}{\hat{h}_{x,y}}\right) \right)^2$, $\hat{y}_c = \sigma(\cdot)$, and $\alpha_{x,y} = \frac{v}{1 - q_{x,y}}$, and $q_{x,y}$ is the Intersection over Union (IoU) value of the predicted vs. ground truth labeled object bounding box value.. $b_{x,y}$ is the ground truth bounding box center location expressed in normalized cell coordinates, $w_{x,y}$ and $h_{x,y}$ are the bounding box width and height respectively, y_c is the ground truth class label for each grid cell (x, y) , $q_{(x,y)\pm 1}$ is the nearest left and right Intersection over Union (IoU) values between labeled and predicted bounding boxes for a given class. To normalize size and shape similarities between ground truth and predicted bounding boxes, ρ^2 is the computed diagonal squared length that encompasses the ground truth and predicted bounding boxes, and $v_{x,y}$ is a ‘normalized shape similarity’ measure that is invariant to bounding box size. The output tensor of YOLOv8 for this study is a $1 \text{ image} \times (4 + C \text{ classes}) \times 8400 \text{ bounding boxes}$ tensor. All overlapping bounding box IoU values, $q_{x,y}$, are retained by normalizing the values through the probability softmax calculation $\hat{q}_{x,y} = \text{softmax}(\cdot)$.

Additional post-processing of the output uses greedy Non-Maximum Suppression (NMS) to iteratively eliminate and separate redundant, overlapping bounding boxes for the same object. Note that the threshold for maximum IoU bounding box elimination is application dependent. This project, the NMS IoU and confidence parameters of 0.75 and 0.25 respectively, are similar values used to obtain optimal performance across several datasets used to originally train the models.

4.2.2 YOLOv11

YOLOv11 introduces several architectural improvements over YOLOv8, focusing on efficiency, feature aggregation, and spatial attention. The backbone replaces the C2f (Cross-Stage Partial with two fusion layers) block with the lighter and faster C3K2 (Cross Stage Partial with kernel size 2) reusable modules, improving feature extraction while reducing computational cost. The SPPF (Spatial Pyramid Pooling - Fast) block is retained for multi-scale receptive field expansion, and a new C2PSA (Convolutional block with Parallel Spatial Attention) module enhances the model’s ability to focus on important spatial regions.

In the neck, YOLOv11 again uses C3K2 blocks to improve multi-scale feature fusion, while C2PSA introduces stronger spatial attention than in YOLOv8. The head employs additional C3K2 and CBS (Conv-BatchNorm-SiLU) layers to refine features before final prediction. The Detect layer outputs the tensor containing bounding boxes, objectness scores, and class probabilities. Overall, YOLOv11 delivers improved accuracy and

speed through lightweight bottlenecks, enhanced spatial attention, and efficient multi-scale feature processing [41].

In summary, anchor-free detection in both YOLOv8 and YOLOv11 removes the need for predefined anchor priors, in theory allowing more flexible localization under curved fisheye projections. Modules such as C3K2 and C2PSA in YOLOv11 improve feature extraction and spatial attention, giving higher accuracy for dense, wide-field traffic scenes (Table 4.1)

Table 4.1 Comparison of YOLOv8 and YOLOv11 architectures and their relevance to fisheye object detection

Version	Key Characteristics and Fisheye Relevance
YOLOv8 [37], [40]	C2f backbone, PAN-FPN hybrid neck, and anchor-free detection. Strong robustness to object scale variation and curved projections in fisheye imagery
YOLOv11 [41]	Lightweight C3K2 blocks, SPPF module, and C2PSA spatial attention block. Improved multi-scale fusion and spatial focus enhance accuracy for small, peripheral, and distorted objects commonly seen in fisheye views

5 Object Detection and Tracking Results

The detection training and performance of the two image representation methods are given below, followed by a summary of the object tracking performance. The overarching comparative performance metrics used in this study are mean average precision (mAP) scores and inter-class Confusion matrices, and inference time. The Confusion Matrix illustrate the inter-class confusion by comparing predicted and true labels, in order to identify systematic misclassifications that affect Recall (for example, a pedestrian object confused with the background). The mAP score is the area under the Precision-Recall curve computed across confidence thresholds within a single, or multiple object classes, where the threshold values correspond to bounding box IoU of predicted vs. ground truth labeled data. The mAP scores were summed for scores at 50% IoU (mAP@50) and between 50% to 95% IoU (mAP@50-95). The latter is a more stringent metric since it considers performance across a range of IoU thresholds. Note that precision, ($Prec = \frac{TP}{TP+FP}$), alludes to “Did it miss anything?”, while recall, ($Rec = \frac{TP}{TP+FN}$), alludes to “did it get it right?”. The F1-score, the harmonic mean of Precision and Recall, indicates a ‘balanced’ measure of detection accuracy that accounts for both missing (FN) and incorrect (FP) object detection.

5.1 YOLO Omniview Multi-Image Detection

The omni-view multi-image detection was tested and evaluated first, using YOLOv8. As eluded in the previous chapter, the ensemble of the multiple image (ROI) views would, in principle, provide better object detection because smaller appearance objects like pedestrians in the original image would take up a larger proportion within the ROI image, thereby providing more discriminative feature details. Also, in theory, the transfer learning could be done more efficiently because the features would be more similar to the original training data objects than using the heavily distorted object appearances in the fisheye image.

YOLOv8 is available in 5 different sizes: nano, small, medium, large and extra-large. The nano, small, medium and large models have 3,157,200, 11,151,080, 25,879,480 and 43,660,680 parameters and 129, 225, 295 and 365 layers respectively. As indicated in [38] and [34], there is a notable mAP score improvement between the nano and small, and small and medium, sized models, compared to medium to large, and large to extra-large, models, while speed performance reductions are more pronounced at the highest accuracy, large and x-large models. As an example, using an initial training and validation training sets (described below) with the medium model yielded across 148 validation images and 818 instances resulted in overall mAP@50 score of 0.843 and mAP@50-95 score of 0.657 but required 330% more processing time (initial + inference + postprocess times) than the small model. The postprocess times (which includes greedy NMS) was the largest contributor at 18ms vs 8.3 ms on a Nvidia Titan X GPU.

5.1.1 YOLOv8 Multi-Image Model Training

The pre-trained YOLOv8 small model was thus chosen for this study due to the increase in speed over the medium and large models with the goal of achieving acceptable detection performance through judicious, reduced class selection for model re-training. The road user traffic classes used to fine-tune train the model were:

1. *Car*
2. *Pedestrian*
3. *Truck*
4. *Pickup-Truck*
5. *Bus*
6. *Van,*
7. *Motorcycle*

The dataset consisted of 1225 randomly extracted images from multiple stored video recordings with 1095 images containing cars. Preliminary experiments with the 7-class categories, using the frozen, untrained YOLOv8 models, the accuracy mAP scores were within 5%, which also reinforced the decision to focus on the small model. Fewer classes for specific private vehicle types are sufficient, as was done with the YOLOv11 implementation herein, but this categorization yielded the best performance both, for training and validation experiments for this approach.

The Roboflow Annotate for YOLO8 tool was used to create the dataset, using the auto-labeling feature for image axis aligned object detection bounding boxes. This can significantly cut down on the time to create the dataset compared to previous YOLO versions. To increase training samples, five image augmentations were performed: 1) Random mirror flip about the vertical axis, and 2) Random rotations within ± 9 degrees to introduce directional traffic variation 3) adjusted random image intensity shifts by $\pm 20\%$, to emulate lighting condition variations, 4) Random Gaussian image blur up to 2.5 pixels, and 5) random additive noise. These augmentations were specifically selected to emulate effects of the large variation in observed lighting conditions and image quality during each day.

In addition to the general data augmentations, Roboflow Annotate introduced mosaic augmentation in the earlier YOLOv4 models, which was also applied to this dataset. This ‘stitches’ four separate image views together to enhance the ability of the YOLOv8 model to learn objects in new locations, containing more ‘occlusion’ scenarios, and different surrounding pixels (Fig. 5.13). After augmentation, the labeled dataset consisted of 2,151 training samples, 188 validation images, and 70 test images. A similar data augmentation method was also applied for fisheye input detection training.

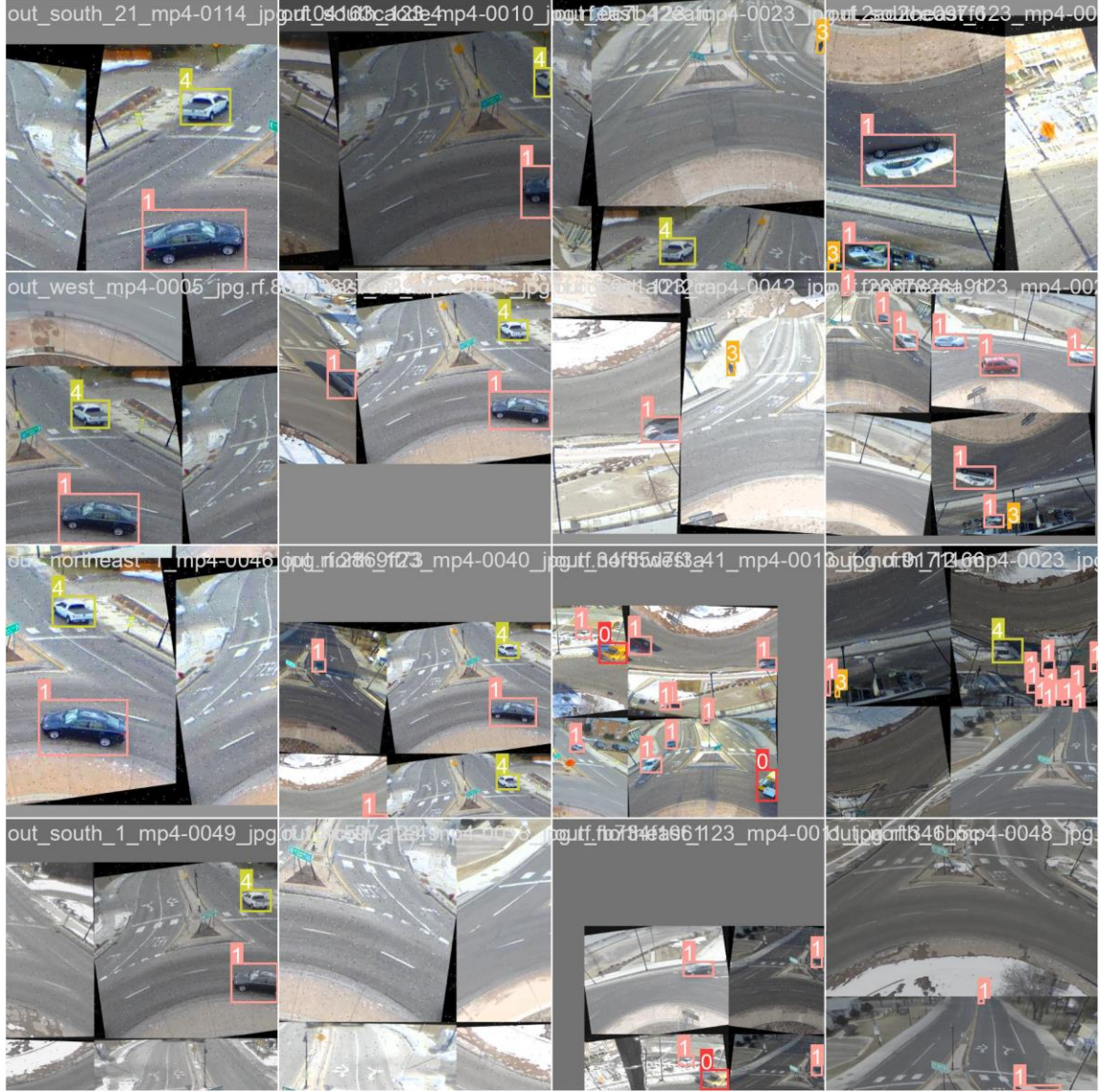


Fig. 5.1: Example of mosaic data augmentation using patches from four of the roundabout training data images.

The chosen parameters for the final model are a batch size of 32, learning rate of .00909 and momentum of 0.9. The model trained for 150 epochs and observed the best performance at epoch 114. Note validation hold out samples flattened out after 150 epochs (Fig. 5.2)

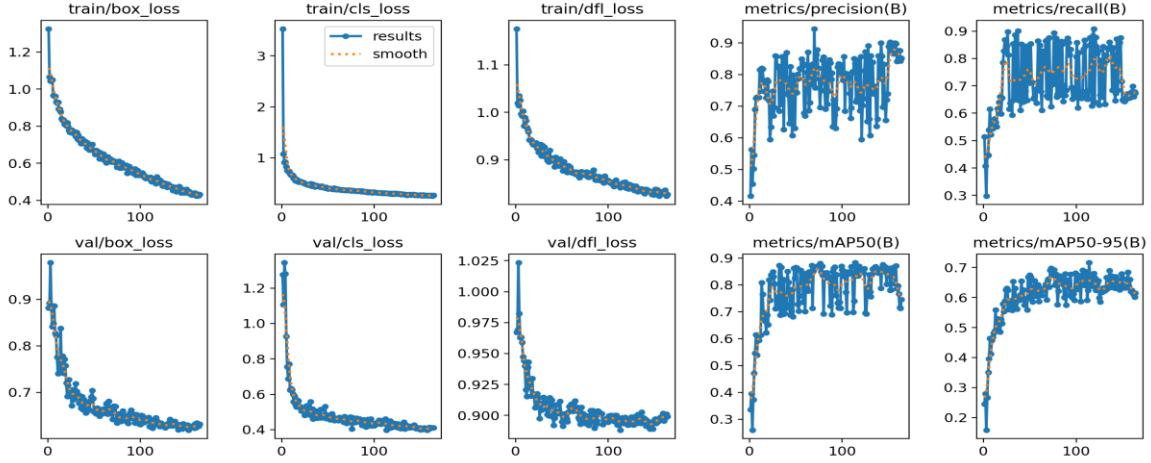


Fig. 5.2: YOLOv8 training results.

5.1.2 YOLOv8 Multi-Image Performance

The final model had a mAP@50 of 0.833 and a mAP50-95 of 0.717. Table 5.1 gives a summary of the validation data statistics and performance metrics. The most significant challenge when training this model approach was the imbalance of classes. Like the training data class composition, the ‘car’ class made up 85% of the total instances in the validation set. For example, the following classes: bus, van, pickup-truck, truck, and pedestrian are in 60-100 images each. This caused the model to not train to identify and detect remaining classes as effectively as the car class, which resulted in large variations in the Precision and Recall performance (Fig. 5.2).

Table 5.1 Class-wise test multi-image performance of YOLOv8

Class	Instances	Precision	Recall	mAP50	mAP50-95
All	736	. 882	. 898	. 911	. 743
Car	566	. 943	. 973	. 977	. 847
Pedestrian	81	. 919	. 877	. 917	. 63
Truck	15	. 785	. 800	. 836	. 709
Pickup-Truck	20	. 898	. 884	. 865	. 782
Bus	32	. 858	. 941	. 966	. 903
Van	21	. 771	. 81	. 818	. 735
Motorcycle	1	1	1	. 995	. 597

The normalized confusion matrix from the validation step is shown in Fig. 5.3, providing insight to Recall performance. The background confusion indicates the contribution of missed detections and accounts for a significant proportion of false negative detections. With the collected video data, there were many cases of very dark, or blurred vehicles that resulted in missed (False Negative) detections. Note ‘motorcycle’ (includes bicycle) class did not have sufficient data for interpretation. As mentioned, the class imbalance meant significantly less labeled pedestrian training and validation data than vehicle data labels. Interestingly, the pedestrian clothing blends in with the road, confusing the detection. Such conditions were exacerbated under low lighting conditions, and the fact under many instances in test videos, pedestrians were still small in scale (Fig.

5.5). However, the detections were reliable when the pedestrians were in the median of the road, which is a critical area for traffic safety (confidence = 0.83).

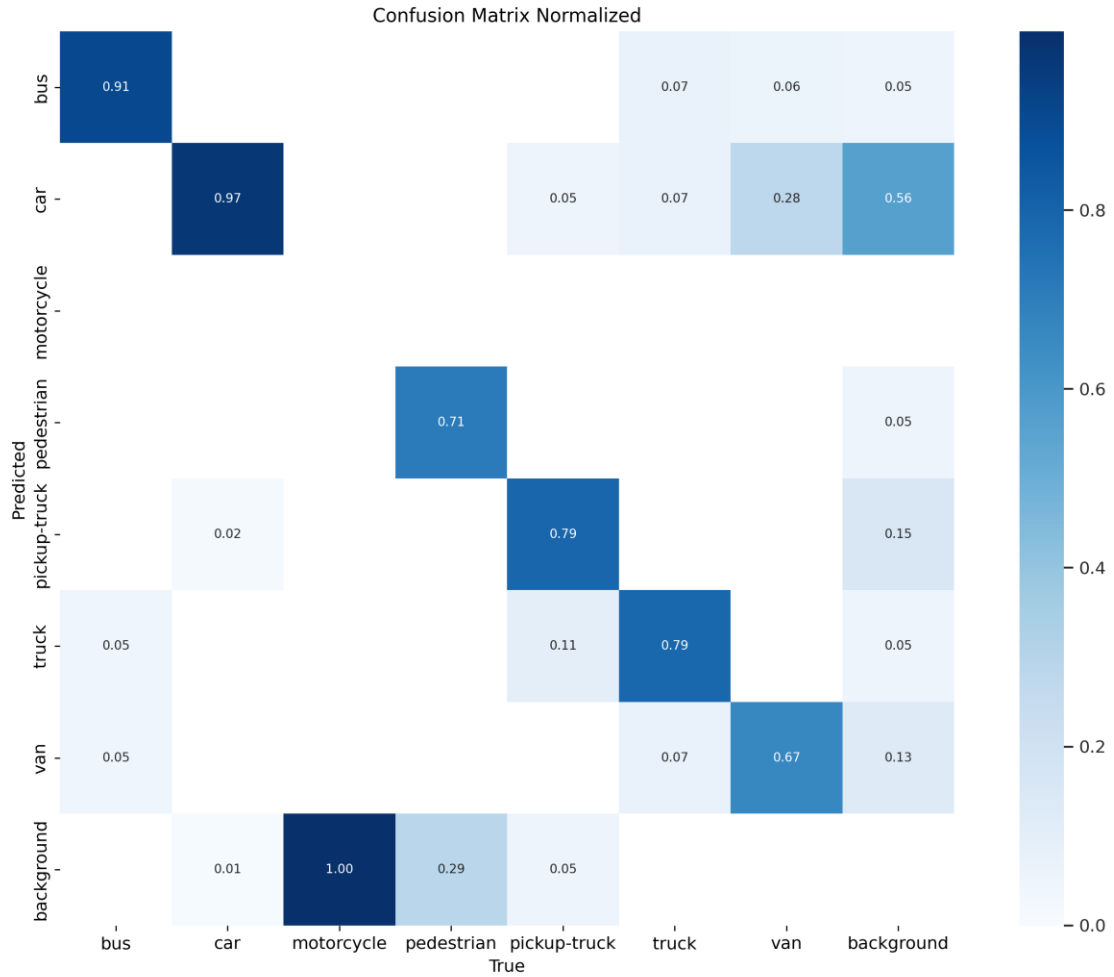


Fig. 5.3: YOLOv8 multi-image detection class confusion matrix

To further highlight the results, the performance of pedestrian detection, occluded and cut-off vehicles are shown in Fig. 5.4. For example, even with intentionally overlapping images vehicles must still be re-identified due to edge occlusion, which the model was observed to handle in most cases, with high class-object confidence scores when at least half the vehicle was within the images. The object detection was robust to occlusion scenarios from vehicle overlap during queuing before and after the roundabout entry yield/stop bar, as well as the foreground artifact arising from the camera pole.



Fig. 5.4: Detection occlusion cases for YOLO8s multi-image detection. TOP: vehicle queuing, foreground occlusions, BOTTOM: vehicle edge detection

The goal for detection computational speed was for each image detection total inference and processing time to be less than 8.33 milliseconds, which would allow detections to be completed within the set video frame rates of 15 fps. The total time averaged 20.74 milliseconds on a Jetson Orin AGX GPU-based edge computing platform including pre- and postprocessing (Table 5.4). Image pre-processing refers to resizing the image and post-processing is associated with the non-maximum suppression (NMS) to prune conflicting bounding boxes described in section 4.2.1. Note that, for future implementations, the frame rate of the most recent model version of the camera can be configured to match the SAE J2735 recommended broadcast minimum of 10 Hz for critical safety applications implementations. The reduced framerate would relax the maximum detection latency over the current implementation by an additional $\left(\frac{1}{10} - \frac{1}{15}\right)/8 \text{ images} \approx 4 \text{ msec}$.

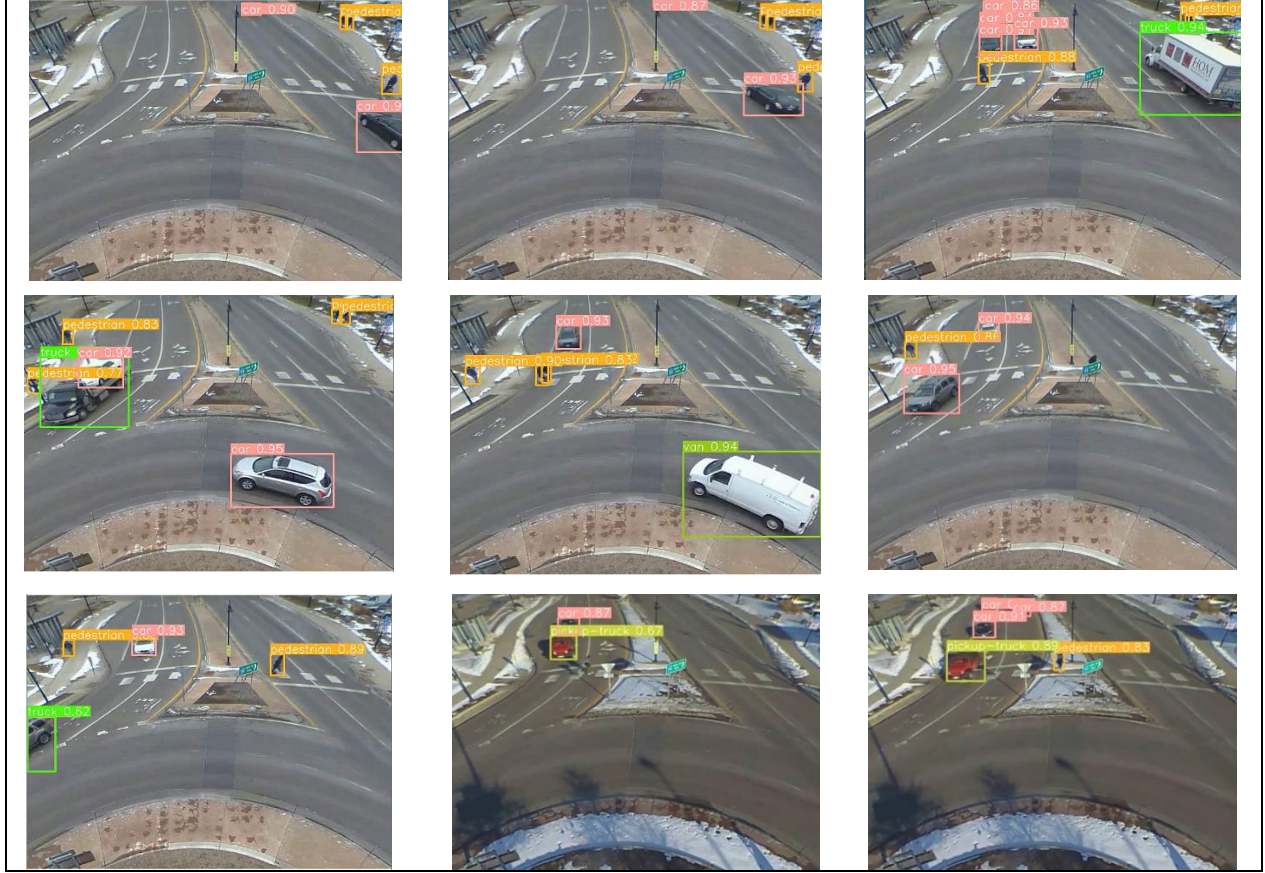


Fig. 5.5: Pedestrian crosswalk detection. TOP: Pedestrian crossing sequence with fail to yield by roundabout exiting vehicles; MIDDLE-Left/BOTTOM: Pedestrian occlusion-based *FN* cases during group crossing sequence from initial merging of pedestrians and roadside sign in refuge island, and BOTTOM-Center/Right: from background confusion.

5.2 YOLO Fisheye Image Detection

As alluded to previously, fisheye cameras introduce technical challenges when used over more traditional narrow field-of-view cameras. The high distortion made axis aligned bounding boxes unsuitable for reliable object detection localization. Objects near the periphery appear heavily stretched which can impact object localization accuracy [30].

There has been recent efforts to develop fisheye-based image labeled datasets that are applicable for object detection directly from unrectified wide panoramic fisheye images. The WoodScape dataset provides approximately 10,000 labeled semantic instances of 40 classes, including vehicles and pedestrians [30]. Another dataset, FishEye8K used 8,000 images with 157,000 labeled bounding boxes across pedestrian, bike, car, bus, and truck classes [28]. However, most of the image data did not closely match the scenario from our deployment or were intended for on-board autonomous vehicle sensing. Training sets from the video recordings using rectified images were thus created to better target the fixed camera sensor deployment environment.

5.2.1 YOLO Fisheye Model Training

Initially, pre-trained oriented bounding box YOLO detection models were tested on the rectified images to visually assess feasibility. The pre-trained YOLOv8 and YOLOv11 nano models were chosen since they have a

significant decrease in detection latencies and memory size footprints over the larger models, making them ideal for edge deployment scenarios such as this application. All runs were performed with the rectified images to train YOLOv8n-OBb and YOLOv11n-OBb (oriented bounding box task heads) under identical training configurations, and with deterministic settings (seed = 0) to ensure reproducibility. This ensures the performance differences between the two models are attributed to architectural variations. The road user traffic classes used to train the pre-trained models were:

1. *Car*
2. *Pedestrian*
3. *Truck*
4. *Bus*

From the videos, 498 images, containing 1,833 labeled oriented bounding boxes were created for training, validation, and testing. The original images were rectified using the calibrated intrinsic distortion model and then rescaled to 640x640 pixels to match the input configuration of the pre-trained models as well as to evaluate the real-time performance on the edge device. The dataset was partitioned into 432 training images (87%), 42 validation images (8%) and 24 test image (5%) sets. The pixel intensities were normalized internally by the YOLO models during training to provide a more consistent illumination scaling across the dataset.

The distribution of annotated objects is dominated by *Car* and *Bus* classes. This imbalance can bias the model toward over-represented categories, reducing recall for the less frequent, *Truck* and *Pedestrian* classes and affecting NMS confidence calibration during training. To mitigate the class imbalance, the training process employed stratified sampling and aggressive data augmentation to improve generalization and balance class representation [42], [43]. Accordingly, offline dataset augmentation expanded the original training set by three times through randomized transformations, including vertical flipping, subtle randomized pixel noise injection (up to 0.1% of the image size), and the opposite, blurring effect with a Gaussian filter (up to 2.5 pixels). The dataset after augmentation increased to 2,000 images, with 1,833 ground-truth class annotated objects. Note that because of the large imbalance of bicycle and motorcycle traffic events, they were not included in the experiments.

Additional data augmentations are performed in-line during model training to further reduce model overfitting. Horizontal image flipping and re-scaling (with a scaling factor up to $\pm 50\%$ the original size), translation ($\pm 10\%$ each direction), Hue-saturation-intensity (HSV) variation, and, like the multi-image detection model, the random mosaic image generation procedure. Randomized erasing of a rectangular image region was also performed with a 40% probability, in order to force the model to learn object identification over broader features rather than highly specific feature patterns (i.e., further reduce overfitting). The training procedure is configured to ‘turn-off’ the mosaic augmentation for the last 10 training epochs to mitigate potential destabilization of learning convergence.

The training was performed for 100 Epochs. The loss functions generally minimized, with mAP scores and Precision/Recall asymptotically reaching maximum values for both models, indicating convergence of model learning with the provided training data. Although not shown, the YOLOv8-OBb showed very similar trends to YOLOv11-OBb (Fig. 5.6).

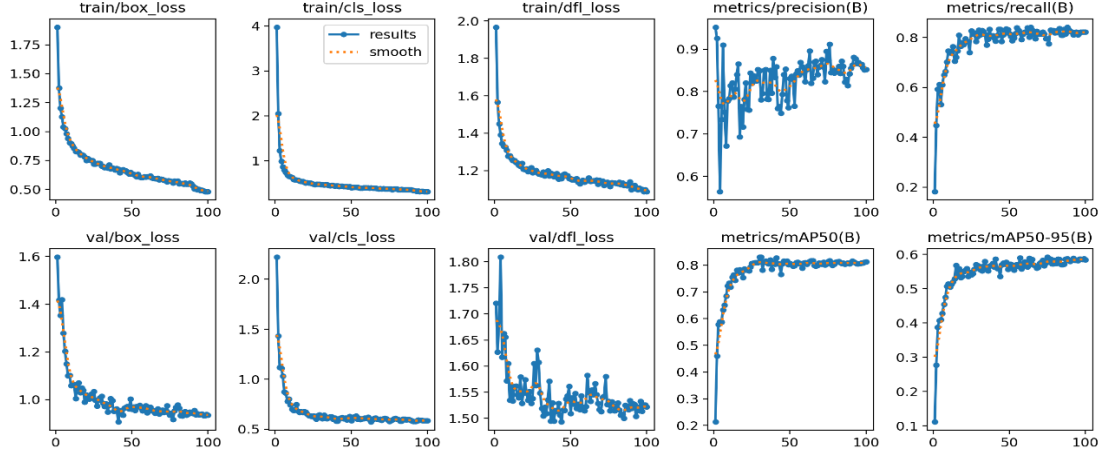


Fig. 5.6: YOLOv11n-OBb training results

5.2.2 YOLO Fisheye Model Performance

Class-wise validation results show that YOLOv11n-OBb achieved higher overall precision, recall, and mAP@50 than YOLOv8n-OBb, while YOLOv8n-OBb yielded a higher overall mAP@50-95 score. Specifically, the YOLOv11n-OBb model achieves a mAP@50 score of 0.832 and an F1 score of 0.83 at a confidence threshold of 0.341. The YOLOv8n-OBb achieves a mAP@50 score of 0.812, with an F1 score of 0.79 at a confidence threshold of 0.33. Note that the F1 curves also indicate performance stability across the range of objectness thresholds; both models indicate relatively flat F1 values across a wide range of confidence thresholds (Fig. 5.7). At the class level, both models performed well on bus and car vehicle detections, whereas pedestrian and truck detection performance remained more challenging due to lower instance counts and greater appearance variability. The confusion matrices in Fig. 5.8 and Fig. 5.9 show that the recall performance with the test data was significantly affected from confusing objects with the background, particularly with trucks and pedestrians (missed object detections). This affected detection recall more so with YOLOv8n-OBb.

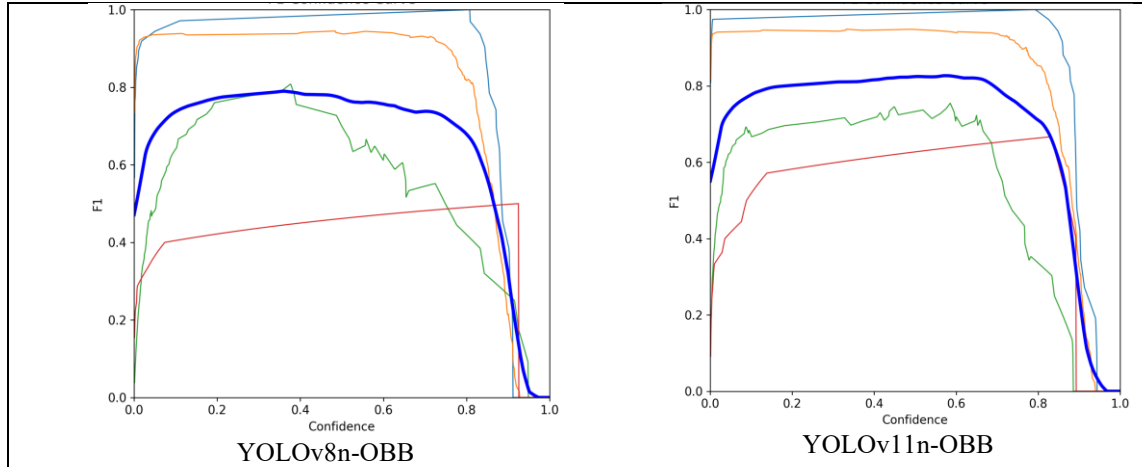


Fig. 5.7: F1-Confidence curves test results.

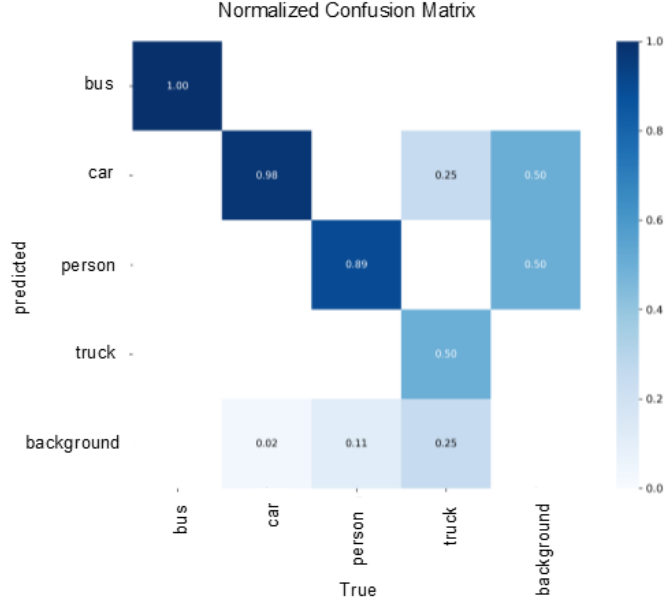


Fig. 5.8: YOLOv11n-OBb test results normalized confusion matrix

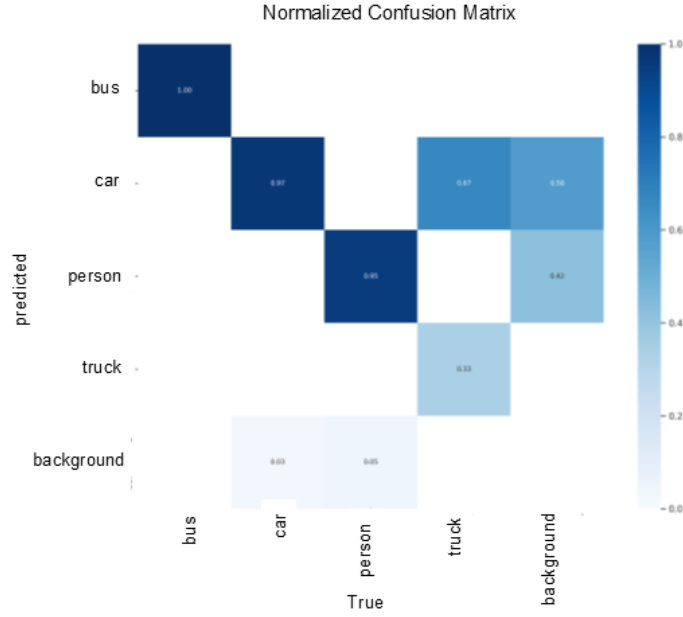


Fig. 5.9: YOLOv8n-OBb test results normalized confusion matrix

Table 5.2: Class-wise test performance for YOLOv8n-OBb

Class	Instances	Precision	Recall	mAP50	mAP50-95
<i>All</i>	221	0.806	0.830	0.827	0.634
<i>Bus</i>	19	0.973	1.000	0.995	0.785
<i>Car</i>	170	0.923	0.976	0.957	0.746
<i>Person</i>	28	0.663	0.842	0.690	0.406
<i>Truck</i>	4	0.667	0.500	0.665	0.598

Table 5.3: Class-wise test performance for YOLOv11n-OB

Class	Instances	Precision	Recall	mAP50	mAP50-95
<i>All</i>	221	0.850	0.840	0.832	0.607
<i>Bus</i>	19	0.968	1.000	0.995	0.811
<i>Car</i>	170	0.932	0.966	0.956	0.726
<i>Person</i>	28	0.718	0.893	0.779	0.380
<i>Truck</i>	4	0.780	0.500	0.597	0.510

Fig. 5.10 through Fig. 5.12 provide a visual comparison of the detection results between models across environmental conditions. The YOLOv11n model demonstrates the higher confidence and fewer missed detections, particularly in the peripheral regions, outside of the roundabout intersection over the varied environmental conditions. The YOLOv11 model shows improved detection confidence and stability to illumination variance and background clutter. Note that the top row within each figure qualitatively demonstrates the performance of the trained models with *unrectified* images. Interestingly, although missed detections are most apparent, the models were still capable of correct objects detections. This supports the earlier notion that the architectures are capable of generalizing distorted global features for object detection, and with additional training with unrectified images, the DL approach could achieve acceptable real-time detection performance, which would theoretically provide a more efficient general object detection for the unrectified top-view fisheye images.

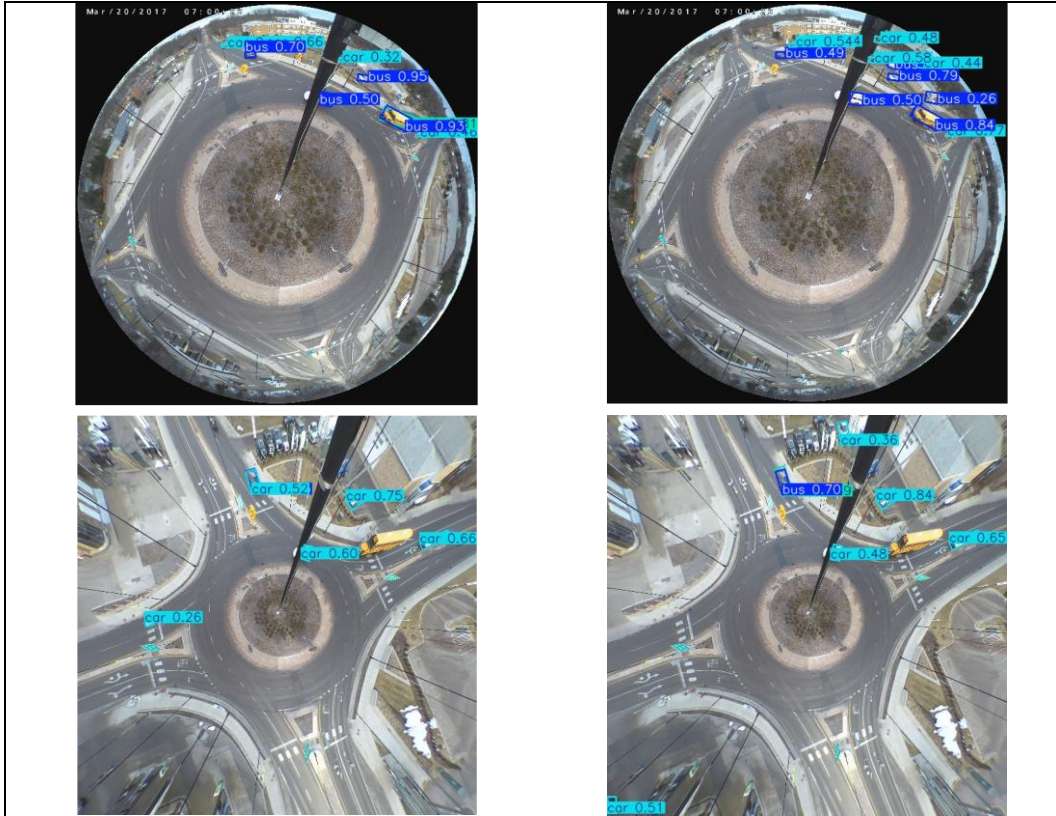


Fig. 5.10: Unrectified (top row) and rectified (bottom row) fisheye images during fair daytime conditions.

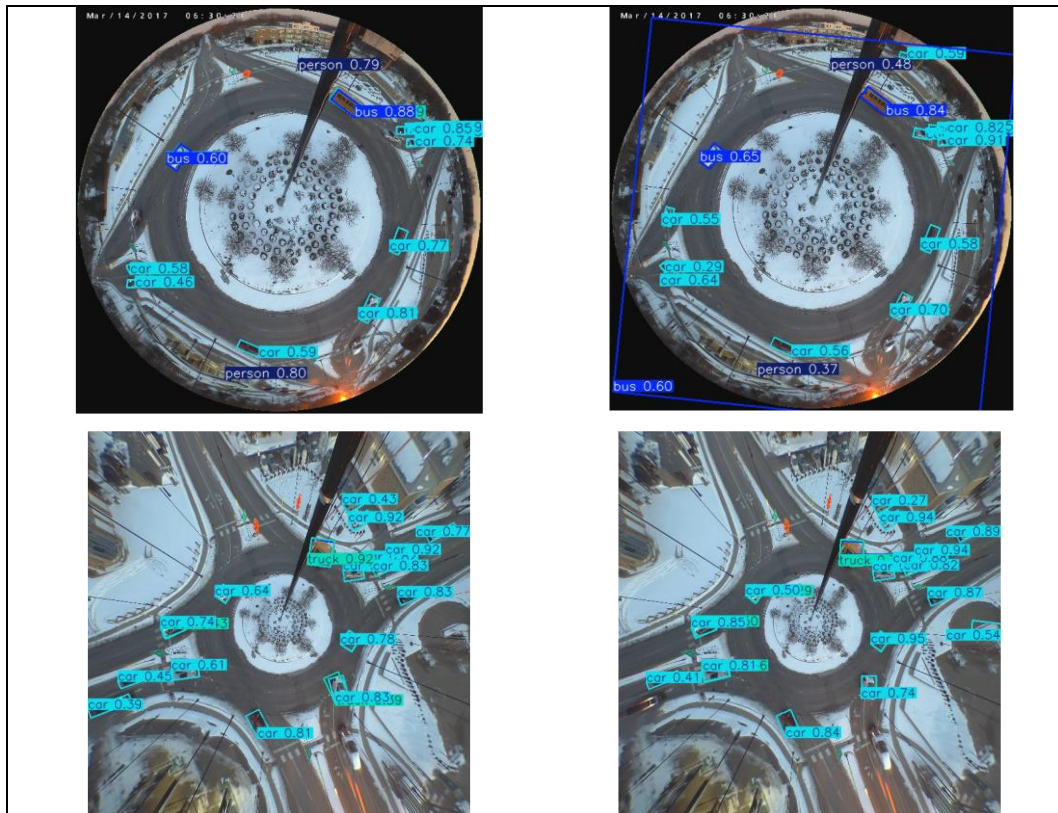


Fig. 5.11: Unrectified (top row) and rectified (bottom row) fisheye images during daytime snowy conditions.

The fisheye model latencies are over 4 times faster than the SAE J2735 BSM target frequency of 10 Hz. However, the multi-image latency may become problematic if only one instance of the model can be run at a time. This may be addressed by instead executing several parallel (at least 6) instances of the forward model detection, queuing multiple constructed images into separate input pipelines for each model instance. This concept could be evaluated in future studies. Note that SAE conformant CV2X wireless broadcast transmission has been reported to contribute an additional latency to receiving devices by 17 to 46 ms [20].

This highlights the importance of evaluating full pipeline latency rather than model inference time alone, particularly for embedded edge deployments. Overall, YOLOv11n-OBB provides improved detection accuracy with minimal additional run-time cost. YOLOv8n-OBB remains a good candidate for latency-critical, or resource-constrained, applications. Although the FP16 weight quantization format has negligible effect on accuracy from other 32 FP formatted structures, further speed optimizations may be possible using an INT-8 (8-bit integer) weight quantization technique but requires specific manual reiterative retraining in order to approach similar performance metrics.

5.4 Object Tracking

Computationally efficient and widely available object tracking algorithms were considered. Q-SORT (Quick-Simple Online and Real-time Tracking), uses a pixel-based Kalman filter for motion prediction and the Hungarian algorithm for object association. The performance suffered when objects were partially occluded by other objects in the scene. ByteTrack [44], indicated better performance under some conditions of object occlusion and is reported to yield comparable performance with state-of-art computationally efficient object tracking methods for human crowds and vehicle heavy traffic scenes [45]. Like the QSORT algorithm, a pixel-based Kalman filter is used for motion prediction. However, rather than a single association step, ByteTrack retains low objectness score bounding boxes from the YOLO output vector in order to complete object associations for overlapping detected objects with the low object confidence scores. The Kalman filter also uses object appearance characteristics (bounding box features).

The SDSM element definition [11], indicates that road user locations are expressed in X,Y positions relative to a road sensor origin (expressed as a latitude, longitude). Therefore, the transformation is formulated that maps image pixel coordinates to such a physical world coordinate location. A multiplicity (94) of WGS84 (*longitude, latitude*) roadway surface features within a 150 foot radius from the center of the intersection, in addition to the camera pole center, were manually digitized from a Google Earth approximated orthographic perspective, and subsequently converted to Minnesota State Plane Easting-Northing Coordinates (X_w, Y_w). The corresponding pixel coordinates were located within a full resolution unrectified fisheye image. The PnP planar surface approximate method was then used to estimate the extrinsic camera $[R|t]$ transformation ($Z_w \cong 0$), using the OpenCV implementation of [46].

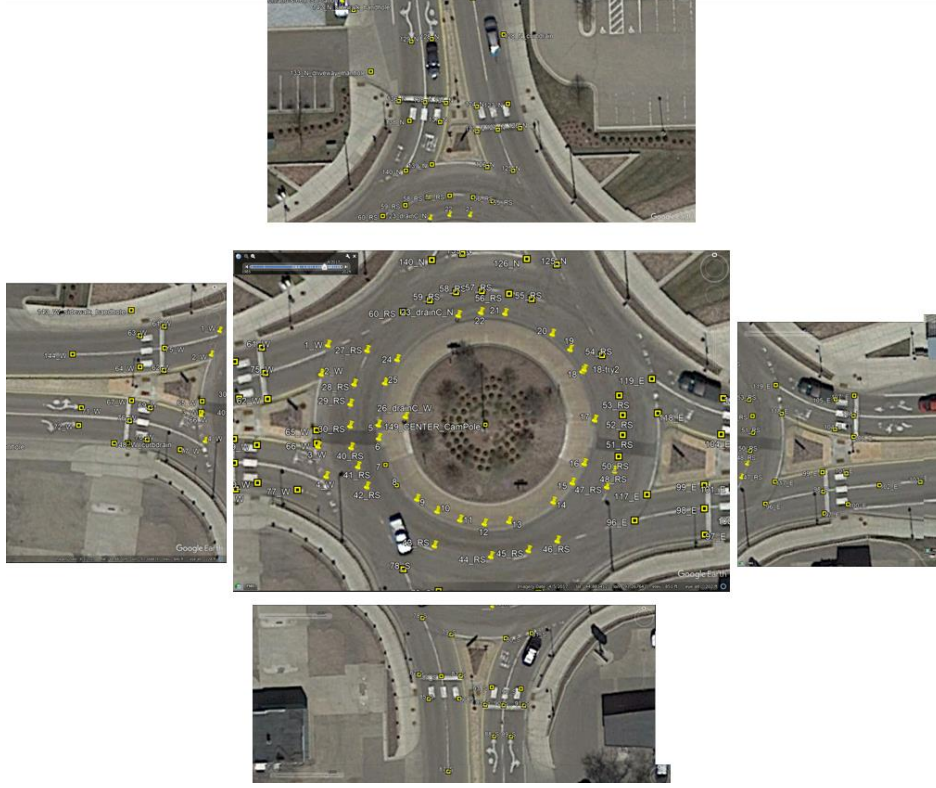


Fig. 5.13: Google Earth view with latitude, longitude calibration points

Occurrences of object track ID becoming ‘lost’ was still observed in some scenarios when objects became partially or fully occluded. For example, during scenarios when multiple vehicles pass behind the camera pole or were occluded by a larger vehicle. To address this, the object tracking in the physical world coordinate space was added improve the robustness of the tracking approach. Tracking corrections done in the physical world coordinates can reduce image perspective, resolution, and scaling artifacts that confound pixel based object tracking and prediction of real-world 3D objects [20], [21]. The object positions in world coordinates were approximated by first computing the (x_c, y_c) pixel centroid of the object oriented bounding box maximum extents (this defines the aligned bounding box) and then back projecting (x_c, y_c) ‘down’ to the road surface, assuming a centroid world coordinate elevation $Z_w \cong \frac{h_c}{2}$ value based on its class type, c . For vehicles, the approach positioned the tracks closer to the center of the lane while traversing the roundabout. A 2nd order A Kalman filter (KF) was then applied to update estimates within the world coordinate system, $(x_w^*, y_w^*, \dot{x}_w^*, \dot{y}_w^*, \ddot{x}_w^*, \ddot{y}_w^*)$. Diagonal starting values for the process, state, and measurement noise covariance noise matrices, $\mathbf{Q}, \mathbf{P}, \mathbf{R}$, were set empirically by observing the filtering convergence to smooth trajectories based on the level of trust between the measurements and state estimates. When a track ID becomes lost, it is buffered (up to 2 seconds, before it is discarded) with the KF predictions. The distances of any initial positions (X_w, Y_w) of a new track are then compared to the lost tracks predicted positions, (x'_w, y'_w) to determine the minimum distance within a given tolerance (empirically set to 25 ft for vehicles, and 10 ft for pedestrians). In the current formulation, the assignment test procedure is applied for object tracked from within the roundabout intersection roadway, under the assumption that it would not be possible for a vehicle or person to ‘disappear’ within the Roundabout intersection area. Speed s , and heading ψ (with the convention of 0 degrees=North), are estimated from the KF velocity state predictions (equations (5.1), (5.2)).

$$s = \sqrt{[\dot{x}_w^*]^2 + [\dot{y}_w^*]^2} \quad (5.1)$$

$$\psi \cong \tan^{-1} \frac{\dot{x}_w^*}{\dot{y}_w^*} \quad (5.2)$$

Figures Fig. 5.14 - Fig. 5.17 visually summarize typical predicted object (vehicle) trajectories, with labeled spatial-temporal correspondences between heading, speed, and position. Examples of ReID tracks are highlighted in Fig. 5.14 and Fig. 5.15. Generally vehicle speeds within the circulatory portion reduce to roughly 20 miles per hour, and, begin to accelerate back to higher main-line speeds before the exit leg. The red lines represent what otherwise would have been a *FP* ReID track occurrence for the same object. Since the heading measurement is a point-wise estimate, the estimate accuracy is affected by the speed of the vehicle due to measurement and state update uncertainties, and the fact that it approaches a singularity if the vehicle stops. In this case, when $s \lesssim 3 \text{ mph} = 4.4 \text{ ft./sec.}$, the heading estimate becomes sporadic (Fig. 5.17). The latter case can be mitigated for example, by introducing criteria which would retain previous headings estimated when the vehicle was traveling at speeds above the threshold. Otherwise, vehicular maneuvers appear to be well represented by the detection system. Note that the units used for calculation and discussion purposes are in US ft. but this can be converted to meters for SDSM message compliance, or transformed efficiently to BSM message compliant geodetic coordinates [47].

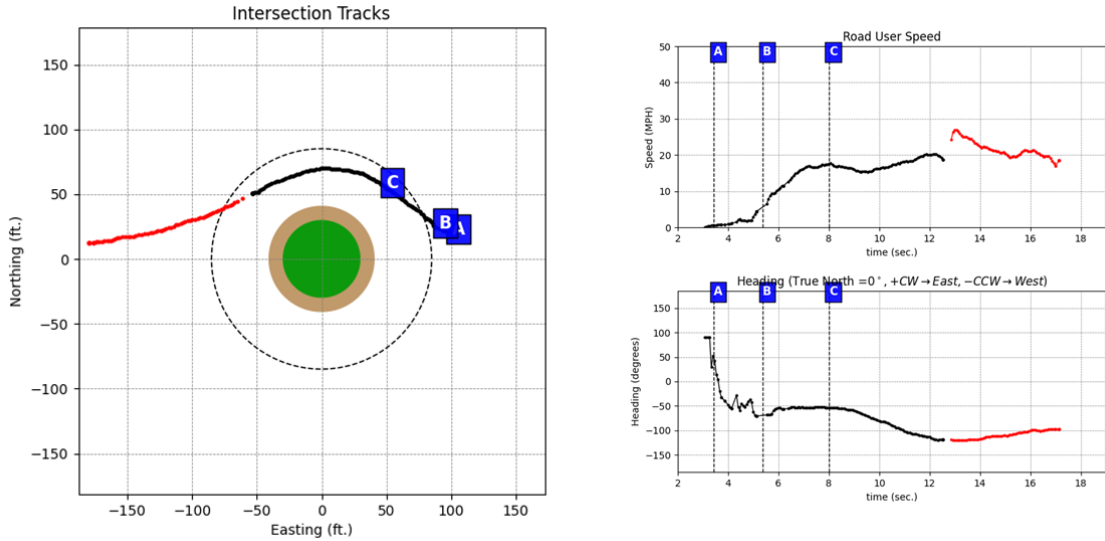
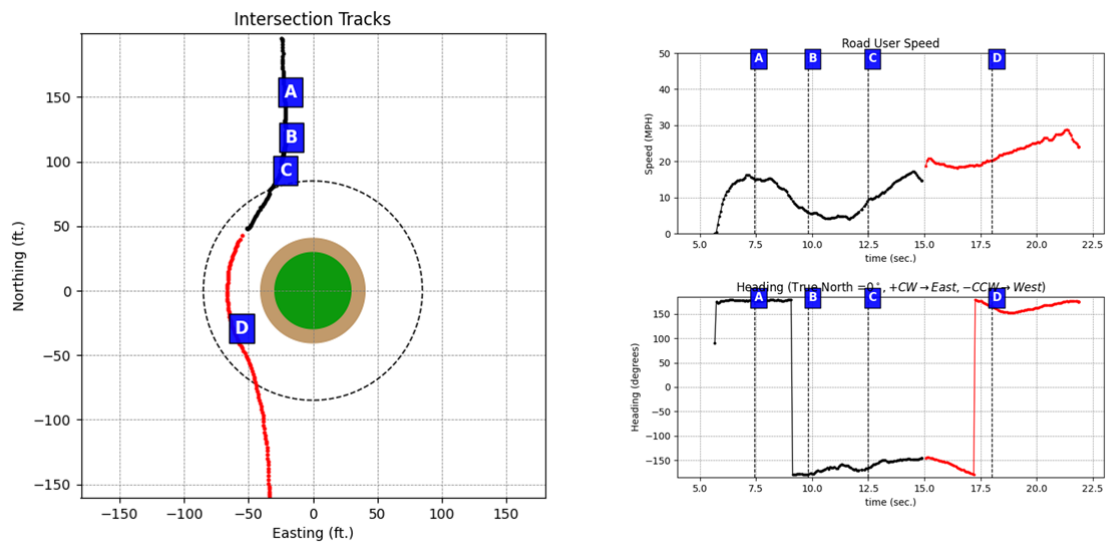
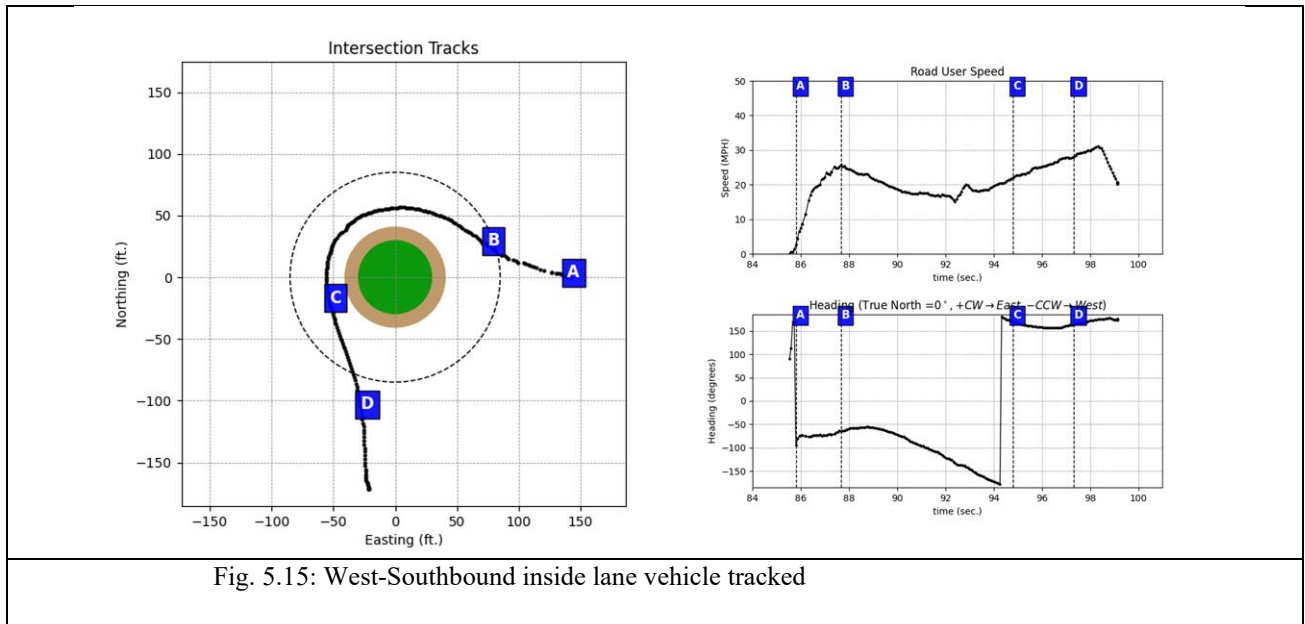


Fig. 5.14: Westbound vehicle tracked recovered the lost track ID



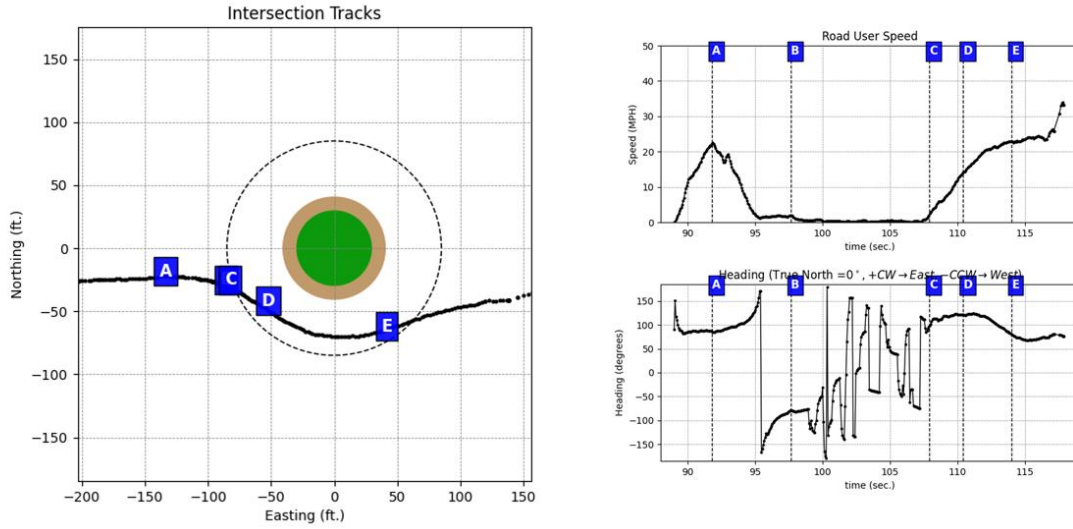


Fig. 5.17: Eastbound stopped vehicle tracked; time points B and C overlap on LEFT intersection diagram

5.4.1 Object tracking performance

Object tracking performance metrics image object tracking performance metrics proposed from literature. In context to this study, there are two important metrics for evaluating tracking performance. Object (physical world) localization errors [21], [22], [23] and preservation of the identity of each tracked object over time. This discussion focuses on the latter, by quantifying tracking association accuracy. If objects are ‘lost’ or confused with other objects, the communicated state information will misrepresent the evolving traffic condition, which may then affect the navigation and control actions taken by the C/AV. In terms of well established multi-object tracking accuracy metrics, the collected data can be used to calculate an Identity Precision metric (IDF1) in literature, defined as the harmonic mean of the Identification Recall and Precision, which emphasizes how long the detection and tracking approach correctly identifies each object trajectory ; a track that is lost, or associated with the wrong Track ID (*FN* and *FP* ReID respectively). This will be elucidated below.

The evaluation compared the predicted YOLO detection tracks IDs associated with manually 164 observed trajectories (objects), which were collected over a 5 minute period within an afternoon rush hour. The variation in track lengths ranged from 6.5 to 37.7 seconds (Fig. 5.19). Track association is characterized by quantifying tracking identification False Negatives (*FN*) and False Positives (*FP*) events with respect to ground-truth observed object trajectories. *FN* events occur for each subsequent frame after a track ID associated with the observed trajectory within the view area is lost. *FP* events occur when a) existing track IDs representing 2 different trajectories switch ID representations (ReID switching), b) newborn track IDs replace an existing *TP* object track ID (ReID replacement), or 3) when track IDs appear for phantom object tracks (e.g., background confusion). A radial boundary of 160 feet from the center of the roundabout intersection was used to clip vehicle tracks (about 2 to 3 car lengths upstream from the cross walks). A complete pedestrian trajectory was defined from curb-to-curb clearance boundaries. The ‘correct’ track ID association with a vehicle trajectory was defined by when the vehicle entered the roundabout to complete its maneuver through the intersection.

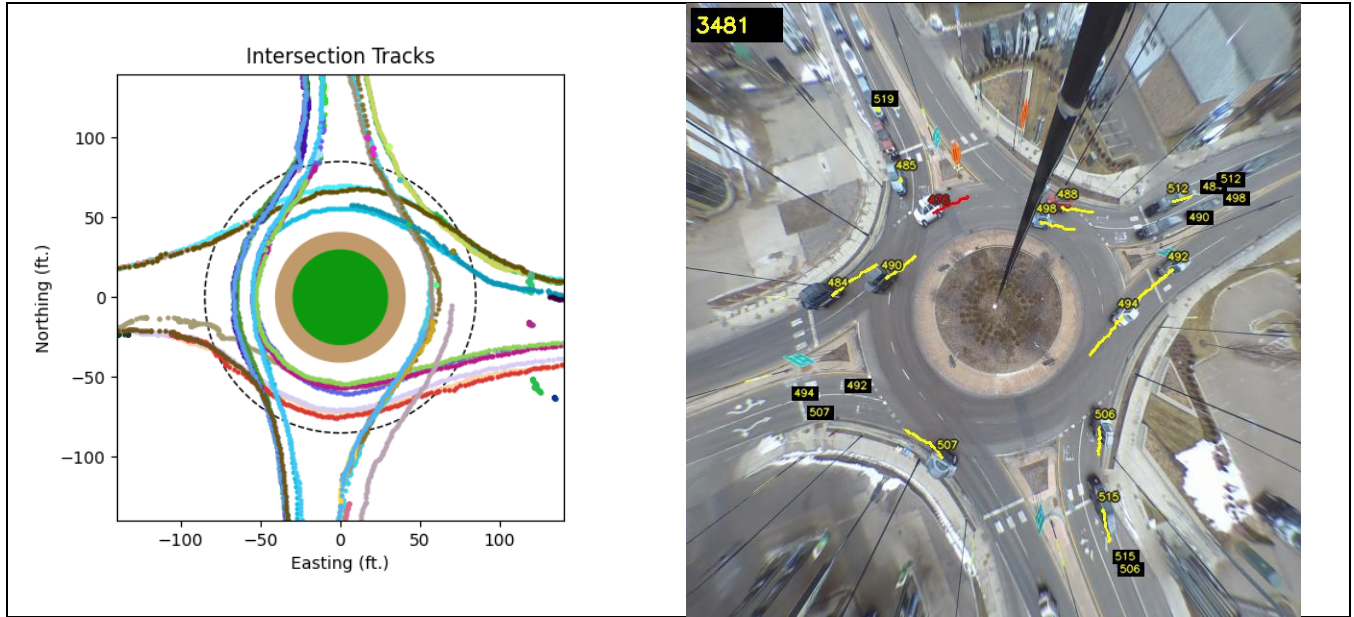


Fig. 5.18: LEFT: Samples of detected track IDs, RIGHT: Real-time object detection track ID observations. The starting location of each track ID (number labels) is displayed for the lifetime of the track. The red tracklet tail indicates resolution of an *FP ReID*, track ID.

The *TP* association for a complete observed trajectory occurred for 72.0% (118) tracks. Most of the tracking association *FP* and *FN* cases occurred within queue formations at the entering legs, where there were significant vehicle-on-vehicle occlusions, and were highly correlated (right hand side of Fig. 5.20). Note that such *FP* and *FN* cases were highly correlated; an assigned track ID would become lost, before another track ID recovered the detected object. Otherwise, within the roundabout intersection there were far less such errors; there was one case of *FP* ReID switching, which recovered to the correct tracks after 10 frames; and 2 cases of ReID replacement where a newborn *FP* replaced a vehicle track within the roundabout. The few (3) pedestrian events that were observed were either missed completely (100%), or otherwise high *FN*-missed detection error rate ($> 50\%$). One pedestrian event was a bicyclist that traversed the crosswalk. Lastly, the IDF1 score for this data is 0.917 compared to 0.912 for Bytetrack in Li & Wu, discussed earlier [48]; their proposed tracking algorithm achieved $IDF1=0.999$.

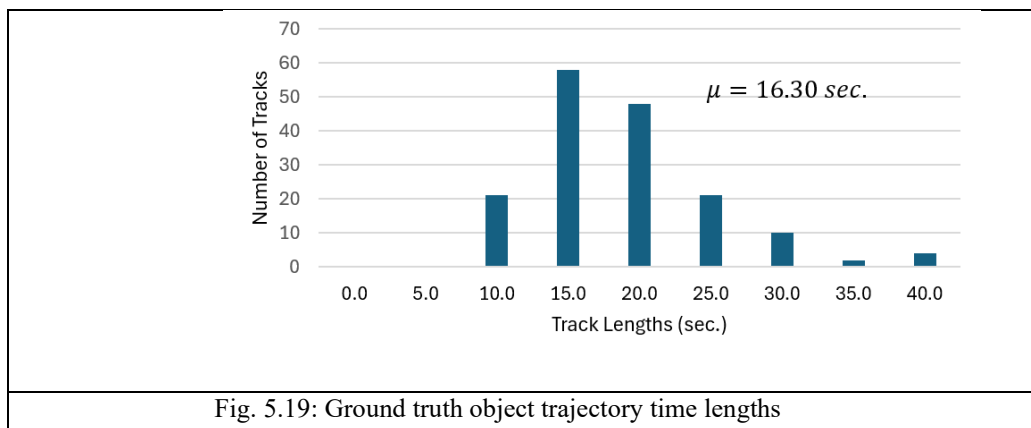


Fig. 5.19: Ground truth object trajectory time lengths

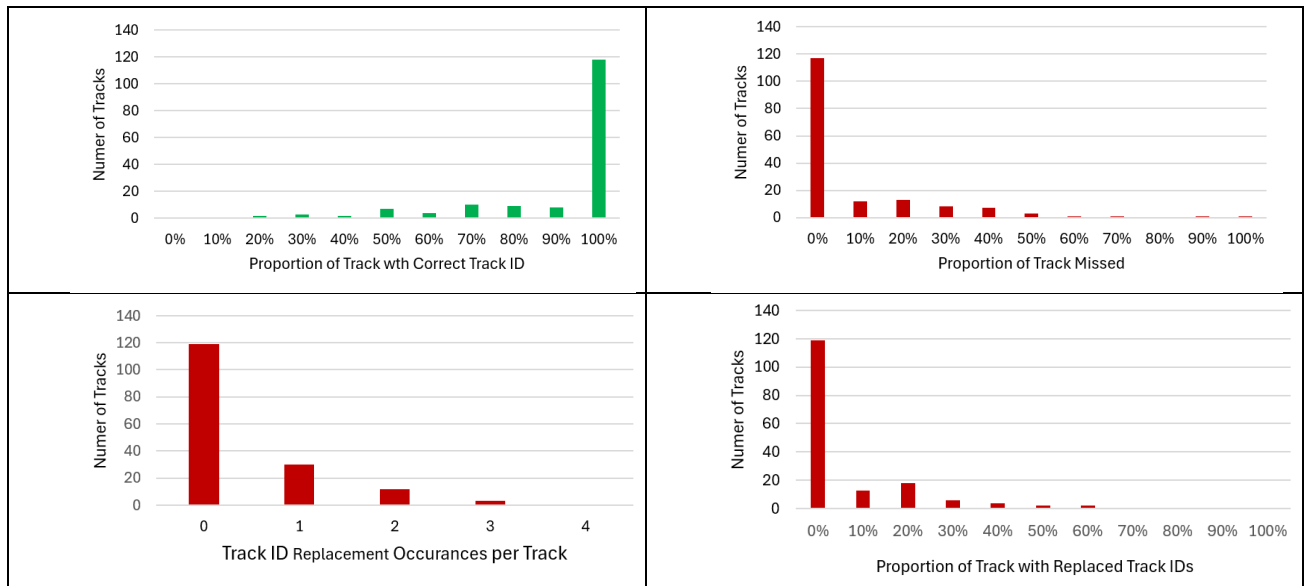


Fig. 5.20: Object tracking association $TP/FN/FP$ accuracy--all classes

Object tracking for vehicles were relatively robust but performance for pedestrians was not satisfactory, or inconclusive at best. As previously, noted there were class imbalances in the datasets that likely contributed to this, as well as image resolution limitations which reduces the discernable features, especially for smaller appearance objects like pedestrians.

6 Conclusions and Recommendations

This objective for this project was to provide roadside hardware and software infrastructure that can provide real-time traffic and pedestrian based movement within a high traffic urban Roundabout. Potentially, the data can be packaged for J2735 Sensor Data Sharing Message broadcasting to connect and autonomous vehicles. In particular, we studied the utility of a single mounted camera sensor, and testing widely available publicly available single-shot DL detection architectures for meeting this objective.

For detection feasibility, two DL models based on YOLO single-shot architectures were investigated. YOLOv8 was tested by synthesizing multiple vantage points from the high-resolution fisheye centered camera into 8 auxiliary image ‘views’ and labeling each of them with axis aligned bounding boxes, with augmentation aimed to train and generalize across all viewpoints. Despite these augmentations, the overall dataset imbalance was likely a limiting factor in detection performance. While synthetic transformations improve robustness to specific perturbations, they do not introduce fundamentally new scene variations or object instances. As a result, increasing the quantity and diversity of real annotated training data—particularly for underrepresented classes such as pedestrians and trucks is expected to yield more significant improvements in generalization and detection accuracy. This is especially important for the detection of vulnerable road users such as pedestrians and bicyclists (and to some extent, motorcyclists or throttle-based e-bike riders). Nevertheless, even with the limited training dataset, the YOLO low resource models had promising performance of

The YOLOv8 model also can potentially robustly learn recognition and detection tasks for varied object sizes and appearances, which we hypothesized would be advantageous for high distortion images such as fisheye optics. In the same, a more recent model, YOLOv11-- which reportedly from other findings indicates that the architecture can be trained more efficiently while rivaling similar performance to YOLOv8, was also tested. Here again, due to the nature of the recorded user traffic, class imbalance affected model performance, which would also benefit from collecting more training data. Nevertheless, direct detection from a single highly distorted fisheye camera was encouraging and this will be pursued further, to reduce computational resources and minimize latency. Additionally, trajectory tracking performance indicated relatively robust performance across different traffic motion behaviors most specifically for vehicles entering, and maneuvering within, the roundabout intersection can be tracked completely (before the crosswalks) 72% of the time, and 90% of the trajectories are 77% of the time.

We plan on evaluating this approach using higher resolution streaming video input to improve detection accuracy and quantify tradeoffs in detection bandwidth and accuracy. Later versions of YOLO (from v10 onward) can support higher input resolutions, at the cost of increased preprocessing, inference, and training time. According to reported performance benchmarks, an input resolution of 1280×1280, which was segmented into the camera view regions for the multi-image detection, may also be sufficient for meeting the real-time requirements for single fisheye image detection [28], [29]. Equally important, future work to implement, real-world data driven 3D tracking would complement this approach by theoretically providing more accurate road user state information. In addition, the tracking location accuracy and latency of trajectory location data should be assessed using a connected/autonomous vehicle platform with high accuracy RTK GPS sensor and BSU radio for transmitting its location to a fixed RSU. In this regard, we plan on deploying a DSRC C2X radio to broadcast the data with the SDSM protocol to determine latency attributed to hardware. The latency can significantly impact the broadcasted reception of the detected and tracked road user data, and will inform future C/AV navigation and control studies

Regarding a phase-II study, the approach in its current form can provide scenario cases for developing AIF-based action policies, to control speed, and to some extent steering; the control actions can be selected for a specific ego vehicle, with predicted track data, in order to create counter factual ‘evidence’ to compare the resulting sequential action control policies.

7 REFERENCES

- [1] L. Rodegerdis, M. Blogg, E. Wemple, E. Myers, M. Kyte, and M. Dixon, “Roundabouts in the United States: National Cooperative Highway Research Program NCHRP Report 572, 2nd Ed.,” Transportation Research Board, WASHINGTON, D.C, 2007.
- [2] L. Rodegerdts, J. Bansen, C. Tiesler, J. Knudsen, and E. Myers, “Roundabouts: An Informational Guide 2nd Ed., National Highway Research Program NCHRP REPORT 672,” Transportation Research Board, 2010.
- [3] “Roundabouts, Carmel, IN.” Accessed: Oct. 04, 2025. [Online]. Available: <https://www.carmel.in.gov/311/Roundabouts>
- [4] W. Hu, “Roundabouts,” IIHS HLDI. Accessed: Oct. 15, 2025. [Online]. Available: <https://www.iihs.org/research-areas/roundabouts>
- [5] J. Hourdos, V. Richfield, and M. Shauer, “Investigation of Pedestrian/Bicycle Risk in Minnesota Roundabout Crossings,” Minnesota Department of Transportation, Saint Paul, Final Report MnDOT 2012-28, Sep. 2012. [Online]. Available: <https://hdl.handle.net/11299/138018>
- [6] J. Hourdos, M. Shauer, and G. Davis, “An Observational Study of Pedestrian and Bicycle Crossing Experience in Two Modern Urban Roundabouts,” in *Transportation Research Board*, Washington D.C.: Transportation Research Board, Jan. 2012, p. 14.
- [7] N. Poudel and P. A. Singleton, “Bicycle safety at roundabouts: a systematic literature review,” *Transport Reviews*, vol. 41, no. 5, pp. 617–642, Sep. 2021, doi: 10.1080/01441647.2021.1877207.
- [8] C. J. Sowers, “Driver Adaptation To A New Two-Lane Roundabout: A New Perspective Using Drones,” Master of Science, University of New Brunswick, New Brunswick, CA, 2017.
- [9] N. P. Belz, “Investigating the role of exiting vehicles and turn indicator usage in gap acceptance at single-lane roundabouts,” *Front. Struct. Civ. Eng.*, vol. 11, no. 1, pp. 8–16, Mar. 2017, doi: 10.1007/s11709-016-0365-7.
- [10] E. Macioszek, “Analysis of Driver Behaviour at Roundabouts in Tokyo and the Tokyo Surroundings,” in *Modern Traffic Engineering in the System Approach to the Development of Traffic Networks*, vol. 1083, E. Macioszek and G. Sierpiński, Eds., in *Advances in Intelligent Systems and Computing*, vol. 1083. , Cham: Springer International Publishing, 2020, pp. 216–227. doi: 10.1007/978-3-030-34069-8_17.
- [11] “SURFACE VEHICLE STANDARD: V2X Communications Message Set Dictionary (J2735).” SAE International, Sep. 2024.
- [12] *michigan-traffic-lab/pyV2XLib*. (Mar. 11, 2026). Python. Mobility Transformation Lab. Accessed: Apr. 03, 2026. [Online]. Available: <https://github.com/michigan-traffic-lab/pyV2XLib>
- [13] L. Walkin, *vlm/asn1c*. (Mar. 21, 2026). C. Accessed: Apr. 03, 2026. [Online]. Available: <https://github.com/vlm/asn1c>
- [14] S. Chen, X. Qin, Z. Boukhers, J. See, W. Sui, and C. Yang, “Pedestrian Collision Prediction Using a Monocular Camera,” in *Proceedings of the 8th international Workshop on Sensor-Based Activity Recognition and Artificial Intelligence*, Lübeck Germany: ACM, Sep. 2023, pp. 1–8. doi: 10.1145/3615834.3615846.

- [15] H. Yang, Y. He, Y. Xu, and H. Zhao, "Collision Avoidance for Autonomous Vehicles Based on MPC With Adaptive APF," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1559–1570, Jan. 2024, doi: 10.1109/TIV.2023.3337417.
- [16] F. Gritschneider, P. Hatzelmann, M. Thom, F. Kunz, and K. Dietmayer, "Adaptive Learning Based on Guided Exploration for Decision Making at Roundabouts," in *Proc. of 2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden: IEEE, Jun. 2016, pp. 433–440.
- [17] D. Chen, B. Zhou, V. Koltun, and P. Krahenbuhl, "Learn by cheating," in *Proceedings of Machine Learning Research*, Osaka, Japan: PMLR, Nov. 2019, pp. 66–75.
- [18] J. Engström, R. Wei, A. D. McDonald, A. Garcia, M. O'Kelly, and L. Johnson, "Resolving uncertainty on the fly: modeling adaptive driving behavior as active inference," *Front. Neurobot.*, vol. 18, p. 1341750, Mar. 2024, doi: 10.3389/fnbot.2024.1341750.
- [19] NHTSA, "Fact Sheet: Improving Safety and Mobility Through Vehicle-to-Vehicle Communication Technology." National Highway Traffic Safety Administration, USDOT, Washington D.C., 2014. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/v2v_fact_sheet_101414_v2a.pdf
- [20] A. Enan *et al.*, "Basic Safety Message Generation through a Video-based Analytics for Potential Safety Applications," *ACM J. Auton. Transport. Syst.*, vol. 1, no. 4, pp. 1–26, Dec. 2024, doi: 10.1145/3643823.
- [21] Z. Zou *et al.*, "Real-time Full-stack Traffic Scene Perception for Autonomous Driving with Roadside Cameras," in *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA: IEEE, May 2022, pp. 890–896. doi: 10.1109/ICRA46639.2022.9812137.
- [22] C. Zhang *et al.*, "A Roadside Cooperative Perception System with Multi-Camera Fusion at an Intersection," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Bilbao, Spain: IEEE, Sep. 2023, pp. 642–649. doi: 10.1109/ITSC57777.2023.10422029.
- [23] M. Adl, X. Guo, A. Mohammad-Alikhani, B. Abdi, R. Ahmed, and A. Emadi, "Roadside Fisheye Vision for Cooperative Perception in V2I-Assisted Automated Driving," *IEEE Open J. Intell. Transp. Syst.*, vol. 6, pp. 1221–1234, 2025, doi: 10.1109/OJITS.2025.3603968.
- [24] O. Zheng, M. Abdel-Aty, L. Yue, A. Abdelraouf, Z. Wang, and N. Mahmoud, "CitySim: A Drone-Based Vehicle Trajectory Dataset for Safety-Oriented Research and Digital Twins," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2678, no. 4, pp. 606–621, Apr. 2024, doi: 10.1177/03611981231185768.
- [25] Z. Islam, M. Abdel-Aty, A. Goswamy, A. Abdelraouf, and O. Zheng, "Modelling the Relationship Between Post Encroachment Time and Signal Timings Using UAV Video data," 2022, *arXiv*. doi: 10.48550/ARXIV.2210.05044.
- [26] F. Kehagia and A. Anagnostopoulos, "Evaluating roundabouts safety performance based on vehicle trajectory analysis through UAVs," presented at the Forum of European Road Safety Research Institutes Conference, The Hague, Netherlands: FERSI, Oct. 2022, pp. 1–7.
- [27] Á. T. Köpeczi-Bócz, T. Mi, G. Orosz, and D. Takács, "YOLOgraphy: Image Processing Based Vehicle Position Recognition," in *16th International Symposium on Advanced Vehicle Control: Proceedings of AVAC'24 - Society of Automotive Engineers of Japan: Lecture Notes in Mechanical Engineering*, G. Mastinu, F. Braghin, F. Cheli, M. Corno, and S. M. Savaresi, Eds., Springer, 2024, pp. 392–398. doi: 10.1007/978-3-031-70392-8_56.

- [28] M. Gochoo *et al.*, “FishEye8K: A Benchmark and Dataset for Fisheye Camera Object Detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2023, pp. 5304–5312.
- [29] H. Rashed *et al.*, “FisheyeYOLO: Object Detection on Fisheye Cameras for Autonomous Driving,” presented at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, Dec. 2020.
- [30] S. Yogamani *et al.*, “WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 9307–9317. doi: 10.1109/ICCV.2019.00940.
- [31] J. Kannala and S. S. Brandt, “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006, doi: 10.1109/TPAMI.2006.153.
- [32] “OpenCV: Fisheye camera model.” Accessed: Apr. 20, 2026. [Online]. Available: https://docs.opencv.org/4.x/db/d58/group__calib3d__fisheye.html
- [33] R. Y. Tsai, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,” *Journal of Robotics and Automation*, vol. RA03, no. 4, pp. 323–344, Aug. 1987.
- [34] “Model Comparisons: Choose the Best Object Detection Model for Your Project.” Accessed: Apr. 20, 2026. [Online]. Available: <https://docs.ultralytics.com/compare/>
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [36] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using YOLO: challenges, architectural successors, datasets and applications,” *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.
- [37] S. Biswas, S. Acharjee, A. Ali, and S. S. Chaudhari, “YOLOv8 based Traffic Signal Detection in Indian Road,” in *2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, Kolkata, India: IEEE, Dec. 2023, pp. 1–6. doi: 10.1109/IEMENTech60402.2023.10423520.
- [38] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8,” May 22, 2024, *arXiv*: arXiv:2305.09972. doi: 10.48550/arXiv.2305.09972.
- [39] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *MAKE*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [40] M. Yaseen, “What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector,” Aug. 28, 2024, *arXiv*: arXiv:2408.15857. doi: 10.48550/arXiv.2408.15857.
- [41] R. Khanam and M. Hussain, “YOLOv11: An Overview of the Key Architectural Enhancements,” 2024, *arXiv*. doi: 10.48550/ARXIV.2410.17725.

- [42] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
- [43] Haibo He and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
- [44] Y. Zhang *et al.*, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” Apr. 07, 2022, *arXiv*: arXiv:2110.06864. doi: 10.48550/arXiv.2110.06864.
- [45] H. Hashempoor and Y. D. Hwang, “FastTracker: Real-Time and Accurate Visual Tracking,” 2025, *arXiv*. doi: 10.48550/ARXIV.2508.14370.
- [46] T. Collins and A. Bartoli, “Infinitesimal Plane-Based Pose Estimation,” *Int J Comput Vis*, vol. 109, no. 3, pp. 252–286, Sep. 2014, doi: 10.1007/s11263-014-0725-5.
- [47] “Geodetic transformation — PROJ 9.8.1 documentation.” Accessed: Jun. 01, 2026. [Online]. Available: <https://proj.org/en/stable/usage/transformation.html>
- [48] X. Li and J. Wu, “Extracting High-Precision Vehicle Motion Data from Unmanned Aerial Vehicle Video Captured under Various Weather Conditions,” *Remote Sensing*, vol. 14, no. 21, p. 5513, Nov. 2022, doi: 10.3390/rs14215513.
- [49] J. Nelson, “How to Resize Images for Computer Vision Without Distortion.” Accessed: Jan. 27, 2026. [Online]. Available: <https://blog.roboflow.com/you-might-be-resizing-your-images-incorrectly/>
- [50] K. Andreas, “Does SAHI with YOLO11 make sense - Discussion - Ultralytics.” Accessed: Feb. 05, 2026. [Online]. Available: <https://community.ultralytics.com/t/does-sahi-with-yolo11-make-sense/1465>

APPENDIX A.

A.1 Outputs

Presentation: “Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users”, March 28 2025, CCAT Global Symposium, Ann Arbor Michigan.

Authors: Ted Morris (presenter), Vassilios Morellas, Varadaraya Ganesh Shenoy

Seminar presentation: “Roadside Vehicle and VRU Detection for Connected Vehicles: Utilization in a multilane Roundabout” , April 10, 2025, CTS Seminar Series, Advanced Topics in Transportation., University of Minnesota, Minneapolis.

Authors Ted Morris (presenter), Vassilios Morellas, Varadaraya Ganesh Shenoy, Ben Shepard

Webinar presentation: “Roundabout Connected/Automated Vehicle Active Inference Control Strategies to Improve Safety for All Users”, November 22, 2024, CCAT Safety Working Group.

Authors Ted Morris(presenter), Vassilios Morellas

A.2 Outcomes

Ben Shepard, M.S. Robotics, Minnesota Robotics Institute, University of Minnesota, Graduated Fall, 2024

Varadaraya Ganesh Shanoy, M.S. Candidate, Robotics, Minnesota Robotics Institute, M.S. Plan A Thesis: “Active Inference Framework for Car-Following Scenario”, Anticipated Graduation: July 2026.

A.3 Impacts

Effects of outcome on the transportation system, or society in general.

1. Increases the body of knowledge and technologies;
2. Improves the physical, institutional, and information resources that enable people to have access to training and new technologies.

A.4 Challenges and lessons learned

Opensource license requirements for public domain models and software used for the project does not align with the more restrictive public licensing structures that are acceptable to the University of Minnesota. Agreements to allow permanent installment infrastructure owned by the University of Minnesota on City/County Right-of-way property. Damage to the infrastructure during the project period added unforeseen delays and costs.