Linking Traffic Condition Metrics with a Cyclist's Workload

MSc Thesis Report

Ruben Terwint

Delft University of Technology

TU Delft

Linking Traffic Condition Metrics with a Cyclist's Workload

MSc Thesis Report

by

Ruben Terwint

Student Name	Student Number
Ruben Terwint	5362679

Supervisors:	Jason K. Moore & Holger Caesar
Daily Supervisor:	Jules Ronné
Project Duration:	May, 2025 - September, 2025
Faculty:	Faculty of Mechanical Engineering, Delft

Cover:	OpenAI DALLE-3
Style:	TU Delft Report Style

TU Delft

Statement on AI Assistance: *During the preparation of this work, the author used OpenAI's DALL·E 3 to generate the cover page image. The image was created based on prompts written by the author. The author also made use of the GPT 4o model in order to review grammar and improve writing. The author reviewed the generated content and takes full responsibility for its appropriateness and inclusion in the final document.*

Summary

This thesis investigates whether traffic patterns captured by a cyclist's camera coincide with moments of rider-reported workload increases, and whether a simple, scalable pipeline using a single forward-facing camera can extract useful signals for workload modelling. We developed an end-to-end system that detects and tracks nearby road users in cyclist point-of-view video, computing traffic features from 984 temporal windows across three urban rides in the city of Delft.

Using a compact set of eight traffic features and logistic regression modelling, we evaluated performance through three complementary approaches. Individual feature analysis revealed consistent directional differences between low and high workload episodes, with six of eight features maintaining the same directional relationships in multivariate analysis. Ranking performance exceeded baseline expectations, achieving within-route ROC-AUC of 0.570 (0.070 above chance) and PR-AUC of 0.450 (0.080 above baseline), meeting established criteria for small effect sizes. Cross-route generalization proved more challenging, with performance dropping to ROC-AUC of 0.517 and PR-AUC of 0.401, though consistently exceeding route-specific baselines across all test routes.

Binary classification demonstrated above-baseline performance, with within-route F1 scores of 0.478 representing a 0.108 improvement over our baseline expectations. Threshold optimization increased F1 to 0.517 by improving recall from 55.0% to 78.3%, though cross-route binary performance remained more limited. The traffic patterns associated with high workload episodes point toward defensive cycling scenarios characterized by increased object density, complex motion patterns, and reduced cyclist velocity in dense urban environments.

These findings establish that camera-derived traffic metrics contain detectable workload-related information while highlighting significant constraints for practical deployment. The modest magnitude of detected relationships and limited cross-route transferability indicate that while the approach demonstrates feasibility, substantial development is needed before robust real-world application. This work provides quantitative evidence for traffic-workload relationships in cycling and establishes methodological foundations for future research in automated cycling safety monitoring.

Contents

1

Introduction

1.1. Overview

Cycling as a mode of urban transportation is continuing to gain significant popularity worldwide due to its sustainability, affordability, and health benefits. Consequently, urban planners and policymakers are increasingly investing in infrastructure aimed at promoting cycling. Despite these advancements, cyclist-centric research focusing on the interactions between cyclists and their urban environment remains limited. This knowledge gap poses a substantial challenge since cyclists are among the most vulnerable road users, continuously exposed to risks associated with traffic dynamics, infrastructure inadequacies, and environmental conditions [76].

One critical yet frequently overlooked factor in cyclist safety and comfort is the mental workload experienced by riders. Mental workload refers to ”the perceived amount of mental and physical effort required to perform a task, reflecting the demands placed on an individual’s cognitive and physical resources” [31]. Increased mental workload can significantly impair decision-making capabilities, situational awareness, and reaction times, subsequently elevating safety risks [74]. Although extensively studied in the context of motor vehicle drivers, mental workload among cyclists remains relatively underexplored, despite cyclists confronting unique cognitive challenges. Unlike motorists, cyclists must simultaneously manage physical exertion while maintaining constant vigilance toward dynamic traffic conditions, changing infrastructure, and varying environmental stimuli, all without protective barriers.

Measuring mental workload is inherently complex, and existing methods typically rely on subjective, performance-based, or physiological assessments [28]. While these techniques effectively indicate when a cyclist’s workload becomes elevated, they do not directly reveal why such elevations occur. This limitation highlights the need to bridge the gap between observing workload increases and identifying their specific root causes.

Recent advancements in machine perception, especially from the perspective of the cyclist (instrumented bicycles), present an innovative solution to this challenge [15]. By continuously capturing and analyzing real-time environmental cues from the cyclist’s point of view, it becomes possible to directly link workload increases to precise external causes. Yet, current methods that implement cyclist perspective sensing are scarce and predominantly utilize costly equipment such as Light Detection And Ranging (LiDAR) sensors, limiting widespread adoption and practical scalability [60].

This thesis addresses this critical gap by presenting a novel, cost-effective approach using only monocular camera footage obtained from eye-tracking glasses worn by cyclists and a Global Navigation Satellite System (GNSS). A tailored analytical pipeline integrating custom fine-tuned You Only Look Once (YOLO)-based object detection, a custom hybrid tracking algorithm, and depth inference techniques was developed [53, 77]. This pipeline allows for the extraction of detailed and cyclist-relevant traffic condition metrics directly linked to changes in subjective mental workload. The ultimate goal of this research is two-fold:

1. Practical Application: Demonstrate that affordable, monocular-camera-based machine perception can reliably capture multiple traffic metrics crucial for understanding cyclist workload.
2. Scientific Insight: Identify a signal between extracted traffic condition metrics and a cyclist's subjective workload, laying the groundwork for bridging cause (environment) and effect (elevated workload).

This research analyses real-world cycling video data collected across different urban routes, lighting conditions, and participants within Delft, Netherlands, ensuring applicability.

1.2. Research Question

From the above contributions, a main and a supporting research question were derived. The main question of this report is:

- *"Do variations in cyclist perspective camera derived traffic condition metrics coincide with (semi)instantaneous perceived workload changes in Urban Cyclists?"*

The supporting methodological research question is the following:

- *"Can a reliable and scalable analytical pipeline, utilizing monocular camera footage from cyclist-worn eye-tracking glasses, be developed to comprehensively extract cyclist-relevant traffic condition metrics without relying on expensive sensors?"*

2

Related Work

2.1. Cyclist's Workload

Cognitive workload refers to the mental and physical effort required by an individual to perform a specific task, consuming cognitive and physical resources [30]. In the context of urban cycling, workload directly affects decision-making, situational awareness, and reaction times, which are critical for cyclist safety due to their vulnerability as unprotected road users [72]. Despite its clear relevance, cyclist mental workload remains understudied compared to other road users such as vehicle drivers.

Cyclist workload is primarily influenced by extrinsic factors that include urban infrastructure, environmental conditions, and traffic conditions [67]. Although all these factors contribute significantly, this research focuses specifically on traffic conditions. Traffic conditions are particularly impactful on cyclists' cognitive workload due to their dynamic nature, requiring continuous monitoring and real-time decisions. Moreover, traffic condition variables such as vehicle proximity, relative speed, and traffic density, can be directly captured and quantified using modern sensor-based approaches. Thus, the clear measurability and direct connection of traffic conditions to cognitive workload provide the strongest motivation for selecting them as the primary focus of this research.

The literature identifies several key traffic condition metrics influencing cyclists' mental workload:

Proximity (Passing Distance): Close passing events by other road users significantly increase perceived risk and cognitive load for cyclists, forcing heightened attention and rapid decision-making [58, 73].

Relative Velocity: High relative velocities between passing vehicles and cyclists reduce available reaction times, increasing cognitive demands and perceived risk during overtaking events [44].

Traffic Density: The density or volume of surrounding traffic increases perceptual load, forcing cyclists to constantly reassess their environment and increasing vigilance demands [72].

Traffic Mix (Road User Composition): The presence of heavy vehicles for example (ex. trucks, buses) significantly increases cognitive workload due to obstructed visibility, aerodynamic effects, and increased collision risk severity [16, 11]. This metric is not limited to the presence of heavy vehicles, it can also encapsulate how the diversity and proportion of different road users affect a bike rider's mental workload due to a decrease in overall order and an increase in unpredictable motions.

These metrics represent critical influences on cyclists' cognitive workload and form the foundation upon which further refined metrics were derived in this thesis. A detailed justification and refinement of these metrics will be presented in the Methodology chapter.

2.2. Mobile Sensing Platforms

To study interactions between cyclists and urban traffic environments, researchers have traditionally used two main approaches: fixed observation methods and mobile sensing platforms. Fixed observa-

tion approaches rely on stationary cameras or sensors at intersections or specific road segments to collect data on interactions between cyclists and vehicles (ex: proximity or time-to-collision metrics) [12, 6]. While effective at capturing high-quality data at specific locations, these methods lack ecological validity because they do not represent the cyclist's actual visual perspective and cannot continuously track individual cyclist experiences over an entire route. Consequently, fixed setups fail to capture critical dynamic information related to where cyclists direct their attention or how sudden and unpredictable interactions truly feel from the rider's perspective.

To address these limitations, a growing number of studies have turned toward mobile sensing platforms, typically instrumented bicycles equipped with sensors such as Light Detection And Ranging (LiDAR), Global Navigation Satellite System (GNSS), Inertia Measurement Units (IMU), and cameras. Examples include the Salzburg Instrumented Bicycle [50], the TU Delft SenseBike [62] and the Chalmers instrumented bicycle [16]. These platforms capture cyclist-centric data in real-world traffic conditions, enabling detailed measurement of overtaking distances, relative vehicle speeds, and environmental contexts (ex: road quality, cycling facilities).

However, despite their promising results, current instrumented bicycles still face critical limitations:

Cost and Complexity: High-end sensing equipment such as LiDAR and advanced stereo cameras significantly increases both the cost, technical complexity of such setups, as well as the overall manoeuvrability of the bicycle due to the added weight of all the instruments, limiting scalability and widespread adoption [24].

Limited Metric Scope per Study: Due to technical complexity and processing demands, many studies focus narrowly on specific metrics (ex: overtaking distances alone), thus lacking a comprehensive, integrated understanding of cyclists' interactions with traffic environments.

Fixed Camera Perspectives: Even mobile setups typically mount cameras rigidly on bike handlebars or frames, capturing fixed viewpoints rather than the true visual perspective and dynamic attention shifts of the cyclist. Thus, they cannot adequately capture cyclists' gaze behaviour or visual attention, which are critical indicators of mental workload [40].

To overcome these limitations, this thesis introduces a novel mobile sensing approach: utilizing monocular video footage obtained from cyclist-worn eye-tracking glasses. This approach inherently captures the cyclist's actual perspective, including dynamic gaze shifts and head movements, providing critical visual context aligned with the cyclist's real-time cognitive experience. This is aligned with a rising trend in which people outside of academic institutions start recording their daily bike rides using head-mounted cameras in order to showcase the dangers they incur from the environment and from their point-of-view. They do this in order to protect themselves, by having evidence in case of a crash or an altercation but also to inform people on the dangerous traffic conditions cyclists face in busy cities such as London [25].

By utilising advancements in computer vision (object detection models, tracking algorithms and monocular depth inference), this method allows extraction of multiple relevant traffic metrics, such as vehicle proximity, relative speeds, and traffic composition using a single, affordable sensor. By significantly reducing cost and complexity while preserving critical cognitive context, this approach represents a practical, scalable, and cyclist-centric evolution of existing mobile sensing platforms. The only additional sensor needed in this setup is the GNSS, which is used to obtain accurate positions and the velocity of the experiment taker.

The following chapter describes the experimental framework and data collection procedures designed to capture both the cyclist's environmental context and subjective workload responses necessary to validate this approach.

3

Data and Experiment Set-up

The data used in this project was captured by a group of students during their bachelor graduation project, who were also studying cyclist's cognitive workload [49]. They are the ones that took part in the experiments, went through different routes around Delft, capturing data from the eye-tracking glasses as well as the GNSS integrated in the bicycle they used, all the while giving auditory cues as to what mental workload level they were experiencing while riding.

3.1. Equipment and Available Data Collection

In order to gather their data, the students rode on the TU Delft Sensebike [69] which is equipped with sensors and modules such as Robosense M1 plus LiDARs and a SparkFun 9DoF Inertial Measurement Unit [62]. The only data collection used in these projects from the bike itself however, is the GNSS data which is collected by the ArduSimple simpleRTK2B GNSS receiver located at the rear of the bike. This means that the data collection could be done with any other bike as long as it comes with an integrated GNSS sensor.

Alongside the GNSS data, the students collected the bulk of their data through the Tobii 3 eye tracking glasses [68], this enabled them to collect video footage from the front facing camera. With the camera also comes a microphone which was used to record the audio workload cues given by the students as they were moving along their route. The eye tracking glasses also record a number of eye metrics alongside the captured footage but these were left outside the scope of this project.

Using this equipment, the students collected a substantial amount of data going through different routes in Delft with all four participants completing the same route per run as well as doing multiple loops within each run in order to reduce bias. They saved eye metrics, video, GNSS and csv files containing their workload cues sampled at 25Hz (which is the same frame rate as the video footage) into different folders for every run they did.

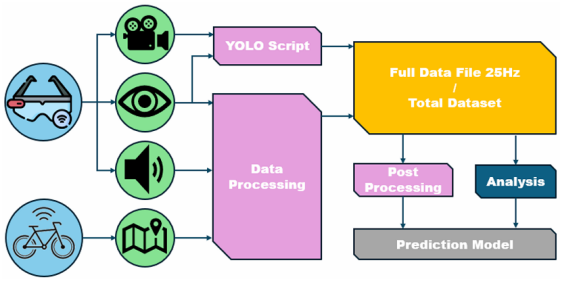


Figure 3.1: Schematic illustration used by the Bachelor students to describe their data collection and processing set-up [49]

The figure above 3.1 was used by the students during their bachelor end project to describe their set-up

and experiment process.

3.2. Data Selection

In order to complete the project, we only needed the video footage, raw GPS data as well as the processed workload data from different runs. A significant problem arose during the data inspection process, it turned out the workload scale the students used ranged from 1-5, the number of workload instances higher than 1 however, were very scarce, the higher scores even being non-existent in the data. In their paper, they report 80% of their full dataset being of workload level 1, 16% of level 2, 3.6% of level 3, 0.4% of level 4 and no level 5 workload levels [49]. Figure 3.2 below shows the reasoning behind their scoring scale:

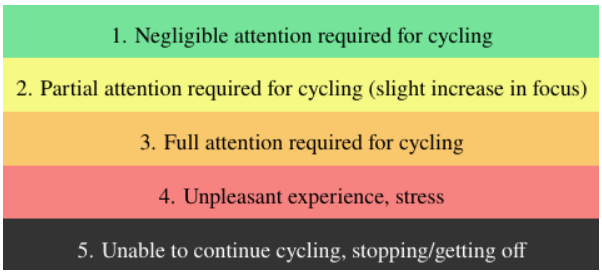


Figure 3.2: Personal subjective workload scale with levels from 1–5 [49].

To deal with this imbalance we decided to turn the workload score into a binary one of low and higher workload, effectively getting rid of the 1-5 scoring. In addition, every data run was inspected in terms of the workload data distribution to see which ones had the highest proportion of high workload instances. Only three of the top 4 videos in terms of high workload prevalence were used for the later analysis and results section, prioritizing statistical significance over completely different data. Samples of other videos were used during the creation of the pipeline as well as model training. The top three videos used for the statistical analysis were the following;

Video 1 is captured by participant 1 is characterised by a large contrast in lighting conditions as it was very sunny that day, causing the surrounding environment and infrastructure to produce large shaded areas. The route choice in this video is divided into two parts, the large and busy bike lane section (which includes few cars and other road users) and the pedestrian dominated city-centre section, providing two distinct environments for the analysis From the video, only an 11 minute 'active' section is used as the rest of the footage is during the set-up or a repeat of the same route. In that 11 minute segment composed of 16500 frames at 25fps, there is an almost identical amount of high vs normal workload data. The workload distribution timeline for Video 1 prior to converting to a binary score is given in figure A.1,

Video 2 is also captured by participant 1 is characterised by grey lighting conditions, much darker than in Video 1 but the lighting level is much more consistent throughout because of the absence of shade. Video 2's route starts directly at the TU Delft campus but it follows a quicker, more direct route to the city centre, focusing their efforts on that area. In their entire run, the participants did approximately three loops of the city centre (these loops are different than those of video 1). Similarly to the first video, only a 12 minute segment of the full video is used for the analysis with the following workload data distribution (see figure A.2).

Video 3 was captured by participant 2 and follows an almost identical route to Video 2 at a slightly different time of day. The main difference with Video 2 is that instead of sampling from the start of their ride at the TU Delft campus and ending it somewhere after the first loop they did in the city centre. The chosen 11 minute segment was entirely within the city centre course in order to capture more high workload instances and have a different route progression from the other videos which would help reduce potential bias in the results. The chosen segment had the following workload distribution (see

figure A.3). Table 3.1 summarises the chosen video segments in terms of frame count, high workload prevalence, route and weather conditions.

Table 3.1: Video segments, frame counts, workload proportion, routes, and weather conditions.

Segment	Frames	High WL	Route	Weather
Video 1 (P1)	17,219	44.3%	Bike route → Delft station → City centre	Extreme sun and shade
Video 2 (P1)	16,823	36.6%	Delft Campus → City Centre	Grey → Sunny
Video 3 (P2)	16,470	29.6%	City Centre	Grey → Sunny

Images 3.3 and 3.4 showcase the extreme lighting contrast due to the weather and specific cycling infrastructure, respectively. These conditions increased the difficulty of the footage analysis conducted in the next section 4, specifically the object detection algorithm suffers greatly from poor visibility due to extreme lighting conditions [56].

Figure 3.3: example image showcasing extreme lighting condition in video 1

Figure 3.4: Example of bad visibility in a tunnel

4

Methodology

4.1. Overview

The goal of this methodology is to extract cyclist-relevant traffic condition metrics from first-person monocular video footage and evaluate their relationship with moment-to-moment mental workload levels. This is achieved through a pipeline composed of five main stages: object detection, object tracking, depth inference, metric extraction, and statistical analysis. Each stage is outlined below.

Object Detection is the foundation of the pipeline, responsible for identifying all relevant traffic participants such as cars, trucks, buses, scooter riders, cyclists, and pedestrians in each video frame. This is a prerequisite for every other stage, as tracking, depth, and metric extraction are all detection-dependent.

Object Tracking is used to temporally link detections across frames, forming continuous object trajectories. This temporal continuity is essential for calculating time-based metrics such as relative velocity and proximity dynamics. Without tracking, these features would be undefined or highly noisy.

Depth Inference provides an estimation of distance between the cyclist and surrounding road users. Since only a monocular camera is used, relative depth must be inferred using deep learning models. These depth estimates are then post-processed and calibrated to approximate metric distances.

Metric Extraction uses the outputs from detection, tracking, and depth to compute interpretable traffic condition variables. These include proximity (ex: closest object per frame), relative speed (derived from tracked objects' motion and depth changes), and road user composition (derived from class distributions). GNSS-based cyclist-velocity (this refers to the speed of the cyclist capturing the footage) is used separately and does not depend on visual detection.

Statistical Analysis links the extracted traffic condition metrics with subjective workload scores. This is performed per video, followed by a combined analysis across datasets. The goal is to assess whether variations in environmental traffic features correlate with changes in cyclist-reported workload.

Each stage is detailed in the sections that follow, including the models used, calibration steps, and design choices made in response to the limitations of monocular footage, detection reliability, and dataset imbalance. Collectively, the pipeline offers a scalable, low-cost method to bridge the gap between external traffic stimuli and internal cognitive responses from the cyclist's perspective.

4.2. Object Detection

In this section, we specify and evaluate the object-detection component that extracts road users from point-of-view cycling videos. We (i) motivate the detector choice (YOLO family), (ii) define a task-appropriate class labelling aligned to our traffic analysis (pedestrian, cyclist = rider+vehicle, car, bus, truck), (iii) describe training and labelling and (iv) report performance on a held-out route with a normalized confusion matrix to interpret error modes that matter for downstream tracking and metric extraction.

4.2.1. Model Selection

The first stage of the pipeline involves detecting relevant road users within each video frame. Given the need for good performance, support for custom training, and proven robustness in urban scenes, the You Only Look Once (YOLO) family of models was selected. YOLO is a one-stage detector that offers a favorable trade-off between speed and accuracy for dense object detection in street-level footage, making it suitable for mobile applications with limited computational resources [71].

The objective was to detect and classify traffic participants that could influence a cyclist's perceived workload. These include pedestrians, cyclists, motorcycles, cars, buses, and trucks. To this end, a YOLO11m model pre-trained on the Common Objects in Context (COCO) dataset was used as a starting point [70]. COCO is a large-scale object detection dataset containing over 200,000 labeled images and 80 object classes commonly encountered in everyday scenes [43]. A subset of these COCO classes was retained for this study: {0: person, 1: bicycle, 2: car, 3: motorcycle, 5: bus, 7: truck}. A confidence threshold of 0.27 was chosen empirically to balance false positives and missed detections.

During the initial testing on the used data, two domain-specific challenges required special attention:

1. **Parked bicycles and motorcycles:** In contrast to parked cars or trucks, stationary two-wheeled vehicles typically do not pose a workload threat to the cyclist, especially when clearly parked on sidewalks or bike racks. However, YOLO's pre-trained model detects all visible objects, including parked bicycles and motorcycles, leading to inflated object counts and noise in proximity-related metrics. This created the need for a strategy to filter out parked bicycles and motorcycles post-detection.
2. **Separation of rider and vehicle:** In YOLO's standard configuration, riders and their vehicles are treated as distinct classes (ex: person + bicycle, person + motorcycle), resulting in two separate bounding boxes. This separation poses a problem when analyzing interactions: for instance, distinguishing a pedestrian crossing from a moving cyclist requires knowing that the bicycle and rider are part of the same entity. Furthermore, applying tracking or motion constraints requires treating a cyclist as one object, not two. This challenge is especially important for differentiating riders from pedestrians during metric extraction.

Figure 4.1: Example YOLO11m output on single frame, in yellow are the bicycle detections, white is the person class, class number and confidence scores are given on top of the detection boxes.

In order to address this, a custom YOLO model was fine-tuned on curated, annotated frames extracted

from different videos in [49]'s dataset. This fine-tuned model was trained with a simplified class structure:

- Cyclist: encompassing both human and vehicle (whether bicycle, motorcycle, or scooter)
- Pedestrian: isolated individuals on foot
- Other road users (cars, buses, trucks) as in the original class list

This approach followed the same definition of cyclist as the labelling rules for a cyclist in the Waymo dataset [65]. This allowed for direct differentiation between dynamic traffic participants (cyclists, pedestrians, vehicles) while filtering out non-relevant elements (ex: parked two-wheelers, bystanders not in the cyclist's path). The fine-tuned model reduced false detections and enabled more precise tracking and metric extraction aligned with the cyclist's perspective and field of view.

4.2.2. Fine-tuning and Labelling Strategy

We fine-tuned a YOLO11m detector on a curated dataset tailored to cyclist-centric traffic perception. Training images were drawn from multiple eye-tracking videos covering distinct routes, weather and lighting conditions. The training images were complemented with selected COCO images containing the relevant road-user classes (car, bus, truck) to mitigate class imbalance in the point-of-view footage from Delft traffic (mostly cyclists and pedestrians).

To maximise scene diversity and reduce temporal redundancy, frames were sampled at random subject to a minimum separation of 50 frames (25 FPS), ensuring that any two samples were at least 2 s apart while preserving class representativeness. The labelling strategy for these training images is outlined below.

The resulting dataset and sampling policy were chosen to improve generalisation to unseen routes and lighting conditions. The performance of the fine-tuned model is also reported in this section.

Labelling Process: The labelling process was performed using the online platform Roboflow [18], which provides a web-based annotation interface and auto-labelling capabilities. Initially, the auto-labeller was prompted using textual descriptions per class, (ex: cyclist class = person on bike, person on motorbike, bike rider) but performance was unreliable and required manual annotation. After manually annotating 400 frames from the sampled dataset, Roboflow's internal model was re-trained on these annotations, which substantially improved auto-labelling accuracy for the remaining frames.

Annotation Guidelines: Label annotations strictly followed guidelines aligned with the research objective: to detect and track only traffic participants in the cyclist's field of view (FoV). Therefore, the following rules were applied:

- Objects partially occluded or only barely visible (ex: torsos only or bike only) were excluded to avoid ambiguous cases and confusing the model during training.
- Seated pedestrians or far away vehicles were not annotated, as they do not directly influence perceived workload and would inflate the later extracted metrics.

Example labelled frames are shown in Figure 4.2.

Figure 4.2: Example frame annotations following defined labelling rules. Only traffic participants clearly within the cyclist's field of view are labelled.

Model Evaluation:

The final custom YOLO11m model was evaluated on a held-out test set consisting of frames not used during training, originating from an entirely new 5th video using the same random sampling strategy. The test set was manually annotated following the same guidelines. Results showed substantial improvements in cyclist and pedestrian detection accuracy and reduced false positives on ambiguous cases. It is important to note that there were no bus class instances in the test set and are therefore are not present in the model results.

We evaluate on 106 images (396 annotated instances) using standard detection metrics. Precision measures the fraction of predicted boxes that are correct (true positives over all positives), and recall measures the fraction of ground-truth objects that are correctly detected (true positives over all ground truths). Formal definitions are provided in Appendix B.2 (Precision/Recall formulas). A detection is considered correct if its Intersection-over-Union (IoU) with a ground-truth box of the same class exceeds a threshold τ. Average Precision (AP) is the area under the precision–recall curve for a class at a given IoU threshold. We report AP@0.50 (often called mAP@0.5), the mean AP across classes at $\tau = 0.50$ (the legacy PASCAL metric), and AP@0.50:0.95 (the COCO metric), the mean AP averaged over IoU thresholds $\tau \in \{0.50, 0.55, \ldots, 0.95\}$ in steps of 0.05, which is more strict and sensitive to localization quality [21, 43].

Table 4.1: Detection results on the cyclist-centric test set (106 images; 396 annotated instances). We report overall precision, recall, and mean Average Precision at IoU thresholds 0.50 (mAP@0.5) and 0.50:0.95 (COCO AP). Per-class AP@0.5 is shown for the classes present in the test set.

Overall metrics	
Precision (P)	0.681
Recall (R)	0.657
mAP@0.5	0.732
mAP@0.5:0.95	0.494

Per-class AP@0.5	
Car	0.816
Cyclist	0.715
Pedestrian	0.744
Truck	0.653

On this cyclist-centric test set, the detector attains $P = 0.681$, $R = 0.657$, mAP@.50 of 0.732, and mAP@0.50:0.95 of 0.494. Under the Microsoft COCO evaluation protocol [43], mAP@0.50:0.95 denotes the mean Average Precision (AP averaged across classes and over IoU thresholds $\tau \in \{0.50, 0.55, \ldots, 0.95\}$ in steps of 0.05); it is stricter than AP.50, which evaluates only at $\tau = 0.50$. Our mAP@0.50:0.95 is comparable in order of magnitude to established real-time baselines reported on the COCO benchmark, for example YOLOv4 reports mAP@0.50:0.95 of 0.435 and mAP@50 of 0.657 [5]. Given the cyclist point-of-view viewpoint and the relatively small training set, these values support the conclusion that the model is well adapted to the target domain.

Figure 4.3 shows the normalized confusion matrix with true classes on the x-axis (columns) and predicted classes on the y-axis (rows). Columns are normalized by the number of ground-truth instances of that class, so each column sums to 1 and can be read as; given a true class X, what fraction was predicted as each class? Diagonal cells therefore indicate correct classifications; off-diagonals indicate misclassification between classes. The background column collects false positives (detections that did not match any ground-truth object at IoU ≥ 0.50), while the background row reflects false negatives (ground-truth objects that were missed by the detector). Most residual errors occur between pedestrian and cyclist, which is expected in crowded scenes with partial occlusions and similar silhouettes; car is comparatively well separated, and truck shows more confusion likely due to lower support and visual similarity to buses (which were excluded at evaluation time).

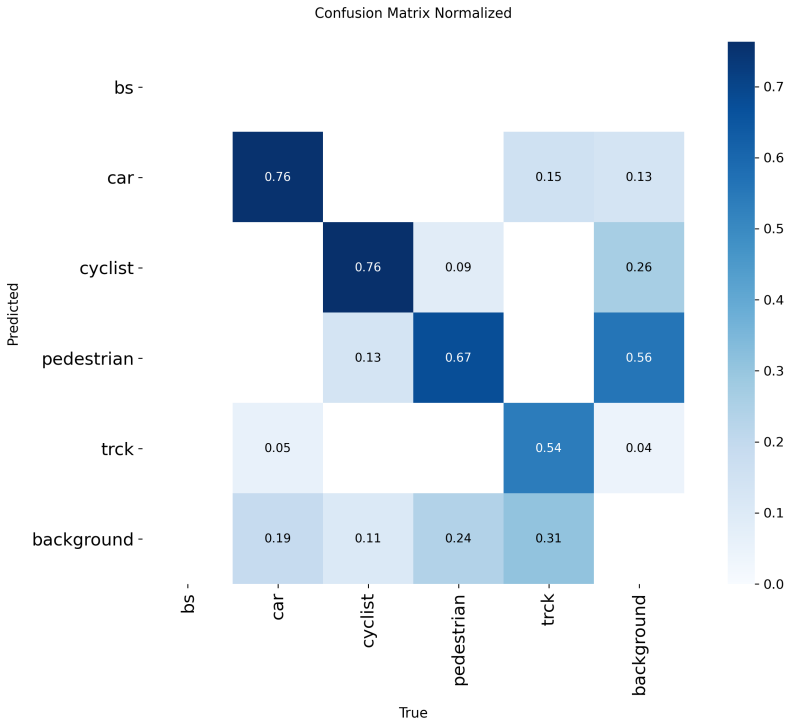


Figure 4.3: Column-normalized confusion matrix on the custom test set (normalization by true class). Values in each column sum to 1 and show how often a true class (x-axis) is predicted as each class (y-axis). One-to-one matching between predictions and ground truth uses IoU ≥ 0.50; the background column/row summarize false positives/false negatives, respectively.For the class names, bs = bus and trck = truck.

During inference, a confidence threshold (joint probability between there being an object at the given position for a given class) of 0.27 was used throughout the pipeline, this means that any inferred detection from the model having a confidence score below the threshold is not a part of the output. This value was chosen empirically to ensure high recall in realistic, often occluded cycling environments. A slightly lower threshold was used to avoid missing important objects in real-world scenarios, such as cyclists emerging from occlusions. The figure 4.4 below shows an example detection output from Video 2 of the analysis set.

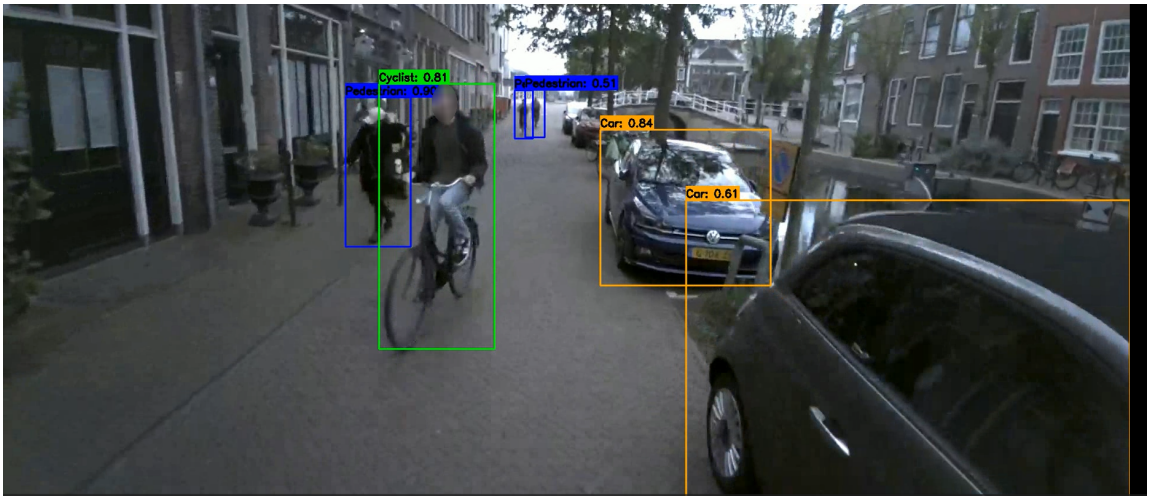


Figure 4.4: Example model output at 0.27 confidence score, classes are given per colour, blue = pedestrian, orange = car, green = cyclist. The confidence score for each detection is given above the bounding box.

Now that the fine-tuned model presented satisfactory results and was able to detect Delft traffic participants, the next step (object tracking) became possible, as without consistent detections across frames it is nearly impossible to track bounding boxes accurately across frames, especially considering the nature of the footage which is riddled with head movements causing changes of perspective and different expected bounding box positions.

4.3. Object Tracking

Off-the-shelf Multi-Object-Tracking (MOT) methods as described in appendix A.3 underperformed on head-mounted cyclist POV video due to rapid head-motion, strong perspective/scale changes, and intermittent occlusions. At the same time, the fine-tuned object-detector produced high-recall, temporally stable detections on the classes of interest. This motivated a detection-driven tracker that minimizes modelling assumptions, relies primarily on geometry, and keeps only the lightest prediction needed to bridge very short frame gaps.

Design principles: The tracker is intentionally simple: (i) prioritise fresh detections over strong dynamical priors, (ii) associate with Intersection over Union between the predicted and ground truth bounding box areas (IoU) B.2 first and fall back to a centre-distance gate that tolerates scale and viewpoint change (distance between bounding box centers), (iii) apply lightweight, damped velocity extrapolation when unmatched, (iv) optionally resurrect short gaps with a conservative interpolator.

Association logic: For each frame, given a set of detections $D = \{d_j\}$ (each d_j contains bounding box coordinates (db_j, a confidence score associated with the prediction and its class) and active tracks $T = \{t_i\}$ (each t_i stores a bounding box tb_i, velocity v_i, age/hits, class, confidence), we perform one-to-one matching:

1. *IoU gate:* For each t_i, compute $\mathrm{IoU}(tb_i, db_j)$ for all unused d_j. If any $\mathrm{IoU} > \tau_{\mathrm{iou}}$ (empirically $\tau_{\mathrm{iou}} = 0.2$), assign the d_j with the highest IoU.

2. *Centre-distance fallback:* If no IoU match, compute centre points

$$c(tb_i) = \left(\tfrac{x_1+x_2}{2}, \tfrac{y_1+y_2}{2}\right), \quad c(db_j) = \left(\tfrac{x_1+x_2}{2}, \tfrac{y_1+y_2}{2}\right).$$

   Let $\Delta c_{\mathrm{dist}} = \|c(tb_i) - c(db_j)\|_2$. This formula denotes the Euclidean distance between track and detection box centers. If $\Delta c_{\mathrm{dist}} < \tau_{\mathrm{cen}}$ (empirically $\tau_{\mathrm{cen}} = 150$ px), and multiple candidates remain, the closest one is accepted.

3. *Update matched tracks:* When $tb_i \leftrightarrow db_j$:
   - We replace t_i's old bounding box coordinates, confidence score and class with the new detection d_j's information .
   - We reset the track age: $\mathrm{age}_i \leftarrow 0$, and increment hits: $\mathrm{hits}_i \leftarrow \mathrm{hits}_i + 1$.
   - *Velocity smoothing.* Let old and new track centres be $\mathrm{c}^{\mathrm{old}}, \mathrm{c}^{\mathrm{new}}$ and instantaneous displacement $\Delta\mathrm{c} = \mathrm{c}^{\mathrm{new}} - \mathrm{c}^{\mathrm{old}}$. Update the velocity with exponential smoothing

$$v_i \leftarrow (1-\lambda)\, v_i + \lambda\, \Delta c, \qquad \lambda = 0.3$$

     which damps noise yet adapts quickly to changes.

4. *Predict unmatched tracks:* If t_i is unmatched:
   - Increase age: $\mathrm{age}_i \leftarrow \mathrm{age}_i + 1$.
   - If $\mathrm{age}_i < \mathrm{max\_age}$ (empirically 4 frames at 25 fps), predict the box forward by shifting with the last velocity, but damp vertical motion to reduce pitch/jitter artefacts:

$$tb_i^x \leftarrow tb_i^x + v_x, \qquad tb_i^y \leftarrow tb_i^y + \alpha_y\, v_y, \quad \alpha_y = 0.35.$$

     Otherwise, drop the track.

5. *Spawn new tracks:* For unused detections with $\mathrm{conf} > 0.27$, create a new track with initial velocity $\mathrm{v} = 0$.

6. *Confirmation:* Only tracks with $\mathrm{hits} \geq \mathrm{min\_hits}$ and $\mathrm{age} = 0$ are exported (empirically $\mathrm{min\_hits} = 3$) to suppress spurious short-lived track IDs.

The overall procedure is greedy (closest bounding box detection is matched to the track) and linear in the number of tracks and detections per frame. An optional post-stage track interpolator can resurrect tracks that disappear for ≤ 3 frames by replaying the last valid state with decayed confidence; in the final experiments this was kept off to avoid hallucinating identities in crowded scenes.

In practice we set a permissive overlap gate of $\tau_{\mathrm{iou}} = 0.20$ to tolerate rapid apparent scale and aspect changes, and fall back to a centre-distance gate of $\tau_{\mathrm{cen}} = 150$ px when overlap is unreliable (ex: front/back views that yield narrow, stacked boxes). Velocities are updated with exponential smoothing, $v_i \leftarrow (1 - \lambda)\, v_i + \lambda\, \Delta c$ using $\lambda = 0.3$, which filters frame-to-frame noise while remaining responsive to direction changes. Unmatched tracks are propagated for at most $\mathrm{max\_age} = 4$ frames at 25 fps; during this short prediction window the vertical component is damped by $\alpha_y = 0.35$ to counter pitch and road vibration. New tracks are spawned only from detections with confidence > 0.27 to favour recall in occluded scenes, and tracks are exported once they have been observed in at least $\mathrm{min\_hits} = 3$ frames to suppress transient false positives.

To summarise this tracker's specificities and why we needed them, head-mounted cyclist video is dominated by camera head-motion rather than smooth, object-centric dynamics. Prioritising fresh detections and using only mild, short-horizon prediction prevents drift when constant-velocity assumptions are violated by rapid viewpoint changes. The two-stage association, IoU then centre distance remains stable across large apparent scale changes and profile $\leftrightarrow$ frontal transitions, where bounding-box overlap can collapse while centres remain coherent. Damping vertical motion reduces spurious fragmentations caused by pitch and road texture, and the combination of a short lifetime with a modest confirmation window lowers identity switches and one-frame ghosts. Together, these choices exploit the detector's high recall while providing just enough geometry to bridge micro-dropouts without inventing motion.

Figure 4.5 shows a representative close-pass sequence. The hybrid tracker maintains a single ID from first appearance at long range through the overtake, preserving the segment most valuable to workload inference, which was not the case with the off-the-shelf trackers A.6.

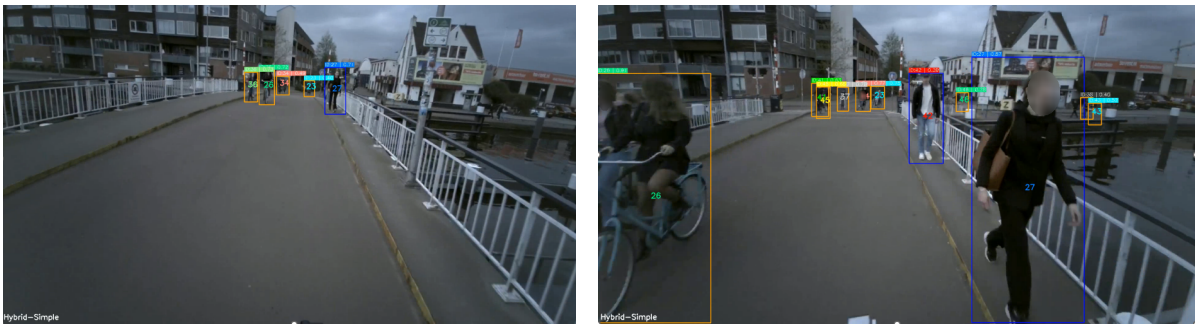


Figure 4.5: Hybrid tracker on a close-pass sequence: a single, continuous ID (number at the center of bbox) is preserved from far to near and through the overtake, enabling reliable proximity/velocity metrics.

The simplicity that improves robustness also imposes limits: without the algorithm learning specific information about each track (appearance cues), ID switches (when a track gets associated to a completely different object from one frame to the next) can occur in dense scenes when multiple nearby objects cross within the centroid gate; tracks do not survive long occlusions ($> \mathrm{max\_age}$) and are re-instantiated with new IDs upon re-entry. These behaviours are accounted for in later stages by (i) focusing on event-level metrics that are less sensitive to identity persistence, (ii) extracting proximity/velocity features over short temporal windows, and (iii) discarding events with insufficient temporal support.

4.4. Monocular Depth Estimation

Traffic analysis from cyclist point-of-view video requires estimating distances to nearby road users to compute proximity metrics and relative velocities. Since our pipeline relies on a single forward-facing camera without additional sensors, depth must be inferred from monocular images alone. This section describes our approach to extracting metric distance estimates from head-mounted footage, addressing the challenges of depth estimation noise, tracking inconsistencies, and the conversion from relative to absolute distance measurements. The resulting depth pipeline provides the spatial foundation for all downstream traffic feature computation.

4.4.1. Model Selection

Several downstream traffic-condition metrics in this work require distance or distance change (ex: proximity metrics and relative approach speeds). With monocular, head-mounted video as the only visual sensor, depth must be inferred from single frames. We adopt Depth Anything V2 (Large) as our monocular depth backbone because it offers fine-grained detail and efficient inference compared to recent diffusion-based approaches [80]. Depth Anything V2 achieves these properties by (i) using synthetic labelled data for teacher training, (ii) scaling model capacity, and (iii) distilling to student models via large-scale pseudo-labelled real images [80]. [1]

Alternative foundations include Mixing Datasets for Accurate Depth Estimation (MiDaS), which output robust relative (scale and shift ambiguous) depth [52, 35], and ZoeDepth, which combines relative pre-training with metric heads for direct conversion [37, 36]. We chose Depth Anything V2 (Large) for its accuracy/efficiency balance and strong outdoor generalization, and use its per-pixel depth as our base signal; conversion to metric distance is handled later via a dedicated calibration step (Section 4.4.4).

4.4.2. Extraction Method

Depth Anything V2 produces a dense depth map for every frame, i.e., a per-pixel estimate of scene depth (in model units). For each tracked object, we reduce the many pixel values inside its bounding box to a single, temporally stable number per frame. After evaluating several summary statistics (median, 75th percentile, 85th percentile), we adopt the median of valid pixels, where ”valid” excludes zero values.

To verify temporal stability, we analyzed a stratified sample of 20 tracks. Rather than random sampling, we divided all tracks by their length (number of frames) into four equal groups (very short to long trajectories) and sampled 5 tracks from each group, ensuring representation across different tracking scenarios. For each selected track, we analyzed up to 30 frames: if a track contained ≤30 frames, we used all frames; if longer, we randomly sampled 30 frames. This yielded 215 frames total across 21.8M pixels. For each track, we computed frame-to-frame variability by calculating the standard deviation of each summary statistic across its frames, then averaged these standard deviations across all 20 tracks. The median yielded the lowest average temporal standard deviation (median: 11.38; 75th percentile: 13.00; 85th percentile: 14.11, in model depth units), confirming superior stability for head-mounted, egocentric video. Accordingly, for every track and frame we store the median depth of all valid pixels within the bounding box; percentile summaries are retained only for diagnostic validation. Conversion from model units to metric distances is detailed in Section 4.4.4.

[1] A teacher–student setup is a form of knowledge distillation in which a large, accurate teacher network first learns the task with full supervision, and its outputs are then used to supervise a typically smaller, faster student network [33]. In Depth Anything V2, the teacher is trained purely on synthetic images that come with perfect depth labels. This teacher is then run on a large number of unlabeled real images to generate pseudo-labels (predicted depth maps). Finally, student models are trained on those pseudo-labeled real images to match the teacher’s depth predictions (ex: via standard regression losses), which transfers the teacher’s knowledge while adapting to real-image statistics and enabling efficient deployment.

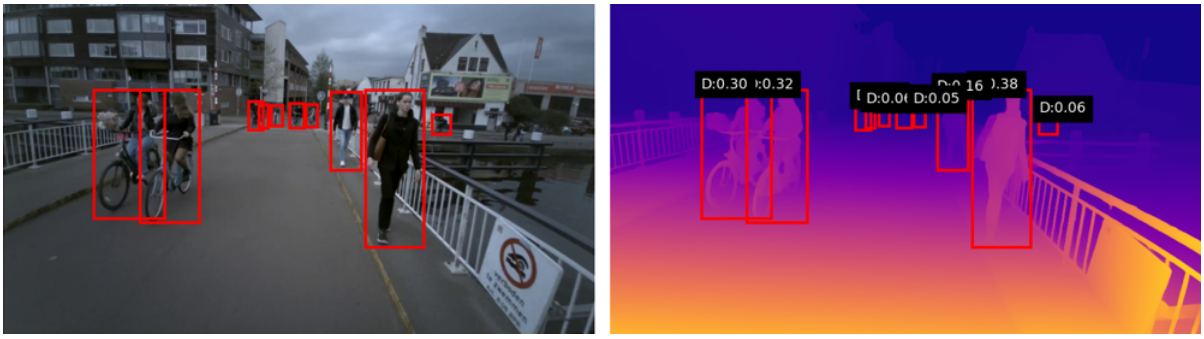


Figure 4.6: On the left we have the original track bounding boxes and on the right, the median extracted depth value per track (Closer = larger depth value).

This result shows consistent depth estimations for the different tracks in that frame, this consistency is lost however when looking at a single track, across consecutive frames. Handling model noise and Depth estimation overshoots between frames will be discussed in the next section.

4.4.3. Handling Noisy Data

Before any conversion to metric distance, we needed to stabilise the relative per–track depth series because two error sources dominated our head–mounted footage: (i) Depth estimation errors between frames and (ii) tracking errors in the form of ID switches. Aggressive smoothing alone risks smearing across identity errors, while conservative filtering leaves residual jitter and unrealistic motion. We therefore applied a staged procedure that is robust to outliers, respects basic motion plausibility, and remains computationally simple.

Depth and Tracking Errors

Across videos, raw per–track depth z_t (median per–box pixel depth at 25 Hz) exhibited:

1. **Estimation noise:** high–frequency fluctuations around the slow approach/retreat trend due to monocular depth sensitivity to lighting, pose, and blur [80, 52].
2. **Tracking induced discontinuities:** short track gaps and occasional identity mistakes, causing abrupt jumps in z_t or piecewise inconsistent segments that should not be smoothed across.

Representative examples are shown in Fig. 4.7 and 4.8.

Aggressive Smoothing

We first clean each track independently (no cross–track operations), building a history per `track_id` from the frame wise detections. The cleaning consists of three passes:

Rolling MAD outlier replacement (Hampel filter): For a small centred window (± 2 frames), compute the local median med_t and median absolute deviation MAD_t for the raw per–track depth z_t (median per–box pixel depth at 25 Hz). If a sample deviates by more than $k\,\mathrm{MAD}_t$ (we used $k{=}1.5$), replace it with med_t:

$$|z_t - \mathrm{med}_t| > k\,\mathrm{MAD}_t \;\Rightarrow\; z_t \leftarrow \mathrm{med}_t.$$

This retains edges while removing impulsive spikes [29].

Reciprocal-space outlier detection: To complement the above MAD outlier filtering, we detect implausible per frame depth changes using a reciprocal transform. Depth estimation outputs exhibit non-linear relationships with physical distance where equal numerical changes represent different physical displacements depending on object proximity [66]. To achieve uniform outlier detection, we apply:

$$r_t \;=\; \frac{K_{\text{filter}}}{z_t}, \qquad v_t \;=\; \frac{|r_t - r_{t-1}|}{\Delta t},$$

with $K_{\text{filter}}{=}300$ and $\Delta t{=}1/25$ s. If the reciprocal-space rate of change exceeds an empirical threshold ($v_t > 6.0$), we correct by interpolation:

$$\text{if } v_t > 6.0: \quad r_t \leftarrow \tfrac{1}{2}(r_{t-1} + r_{t+1}), \quad z_t \leftarrow \frac{K_{\text{filter}}}{r_t}.$$

This preprocessing removes obvious errors uniformly across depth ranges, with the threshold value determined empirically to balance artifact removal against preservation of genuine motion patterns.

Weighted–median temporal smoothing: After outlier removal, residual measurement jitter and small inconsistencies remain. We apply a sliding weighted–median filter (window size $W{=}10$ frames) with higher center weighting ($3\times$) to ensure temporal consistency while preserving motion edges better than linear averaging [38]. This step prepares clean, consistent depth series for subsequent kinematic filtering.

Kalman Filtering

After robust cleaning, we apply a 1D Kalman filter on depth to impose smooth kinematics and bridge very short gaps. The state is $x_t = [d_t,\ \dot{d}_t,\ \ddot{d}_t]^\top$ (depth, depth–rate, acceleration). With a unit–frame time step [3].

$$x_t \ = \ Fx_{t-1} + w_t, \qquad F = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad y_t \ = \ Hx_t + v_t, \quad H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

We use the cleaned depth as the measurement y_t, with $w_t \sim \mathcal{N}(0, Q)$ and $v_t \sim \mathcal{N}(0, R)$. Following each prediction we clamp acceleration to $|\ddot{d}_t| \le a_{\max}$ (we used $a_{\max}{=}1.0$ in depth units per frame^2) to prevent unphysical transients:

$$\ddot{d}_t \leftarrow \mathrm{clip}(\ddot{d}_t,\ -a_{\max},\ a_{\max}).$$

We initialise d_0 with the first cleaned observation, use modest process noise on acceleration, and a measurement noise tuned per sequence. Because tracks are processed independently, the filter naturally resets when a new `track_id` starts.

Why this staging works: The Hampel filter removes statistical outliers within local windows; the reciprocal velocity check prevents implausible frame-to-frame changes; the weighted median ensures temporal consistency across larger scales; the Kalman filter then enforces smooth kinematics. This multi-scale approach addresses different error characteristics systematically rather than relying on any single filtering method.

In figure 4.7 we compare the raw depth signal from a noisy but seemingly consistent track ID, the peaks that are shown are noise overshoots. As an object draws closer, especially in mid range, the model tends to overestimate depth, which will cause a large jump in between two frames. This problem benefits the most from aggressive smoothing. The second graph shows the smoothed depth (in green) as well as the smoothed depth with additional kalman filtering (in blue) which ensures smoother trends compared to only relying on the aggressive smoothing. This filtering outputs a much more realistic depth change than the raw data.

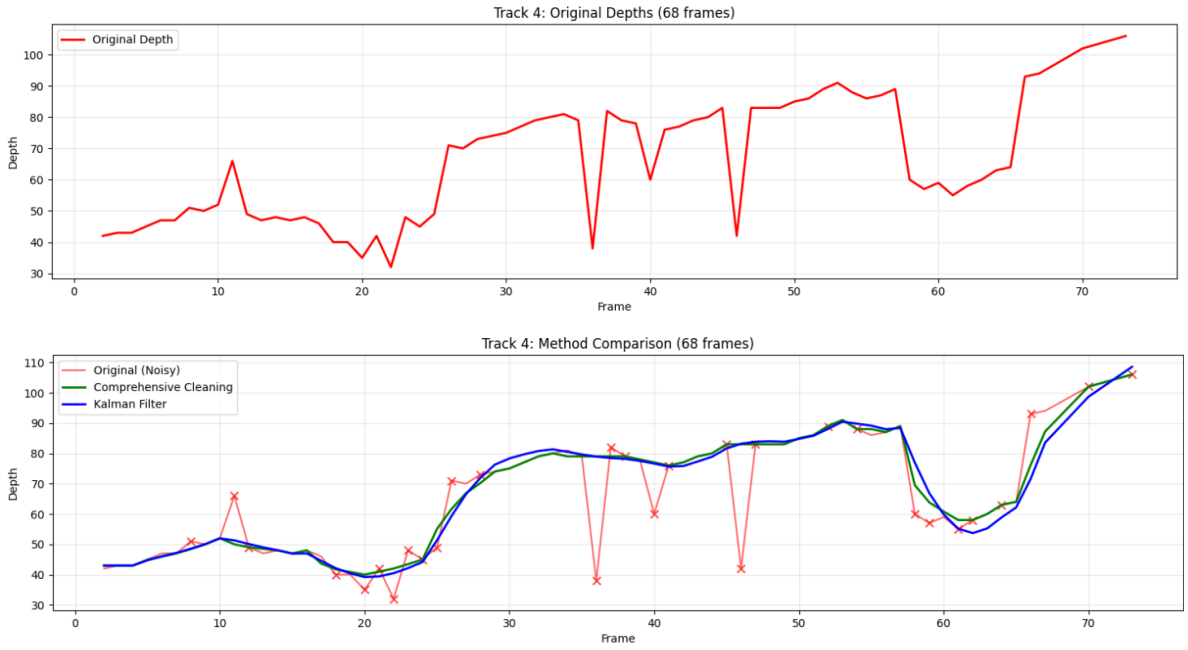


Figure 4.7: Noisy Depth (median extracted pixel depth across 75 frames) data from a short Track ID

In figure 4.8 the first graph shows the raw depth signal from an extended track which lasts almost 12 seconds (292 frames at 25fps) and most likely does suffer from an ID switch as this has a higher chance of happening with longer tracks. Nevertheless, it is difficult to discern with certainty whether an overshoot was caused by depth estimation or by a tracking error, especially for longer trends. This is why the smoothing stays somewhat conservative in order to preserve motion information from a potential tracking error while trying to reduce depth estimation noise as much as possible.

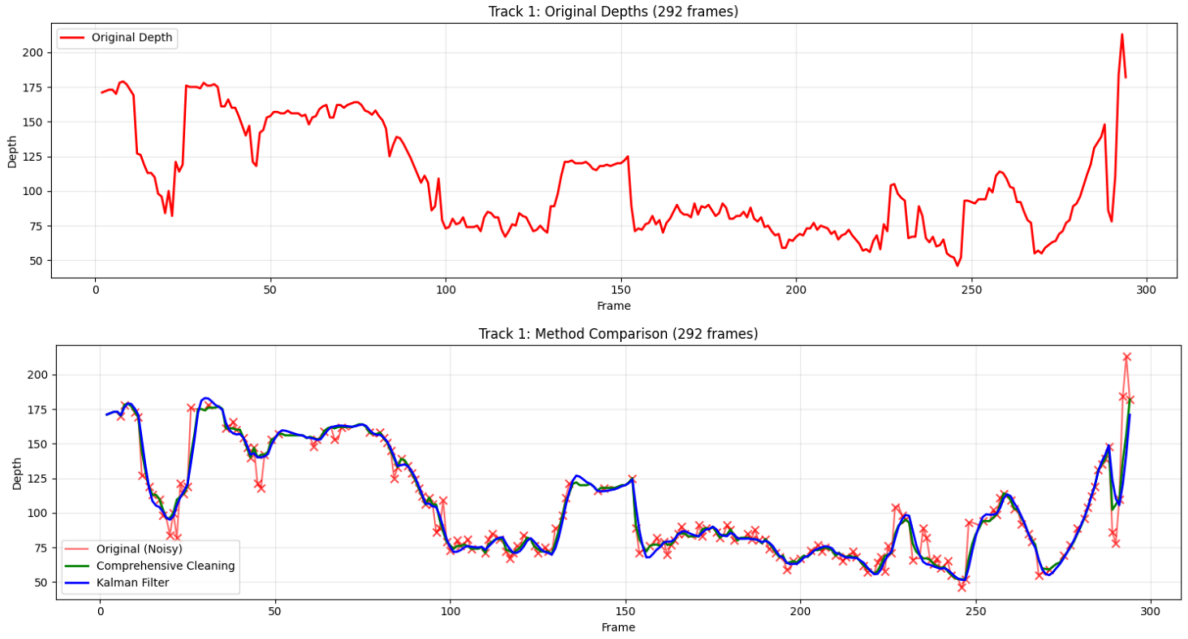


Figure 4.8: Noisy Depth (median extracted pixel depth across 293 frames) data from a longer track with possible ID switches

Table 4.2 reports the filtering effectiveness for all the three videos across all their tracks. In order to better understand the filtering summary in this table, it is important to note that depth values vary

between 0 and 255 (in arbitrary units). With most frames being corrected (around 94% of of each video segment), and the average depth correction for a single frame being between 2 and 3.5, this confirms that the filtering is being applied consistently.

Table 4.2: Effect of depth-signal cleaning per video (absolute change in per-frame relative depth).

Video	Total tracks	Changed (count)	Changed (%)	Avg. change	Max change
Video 1	63,675	59,600	93.6	3.501	139.703
Video 2	59,585	56,224	94.4	2.713	126.134
Video 3	65,021	61,157	94.1	2.263	108.008

4.4.4. Converting to absolute metric distance

Downstream traffic metrics, especially velocity based metrics are most interpretable in meters. The depth network provides a per–pixel relative value d_{rel} (larger = closer, up to a global scale/shift). We therefore need a mapping from d_{rel} to an *absolute* distance d_{m} in meters.

Options considered (and why they were rejected for head-mounted POV):

Stereo / LiDAR / radar fusion (absolute sensors): Instrumented bicycles and automotive platforms attach sensors that provide metric range directly; depth networks are then used only for densification or semantics [14]. This is reliable but outside the scope of a low-cost, monocular, head-mounted setup.

Pose from n points (PnP): While urban environments contain numerous visible objects, PnP requires establishing correspondences between specific scene points with known metric 3D coordinates and their 2D image projections across multiple frames [79, 42]. This creates several fundamental challenges: (i) the same physical points must remain visible and trackable across frames as the cyclist moves through the scene, (ii) these points require prior surveying to establish their metric coordinates, and (iii) robust 2D-3D correspondences must be maintained despite changing viewpoints, occlusions, and lighting conditions. In our dynamic cycling footage, objects continuously enter and exit the field of view, making it impractical to maintain the stable correspondences required for reliable PnP solutions. Additionally, any calibration would be scene-specific, requiring recalibration for each route, which contradicts our objective of developing a generalizable monocular traffic analysis framework.

Planar ground homography / inverse perspective mapping (IPM): This approach attempts to estimate object distances by assuming the road surface forms a flat plane and using geometric relationships to map image coordinates to ground plane positions [45, 32]. The method faces several critical limitations for our case. First, the flat ground assumption is systematically violated by road curvature, slopes, intersections, and elevation changes. Second, the technique is extremely sensitivity to camera orientation changes, small head movements during cycling create large distance estimation errors, particularly for distant objects where the geometric leverage amplifies any pitch or roll variations. Third, and most fundamentally, the method only provides distances to the ground plane, not to traffic objects themselves. While IPM can theoretically estimate where a vehicle's base contacts the road, the visible portions of traffic objects (vehicle bodies, cyclist torsos, pedestrian heads) extend above this ground plane at unknown heights. Determining actual object depth requires additional assumptions about object geometry or height, reintroducing the measurement uncertainties that IPM was meant to avoid. Finally, the approach requires precise horizon detection for each video sequence, contradicting the goal of developing a generalizable monocular framework for diverse cycling scenarios.

Choice of one global scale in this work: Given the limitations above, we adopted a simple, reproducible calibration using a single global scale K that maps relative depth to meters via

$$d_{\text{m}} = \frac{K}{d_{\text{rel}}}.$$

The scale parameter was determined empirically by testing a range of plausible values and selecting $K{=}235$ based on minimizing relative error when validated against reference measurements. These reference measurements utilized standardized Dutch cycling infrastructure elements, specifically the spacing between white centerline markings that separate bidirectional bike lanes. Such markings follow regulated design standards, providing consistent reference distances that can be measured using satellite imagery for validation purposes.

This approach trades precision for practical implementability: systematic errors become quantifiable, the pipeline remains computationally simple, and no per-frame calibration is required. The method achieves sufficient accuracy for traffic behavior analysis while maintaining generalizability across different cycling routes.

Calibration Validation Results Validation against 16 reference measurements from 5 frames at different points in video 1 yielded a mean absolute error of 1.71 ± 1.46 meters and mean relative error of 16.9 ± 14.0%. Approximately 75% of measurements fell within 25% relative error. Error distribution

shows considerable variance (0-50% relative error) but no systematic bias, indicating that while individual predictions may vary, the global scale provides reasonable average performance across the measurement range. Because a single global scale is used, some measurement-to-measurement variation remains; this trade-off was preferred over per-frame fitting to maintain reproducibility and computational simplicity. Figure 4.10 shows an example calibration to google earth reference image pair.

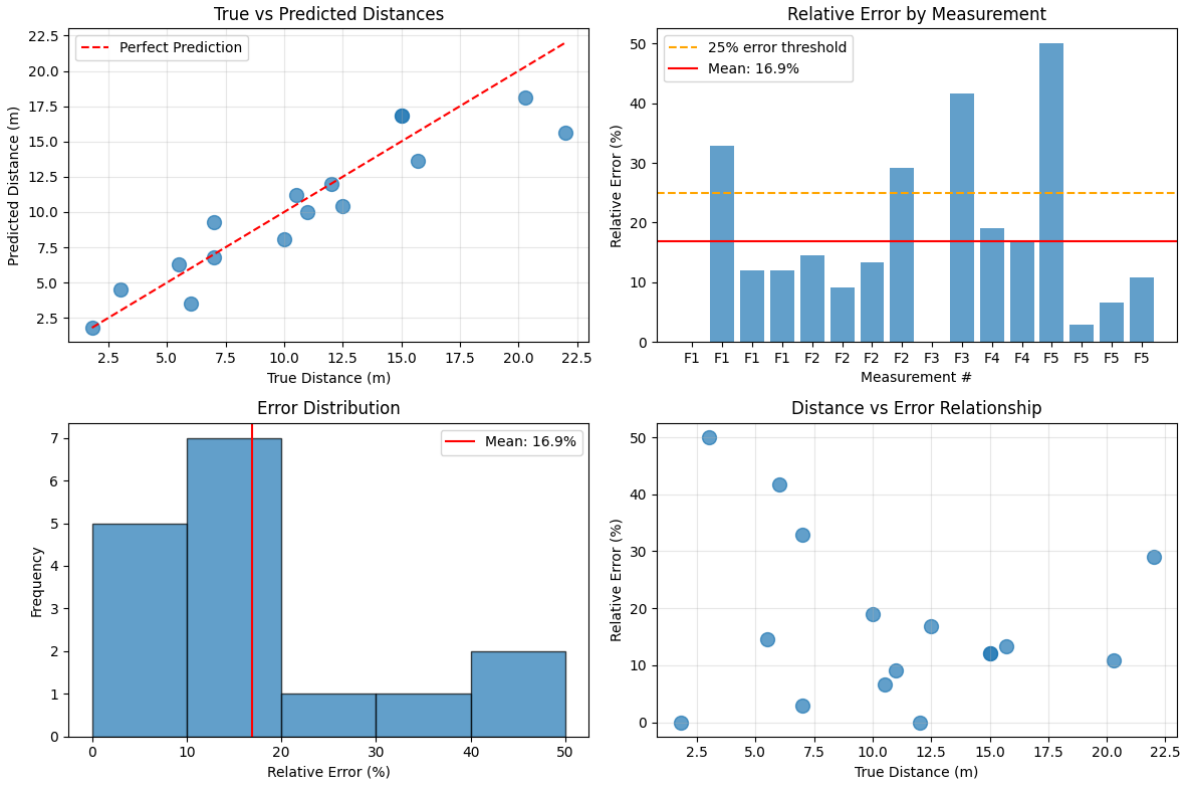
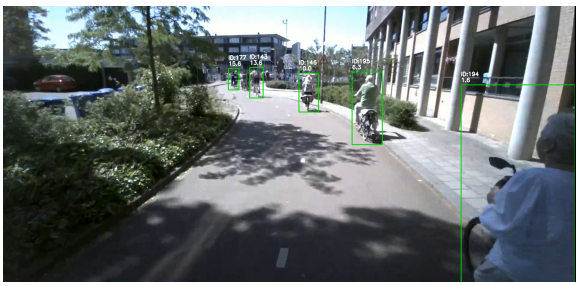


Figure 4.9: Calibration sanity check using a single global scale K. Top-left: predicted vs. true distance (red dashed line = ideal). Top-right: per-measurement relative error with mean (red) and a 25% reference (orange) F1-F5 corresponds to the specific frame for which these references were taken. Bottom-left: error distribution. Bottom-right: relative error vs. true distance.



(a) Frame with tracks and measured d_m.

(b) Top-down reference with ground-truth spacings.The orange lines represent the line distances between the cyclist and observed tracks in the image on the left.

Figure 4.10: Single-scale calibration example: one annotated frame and its google earth view reference

Notes and limitations: (i) The reference frames were selected to avoid obvious depth failures, so the reported error is a slightly optimistic lower bound. (ii) A global K ensures consistent, easy-to-interpret bias but cannot compensate local pose changes or scene-specific geometry. (iii) For applications requiring sub-meter accuracy at mid-range, additional per-session calibration cues or stereo would be advisable; for the present workload study, the above accuracy was sufficient to rank proximity and approach events reliably.

4.5. Metric Extraction

With per–track distances now expressed in meters and temporally cleaned (Section 4.4), we extract traffic-exposure metrics that map to known drivers of cyclist workload: who is around the rider, how close they are, how much of the forward view they occupy, and how quickly interactions evolve along the line of sight. We group outputs into:

1. **Basic presence and proximity** (counts, heavy-vehicle share, near-field object counts, field-of-view clutter).
2. **Velocity metrics** (longitudinal approach/recede speeds and their directional composition).
3. **Cyclist Velocity** metric extracted from the available GPS data.

The choices are grounded in prior work: close passes and heavy-vehicle exposure are associated with stress and risk; visual scene clutter increases attentional demand; and high approach speeds compress reaction time [73, 14, 28, 48].

4.5.1. Extracting Basic Metrics

For each video frame t, we have a set of active tracks with class label c_k, bounding box $b_k = [x_1, y_1, x_2, y_2]$, and corrected distance d_k (m). To reflect the cyclist's forward focus, we also define a central horizontal band spanning $[20\%, 80\%]$ of the image width.

Counts and shares: We report the total number of actors and key composition ratios: (i) *heavy-vehicle share* (buses+trucks over all actors), motivated by their outsized safety impact and occlusion effects [28]; (ii) *pedestrian proportion*, capturing the prevalence of unpredictable, low-speed actors in shared spaces.

Near-field tiers (3 m / 5 m): We count the number of objects within 3 m and 5 m of the cyclist (Tier 1, Tier 2). Fixed near-field cut-offs emphasise interactions most likely to draw attention, induce evasive manoeuvres, or trigger stress (3–5 m aligns with common passing/interaction distances in urban cycling [73]).

Field-of-view (FoV) clutter: To approximate instantaneous *visual load*, we compute the fraction of pixels covered by tracked bounding boxes ("clutter ratio"). We report clutter over the full frame and over the central band, both overall and censored to near-field (< 5 m, < 3 m). Unlike raw counts, clutter naturally weights large/near objects more and is comparatively robust to short ID switches because area integrates over small box jitters. The central-band variant downweights edge artefacts during head turns.

Basic metric design: These metrics capture complementary aspects of the traffic environment: raw counts reflect density, composition ratios indicate traffic mix characteristics, near-field tiers emphasize proximal interactions, and visual clutter quantifies instantaneous scene complexity through pixel coverage.

4.5.2. Extracting Velocity Metrics

Depth noise and occasional ID switches make single-step finite differences unreliable. We therefore estimate *longitudinal* relative speeds (along the camera forward axis) with a multi-stage, consensus-based approach that only reports a velocity when there is sustained, self-consistent motion evidence. Practically, the mere presence of a velocity estimate already flags a meaningful approach/recede episode likely to matter for workload.

Windowed derivatives with physical limits: For each sufficiently long track, we slide a short temporal window (ex: 10 frames at 25 Hz) and compute frame-to-frame longitudinal speeds from metric distance:

$$v_{i \to i+1} \;=\; \frac{d_{i+1} - d_i}{\Delta t} \quad [\mathrm{m/s}], \qquad v_{\mathrm{km/h}} = 3.6\, v.$$

We remove impossible spikes using *class-specific physical limits* (ex: pedestrians ≤ 25 km/h, cyclists ≤ 50 km/h, cars/buses/trucks ≤ 60–70 km/h), which removes outliers from residual depth jitter or brief mismatches.

Consensus grouping (noise rejection): Within each window, remaining speeds are clustered on a line (simple proximity threshold) and we keep the *largest group* of mutually similar values. From this group we compute: (i) magnitude $\bar{v}$ (mean absolute relative speed); (ii) signed speed $\tilde{v}$ (mean with sign); (iii) direction label: *approaching* if $\tilde{v} < 0$ (distance decreasing), *receding* if $\tilde{v} > 0$, else *uncertain* when signs are mixed; (iv) direction confidence as the normalised difference between positive/negative counts. This step rejects isolated spikes while preserving genuine, sustained motion.

De-duplication across overlapping windows: A long track yields multiple overlapping windows. We group consecutive windows with small gaps, optionally discard the bottom $\sim$30% by a quality score (consensus size, direction confidence, data coverage), and produce a single, weighted-average measurement per group. This reduces redundancy and prevents over-counting during long interactions.

Frame-level velocity series: Finally, we map consolidated measurements back onto the frames they span and aggregate per frame: overall average radial speed, averages by direction (approaching/receding/uncertain), counts by direction, speed dispersion (standard deviation), and a simple *movement complexity* index: the entropy of the directional mix, normalised to $[0, 1]$ so that higher values indicate more conflicting motion flows. These signals capture how quickly the scene is "closing in" and how coherent or chaotic its motion is. This metric is defined mathematically below [61]:

Let $n_{\mathrm{app}}(t)$, $n_{\mathrm{rec}}(t)$, $n_{\mathrm{unc}}(t)$ be counts of episodes per direction and $n_{\Sigma}(t)$ their sum; proportions $p_c = n_c/n_{\Sigma}$. Define *movement complexity* (normalised entropy)

$$H(t) = -\frac{\sum_{c\in\{\mathrm{app,rec,unc}\}} p_c \log p_c}{\log 3} \quad \in [0, 1],$$

with $0 \log 0 := 0$ by convention in order to not cause an error when $p_c = 0$.

Scope and limitations: These are *longitudinal* speeds (only from depth changes, perpendicular to the image plane); purely lateral swerves without range change are not captured (appropriate for headway/closing dynamics but conservative for side-passes). The consensus pipeline trades some temporal granularity for robustness to depth jitter and short ID switches.

GNSS data and Cyclist-Velocity

To recover the cyclist's speed, we read the satellite-position (GNSS) logs from the recording, fill small gaps by simple linear interpolation, and then compute speed from the change in latitude/longitude over time (details in the next paragraph). The receiver is an ArduSimple simpleRTK2B [62]. In ordinary standalone use (no RTK corrections) its stated horizontal position accuracy is about 1.5 m circular error probable, meaning that roughly half of all fixes fall within a 1.5 m radius of the true position [78]. This positional uncertainty is the main source of error in the derived speed.

Instantaneous speed is then obtained from consecutive resampled latitude/longitude pairs using the great-circle distance between points and a fixed time step Δt [39]:

$$v_{\mathrm{cyc}}(t_i) = \frac{\mathrm{dist}_{\mathrm{gc}}\big(\phi_{i-1}, \lambda_{i-1};\, \phi_i, \lambda_i\big)}{\Delta t} \quad \text{with} \quad \Delta t = 1/25 \text{ s}.$$

To reduce GNSS jitter, the instantaneous result is smoothed with a short (5-sample) centered moving average, yielding $v_{\mathrm{cyc}}^{\mathrm{smooth}}$. In practice this produces a slightly "inertial" velocity signal: small frame-to-frame changes are absorbed by the smoother, and only sustained speed changes produce a visible update. This is desirable for our use case (robust subtraction from relative velocities) but implies a temporal lag relative to sudden speed changes.

Protocol to Evaluate relative velocity calculations: To verify the calculated relative-velocity estimates, we used short windows in a video where clearly stationary tracked objects (stationary Cyclists

and parked cars in this case) had a relative velocity calculation associated to them. For each stationary track and target frame range, we extracted the smoothed cyclist GNSS speed over the same frame range and we expect:

$$v_{\mathrm{rel}} \approx v_{\mathrm{cyc}}^{\mathrm{smooth}},$$

since for a static object the cyclist motion fully explains the relative change in metric distance. We compared this to the measured relative velocity magnitude and reported absolute errors.

Table 4.3: Relative-velocity sanity check using stationary objects (13 episodes).

Metric:	Mean	Median	Std. dev.	Min	Max
Absolute error (km/h):	4.45	3.68	2.88	0.16	10.11

Error buckets (count, %):	Excellent < 2 km/h	2 (15.4%)
	Good 2–5 km/h	6 (46.2%)
	Fair 5–10 km/h	4 (30.8%)
	Poor ≥ 10 km/h	1 (7.7%)

In table 4.3 each “episode” is a short, continuous segment where the tracked object is stationary. For every episode we compute the framewise absolute error $|v_{\mathrm{rel}} - v_{\mathrm{cyc}}^{\mathrm{smooth}}|$ and average it over the episode; the table then summarizes these per-episode averages across all $n{=}13$ episodes, in km/h. The bucket list below the table simply counts how many episodes fall into common error ranges to give a quick sense of dispersion.

Interpretation and limitations: Errors in the 3-5 km/h range were most common and are acceptable given (i) monocular depth noise, (ii) occasional short tracks or partial occlusions, and (iii) small timing mismatches between $v_{\mathrm{cyc}}^{\mathrm{smooth}}$ and the consensus velocity windows. Larger errors occurred when the velocity episode was very short, when the object’s track contained residual depth spikes, or when the GNSS smoother lagged a sharp pedal-induced speed change. Overall, this check supports that the consensus relative-velocity pipeline, once corrected by GNSS derived cyclist speed, provides physically plausible magnitudes and directions for downstream workload analysis.

4.5.3. Metrics and Workload Data:

In the final metric selection, several metrics are intentionally redundant variants (ex: area ratios at < 3 m vs. < 5 m; whole frame vs. center band) to allow the statistical analysis to identify which specificity levels matter most; later we address multicollinearity and feature selection.

Basic (appearance/proximity) metrics. These summarize instantaneous “who/where” context per frame: counts, class shares, and normalized bounding-box area (a proxy for visual load) across the full image and the central field of view (central 60% width).

Metric name	Group	Type	Definition / intuition
total_tracks	Volume	count/frame	Number of tracked road users visible in the frame.
heavy_vehicle_share	Traffic mix	ratio [0-1]	Share of buses/trucks among all tracks (exposure to large vehicles).
pedestrian_proportion	Traffic Mix	ratio [0-1]	Share of pedestrians among all tracks (unpredictable micro-motions).
tier_1_objects	Proximity	count/frame	Objects within < 3 m (near-field).
tier_2_objects	Proximity	count/frame	Objects within < 5 m (very near/near).
fov_clutter_all	Visual Density	area ratio	Sum of all bbox areas divided by image area (visual clutter proxy).
fov_clutter_5m	Visual Density	area ratio	As above, restricted to objects with distance < 5 m.
fov_clutter_3m	Visual Density	area ratio	As above, restricted to objects with distance < 3 m.
center_fov_clutter_all	Visual Density	area ratio	Clutter within central 60% of image width (task-relevant region).
center_fov_clutter_5m	Visual Density	area ratio	Center clutter for objects < 5 m.
center_fov_clutter_3m	Visual Density	area ratio	Center clutter for objects < 3 m.

Table 4.4: Basic (appearance/proximity) metrics computed per frame.

Velocity and Cyclist-velocity metrics: These capture the dynamics around the rider. Relative-velocity features come from the consensus windowing scheme (Sec. 4.5.2), so any non-zero value implies a sustained, reliable motion estimate for at least one object in that frame. Cyclist-velocity comes from the GNSS pipeline resampled to 25 Hz and smoothed (Sec. 4.5.2).

Metric name	Group	Type	Definition / intuition
ego_velocity_kmh_smooth	Ego Velocity	speed [km/h]	Smoothed cyclist speed from GPS, aligned at 25 Hz.
avg_relative_velocity	Velocity	speed [km/h]	Mean magnitude of active relative-velocity estimates in frame.
num_velocity_measurements	Velocity	count/frame	Number of active velocity windows contributing in frame (sustained motion evidence).
approaching_velocity_avg	Velocity	speed [km/h]	Mean of approaching-only velocities (objects getting closer).
receding_velocity_avg	Velocity	speed [km/h]	Mean of receding-only velocities.
uncertain_velocity_avg	Velocity	speed [km/h]	Mean of velocities with low direction confidence.
num_approaching	Velocity	count/frame	Count of approaching measurements in frame.
num_receding	Velocity	count/frame	Count of receding measurements in frame.
num_uncertain	Velocity	count/frame	Count of uncertain-direction measurements in frame.
velocity_std	Velocity	std. [km/h]	Standard deviation of all velocity magnitudes in frame (spread).
movement_complexity	Flow Order	entropy [0–1]	Directional entropy over {approach, recede, uncertain}; 0=single mode, 1=balanced mix.

Table 4.5: Velocity/ego-motion metrics mapped to frames; non-zero values imply sustained windows passing the consensus filters.

Workload series and alignment: The workload labels come from [49]'s protocol (Sec. 3.1): riders produced step-like audio cues while cycling; these were transcribed to a per-frame series at 25 Hz. As described earlier, the original 1-5 scale was binarized to workload_high ($1 =$ higher workload) to counter severe class imbalance and to emphasize onset/offset dynamics. For each video, we trimmed the workload series to the analyzed frame span and merged it with the metrics.

Final frame-level dataset: For each video segment, the merged dataframe contains 25 columns:

- 22 predictor metrics (Tables 4.4–4.5),
- the frame index (`frame_idx`),
- the original ordinal label (`workload_original`) and the binary target (`workload_high`).

These datasets are the input to the results methodology in the next section, where we prepare the data for rigorous statistical analysis.

4.6. Results Methodology

This section describes our methodology for examining the relationship between camera-derived traffic metrics and perceived cycling workload. We transform high-frequency sensor data into analysis-ready episodes, develop interpretable predictive models, and establish rigorous validation protocols. The approach balances statistical rigor with practical interpretability, progressing from raw frame level metrics to quantified traffic-workload relationships through systematic data preprocessing, feature engineering, and multi-perspective evaluation. Our methodology addresses key challenges in time-series analysis (temporal dependence), imbalanced classification (class prevalence effects), and spatial generalization (route-specific patterns) to provide robust evidence about the detectability and practical utility of traffic-based workload prediction.

4.6.1. Data Pre-processing

The raw data obtained from the metric extraction is frame by frame traffic information recorded at 25 Hz (one row every 0.04 s) and a binary workload annotation extracted at the same frequency as described in 3.2. To make these signals usable for statistical modelling, we need to ensure independence between the rows of the dataset, which is not the case at 25 Hz since the metric information between frames doesn't have sufficient time to change significantly. To manage this, we convert them into simple, self-contained episodes: non-overlapping 2 s windows, each with a small set of traffic features and a single workload label corresponding to that window. This section will describe the chosen process to transform our frame-level metric data, obtaining a rigorous analysis-ready, final dataset.

Data sources and combination

Each of the three rides (Video 1, Video 2, Video 3 from 3.2) contain a time-ordered sequence of frames with synchronized metrics and a time-ordered sequence of workload labels. We keep the native chronology of every ride and do not shuffle rows. The three rides are then concatenated into one table and we add a `video_id` column so that later analyses can:

1. Evaluate performance from a prediction model within a route (by splitting each route into adjacent blocks).
2. Test cross-route generalization from the model (by holding one route out entirely).

Table 4.6 summarises the raw datasets from the three individual video segments and their specificities, as well as the combined dataset. The main difference to keep in mind is that, unlike Videos 1 and 2, which have a clear route progression going from protected bike lane paths into a dense city centre, Video 3 is contained only within the city centre.

Table 4.6: Video segments, frame counts, workload proportion, routes, and weather conditions.

Segment	Frames	High WL	Route	Weather
Video 1 (P1)	17,219	44.3%	Bike route → City centre	Extreme sun and shade
Video 2 (P1)	16,823	36.6%	Delft Campus → City Centre	Grey → Sunny
Video 3 (P2)	16,470	29.6%	City Centre	Grey → Sunny
Combined Video	50,658	37%	Mixed	Mixed

From many metrics to a small, clear feature set

The metric-extraction stage 4.5 produced several families of traffic measures. In plain terms:

- *Density & proximity:* how much is around the cyclist and how close (ex: `total_tracks`, `tier_2_objects`, `num_uncertain`, `num_velocity_measurements`).
- *Scene complexity:* how visually cluttered the field of view is (ex: `fov_clutter_all`).
- *Motion composition:* how surrounding traffic moves relative to the cyclist (ex: `receding_velocity_avg`, `avg_relative_velocity`).

From the main families, a number of children metrics were made, this means that from the set of 22 metrics, many hold redundant information and it was decided to select a core group of 8 metrics which would be used to describe traffic conditions and provide results for a statistical analysis. This set was chosen based on the following criteria:

- *Parsimony and clarity*: one representative mean summary per concept to avoid redundancy and collinearity.
- *Acceptable zero-inflation*:As mentioned above, many metrics are specified versions of a main family (ex:`center_fov_clutter_3m`) and they appear much less frequently in the data making them highly zero-inflated.
- *Coverage*: keep diversity across concept groups (density, scene, motion).

Table 4.7 summarises the key metric group kept for the subsequent analysis. In contains four count metrics as well as four continuous metrics, representing all the different families with limited redundancy. The cyclist's own speed obtained from GNSS is intentionally left out of the set as it reflects a physiological reaction from the cyclist to an external stressor and we don't want it to influence the results.

Table 4.7: Core-8 feature set

Core-8
`total_tracks`
`tier_2_objects`
`num_uncertain`
`num_velocity_measurements`
`fov_clutter_all`
`receding_velocity_avg`
`avg_relative_velocity`
`movement_complexity`

Transforming count metrics (log1p), before aggregation: Four of the Core-8 features are counts (`total_tracks`, `tier_2_objects`, `num_uncertain`, `num_velocity_measurements`). Counts are naturally more skewed and often include zeros. To make their scale more comparable and to prevent occasional bursts from dominating an episode, we apply a simple *log-plus-one* transform to each frame:

$$\tilde{x} \;=\; \log(1 + x).$$

This has three practical benefits we rely on later: (i) it handles zeros cleanly; (ii) it compresses large values so an extra object at high density does not count as much as at low density; and (iii) it stabilizes variance so that linear models can use the feature more effectively. We only apply this transform to the four count variables; continuous measures (ex: `fov_clutter_all`, velocities, `movement_complexity`) are left on their native scale.

Why we aggregate frames into 2 s windows

Individual frames arrive every 0.04 s. At that time scale, successive rows in the dataset hold almost the same information. Treating them as independent would overstate how much data we have and can leak information between training and testing. We therefore:

- group frames into non-overlapping 2 s windows (50 frames) to reduce temporal dependence and align with human cognitive processing timescales. Research on attention and workload indicates that subjective workload assessments integrate information over 1-3 second intervals [31, 74], and

- avoid overlap so that no frame appears in two samples, preventing accidental information sharing.

When we group the frames into the windows we effectively reduce the dataset from 50,658 frame metrics to 1009 non-overlapping 2 second windows which is a substantial reduction in available data, but it ensures its quality and prevents information overlap.
To keep features easy to interpret and the analysis compact, each window is summarized with a single *mean* per metric:

- For the four count features, we first transform each frame with $\tilde{x} = \log(1 + x)$ and then take the arithmetic mean across the 50 frames. We denote these as `{count}_log1p_mean`.
- For the continuous features (`fov_clutter_all`, `receding_velocity_avg`, `avg_relative_velocity`, `movement_complexity`), we take the arithmetic mean across the 50 frames and denote them as `{metric}_mean`.

The mean gives a direct interpretation of a typical level over two seconds and combined with the log1p step for counts, is robust to short spikes.

Table 4.8: Core-8 traffic features used in the analysis. Count variables are first transformed with $\log(1 + x)$ at the frame level and then averaged over the 2 s window; continuous variables are averaged directly over the 2 s window.

Name	Short definition (with units/transform)
`total_tracks_log1p_mean`	Number of tracked objects in field of view (count → log1p, 2 s mean).
`tier_2_objects_log1p_mean`	Count of objects within 5m (count → log1p, 2 s mean).
`num_uncertain_log1p_mean`	Count of object velocities with uncertain direction (count → log1p, 2 s mean).
`num_velocity_measurements_log1p_mean`	Objects with reliable velocity estimates (count → log1p, 2 s mean).
`fov_clutter_all_mean`	Sum of bbox areas divided by image area (unitless 0–1, 2 s mean).
`receding_velocity_avg_mean`	Mean speed of objects moving away from the cyclist (km/h, 2 s mean).
`avg_relative_velocity_mean`	Mean absolute relative speed between cyclist and objects (km/h, 2 s mean).
`movement_complexity_mean`	Directional entropy of objects (unitless, 2 s mean).

Aligning labels to traffic: choose the lag before finalizing windows

Once we have windowed traffic features, we still need to attach a single workload label to each window. Since we can't use the mean for the windowed workload label, we use the center label, in other words we use the workload label at the center of that window. Before aligning the labels and metrics however, a test was conducted to investigate the temporal relationship between the workload labels and the captured metrics. The raw labels are spoken cues and therefore behave like a step function. When analysing the videos we noticed that the participants often anticipated workload changes before the traffic metrics would be able to capture a meaningful change. If we paired traffic at time t with the label spoken at exactly t, we might be misaligned: the label could actually refer to traffic from a short time later.

In order to test this, we introduce a small, positive lag L (in seconds) to shift the label stream forward so that traffic at time t is compared with the label that was spoken shortly before t. This aligns current traffic conditions with workload that was anticipated and announced before the measurable complexity occurred.

The images below 4.11 and 4.12 constitute an example of the anticipatory behaviour discussed above. In the first image, the participant announces an increase in workload as he is anticipating traffic crossing the bridge in front of him and blocking his trajectory. The traffic metrics would only see a meaningful change seconds later in the second image demonstrating the temporal misalignment between cues and metrics.

Figure 4.11: Moment at which participant gives high workload cue

Figure 4.12: Moment where metrics would capture information

How the lag is chosen: We test a small set of plausible delays $L \in \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5\}$ s in three steps:

1. **Shift labels at the frame level:** For each candidate L, move the label time-series forward by L seconds (so a frame at time t is paired with the label uttered at $t+L$).
2. **Rebuild windowed datasets:** Using the same 2 s non-overlapping windows metric means and the same center-state rule for the workload labels (defined below), we build a windowed dataset for that L.
3. **Score alignment quality.** For each candidate lag L, we ask for each feature: "Do its values tend to be different in High workload windows than in Low workload windows, or not?" We answer this with the *Mann-Whitney U test* computed separately per feature:

- This test takes all the values of one feature and orders them from smallest to largest. If High and Low windows are truly similar, their values will be intermixed across this ordering. If one group tends to have bigger (or smaller) values, its observations will sit higher (or lower) in the ordering. The test measures how separated those two groups are in this ordered list. It does not require normal (bell shaped) data and works well with small samples which is needed in our case of 1009 total windows.[46]
- From the test we also obtain its p values per feature, they represent the chance of seeing a distinction at least this strong just by luck if there was no real difference between High and Low. We call a feature "discriminative" at a lag L if $p < 0.05$. This test is described mathematically in B.1.5.
- For each lag L, we compute the the percentage of features that are discriminative ($p < 0.05$). The lag with the highest percentage of discriminative features is chosen as $L^\star$. If two lags are similar, we prefer the smaller lag for simplicity and to avoid over-fitting the choice to noise.

We then select a single lag for all videos $L^\star$ that maximizes separation. That $L^\star$ is fixed for all subsequent steps.

Table 4.9: Lag sweep summary per route using core 8 features. A feature is "discriminative" if Mann–Whitney $p < 0.05$ for High vs. Low workload at that lag. We pick the smallest lag with the highest discrimination rate.

Video	Lag (s)	Windows	High WL %	Discriminative (/8)	Rate %
Video 1	0.0	344	44.5%	5/8	62.5%
Video 1	**1.0**	343	44.3%	**7/8**	**87.5%**
Video 2	**0.0**	339	36.9%	**6/8**	**75.0%**
Video 2	1.0	338	36.4%	5/8	62.5%
Video 3	0.0	329	29.5%	1/8	12.5%
Video 3	**1.0**	328	29.3%	**2/8**	**25.0%**

Notes. (i) For Videos 1 and 3, 1.5 s ties 1.0 s; we select the smaller lag (1.0 s). (ii) "Windows" and High-WL rates show minor change from shifting labels.

From table 4.9 we see that Video 1 improved from 62.5% to 87.5% discriminative features at 1 s, and

Video 3 improved from 12.5% to 25% at 1 s. Video 2 did not improve (75% to 62.5%), but the net effect across routes was positive, so we fixed a single 1 s lag for all videos to keep the pipeline comparable. Also note that the number of discriminative features for Video 3 is much lower due to the fact that its segment is only contained within the city center and lacks the significant route differences the other two have.

Final windows and labels (using the chosen lag $L^\star$ = 1s)
With $L^\star$ fixed, we finalize the window samples:

- **Apply the lag:** We shift the frame-level workload label stream forward by 1 second.
- **Build 2 s non-overlapping windows:** We divide each route into consecutive 2 s segments (50 frames), preserving time order and ensuring no frame is reused across windows.
- **Center-state labelling:** The label of a window is simply the lagged label at the middle frame of that window. This ties each episode to a single, unambiguous state.
- **Exclude WL transition:** When the (lagged) Workload label flips from low→high or high→low, the traffic metrics inside a nearby window can be a mixture of two states. To avoid ambiguous targets, we drop any window whose center falls within ± 0.5 s of a flip. This 0.5-second buffer represents a conservative estimate of workload transition duration, ensuring that retained windows contain homogeneous workload states while preserving 97.5% of the dataset (Table 4.10).

Table 4.10: Final per-route counts after applying a 1 s label lag, 2 s non-overlapping windows, center-state labeling, and a ± 0.5 s exclusion buffer around label flips.

Route	# WL Transitions	Pre-buffer Windows	Excluded	Kept	High WL %
Video 1	24	343	9 (2.6%)	334	44.3
Video 2	13	338	8 (2.4%)	330	36.4
Video 3	16	328	8 (2.4%)	320	28.7
Combined	**53**	**1009**	**25 (2.5%)**	**984**	**36.6**

Table 4.10 shows the final number of windows and high workload proportion after having applied a transition filter which removes windows with a highly mixed workload state.

Before moving on to the results methodology, the flowchart below summarises all the steps taken in the dataset pre-processing pipeline.

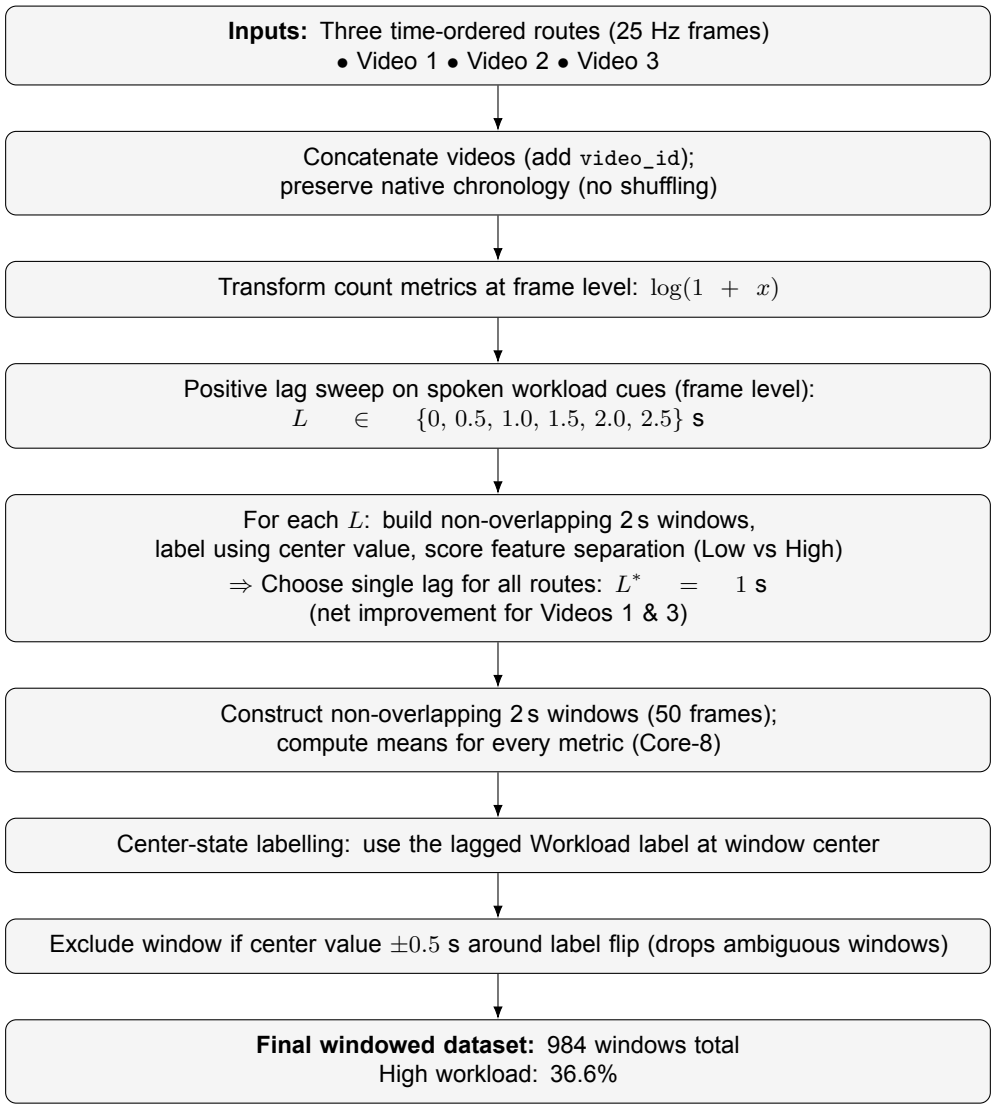


Figure 4.13: Processing pipeline for windowed dataset construction.

What the modeling stage receives

The modeling stage described in the next section takes the final table of windowed metrics:

- Eight traffic features (the Core-8 means).
- One binary label per window (lagged, center-based).
- A video_id to enable within-route and cross-route evaluation.

With this dataset, we can now move from feature engineering to predictive modelling. The next section outlines how we train and evaluate models on these windowed features and how we quantify predictive performance.

4.6.2. Modelling and Evaluation

The aim of this statistical methodology is to examine to what extent variations in camera derived traffic metrics coincide with near-instantaneous changes in perceived workload. “Coincide” here means that when the traffic metrics shift, the probability of reporting high workload shifts in a consistent direction within the same 2 s episode.

To be able to turn “coincide” into measurable quantities, we use a simple, interpretable model Logistic Regression model and look at alignment through three lenses, from least strict to strictest:

1. *Single-feature contrast:* For each traffic feature on its own, do typical values differ between Low and High workload episodes, or do they look similar?
2. *Ordering ability:* How well do traffic features work together in a multivariate model to distinguish workload levels? We examine both the individual feature contributions (through fitted coefficients) and the model’s overall ability to rank episodes. When the model scores two episodes (one Low and one High), does the High episode tend to receive a higher score? This checks whether there is a consistent ranking signal without forcing a binary yes/no decision.
3. *One cut-off rule:* If we must convert scores into a binary decision using a single threshold, how well can we balance missed High episodes against false alarms? This is the strictest view because it compresses a smooth score into a binary decision.

Primary model: Logistic Regression (LR)
From the eight window-level features, logistic regression outputs a number between 0 and 1: the model’s estimated probability that the window is high workload. Internally, it forms a weighted sum of the features and passes it through an sigmoid function so the output is a valid probability. Formally,

$$\Pr(y{=}1 \mid \mathbf{x}) \;=\; \sigma\Big(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}\Big), \quad \text{with } \sigma(z) = \frac{1}{1+e^{-z}}.$$

Each coefficient β_j says how a one-unit increase in feature x_j changes the *log-odds* of high workload.

Why this model?

- *Interpretability:* Signs show direction (positive $\rightarrow$ higher odds of high workload; negative $\rightarrow$ lower odds). Magnitudes can be read as odds ratios, with $\exp(\beta_j)$ indicating how much the odds change for a one-unit increase in the predictor (values > 1 mean higher probability, values < 1 mean lower).
- *Parsimony:* We use a compact, means-only Core-8 feature set and have a modest sample size; a linear model is an appropriate, robust baseline.
- *Stability with time-aware splits:* A linear decision boundary is less likely to learn noise within one route when we test on another route later.

What the model uses:

- *Features.* The Core-8, means-only, traffic-only features from Sec. 4.6.1.
- *Scaling:* Standardise each feature inside the training fold only (mean = 0, sd = 1); apply that transform separately to the validation data. This prevents “peeking” at validation statistics.
- *Class imbalance:* ”High” windows are fewer than ”Low” (36.6% prevalence); we set `class_weight=balanced` to automatically weight classes inversely proportional to their frequency, effectively upweighting the minority class by a factor of 1.73. This approach was chosen over manual threshold adjustment or resampling methods to maintain the original data distribution while improving minority class detection.
- *Regularisation:* We use L2 (ridge) regularization with the default penalty strength (C=1.0), which provides coefficient with smaller weights while maintaining feature interpretability, this is preferable to L1 regularization that would perform feature selection and complicate coefficient interpretation.The `lbfgs` optimizer is the scikit-learn default for logistic regression and performs well for our dataset size. We set `max_iter=2000` based on convergence monitoring during development; all models converged within 200 iterations across all folds. Hyperparameters remain fixed to maintain simplicity and comparability across validation splits.

Training and validation

We evaluate model performance using two complementary validation schemes that test different aspects of generalization:

- *Within-route performance: Stratified 5-fold cross-validation.* We divide our combined dataset into 5 folds using stratified sampling, which ensures two key properties: (1) each fold contains samples from all three routes, and (2) each fold maintains the same proportion of high vs. low workload periods as the overall dataset. To achieve this, we create stratification groups by combining route identity with workload class (creating 6 groups: Route1_High, Route1_Low, Route2_High, etc.), then sample proportionally from each group into every fold. This approach tests whether the model can identify predictive patterns that generalize across different moments within our route set. For example, a model trained on 80% of samples from varied time periods across all routes should predict the remaining 20% drawn from different time periods. Since our 2-second windows provide temporal independence, this maximizes training diversity while testing signal detectability across varied driving conditions within known routes.
- *Cross-route generalisation: Leave-One-Route-Out.* Train on two entire routes; test on the third route held out in its entirety. This tests whether predictive patterns learned from one set of driving environments can transfer to completely novel routes with different geometric, traffic, and contextual characteristics. Scaling is fit on the training routes only and then applied to the test route.

These two validation schemes answer distinct questions: stratified 5-fold asks ”Is there a detectable workload signal within our route set?” while leave-one-route-out asks ”Do workload patterns generalize to new cycling scenarios?” Together, they provide evidence for both signal detectability and spatial transferability.

Evaluation metrics

We evaluate classification performance using three complementary metrics that capture different aspects of model behaviour. These metrics are described mathematically in B.2.

F1 Score (threshold-dependent): The harmonic mean of precision and recall for the High workload class, computed at a fixed decision threshold. This metric balances two competing objectives: correctly identifying high-workload periods (recall) while minimizing false detections (precision). The harmonic mean ensures that good F1 performance requires both objectives to be reasonably satisfied, a model with perfect recall but terrible precision (or vice versa) will have poor F1. For our imbalanced dataset (36.6% High), we use a *prevalence-matched random* baseline, a chance classifier that predicts “High” with probability equal to the positive-class prevalence (0.366). This is a no-signal baseline because it preserves the observed class imbalance (rather than always predicting the majority class for example) and reflects what performance looks like when labels are guessed in proportion to how often they occur. For such a classifier the expected $F1$ equals the prevalence, so $F1_{\text{rand}} \approx 0.37$.

ROC-AUC (threshold-independent): The area under the Receiver Operating Characteristic curve, which plots true positive rate against false positive rate across all possible decision thresholds. This metric evaluates the model’s ability to rank high-workload windows higher than low-workload windows, independent of any specific threshold choice. ROC-AUC is unaffected by class imbalance, with random classifier performance always equal to 0.5.

PR-AUC / Average Precision (threshold-independent): The area under the precision-recall curve, which plots precision against recall across all decision thresholds. Unlike ROC-AUC, this metric focuses specifically on the model’s performance on the positive (high-workload) class and is sensitive to class imbalance. *Practical interpretation:* ”As we adjust the decision threshold to detect more high-workload periods (increasing recall), how well can the model maintain precision?” This metric is particularly informative when the positive class is relatively rare, as it directly measures how much precision we must sacrifice to achieve higher recall. A random classifier achieves PR-AUC equal to the positive class prevalence (≈ 0.37 in our case).

Together, these metrics provide a comprehensive view: F1 answers how good the model is for default deployment, ROC-AUC answers how well the model discriminates overall, and PR-AUC answers how useful the model is for finding rare high-workload events. All performance is evaluated relative to appropriate random baselines that account for class imbalance effects.

Uncertainty around the estimates. For cross-validation, each score is summarised as mean $\pm$ a 95% t-interval across the valid folds:

$$\bar{m} \pm t_{0.975,\, n-1} \frac{s}{\sqrt{n}},$$

where $\bar{m}$ is the fold-average, s is the sample standard deviation, and n is the number of valid folds. This interval is an uncertainty band, not a hypothesis test.

Turning probabilities into decisions: the threshold

The model outputs a probability $\hat{p}$ for High. To turn $\hat{p}$ into a label we need a decision threshold.

Primary policy: We use a fixed threshold of 0.5 for all within-route and cross-route evaluations. One shared cut-off keeps fold- and route-level results directly comparable and avoids tailoring the threshold to each split.

Supporting check: training-based threshold calibration. We examine whether optimizing the decision threshold can improve the model's ability to detect high-workload periods. Within each of the five blocked folds, we train the logistic regression on the training portion, then evaluate all possible thresholds on these same training predictions to identify the one that maximizes classification performance. This yields five fold-specific optimal thresholds, which we average to obtain a single calibrated threshold. This aggregated threshold is then applied across all validation folds for evaluation. The purpose is to assess whether threshold tuning using only information available at training time and aggregated across temporal blocks can enhance detection of the high-workload class compared to the standard 0.5 threshold. This approach maintains proper train-validation separation while leveraging the stability of cross-fold averaging for threshold selection.

Interpreting the fitted model

Logistic regression also provides a compact, human-readable summary of relationships:

- *Standardised coefficients:* Because we scale features inside each training fold, coefficients are on comparable units: a larger absolute value means a stronger association in the model.
- *Odds ratios:* For feature j, $\exp(\beta_j)$ is the multiplicative change in the odds of high workload for a $+1$ standard-deviation increase in that feature, holding the others fixed (ex: 1.50 = 50% higher odds; 0.67 $\approx$ one-third lower).
- *How we present them:* In the Results we show a table of the fitted coefficients with odds ratios annotated. Given the modest sample size, we treat these as descriptive, not as "significant" or causal.

Mapping analytical approaches to evaluation metrics

To systematically examine the relationship between traffic metrics and workload, we implement the three analytical approaches outlined at the start of the sub-section using established statistical methods:

- *Single-feature contrast analysis:* To assess whether individual traffic features show different patterns between high and low workload episodes, we compare the mean values of each feature across the two workload conditions using the full combined dataset. This provides a descriptive overview of which traffic metrics are associated with higher or lower perceived workload.
- *Ranking performance evaluation:* To evaluate how traffic features work together to distinguish workload levels, we fit a logistic regression model using all features simultaneously. We examine both the fitted coefficients (which show each feature's contribution while controlling for others) and the model's ranking performance using threshold-independent metrics: ROC-AUC (which treats both classes symmetrically) and PR-AUC (which focuses on performance for the minority high-workload class). The standardized coefficients reveal the direction and strength of each

feature's association with high workload in the multivariate context, while the ranking metrics assess whether the model consistently assigns higher scores to high-workload episodes.

- *Binary decision performance:* To evaluate performance when the model must make definitive high/low classifications, we use the F1 score computed at a fixed 0.5 threshold. This measures how well the model balances the competing objectives of detecting genuine high-workload episodes (recall) while avoiding false alarms (precision).

Each approach provides complementary evidence: feature-level analysis reveals which individual traffic metrics are most informative, ranking metrics assess the model's fundamental discriminative ability, and binary classification metrics evaluate practical deployment performance.

Summary and transition to results

This methodology transforms 50,658 raw video frame metrics into a final dataset of 984 independent 2-second averaged window metrics, each characterized by 8 traffic metrics and a single workload label. The processing pipeline addresses temporal dependence through windowing, optimizes feature-label alignment via lag testing, and excludes ambiguous transition periods to ensure clean targets.

The analysis proceeds through three complementary analytical approaches: descriptive feature comparison reveals individual metric patterns, multivariate logistic regression quantifies combined predictive relationships, and performance evaluation assesses practical deployment viability. Two validation schemes test different generalization aspects, the stratified cross-validation examines within-route signal detectability while leave-one-route-out tests cross-route transferability.

The following Results section presents findings from this analytical framework, progressing from individual feature insights through model performance to practical deployment considerations.

5

Results

This chapter reports how camera-derived traffic metrics relate to near-instantaneous perceived workload, directly addressing RQ1 and indirectly addressing RQ2 by using metrics from an end-to-end monocular pipeline. We analyse a windowed dataset of 984 independent 2-s episodes drawn from three urban rides (high-workload prevalence per held-out route: 44.3%, 36.4%, 28.7%) using a compact Core-8 traffic feature set.

The results are shown in three steps: (i) single-feature contrasts quantify directional differences between high vs. low workload episodes; (ii) a multivariate logistic regression summarises feature contributions and evaluates threshold-independent ranking performance (ROC-AUC, PR-AUC) under stratified 5-fold within-route and leave-one-route-out validation; and (iii) binary decision performance is assessed with F1 at a fixed 0.5 threshold alongside a training-only calibrated threshold.

To ensure fair interpretation under class imbalance, all scores are reported against appropriate random baselines ((ROC–AUC = 0.5; PR–AUC = positive-class prevalence; F1 compared to a *prevalence-matched random* baseline with expected F1 equal to high workload prevalence) including route-specific prevalences for cross-route results.

5.1. Analytical Approach 1: Individual Feature Patterns

To establish whether individual traffic metrics contain workload-related information, we compare the mean values of each Core-8 feature between low and high workload episodes across our complete dataset of 984 windows. This descriptive analysis reveals which traffic characteristics are most strongly associated with perceived workload changes.

Table 5.1 presents the mean values for each traffic feature, ordered by the magnitude of difference between workload conditions. The analysis reveals systematic patterns: six of eight features show higher values during high workload episodes, while two show lower values.

Table 5.1: Individual feature patterns: mean values by workload level. Features ordered by magnitude of difference between high and low workload episodes.

Traffic Feature	Low WL	High WL	Difference	Direction
receding_velocity_avg	4.034	4.320	+0.286	+7.1%
avg_relative_velocity	9.604	9.358	-0.246	-2.6%
tier_2_objects	0.622	0.672	+0.050	+8.1%
total_tracks	1.408	1.446	+0.038	+2.7%
num_velocity_measurements	1.389	1.407	+0.018	+1.3%
num_uncertain	0.588	0.604	+0.016	+2.8%
movement_complexity	0.411	0.425	+0.013	+3.3%
fov_clutter_all	0.096	0.086	-0.010	-10.1%

The strongest individual discriminators are velocity-related metrics: receding velocity shows the largest increase (+7.1%) in high workload conditions, while the average relative velocity exhibits the largest decrease (-2.6%). These patterns in velocity metrics suggest a relationship between traffic motion characteristics and perceived workload.

Object density metrics all trend upward during high workload episodes. The number of nearby objects (tier 2 objects) increases by 8.1%, total tracked objects rise by 2.7%, and objects with uncertain motion direction increase by 2.8%. Movement complexity also increases by 3.3%, indicating more varied directional patterns in the traffic environment.

On the other hand, field-of-view clutter decreases by 10.1% during high workload periods despite increases in object counts. This suggests that high workload scenarios may involve smaller or more distant objects, which is consistent with there being a higher proportion of pedestrians in the city centre, since they have the smallest bounding box areas among classes and that's where most of the high workload moments are concentrated.

In order to further corroborate this feature analysis, figure 5.1 shows the box plot representation of each feature's distribution in low and high workload. Although most of the differences are subtle, we do find that the median value (black line in the center of the boxes) does follow the same changes as the mean; it is higher for all features during high workload, apart from the average relative velocity and the field of view clutter. Movement complexity however, seems to have a slightly lower median in high workload but it does have higher interquartile ranges (middle 50% data spread) within high workload episodes. Additionally the Field of View clutter feature has the most outlier values by far (black circles) which is indicative of cyclist point-of-view footage, most of the time these values are quite low but they spike when objects pass by as their bounding boxes reach their largest total area. The lower field of view clutter values in high workload could actually indicate a smaller number of passing events in congested traffic.

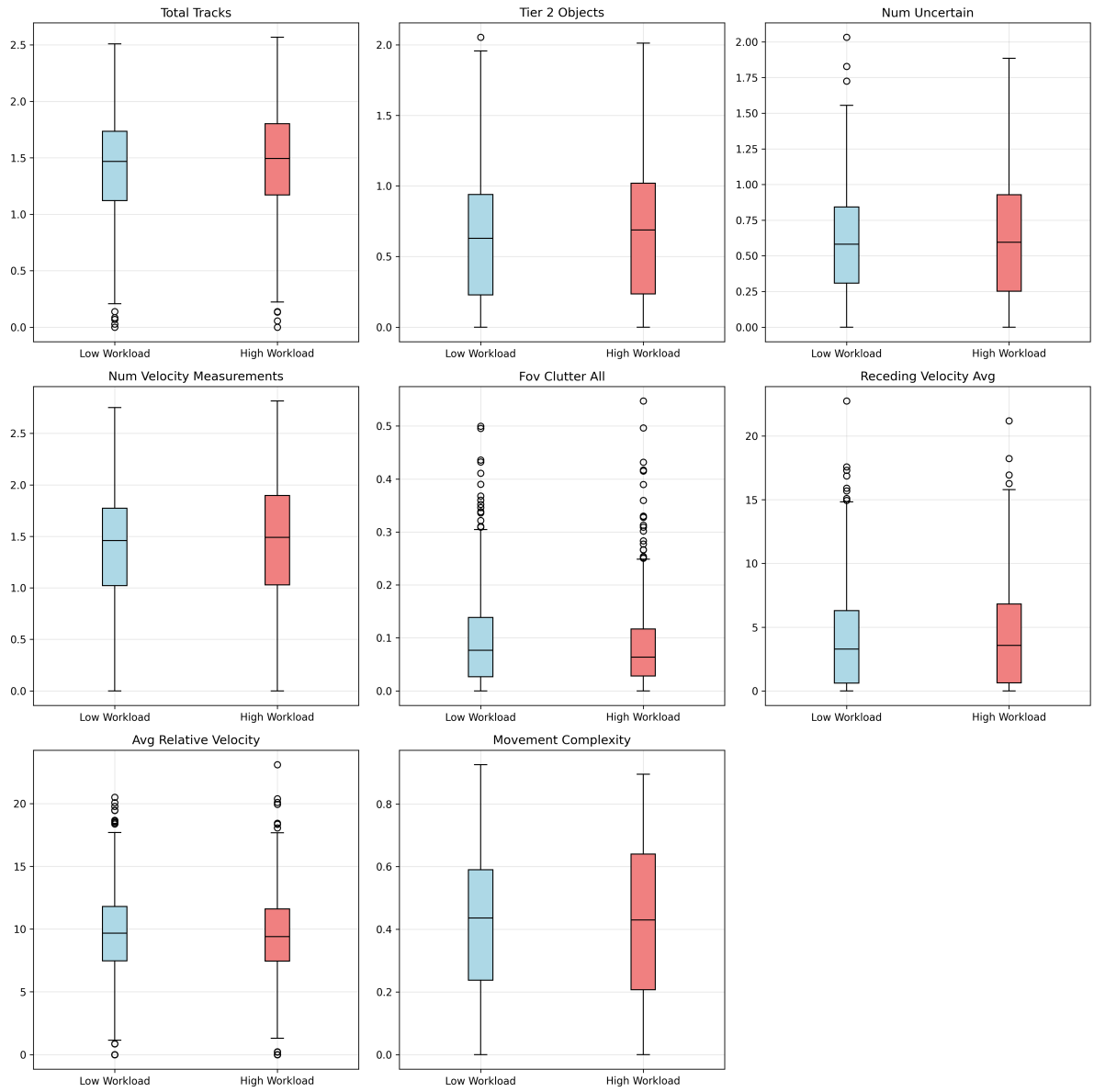


Figure 5.1: Box plots showing distributional differences for each feature between high and low workload with the black line representing the median value

In summary,these individual patterns suggest a coherent underlying traffic scenario associated with high workload conditions: higher object density with more complex movement patterns, combined with specific velocity relationships that indicate particular traffic flow characteristics. The consistency of these directional patterns across complementary metrics provides initial evidence that camera-derived traffic features contain meaningful workload-related information. We examine whether these individual patterns persist when features are combined in a multivariate model.

5.2. Analytical Approach 2: Multivariate Ranking Performance

Having established that individual traffic features show some patterns with workload levels, we now examine how these features perform when combined in a multivariate model. This analysis addresses two key questions: which features contribute most strongly to workload prediction when considered jointly, and how well can the combined model rank episodes by workload probability. We first examine the logistic regression coefficients to understand feature contributions, then evaluate the model's ranking performance under two validation schemes.

5.2.1. Feature contributions in the combined model

To understand how traffic features work together to distinguish workload levels, we fit a logistic regression model using the stratified 5-fold cross-validation (CV) approach described in Section 4.6.2. The coefficients presented here are obtained from training the model on the complete combined dataset using the same preprocessing and scaling procedures outlined in the methodology. These standardized coefficients indicate each feature's relative contribution to workload prediction when all features are considered jointly.

Table 5.2 presents the standardized coefficients their corresponding odds ratios and their associated p-values testing for statistical significance. Positive coefficients indicate features associated with higher workload probability, while negative coefficients suggest associations with lower workload probability. Statistical significance of individual predictors was assessed using the Wald test which evaluates the null hypothesis that each regression coefficient equals zero ($H_0 : \beta_i = 0$), indicating no linear relationship between the predictor and log-odds of high workload. For each coefficient, a z-statistic is calculated by dividing the estimated coefficient by its standard error, which follows a standard normal distribution under the null hypothesis [34]. The mathematical description of this test is given in B.1.3.

Table 5.2: Logistic regression coefficients and odds ratios for traffic features. Coefficients represent the change in log-odds of high workload for a +1 standard deviation increase in each feature, holding all other features constant. P-values are obtained from the Wald test.

Feature	Coefficient (β)	Odds Ratio (e^{β})	P-value
tier_2_objects_log1p_mean	+0.4645	1.5912	0.0002
total_tracks_log1p_mean	+0.2472	1.2805	0.0438
receding_velocity_avg_mean	+0.0897	1.0939	0.3320
movement_complexity_mean	+0.0591	1.0609	0.7182
num_uncertain_log1p_mean	-0.0286	0.9718	0.8275
avg_relative_velocity_mean	-0.0915	0.9125	0.3085
num_velocity_measurements_log1p_mean	-0.2871	0.7505	0.1524
fov_clutter_all_mean	-0.4451	0.6408	$\approx 10^{-4}$

Table 5.2 indicates that the two strongest positive predictors are object density metrics: `tier_2_objects` (β = +0.465, OR = 1.59) and `total_tracks` (β = +0.247, OR = 1.28). A one standard deviation increase in nearby objects (within 5m) increases the odds of high workload by 59%, while a similar increase in total tracked objects increases odds by 28%. These coefficients align with the individual feature patterns, confirming that higher object density consistently predicts elevated workload.

Velocity-related features show both consistent and divergent patterns compared to individual analysis. `Receding_velocity_avg` and `avg_relative_velocity` maintain their individual directions with coefficients of +0.090 and -0.0915 respectively, preserving the same relationships observed in descriptive analysis. However, two features exhibit reversed relationships in the multivariate context: `num_velocity_measurements` becomes strongly negative (β = -0.287, OR = 0.75) despite showing a positive individual pattern, and `num_uncertain` shifts to slightly negative (β = -0.0286) from its individual positive pattern. These reversals suggest that when controlling for other traffic characteristics, having many reliable velocity measurements or uncertain objects may actually indicate lower workload scenarios, possibly reflecting situations where systematic tracking (many velocity measurements) and predictable motion patterns (uncertain direction due to similar speeds) occur in structured traffic environments, though the specific mechanism requires further investigation.

`Fov_clutter_all` exhibits the strongest negative association (β = -0.445, OR = 0.64), meaning a one standard deviation increase in visual clutter reduces high workload odds by 36%. This coefficient amplifies the counter-intuitive pattern observed in individual feature analysis, suggesting that when other traffic characteristics are accounted for, higher visual clutter actually predicts lower perceived workload.

The logistic regression coefficients show notable consistency with individual feature patterns: six of the eight features maintain the same directional relationships observed in the descriptive analysis. However, the two reversed features (`num_velocity_measurements` and `num_uncertain`) highlight how multivariate modeling can reveal that individual correlations may be confounded by other variables. When

all features are considered jointly, the model identifies which aspects of traffic are independently predictive of workload, potentially revealing that some individual patterns were driven by their correlation with other, more fundamental traffic characteristics.

At the conventional $\alpha = 0.05$ level, three predictors show evidence of a non-zero effect: `tier_2_objects` ($p = 0.0002$), `total_tracks` ($p = 0.0438$), and `fov_clutter_all` ($p \approx 10^{-4}$). The remaining coefficients are not statistically significant so once the other features are controlled for, their estimated effects are indistinguishable from zero in this sample (Wald test as described above).

This pattern is expected for three main reasons. Firstly, many traffic features move together (multicollinearity), which lowers individual significance even when the set of predictors is useful. Secondly, when analysed jointly, single features often contribute little unique information because several of them proxy the same underlying "traffic complexity" signal. Thirdly, with eight predictors and a finite sample, the Wald test becomes conservative: stronger individual effects are needed to pass the significance threshold. An additional reason could be potential non-linearity between the feature workload relationship, the logistic regression tests linear effects on the log-odds scale so non-linear relationships can appear non-significant.

To conclude, the significant predictors are the `fov_clutter_all`, `tier_2_objects` and `total_tracks`. Motion-based features do not reach significance. This seems to fit the broader picture that visual–spatial load (how many objects are present to monitor) is more consequential than fine-grained velocity cues for distinguishing workload in this dataset.

5.2.2. Model ranking ability validation

We assess whether the logistic-regression model can *rank* episodes by workload level using two complementary schemes: (i) stratified 5-fold cross-validation where folds mix segments from all three routes (pooled view), and (ii) leave-one-route-out (LORO) where an entire route is held out for testing (cross-route generalisation).

Table 5.3: Ranking performance of the logistic-regression model. The first row reports stratified 5-fold cross-validation (pooled across routes) as mean $\pm$ 95% t-interval across folds. LORO rows report scores on each held-out route (no CI). High WL % indicates the proportion of high-workload episodes in the held-out set.

Validation scheme	ROC-AUC	PR-AUC	n_{test}	High WL %
Stratified 5-fold CV	0.570 ± 0.057	0.450 ± 0.072	—	37
LORO Held-out Route 1	0.469	0.451	334	44.3
LORO Held-out Route 2	0.537	0.383	330	36.4
LORO Held-out Route 3	0.546	0.367	320	28.7
LORO Average	**0.517**	**0.401**	—	—

Under stratified 5-fold CV, the model achieves ROC-AUC $= 0.570 \pm 0.057$ and PR-AUC $= 0.450 \pm 0.072$. Relative to random classifier baselines (ROC-AUC $= 0.5$; PR-AUC ≈ 0.37, matching 37% high workload in Table 5.3), this represents improvements of $+0.070$ and $+0.080$, respectively. This means that when comparing a random high and a random low episode drawn from mixed segments, the model ranks the high one above the low 57% of the time, and it recovers the positive class with moderate precision–recall trade-offs that exceed the appropriate baseline.

When asked to generalise to an entirely unseen route, average performance drops to ROC–AUC $= 0.517$ and PR–AUC $= 0.401$ (Table 5.3). For LORO, PR–AUC should be evaluated against each route's own prevalence baseline (last column): Route 1 baseline $= 0.443 \Rightarrow 0.451 - 0.443 = +0.008$; Route 2 baseline $= 0.364 \Rightarrow 0.383 - 0.364 = +0.019$; Route 3 baseline $= 0.287 \Rightarrow 0.367 - 0.287 = +0.080$. On average, PR–AUC exceeds route-specific baselines by approximately $+0.036$, while ROC–AUC remains $+0.017$ above its random baseline of 0.5. Differences by held-out route are expected: Route 1 follows a different route, while Routes 2–3 share a broader itinerary but still cover non-identical segments. Two non-exclusive explanations fit the pattern: (i) the test route's traffic mix and geometry differ from the training routes; and/or (ii) Route 1 may span a broader variety of conditions, making it a good training source (helpful for others) but harder when held out. We therefore avoid strong causal claims and simply note that the ranking signal is present but attenuates under a leave-one-route-out (LORO)

split. A visual representation of the trained model's ranking ability for the LORO method is shown in A.5.

In summary, the model exhibits consistent above-baseline ranking performance in the pooled view and retains detectable ranking ability on held-out routes. This supports the claim that camera-derived traffic variations and perceived workload coincide to a detectable extent, while also highlighting the limits of cross-route transfer.

5.3. Analytical Approach 3: Binary Decision Performance

The final analytical approach evaluates how well the model performs when required to make definitive high/low workload classifications. This is the most demanding test as it compresses probabilistic predictions into a binary decision. We examine both default threshold performance and the potential for improvement through threshold calibration.

Default threshold performance

Table 5.4 presents F1 scores and supporting metrics using the standard 0.5 decision threshold across both validation schemes.

Table 5.4: Binary classification performance at 0.5 threshold. Within-route results show mean ± 95% confidence intervals; cross-route results show individual route performance and average. High WL % indicates the proportion of high-workload episodes in the held-out set.

Validation Scheme	F1 Score	Precision	Recall	High WL %
Within-route (Stratified 5-fold CV)				
All 3 Routes	0.478 ± 0.038	0.423	0.550	37
Cross-route (Leave-one-route-out)				
Route 1 held out	0.477	0.418	0.554	44.3
Route 2 held out	0.473	0.401	0.575	36.4
Route 3 held out	0.331	0.380	0.293	28.7
LORO Average	**0.427**	**0.400**	**0.474**	—

Within-route performance achieves $F1 = 0.478 \pm 0.038$ with precision 0.423 and recall 0.550. Relative to a *prevalence-matched random* baseline ($F1_{\text{rand}} =$ pooled High prevalence ≈ 0.37, see Table 5.4), this is an improvement of $\approx +0.11$. The model identifies 55% of High-workload episodes at this threshold while keeping precision above the random reference, though there is just under one false alarm per correct prediction.

Cross-route binary decisions show similar patterns to ranking performance: a performance drop to average F1 $= 0.427$ and notable variability across routes. Interpreted against each route's own prevalence-matched random baseline $\text{F1}_{\text{rand}} \approx p_r$ (the held-out route's high-workload proportion shown in Table 5.4), the improvements are: Route 1 $+0.034$ (0.477 vs 0.443), Route 2 $+0.109$ (0.473 vs 0.364), and Route 3 $+0.044$ (0.331 vs 0.287). Route 3 presents the greatest challenge (F1 $= 0.331$), with particularly low recall (29.3%), suggesting the model struggles to detect high-workload episodes in the city-center-only environment when trained on more varied route conditions, despite remaining above its own baseline.

Threshold optimization potential

Table 5.5 shows the results of training-based threshold calibration, where we optimize the decision threshold using training data to maximize F1 performance.

Table 5.5: Threshold optimization results using stratified cross-validation. Training-based calibration finds the F1-optimal threshold on training data within each fold.

Threshold Approach	F1 Score	Precision	Recall	Threshold
Default (0.5)	0.478	0.423	0.550	0.500
Training-calibrated	0.517	0.388	0.783	0.444 ± 0.019
Improvement	**+0.039**	**-0.035**	**+0.233**	//

Training-based threshold calibration improves F1 performance from 0.478 to 0.517 (+3.9 percentage points) by lowering the decision threshold to 0.444 ± 0.019. This improvement comes through a classic precision-recall trade-off: recall increases substantially (+23.3 percentage points) at the cost of reduced precision (-3.6 percentage points). The calibrated model detects 78.3% of high workload episodes compared to 55.0% at the default threshold, but generates more false positives in the process.

Binary decision assessment

Binary classification represents the most challenging application of our traffic-workload model, requiring definitive decisions rather than probabilistic rankings. The results demonstrate:

Above-baseline performance with variable improvements: F1 scores of 0.478 (within-route) and 0.427 (cross-route) represent improvements of $\approx +0.11$ and $\approx +0.062$ (this represents the average improvement across the three routes) respectively over the pooled prevalence-matched random baseline (≈ 0.37). Route-specific baselines show heterogeneous gains: $+0.034$ (Route 1), $+0.109$ (Route 2), $+0.044$ (Route 3). While acknowledging significant room for improvement, These results indicate that binary decisions consistently exceed chance expectations, with the model identifying approximately half of high-workload episodes while maintaining precision above baseline levels.

Threshold calibration provides measurable improvement: The 3.9 percentage point F1 improvement (from 0.478 to 0.517) demonstrates that careful threshold selection could enhance practical deployment, particularly in applications where higher recall (detecting more high workload episodes) is preferred over precision. This optimization maintains the above-baseline performance while better balancing precision-recall trade-offs.

Cross-route challenges persist: The performance drop from within-route to cross-route validation (ΔF1 = -0.051) reinforces the finding that workload patterns contain route-specific characteristics that limit generalization to novel environments. However, cross-route performance remains meaningfully above random baseline, indicating that core traffic-workload relationships transfer across routes despite reduced effectiveness.

These binary decision results complement the ranking performance findings, confirming that camera-derived traffic metrics contain detectable workload-related information while highlighting the practical constraints for real-world deployment applications.

Overall these analyses demonstrate that camera-derived traffic metrics contain detectable workload-related information across multiple analytical approaches, with performance varying significantly between within-route and cross-route evaluation conditions. The implications of these findings for practical applications are discussed in the following section.

6

Discussion

6.1. Key findings

This section addresses both research questions by examining the evidence for traffic-workload relationships and evaluating the technical feasibility of monocular measurement. The findings are organised around the core research questions, with additional analysis of feature patterns and decision-making performance.

6.1.1. Answer to RQ1 (Do camera-derived traffic metrics coincide with perceived workload?)

Within-route signal is present: Using the protocol in Sec. 4.6.2 and the compact Core-8 traffic feature set, the logistic model separates higher from lower workload within routes better than chance. Both ranking metrics rise above their random baselines (ROC-AUC = 0.570 vs 0.5 baseline, +0.070 improvement; PR-AUC = 0.450 vs 0.37 baseline, +0.080 improvement), indicating that camera-derived traffic cues carry usable signal with balanced precision-recall trade-offs. The within-route ROC-AUC of 0.570 meets established criteria for a small effect size in predictive accuracy research [55], suggesting that the observed traffic-workload relationship, while modest, represents a detectable signal beyond measurement noise. Individual feature analysis revealed directional differences between high and low workload episodes, with logistic regression coefficients maintaining the same directional relationships for six of the eight features, confirming that the patterns represent genuine associations rather than noise.

Cross-route generalisation is modest but consistently above route-specific baselines: When an entire route is held out (LORO), performance decreases substantially but remains above baseline. Cross-route performance drops to ROC-AUC = 0.517 ($\Delta = -0.053$) and PR-AUC = 0.401 ($\Delta = -0.049$), suggesting that the measured traffic metrics contain route-specific characteristics that limit transferability. Furthermore, the PR-AUC, which must be read against each route's own high-workload prevalence, shows that all three held-out routes exceed their route-specific baselines: Route 1 (+0.008), Route 2 (+0.019), and Route 3 (+0.080), with the strongest gain on the city-centre-only recording (Video 3) and smaller margins on the other two. This pattern is consistent with our data: Video 1 follows a different itinerary from the others, while Video 3 concentrates on dense, uniform inner-city segments with the lowest proportion of high-workload labels. Overall, the learned signal transfers across routes in a limited but detectable way, indicating the need for infrastructure context in practical applications.

What the features suggest about high-workload scenarios: The alignment between descriptive patterns and logistic regression coefficients suggests a typical traffic scenario underlying high workload episodes. High workload episodes are characterized by increased object density (more total tracks, more nearby objects), complex motion patterns (higher movement complexity, more uncertain velocities), reduced visual clutter despite higher object counts, and specific velocity relationships (higher

receding velocity, lower average relative velocity). These patterns point to a plausible picture of defensive riding in dense urban environments where progress is constrained.

The motion terms indicate higher receding velocity and lower average closing speed, consistent with riders regulating speed when progress is constrained or perceived risk is higher. This aligns with studies showing that riders judge scenes by approach velocities and adjust speed to keep perceived demands manageable [41, 54]. The density terms (more nearby actors and higher total tracked participants) fit the visual scene of inner-city segments where pedestrians and cyclists cluster around the rider. Figure 6.1 which is taken from video 1 illustrates such a scenario: a dense city center scene where the participant has just announced elevated workload while being blocked by a pedestrian ahead and approached by a car, with his velocity being at zero, explaining the observed velocity patterns.

Although prior work links visual clutter to elevated attentional demand in driving [57, 19], our multivariate model yields a negative association between our clutter metric and reported workload. A plausible explanation is measurement-specific: the field of view metric is an area ratio, so scenes dominated by many small agents (pedestrians and cyclists) can raise density without occupying much pixel area, whereas large vehicles may increase pixel coverage but are often parked or segregated from cycling areas. Another possible reason for this could be explained by the fact that field of view clutter is characterised by many outliers; this is because this metric spikes during passing events, when objects are very close and their bounding box becomes very large. This could reinforce the idea that in high workload episodes, there are fewer passing events due to congested traffic, and the field of view metric becomes smaller than in high workload episodes. Additionally, two variables reverse sign relative to their univariate contrasts (number of velocity measurements and uncertain headings go from positive effect sign to negative), which can be interpreted as representing dense yet orderly bicycle streams once density and motion are controlled: many stable tracks naturally yield more velocity estimates, and riders in front moving at similar speeds often produce uncertain directions without implying greater strain. Overall, we interpret the multivariate pattern as consistent with higher workload arising in crowded, slow-progress situations with nearby opposing trajectories [41, 54], while noting these are correlational findings requiring causal validation in future work.

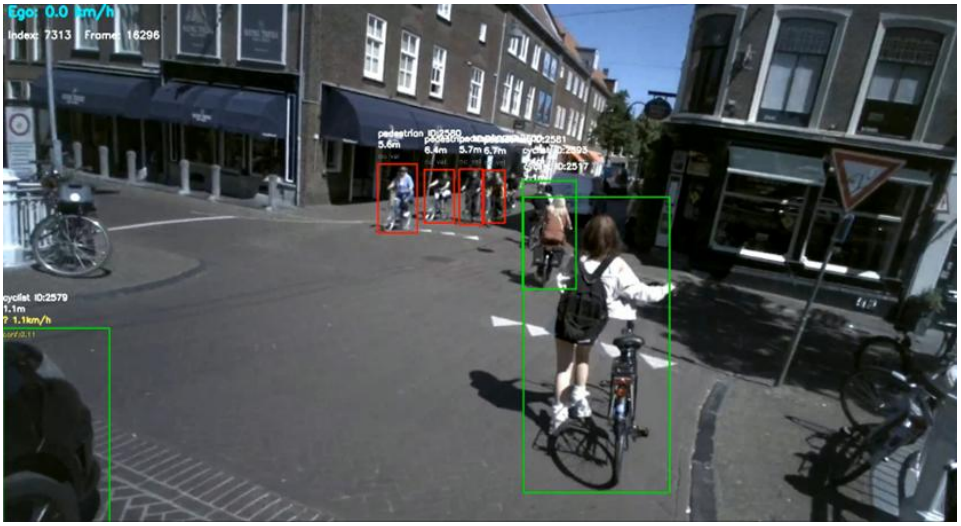


Figure 6.1: Illustration of the inferred high-workload traffic scenario: dense city center with many small objects creating complex motion patterns while the cyclist moves slowly. Detections (in green = cyclist, red = pedestrian) are drawn including their class name and tracking ID at the top as well as their proximity in meters. The relative velocity of the car is given in yellow. In the top left is the ego-velocity of the cyclist.

From rankings to decisions: At a fixed decision threshold, within-route binary classification achieves F1 = 0.478 (+0.108 above our prevalence-matched random baseline baseline), reflecting a workable balance between missed and false detections, while cross-route F1 = 0.427 (with an average improvement of +0.062 above each route's random baseline) is lower but still exceeds chance when compared to each route's own prevalence (Table 5.4). Training-only threshold calibration (Table 5.5) shifts the

operating point toward higher recall at a tolerable precision cost, improving F1 from 0.478 to 0.517 (+0.039 gain) by increasing recall from 55.0% to 78.3% while reducing precision from 42.3% to 38.8%. In practice, the choice of threshold is application-specific: safety auditing may prefer higher sensitivity, whereas on-bike alerts may prioritise precision. These results demonstrate above-baseline practical performance with substantial room for improvement, particularly for cross-route deployment scenarios.

6.1.2. Answer to RQ2 (Can this be done with a scalable monocular pipeline?)

End-to-end monocular extraction is technically viable: All predictors are obtained from eye-tracking video (detection, tracking, monocular depth, consensus velocities) and bicycle GPS only, with design choices aimed at robustness (area/count metrics; central-band clutter; consensus windows). This satisfies the RQ2 constraint of relying on scalable monocular inputs.

Measurement quality is sufficient for downstream use: The single-scale calibration and consensus relative-velocity procedure produce physically plausible distances/speeds; residual error sources (depth noise, short tracks, GNSS calculated velocity lag due to smoothing) are documented and acceptable for ranking proximity and approach/recede episodes [26].

6.2. Limitations

Several methodological and technical constraints affect the interpretation and scope of these findings. These limitations span data collection design, measurement quality, and analytical choices that shape what relationships can be detected and how broadly the results generalise.

6.2.1. Data & design validity

Window aggregation and effective sample size: Adopting short aggregation windows improves label reliability and better reflects how workload is experienced; however, it also reduces the number of analysable episodes after filtering and alignment. The resulting dataset is small relative to the complexity of the phenomenon, which limits statistical power, widens uncertainty bands, and makes cross-route estimates especially sensitive to prevalence differences [51]. In practice, this means that effect sizes and generalisation gaps should be read as indicative rather than definitive, and future work should prioritise collecting balanced data from more routes and different participants.

Labelling and temporal alignment: Workload labels originate from step-like verbal cues, then are mapped to frames. Moving from frame-level labels to short windows improves robustness, and the lag adjustment helps compensate for the data-specific cue timing; nevertheless, residual misalignment remains plausible. Anticipatory cues and delayed cues attenuate instantaneous associations and can invert directions for fast-varying scene metrics. This is a form of label noise, known to bias supervised estimates and shrink real effects [23]. Even physiological markers such as pupil dilation exhibit subsecond latencies to cognitive demand [47], underscoring the general challenge of time-locking labels to visual evidence.

Evaluation design and baselines: Within-route performance is estimated with stratified cross-validation and cross-route performance with leave-one-route-out; under class imbalance, scores must be read against appropriate baselines (ROC–AUC against 0.5; PR–AUC and F1 against the positive-class prevalence) [13]. We follow this practice, including route-specific prevalences for LORO. For the stratified cross-validation however, residual optimism may remain since neighbouring windows still share some context across folds.

6.2.2. Sensing & measurement

Detector/Tracker noise cascades into depth and velocity: The depth inference model produces a substantial amount of noise between frames in the form of overshoots, and because of imperfect detections and tracking which produces ID switches, it is much more difficult to correct and aggressively filter the noise from these incoherent signals as we might be removing important information from an actual track's motion having the same ID as the previous. This ultimately prevents us from obtaining high quality depth information and therefore also forces the use of conservative consensus window based velocity calculations. .

Monocular depth is scale–ambiguous: Using a single global scale K enforces metric consistency but cannot correct local pose and geometry changes or dynamic scene violations [20]. Sub-meter accuracy at mid-range typically requires a calibrated baseline (stereo) or additional cues (ex: LiDAR) [59]. Improving calibration and adding weak geometric priors would help reduce depth noise and simplify relative-velocity estimation.

End-to-end consequences: Because detection $\rightarrow$ tracking $\rightarrow$ depth $\rightarrow$ velocity errors compound, a small improvement upstream often yields a large improvement downstream. In practical terms: fewer ID switches and tighter depth noise directly translate to more reliable approach/recede signals, which were valuable features in our core set.

6.2.3. Feature construction & selection

Zero-inflation and censoring choices: Screening by zero inflation (not selecting specified features with many zeros during the pre-processing stage) and log(1+x) transforms make sense for robustness, but they also shape which relations are detectable. In other words we risk missing a feature-workload relationship where the presence of non-zero value in a specific feature increases the odds of detecting high workload.

Only longitudinal relative velocity is used: The current pipeline estimates approach/recede (range-rate) from depth change but omits tangential components (image-plane x/y). Lateral motion would help deduce time-to-collision and time-to-closest-approach metrics, especially in pedestrian-rich scenes [41]; incorporating full 2D relative velocity would likely improve workload prediction.

6.2.4. Scope relative to RQ2

Pipeline is scalable but not universal: The monocular pipeline (detection, tracking, depth, GPS sync) can run with off-the-shelf components and modest tuning, satisfying RQ2. Robust cross-city deployment will, however, require stronger calibration and/or explicit domain adaptation to maintain accuracy across object detections and depth measurements.

6.3. Future Research

The findings and limitations identified above suggest several promising directions for extending this work. These opportunities range from technical improvements that could enhance measurement precision to broader applications that could increase practical impact for cycling safety and urban planning.

6.3.1. Expanding Beyond Traffic: Infrastructure-Aware Workload Modelling

The current study's focus on traffic condition metrics represents only one dimension of the cyclist's environmental experience. A natural and crucial extension would be to incorporate cycling infrastructure characteristics into the predictive framework. This research has shown that traffic metrics coincide with workload, but this link is weak and it suggests that contextual factors beyond pure traffic conditions are essential.

Future work should develop infrastructure-aware metrics that encode the cycling environment's structural properties per frame or temporal window. These could include:

- **Lane separation type:** protected vs. mixed traffic vs. shared space classification.
- **Infrastructure transitions:** points where cyclists move between different facility types.
- **Contextual hazard zones:** shopping areas, route crossings, surface quality.

By combining traffic dynamics with infrastructure context, models could learn to differentiate "busy but safe" scenarios (ex: high cyclist density in protected lanes) from "busy and risky" situations (ex: mixed traffic with poor sight lines). This multi-dimensional approach would directly address the route-dependency observed in the current results and provide actionable insights for urban planning.

6.3.2. Enhancing Measurement Precision Through Technical Improvements

While this research successfully demonstrated that monocular sensing can provide a signal for workload prediction, several technical enhancements could significantly improve measurement quality and expand the range of detectable interactions.

Additional Motion Analysis: The current pipeline captures longitudinal relative velocity (approach/recede) but misses lateral motion components. Incorporating full 2D relative motion tracking would enable computation of time-to-collision and closest-approach metrics, particularly valuable in pedestrian-rich environments.

Additionally, incorporating time-to-collision (TTC) estimation would provide crucial temporal information about approaching objects. TTC can be calculated from monocular vision using the tau variable ($\theta / \frac{d\theta}{dt}$), which measures the rate of optical expansion without requiring absolute distance measurements [41]. In practice, this can be implemented by tracking the expansion rate of object bounding boxes over time, where θ represents the angular size of the object and $\frac{d\theta}{dt}$ represents its rate of change. This metric can be computed directly from the object detection and tracking pipeline already established. While monocular TTC estimation has known limitations for small, distant objects [27], the close-range interactions typical in urban cycling scenarios would be well-suited to this approach.

Improved Depth and Tracking: The challenges with monocular depth noise and ID switches, while managed through robust filtering, represent clear targets for improvement. Fine-tuning depth estimation models specifically for head-mounted cycling footage and improving the current hybrid tracking approach could reduce noise propagation and enable more sophisticated proximity analysis.

6.3.3. Physiological Integration and Label Quality Enhancement

The reliance on verbal workload cues, while providing interpretable labels, introduces temporal misalignment challenges that likely attenuate true relationships, as mentioned above.

Future research should explore multi-modal workload sensing that combines the environmental metrics established here with continuous physiological indicators:

- **Eye-tracking integration:** Since footage is already captured via eye-tracking glasses, pupil dilation and gaze pattern analysis could provide more precise workload timing.
- **Wearable sensors:** Heart rate variability, skin conductance, and other autonomic indicators could supplement or replace verbal cues.
- **Hybrid labelling approaches:** Combine sparse but precise event-based annotations (using verbal cues for major events) with continuous physiological baselines.

This multi-modal approach would not only improve label quality but it would also enable investigation of different types of cognitive load (visual attention vs. decision-making vs. physical effort).

6.3.4. Scaling Toward Real-World Deployment and Impact

The demonstrated feasibility of monocular workload detection opens pathways toward practical applications that could meaningfully improve cyclist safety and urban planning.

Large-Scale Data Collection: The idea of this scalable pipeline established in this work could enable crowdsourced workload mapping across cities. Future research should build upon this accessible and cheap research strategy in order to allow for mass data collection and workload mappings across different cities and hundreds of riders.

Evidence-Based Infrastructure Design: Perhaps most importantly, this research methodology could inform cycling infrastructure improvements in the future. By systematically measuring workload responses across different infrastructure types and traffic conditions, transportation planners could make evidence-based decisions about cyclist facility design. The interpretable relationship between specific traffic patterns and workload provides a direct bridge from measurement to policy.

7

Conclusion

This thesis examined whether camera-derived traffic metrics coincide with near-instantaneous perceived workload (RQ1) and whether an affordable monocular pipeline can produce those metrics reliably enough for modelling (RQ2). We implemented an end-to-end, cyclist-centric metric extraction pipeline from eye-level video and analysed short windowed episodes with a compact feature set.

Across complementary evaluations, we observed coherent directional patterns and multivariate relationships that translate into detectable performance improvements. Within-route analysis revealed that the model can distinguish between high and low workload episodes with ROC-AUC reaching 0.570, representing a 0.070 improvement above chance performance, while PR-AUC achieved 0.450, exceeding its baseline by 0.080. These values meet established criteria for small effect sizes in predictive research. Binary classification performance demonstrated similar patterns, with F1 scores reaching 0.478, which represents a 0.108 improvement above our baseline expectations. When we calibrated the decision threshold to optimize recall, F1 performance improved further to 0.517 by increasing recall from 55.0% to 78.3%, though at the cost of reduced precision.

Cross-route generalization presented greater challenges, as expected when testing on entirely unseen environments. Performance dropped to ROC-AUC of 0.517, PR-AUC of 0.401 and F1 of 0.427, representing decreases of 0.053, 0.049 and 0.051 respectively from within-route performance. Nevertheless, the model consistently exceeded route-specific baselines across all three test routes.

The feature relationships paint a coherent picture of defensive riding in dense urban environments, where high workload episodes emerge during situations characterized by increased object density, complex motion patterns, and reduced cyclist velocity. However, this study remains correlational and we do not claim causality between traffic conditions and perceived workload.

Several factors constrain the strength of our results and point toward necessary improvements. Our dataset comprised 984 temporal windows drawn from only three routes and one participant, which limits both statistical power and generalizability. Window aggregation improved label robustness but reduced the effective sample size, while step-like verbal cues introduced residual timing uncertainty despite lag adjustment. The simple logistic regression model, while interpretable, does not exploit temporal structure or explicit infrastructure context that could enhance prediction accuracy.

These observations suggest several paths for future work. Augmenting traffic cues with infrastructure-aware descriptors could help models distinguish between busy but manageable and busy but demanding scenes across different cycling facilities. Refining measurement of motion, depth, and lateral velocity information may capture additional workload-relevant signals, while improved temporal alignment and calibration could reduce noise in the ground truth labels. Most importantly, scaling data collection across multiple riders, routes, and cities would enable more robust evaluation of generalization capabilities.

Taken together, this work demonstrates the feasibility of monocular, cyclist-centric sensing for workload inference while establishing transparent baselines and evaluation practices under class imbalance.

While the detected signal remains modest in magnitude, our findings provide quantitative evidence that camera-derived traffic metrics contain detectable workload-related information, laying groundwork for future development of cycling safety monitoring and urban planning applications.

References

[1] Nir Aharon. *BoT-SORT official repository*. GitHub. https://github.com/NirAharon/BoT-SORT. 2022.

[2] Nir Aharon, Ron Orfaig, and Ben-Zion Bobrovsky. "BoT-SORT: Robust Associations Multi-Pedestrian Tracking". In: *arXiv:2206.14651* (2022).

[3] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Standard reference for nearly-constant-acceleration (NCA) motion models. New York: John Wiley & Sons, 2001. isbn: 9780471416555.

[4] Alex Bewley et al. "Simple Online and Realtime Tracking". In: *arXiv:1602.00763* (2016).

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *arXiv* (2020). Reports AP $= 43.5\%$ and $\text{AP}_{50} = 65.7\%$ on MS COCO. eprint: 2004.10934.

[6] Maik Boltes et al. "Empirical Results of Pedestrian and Evacuation Dynamics". In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. Berlin, Heidelberg: Springer, Oct. 25, 2018, pp. 1–29. doi: 10.1007/978-3-642-27737-5_706-1. url: https://link.springer.com/referenceworkentry/10.1007/978-3-642-27737-5_706-1.

[7] Mikel Broström. *BoxMOT (PyPI package)*. https://pypi.org/project/boxmot/. 2025.

[8] Mikel Broström. *BoxMOT: Pluggable SOTA multi-object tracking modules*. GitHub repository. https://github.com/mikel-brostrom/boxmot. 2025.

[9] Jinkun Cao. *OC-SORT official repository*. GitHub. https://github.com/noahcao/OC_SORT. 2023.

[10] Jinkun Cao et al. "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking". In: *CVPR*. 2023.

[11] Álvaro Caviedes and Miguel A. Figliozzi. "Modeling the impact of traffic conditions and bicycle facilities on cyclists' on-road stress levels". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 58 (2018), pp. 488–499. doi: 10.1016/j.trf.2018.06.032.

[12] Chao Chen et al. "How bicycle level of traffic stress correlate with reported cyclist accidents injury severities: A geospatial and mixed logit analysis". In: *Accident Analysis & Prevention* 108 (2017), pp. 234–244. doi: 10.1016/j.aap.2017.09.001. url: https://www.sciencedirect.com/science/article/pii/S0001457517303160.

[13] Jesse Davis and Mark Goadrich. "The relationship between precision-recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[14] Marco Dozza, Gabriella Bianchi Piccinini, and Julia Werneke. "Using naturalistic data to assess e-cyclist behavior". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 41 (2016), pp. 217–226. doi: 10.1016/j.trf.2015.04.003.

[15] Marco Dozza and Julia Werneke. "Introducing naturalistic cycling data: What factors influence bicyclists' safety in the real world?" In: *Transportation Research Part F: Traffic Psychology and Behaviour* 24 (2014), pp. 83–91. doi: 10.1016/j.trf.2014.04.003.

[16] Marco Dozza et al. "How do drivers overtake cyclists?" In: *Accident Analysis & Prevention* 88 (2016), pp. 29–36. doi: 10.1016/j.aap.2015.12.008.

[17] Yunhao Du, Zhicheng Zhao, Yang Song, et al. "StrongSORT: Make DeepSORT Great Again". In: *arXiv:2202.13514* (2022).

[18] B. Dwyer, J. Nelson, T. Hansen, et al. *Roboflow (Version 1.0) [Software]*. Computer vision. 2025. url: https://roboflow.com.

[19] Jessica Edquist et al. *Visual clutter in road environments: What impact on driving performance?* Tech. rep. MUARC Report 299. Victoria, Australia: Monash University Accident Research Centre, 2011.

[20] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network”. In: *Advances in Neural Information Processing Systems (NIPS) Workshop / arXiv preprint* (2014). eprint: 1406.2283. url: https://arxiv.org/abs/1406.2283.

[21] Mark Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338. doi: 10.1007/s11263-009-0275-4.

[22] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[23] Benoît Frénay and Michel Verleysen. “Classification in the Presence of Label Noise: A Survey”. In: *Neurocomputing* 160 (2015), pp. 12–34. doi: 10.1016/j.neucom.2014.10.081.

[24] April Gadsby and Kari Watkins. “Instrumented bikes and their use in studies on transportation behaviour, safety, and maintenance”. In: *Transport Reviews* 40.6 (2020), pp. 774–795. doi: 10.1080/01441647.2020.1769227.

[25] Global Cycling Network. *Are Our Roads Designed To Be Dangerous? 2 Hour Compilation*. YouTube video. url: https://www.youtube.com/watch?v=%3CVIDEO_ID%3E.

[26] Clément Godard et al. “Digging Into Self-Supervised Monocular Depth Estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 3828–3838. doi: 10.1109/ICCV.2019.00393.

[27] Rob Gray and David Regan. “Accuracy of estimating time to collision using binocular and monocular information”. In: *Vision Research* 38.4 (1998), pp. 499–512. doi: 10.1016/s0042-6989(97)00230-7.

[28] Khandker Nurul Habib, Luis L. Losada-Rojas, and Nicholas N. Ferenchak. “Review of the Impacts of Human Factors on Cycling: Perceptions, Workload, and Behavior”. In: *Transportation Research Record* 2678.11 (Nov. 2024), pp. 979–993. doi: 10.1177/03611981241242766.

[29] Frank R. Hampel. “The Influence Curve and Its Role in Robust Estimation”. In: *Journal of the American Statistical Association* 69.346 (1974), pp. 383–393.

[30] Sandra G. Hart. “NASA-Task Load Index (NASA-TLX); 20 Years Later”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 50. 9. 2006, pp. 904–908. doi: 10.1177/154193120605000909.

[31] Sandra G. Hart and Lowell E. Staveland. “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research”. In: *Advances in Psychology* 52 (1988), pp. 139–183.

[32] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004.

[33] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the Knowledge in a Neural Network”. In: *arXiv preprint arXiv:1503.02531* (2015). arXiv: 1503.02531 [stat.ML]. url: https://arxiv.org/abs/1503.02531.

[34] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley, 2013. doi: 10.1002/9781118548387.

[35] Intel ISL. *MiDaS (Monocular Depth Estimation) on PyTorch Hub*. https://pytorch.org/hub/intelisl_midas_v2/. 2023.

[36] Ya Jin, Ming Song, and Xin Yang. *ZoeDepth repository*. https://github.com/isl-org/ZoeDepth. 2023.

[37] Ya Jin, Ming Song, Xin Yang, et al. “ZoeDepth: Zero-Shot Transfer by Overfitting and Postgreps”. In: *arXiv preprint arXiv:2302.12288* (2023).

[38] B. Justusson. “Median Filtering: Statistical Properties”. In: *Two-Dimensional Digital Signal Processing II*. Springer, 1981, pp. 161–196.

[39] Charles F. F. Karney. “Algorithms for geodesics”. In: *Journal of Geodesy* 87.1 (2013). Accurate great-circle (geodesic) distances on WGS-84, pp. 43–55. doi: 10.1007/s00190-012-0578-z.

[40] Fatima Kchour, Salvatore Cafiso, and Giuseppina Pappalardo. “Understanding Cyclists’ Visual Behavior Using Eye-Tracking Technology: A Systematic Review”. In: *Sensors* 25.1 (2024), p. 22. doi: 10.3390/s25010022.

[41] David N. Lee. “A theory of visual control of braking based on information about time-to-collision”. In: *Perception* 5.4 (1976), pp. 437–459. doi: 10.1068/p050437.

[42] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EPnP: An accurate O(n) solution to the PnP problem”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 1–8.

[43] Tsung-Yi Lin et al. “Microsoft COCO: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[44] Carlos Llorca et al. “Motor vehicles overtaking cyclists on two-lane rural roads: Analysis on speed and lateral clearance”. In: *Safety Science* 92 (2017), pp. 302–310. doi: 10.1016/j.ssci.2015.11.005.

[45] Hanspeter A. Mallot et al. “Inverse perspective mapping simplifies optical flow computation and obstacle detection”. In: *Proceedings of the International Conference on Computer Vision*. 1991, pp. 132–135.

[46] Henry B. Mann and Donald R. Whitney. “On a Test of Whether One of Two Random Variables Is Stochastically Larger Than the Other”. In: *Annals of Mathematical Statistics* 18.1 (1947), pp. 50–60.

[47] Sebastiaan Mathôt. “Pupillometry: Psychology, Physiology, and Function”. In: *Journal of Cognition* 1.1 (2018), p. 16. doi: 10.5334/joc.18.

[48] Maaza C. Mekuria, Peter G. Furth, and Hilary Nixon. *Low-Stress Bicycling and Network Connectivity*. Tech. rep. Report 11-19. Mineta Transportation Institute, San José State University, 2012. url: https://scholarworks.sjsu.edu/mti_publications/107/.

[49] Jur Nelissen et al. *Cognitive Workload Among Cyclists*. Bachelor End Project report. CoR BEP Group 12; supervisors: H. Caesar and J. E. N. M. Ronné. Delft, The Netherlands: Delft University of Technology, June 13, 2025.

[50] Armin Niedermüller. “Salzburg Bicycle LiDAR Data Set”. Thesis. Salzburg University of Applied Sciences, Feb. 15, 2023.

[51] Joaquin Quiñonero-Candela et al., eds. *Dataset Shift in Machine Learning*. Cambridge, MA: MIT Press, 2009.

[52] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-Dataset Transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8 (2023), pp. 9653–9668.

[53] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[54] David Regan and Stanley J. Hamstra. “Visual judgements of time to contact”. In: *Perception & Psychophysics* 54.2 (1993), pp. 179–186. doi: 10.3758/BF03211747.

[55] M. E. Rice and G. T. Harris. “Comparing effect sizes in follow-up studies: ROC area, Cohen’s d, and r”. In: *Law and Human Behavior* 29.5 (2005), pp. 615–620. doi: 10.1007/s10979-005-6832-7. url: https://doi.org/10.1007/s10979-005-6832-7.

[56] José A. Rodríguez-Rodríguez et al. “The Impact of Noise and Brightness on Object Detection Methods”. In: *Sensors* 24.3 (2024), p. 821. doi: 10.3390/s24030821.

[57] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. “Measuring visual clutter”. In: *Journal of Vision* 7.2 (2007), p. 17. doi: 10.1167/7.2.17.

[58] Elisabeth Rubie, Narelle Haworth, and Naohide Yamamoto. “Passing distance, speed and perceived risks to the cyclist and driver in passing events”. In: *Journal of Safety Research* 87 (2023), pp. 86–95. doi: 10.1016/j.jsr.2023.09.007.

[59] Daniel Scharstein and Richard Szeliski. “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”. In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 7–42. doi: 10.1023/A:1014573219977.

[60] Andrea Seconi et al. *SaBi3D: Car-To-Bike Overtaking Dataset from LiDAR Surveys at Urban Intersections*. arXiv preprint arXiv:2406.10688. 2024. url: https://arxiv.org/abs/2406.10688.

[61] Claude E. Shannon. “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[62] Jan Peter Simons. “SenseBike Dataset – Addressing LiDAR Domain Gaps through the Introduction of a Novel Dataset from a Bicycle’s Perspective”. Supervised by Holger Caesar. Master’s Thesis. Delft University of Technology, 2024. url: https://sites.google.com/it-caesar.de/homepage/team.

[63] Vukašin Stanojević. *BoostTrack official repository*. GitHub. https://github.com/vukasin-stanojevic/BoostTrack. 2024.

[64] Vukašin Stanojević and Branimir Todorović. “BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking”. In: *Machine Vision and Applications* 35.3 (2024), p. 42.

[65] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR 2020. IEEE/CVF. 2020.

[66] Richard Szeliski. *Computer Vision: Algorithms and Applications*. 2nd ed. Chs. on planar homographies and parallax. Springer, 2022.

[67] Ruben Terwint. *Cyclist’s Workload in Urban Environments: A Literature Review*. Literature review. Faculty of Mechanical Engineering; supervisors: Holger Caesar and Jason Moore; student no. 5362679. Delft, The Netherlands: Delft University of Technology, May 2025.

[68] Tobii. *Tobii Pro Glasses 3*. Product page. 2025. url: https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3 (visited on 05/26/2025).

[69] TU Delft. *How a single bike can improve safety for all cyclists*. Web article. 2024. url: https://www.tudelft.nl/en/stories/articles/how-a-single-bike-can-improve-safety-for-all-cyclists (visited on 05/26/2025).

[70] Ultralytics. *COCO Dataset — Pretrained YOLO11 Models (n/s/m/l/x)*. https://docs.ultralytics.com/datasets/detect/coco/. States that YOLO11n/s/m/l/x models are pretrained on COCO. 2024. (Visited on 08/28/2025).

[71] Ultralytics. *YOLOv8 Models*. Documentation. Jan. 10, 2023. url: https://docs.ultralytics.com/models/yolov8/ (visited on 08/17/2025).

[72] Sergio A. Useche et al. “Infrastructural and Human Factors Affecting Safety Outcomes of Cyclists”. In: *Sustainability* 10.2 (2018), p. 299. doi: 10.3390/su10020299.

[73] Ian Walker. “Drivers overtaking bicyclists: Objective data on the effects of riding position, helmet use, vehicle type, and apparent gender”. In: *Accident Analysis & Prevention* 39.2 (2007), pp. 417–425. doi: 10.1016/j.aap.2006.08.010.

[74] Christopher D. Wickens and Justin G. Hollands. *Engineering Psychology and Human Performance*. 4th. Pearson Prentice Hall, 2008.

[75] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. “Simple Online and Realtime Tracking with a Deep Association Metric”. In: *arXiv:1703.07402* (2017).

[76] World Health Organization. *Road traffic injuries*. Fact sheet. Updated 13 Dec 2023; accessed 28 Aug 2025. 2023. url: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[77] Lihe Yang et al. *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data*. arXiv preprint arXiv:2401.10891. 2024. doi: 10.48550/arXiv.2401.10891. url: https://arxiv.org/abs/2401.10891.

[78] *ZED-F9P-04B: High precision GNSS module — Data sheet (UBX-21044850, R05)*. Table 3 lists horizontal position accuracy: 1.5 m CEP (PVT), 1.0 m CEP (SBAS), 0.01 m + 1 ppm (RTK). u-blox. Mar. 21, 2024. url: https://content.u-blox.com/sites/default/files/ZED-F9P-04B_DataSheet_UBX-21044850.pdf.

[79] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334.

[80] Jun Zhao, Wei Li, Ming Zhu, et al. “Depth Anything V2: Distilling Synthetic Monocular Depth for Real-World Data”. In: *NeurIPS*. 2024.

A

Further Details

A.1. Workload Balance and Per-Video Diagnostics

The three figures A.1, A.2, A.3 below show the Workload distributions for each chosen video segment which are all around 11-12 minutes long and sampled at 25 Hz with the original scoring given by the BEP students (before turning into binary scoring).

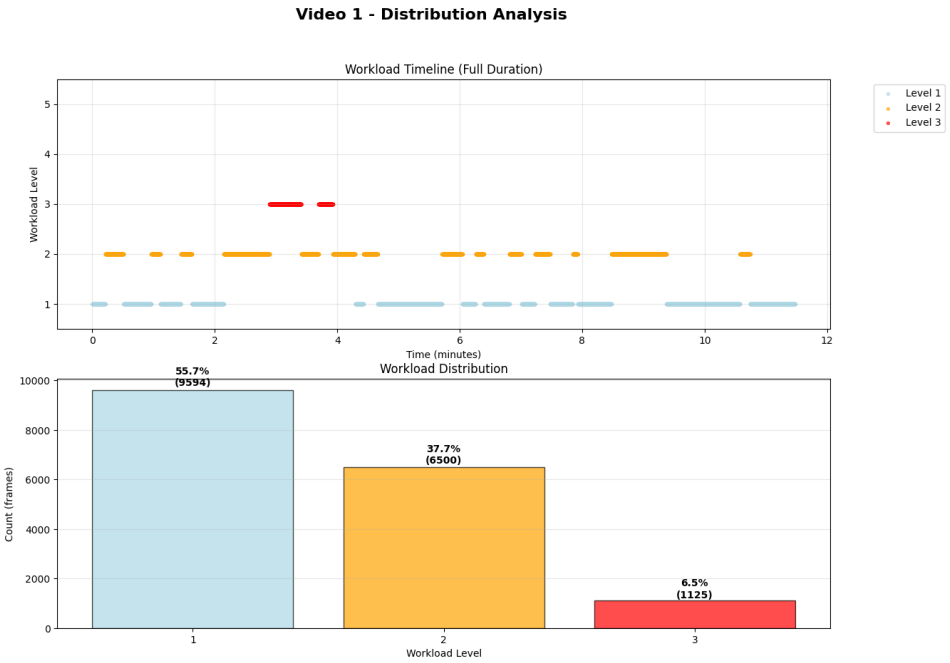


Figure A.1: Workload timelines and balance for Video 1

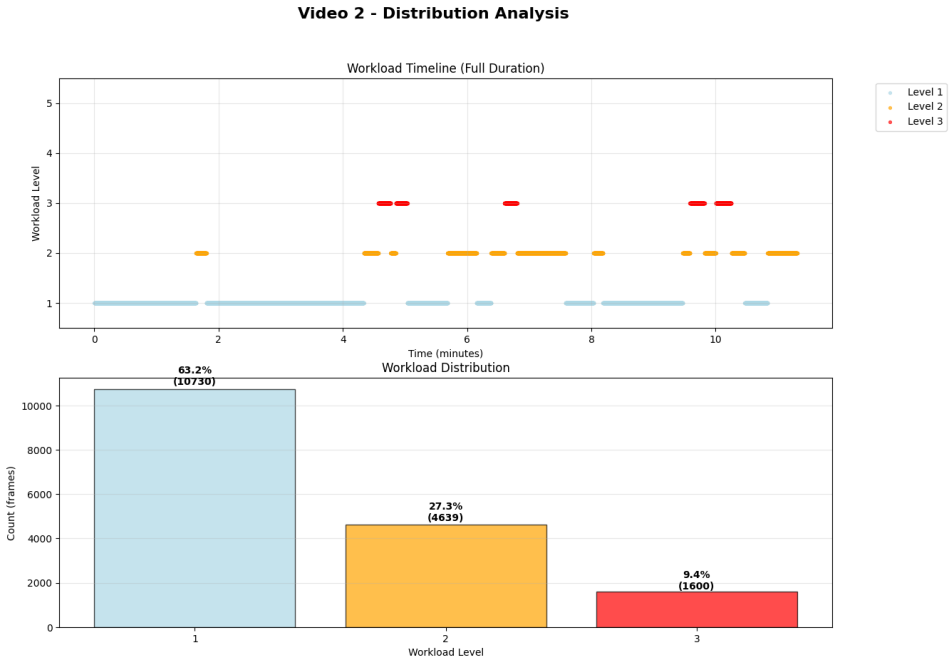


Figure A.2: Workload timelines and balance for Video 2

The distributions in A.2 clearly show the spoken about route progression aspect, where the sample starts on Delft Campus and progresses to the City Center.

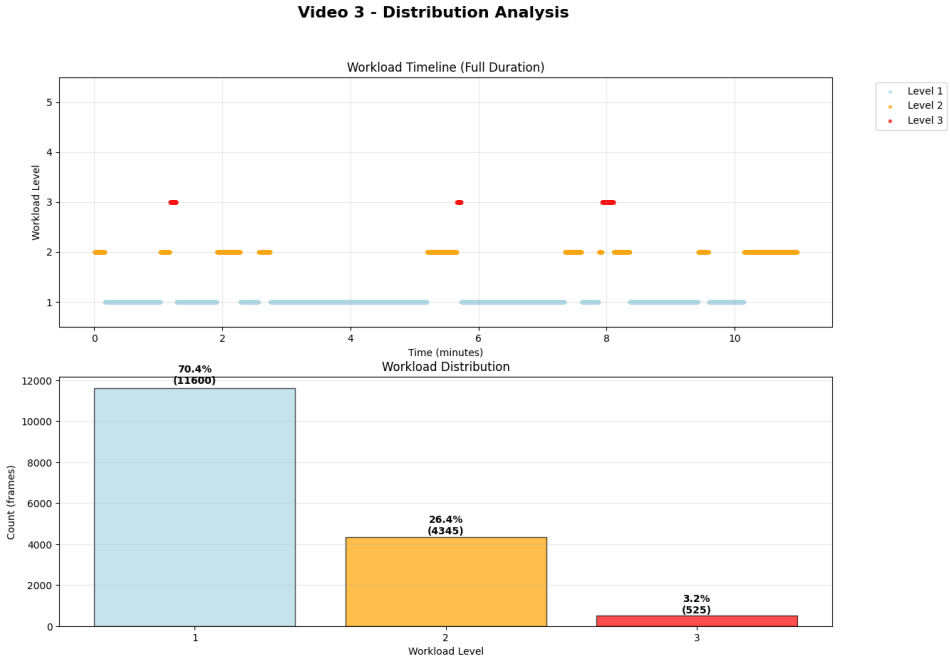


Figure A.3: Workload timelines and balance for Video 3

A.2. Attempt at rider-bike merge before YOLO fine-tune

To address the bike/motorcycle and rider separation, an intermediate approach was tested: post-processing the raw YOLO outputs using a rider-cycle merging algorithm. Two variants were implemented:

- Intersection over Union (IoU)-based merging, where a person and a bicycle or motorcycle are merged if their bounding boxes overlap above a certain threshold.
- Centre-proximity merging, where merging is triggered if the bounding box centers are within a specified pixel distance (ex: 120 pixels), accounting much better for front/back views of cyclists/scooter riders where the bounding boxes are very narrow as well as being stacked on top of each other (low overlap), making it much more stable than the IoU merge.

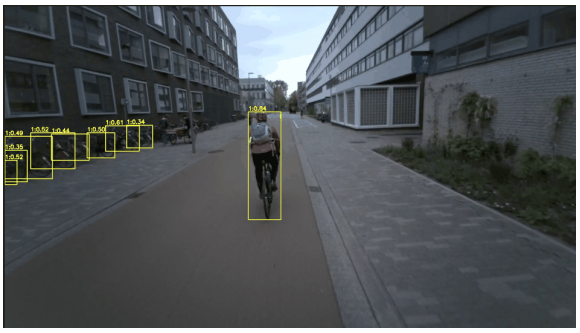


Figure A.4: Successful merge on cyclist (yellow = bicycle class, class number and confidence score indicated on top of bounding boxes)

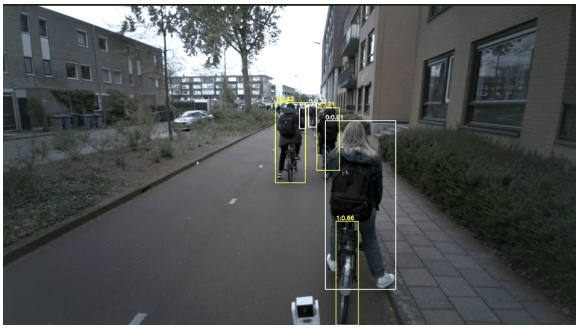


Figure A.5: Merge working only for 2/3 of the detected objects fully in view

Despite some promising examples (see Figures A.4 and A.5), the merging strategy proved unreliable across varied perspectives, occlusions, and object sizes. The wide variation in bounding box aspect ratios caused by changes in distance and object orientation (ex: profile versus frontal views) led to inconsistent merging. Loosening the parameters to improve recall introduced significant noise, while stricter thresholds missed valid merges. In addition, the computational cost and complexity of integrating these corrections into the pipeline in real time made this method unsustainable.

A.3. Initial exploration of state of the Art Multi Object Trackers (MOT)

The purpose of tracking in this pipeline is to link per–frame detections into temporally consistent trajectories so that time-dependent metrics (ex: relative speed) can be computed. We first evaluated state-of-the-art multi-object trackers available through the BoxMOT framework using the fine-tuned YOLO detections as input. Four representative trackers were compared: OC-SORT, BoT-SORT, StrongSORT, and BoostTrack. Each was configured and tuned with and without appearance re-identification (ReID), with and without global motion compensation (GMC) when available, and with both IoU and centre-distance based association where supported [8, 7]. Post-processing steps such as short-gap interpolation and track validation were also tested.

Trackers considered (high-level summary). All four trackers are built on the SORT/DeepSORT family and use a linear Kalman filter with a constant-velocity state in image coordinates (unless explicitly disabled), combined with geometric gating, a Hungarian association algorithm and, in some cases, appearance cues and global motion compensation.[4, 75]

- OC-SORT: observation-centric SORT that reduces sensitivity to noisy detections by updating the motion state with the latest observations and using geometry-based association (IoU and centre distance). Uses the Kalman constant-velocity model in the image plane [9, 10].
- BoT-SORT: integrates appearance embeddings (ReID), IoU association, and optional global motion compensation to discount camera motion; two-stage association can exploit low-score detections to recover missed matches. Uses the same Kalman constant-velocity backbone[2, 1].
- StrongSORT/StrongSORT++: an improved DeepSORT variant with stronger ReID features, enhanced track management, and optional global motion compensation; designed for crowded scenes and longer occlusions. Also based on the Kalman constant-velocity model [17].
- BoostTrack: augments motion-plus-IoU association with a boosting step that attempts to revive or bridge short track gaps using appearance and geometric cues, on top of a Kalman constant-

velocity motion model [64, 63].

A concise comparison of main cues is provided in Table A.1; parameters such as IoU gates, max_age, min_hits, ReID cosine-distance thresholds, and global motion compensation options were tuned per tracker.

Table A.1: Trackers evaluated and principal cues used for association. All use a linear Kalman filter with a constant-velocity state in the image plane(implementations via BoxMOT, [8]).

Tracker	Motion model	Geometric cue	Appearance cue	GMC
OC-SORT	Kalman (const. vel.)	IoU / centre distance	–	–
BoT-SORT	Kalman (const. vel.)	IoU	ReID	optional
StrongSORT	Kalman (const. vel.)	IoU	ReID (strong)	optional
BoostTrack	Kalman (const. vel.)	IoU / heuristics	ReID	–

Outcome on cyclist POV video. Despite careful parameter tuning, the resulting trajectories were under-specified for my use case. The most frequent failure modes were:

1. Premature track termination and late birth: tracks would appear only when objects were already near, or would drop just before or during a close pass, losing precisely the segments needed for proximity and approach-rate metrics.
2. Unstable ReID cues: appearance descriptors were unreliable for distant, small, or motion-blurred targets and under rapid viewpoint change; enabling ReID often increased ID switches instead of reducing them.
3. Ineffective global motion compensation: compensation modules based on planar homographies could not reliably cancel the head-mounted camera motion in near-field, non-planar urban scenes with strong parallax; the residual error sometimes degraded association.

Why off-the-shelf trackers struggle on head-mounted cyclist footage. These trackers are highly effective in surveillance or vehicle-mounted datasets where camera motion is static or smoothly varying and object scale changes are moderate. In contrast, head-mounted cyclist POV introduces:

- Rapid rotational head-motion and micro-jitter: quick yaw or pitch during shoulder checks and continuous vibration from the road cause large, non-linear apparent motion in the image between consecutive frames. This violates the constant-velocity assumption of the Kalman model and pushes the predicted box outside the geometric gate, breaking short-horizon association [4, 75].
- Strong perspective and scale change: as objects approach, their projected size grows rapidly and their apparent shape changes with viewpoint. IoU between the predicted box and the next detection can drop sharply even when it is the same physical object, because the bounding box inflates and aspect ratio changes nonlinearly frame to frame. Centre-distance is more tolerant but still affected by large apparent jumps due to camera rotation.
- Frequent partial occlusions and truncation: during head turns, targets are often cropped by image borders or hidden behind nearby objects for several frames. If the gap exceeds the tracker's max_age, the track is terminated; when the target reappears with different scale and position, association frequently fails and a new ID is created.
- Non-planar scenes with near-field parallax: global motion compensation commonly estimates a single *homography* to align successive frames.[1] This approach assumes a roughly planar scene or gentle camera motion. In cyclist POV, nearby objects lie at very different *depths* from the camera, so their apparent displacements differ under translation; this depth-dependent differential motion is *parallax*.[2] As a result, a single homography cannot correctly stabilize all regions, leaving residual motion that harms association [66].

[1] A homography is a single (3x3) projective transform that maps image points between two views under a planar-scene assumption or pure camera rotation about its optical centre. It is widely used to remove global background motion before association.

[2] Parallax is the phenomenon where objects at different depths move by different amounts in the image when the camera translates. In non-planar, near-field urban scenes, a single homography cannot simultaneously align near and far regions.

Figure A.6: Example of a typical failure case with off-the-shelf trackers (StrongSort in this case): track (car) dies as it enters proximity due to scale change from far to near and failed detection matching.

Motivation for a simpler, detection-driven tracker. Because the fine-tuned detector achieved high recall on the target classes, We adopted a different strategy: reduce model complexity and rely more directly on the detections themselves for association, favouring geometry that is robust to the above issues. This led to the hybrid tracker described in the next subsection, which combines IoU association with a centre-distance fallback, lightweight velocity prediction, and short-gap interpolation tuned to the cyclist POV dynamics.

A.4. Correlation Analysis of Predictor Variables

To assess the relationships between predictor variables in the final dataset, a Pearson correlation matrix was computed for all continuous predictors used in the logistic regression models. Figure A.7 presents the correlation coefficients between all eight predictor variables.

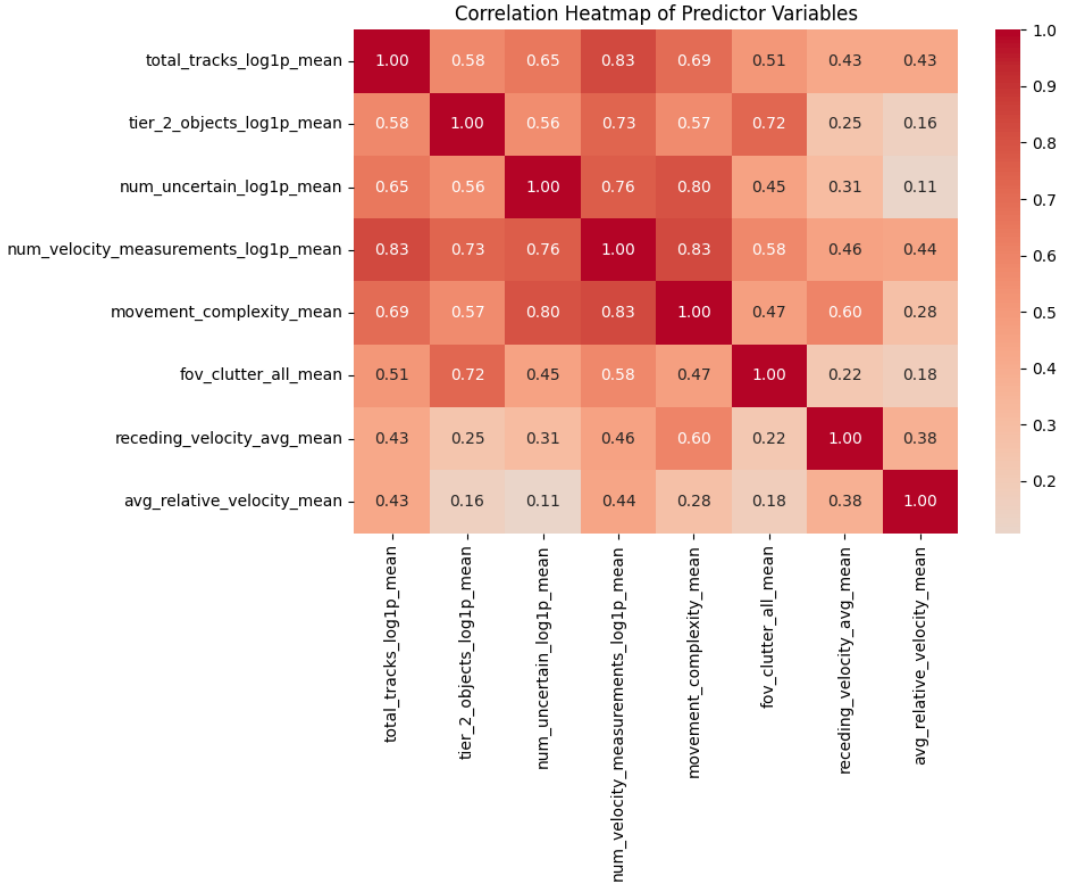


Figure A.7: Correlation matrix heatmap of predictor variables. Values represent Pearson correlation coefficients, with darker red indicating stronger positive correlations and blue indicating negative correlations.

The correlation analysis reveals several notable patterns in the relationships between predictor variables. The strongest correlations were observed between total tracks and number of velocity measurements (r = 0.83), number of uncertain objects and movement complexity (r = 0.80), and number of velocity measurements and movement complexity (r = 0.83). Multiple variables also showed moderate correlations, particularly among the object detection and tracking metrics, with correlation coefficients ranging from 0.56 to 0.76.

The presence of moderate to high correlations between predictor variables indicates multicollinearity, which is expected given that these measures all relate to traffic complexity and that they are obtained through similar measures. Despite these correlations, the variables were retained in the analysis as each captures a theoretically distinct aspect of the cycling environment that may contribute uniquely to workload perception beyond their statistical overlap. Additionally, the logistic regression models employed L2 regularization, which helps mitigate multicollinearity effects by shrinking correlated coefficients and reducing overfitting.

The exploratory nature of this study necessitates the inclusion of conceptually distinct measures even when they show statistical correlation, as the goal is to identify which aspects of traffic complexity are most predictive of workload. Furthermore, in naturalistic cycling contexts, traffic complexity measures are inherently correlated, and isolating completely independent predictors would not accurately reflect the actual riding environment.

A.5. Visual Analysis of LORO Model Performance

To complement the quantitative LORO results presented in section 5.3, we provide visual representations of model predictions across the three held-out routes. These figures illustrate the temporal patterns of workload prediction and reveal route-specific characteristics that influence cross-route generalization performance.

A.5.1. Interpreting the Visualization

Both figures display model predictions as continuous probability scores (blue/orange/green lines) alongside actual high-workload episodes (red segments at top and bottom). The horizontal dashed line at 0.5 represents the decision threshold for binary classification. Predictions above this threshold are classified as high workload, while those below are classified as low workload.

In Figure A.8, the continuous probability traces reveal the model's confidence throughout each route. High-workload episodes are marked as red segments, allowing direct visual assessment of whether the model assigns higher probabilities during these periods. The accuracy percentages indicate overall classification performance when applying the 0.5 threshold.

Figure A.9 provides a complementary segment-level analysis where predictions are averaged within each workload episode (high or low). This aggregation approach tests whether the model can distinguish between entire workload segments rather than individual temporal windows. Green dashed lines indicate correct predictions (high-workload segments predicted above 0.5, low-workload segments predicted below 0.5), while yellow dashed lines indicate incorrect predictions. Both window-level accuracy (matching Figure A.8) and segment-level accuracy are reported.

The LORO model performance varies considerably across the three held-out routes. For temporal window predictions, Route 1 achieves 46.1% accuracy (154/334 correct predictions), Route 2 achieves 53.3% accuracy (176/330 correct predictions), and Route 3 achieves 65.9% accuracy (211/320 correct predictions).

At the segment level, where predictions are aggregated within workload episodes, the model correctly classifies 44.0% of segments in Route 1 (11/25 segments), 50.0% of segments in Route 2 (7/14 segments), and 58.8% of segments in Route 3 (10/17 segments). The segment-level analysis reveals that Route 1 contains 12 high-workload and 13 low-workload segments, Route 2 contains 7 high-workload and 7 low-workload segments, and Route 3 contains 9 high-workload and 8 low-workload segments across the respective route durations.

A.5.2. Route-Specific Patterns

The visualizations reveal distinct performance patterns across routes, with accuracy ranging from 46.1% (Route 1) to 65.9% (Route 3). Route 1 shows the most variable probability traces with frequent threshold crossings, reflecting the diverse traffic conditions along its mixed urban-suburban itinerary, resulting in the lowest classification accuracy. Route 2 demonstrates more stable predictions, particularly during extended high-workload segments, achieving intermediate performance (53.3% accuracy). Route 3 exhibits the most consistent above-threshold predictions during high-workload episodes, achieving the highest accuracy (65.9%) despite having the lowest proportion of high-workload labels (28.7%).

These visual patterns support the quantitative finding that cross-route generalization varies substantially by route characteristics, with the model showing particular sensitivity to the traffic patterns and infrastructure contexts that define each route's unique cycling environment.

Figure A.8: Temporal visualization of LORO model predictions across three held-out routes. Continuous lines show predicted workload probabilities over time, with red segments indicating actual workload episodes. The horizontal dashed line at 0.5 represents the classification threshold. Accuracy percentages reflect binary classification performance at this threshold.

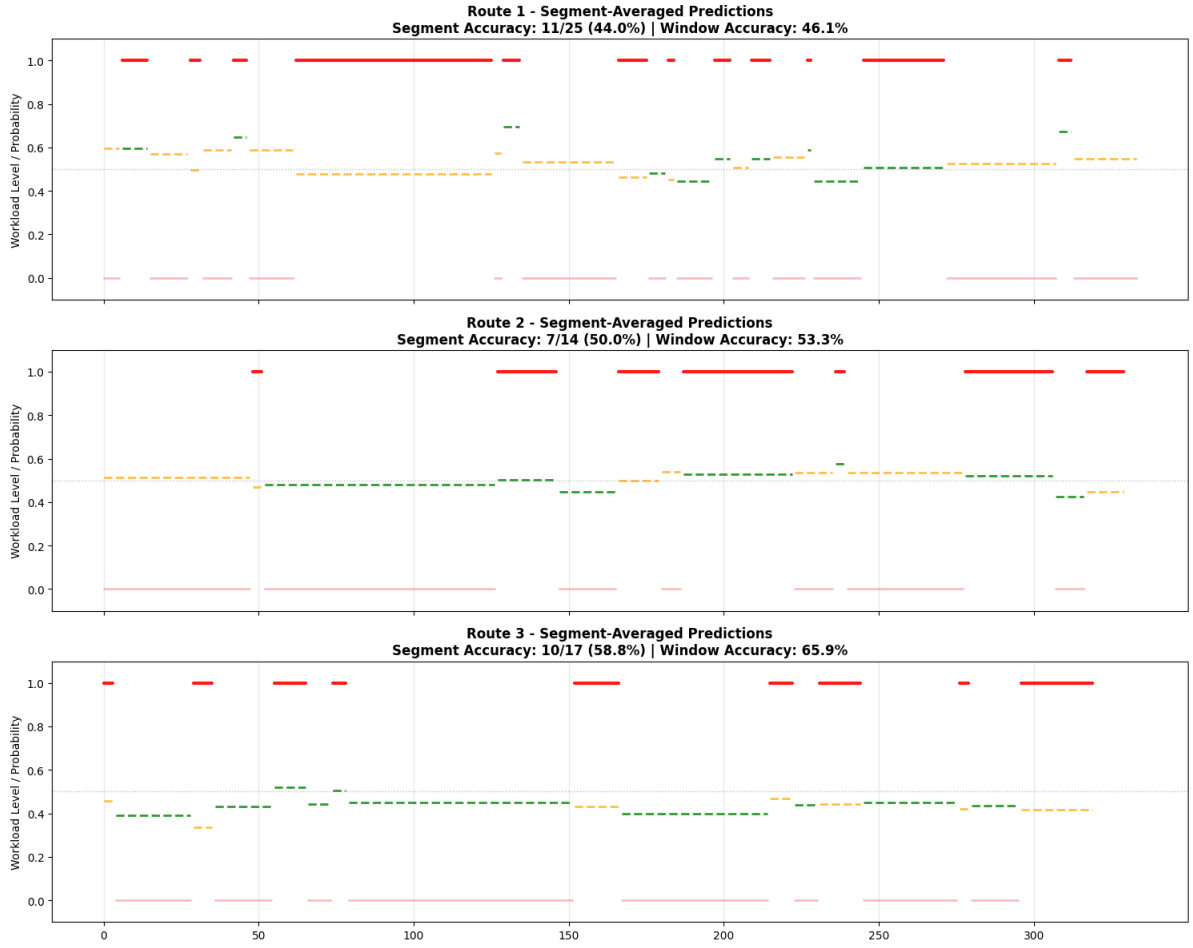


Figure A.9: Segment-averaged LORO predictions showing model performance at the workload episode level. Predictions are averaged within each high-workload (red) and low-workload (pink) segment. Green dashed lines indicate correct predictions relative to the 0.5 threshold, while yellow dashed lines indicate incorrect predictions. Both window-level and segment-level accuracy metrics are provided to assess temporal versus episodic classification performance.

B

Formulas

B.1. Statistical Analysis Formulas

B.1.1. Logistic Regression Model

In logistic regression, we model the log-odds of the outcome as:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} \tag{B.1}$$

where $\pi_i = P(Y_i = 1|\mathbf{x_i})$ is the probability of high workload for observation i.

B.1.2. Parameter Estimation

Coefficients are estimated using maximum likelihood estimation (MLE). The variance-covariance matrix of coefficient estimates is:

$$\text{Var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1} \tag{B.2}$$

where $\mathbf{X}$ is the design matrix and $\mathbf{W}$ is a diagonal matrix with elements $w_i = \pi_i(1-\pi_i)$.

The standard error of the j-th coefficient is:

$$SE(\hat{\beta}_j) = \sqrt{[(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}]_{jj}} \tag{B.3}$$

B.1.3. Wald Test Statistic

For testing $H_0 : \beta_j = 0$ against $H_1 : \beta_j \neq 0$, the Wald test statistic is:

$$z_j = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \tag{B.4}$$

Under the null hypothesis, $z_j \sim N(0,1)$.

B.1.4. P-value Calculation

The two-sided p-value is:

$$p\text{-value} = 2 \times [1 - \Phi(|z_j|)] \tag{B.5}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

A significant p-value ($p < 0.05$) indicates rejection of the null hypothesis, suggesting the predictor has a statistically significant relationship with the log-odds of the outcome, conditional on all other predictors in the model. [34]

B.1.5. Mann-Whitney U test

The Mann–Whitney U test [46] is defined as follows: Pool the samples, rank them, and let R_1 be the sum of ranks for the $Y = 1$ group. Then

$$U_1 \;=\; n_1 n_0 + \frac{n_1(n_1+1)}{2} - R_1, \qquad U_0 \;=\; n_1 n_0 - U_1, \qquad U = \min(U_0, U_1).$$

For large samples, the normal approximation gives

$$\mu_U = \frac{n_1 n_0}{2}, \qquad \sigma_U = \sqrt{\frac{n_1 n_0 (n_1 + n_0 + 1)}{12}}, \qquad Z = \frac{U - \mu_U}{\sigma_U},$$

(with the standard tie correction to σ_U when ties occur), and the two-sided p-value is $p = 2\big(1 - \Phi(|Z|)\big)$, where Φ is the standard normal CDF.

B.2. Evaluation Metrics

Let TP = true positives, FP = false positives, TN = true negatives, FN = false negatives for the high-workload class. Let n^+ and n^- denote the number of positive and negative samples, respectively.

F1 Score

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

Intersection over Union (IoU). Given a predicted bounding box B_p and a ground-truth box B_g, the IoU is

$$\mathrm{IoU}(B_p, B_g) = \frac{\mathrm{area}(B_p \cap B_g)}{\mathrm{area}(B_p \cup B_g)} = \frac{\mathrm{area}(B_p \cap B_g)}{\mathrm{area}(B_p) + \mathrm{area}(B_g) - \mathrm{area}(B_p \cap B_g)} \in [0, 1].$$

It equals 0 for disjoint boxes and 1 for identical boxes. A detection is considered a match at threshold τ (e.g., τ=0.50) only if its class is correct *and* $\mathrm{IoU} \geq \tau$. (For segmentation, replace “area of box” with the number of pixels in the respective masks.)

ROC-AUC For a binary classifier producing scores s_i for samples $i = 1, \ldots, n$:

$$\text{ROC-AUC} = \frac{1}{n^+ n^-} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbb{I}(s_i > s_j)$$

where $\mathcal{P}$ and $\mathcal{N}$ are the sets of positive and negative sample indices, and $\mathbb{I}(\cdot)$ is the indicator function. This counts the fraction of positive-negative pairs where the positive sample receives a higher score [22].

PR-AUC (Average Precision) Let (r_k, p_k) be precision-recall pairs at thresholds t_k, ordered by increasing recall $r_1 \leq r_2 \leq \cdots \leq r_K$:

$$\text{PR-AUC} = \sum_{k=2}^{K} (r_k - r_{k-1}) p_k$$

This computes the area under the precision-recall curve using the interpolated/step-wise rule [13]. For a random classifier in an imbalanced dataset, PR-AUC $= \frac{n^+}{n^+ + n^-}$ (the positive class prevalence).