Application of machine learning models for tracking and counting heavy vehicles on highways

Brenner dos Santos Araujo
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0009-0001-9422-961X

Acbal Rucas Andrade Achy
PPGEEC – UFRB
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0002-4063-0774

Daniel José Cerqueira Brito
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0009-0006-4018-255X

Tiago Palma Pagano
PPGEEC – UFRB
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0003-2457-9064

Weiner Gustavo Silva Costa
CETEC – PaveLab
UFRB
Cruz das Almas, Brasil
ORCID: 0000-0001-6151-8805

Mario Sergio Souza de Almeida
CETEC – PaveLab
UFRB
Cruz das Almas, Brazil
ORCID: 0000-0002-0222-4804

***Abstract*—Accurate traffic volume estimation is essential for road infrastructure planning, particularly for the design and maintenance of asphalt pavements. This study presents a comparative evaluation of two object tracking algorithms, BoT-SORT and ByteTrack, applied to the task of monitoring heavy vehicles in real-world road environments using side-view videos. Detection was performed using both pre-trained and customized YOLOv8 models, with the latter tailored to the vehicle categories defined by the DNIT traffic manual. A single-line counting technique was employed to mitigate false negatives caused by occlusions. The experiments were conducted using real data from the BR-110 highway, and performance was evaluated based on precision, recall, and F1 score metrics. The results demonstrate that customized YOLOv8 models, particularly when combined with ByteTrack, achieved superior performance, with a significant reduction in false positives. Previous studies indicate that camera positioning and model adaptation to specific data are critical factors for enhancing detection accuracy in side-view scenarios, especially in environments subject to occlusions and heavy traffic. In this study, we observed that the use of customized models led to improved performance; however, no comparative experiments were conducted with different camera positions, which limits the generalizability of the findings.**

***Index Terms*—YOLO, Tracking, CNN, ByteTrack, BoT-SORT, DNIT, Traffic, Camera.**

I. Introduction

Traffic monitoring plays a crucial role in transportation engineering, particularly in the design and maintenance of asphalt pavements. The number of vehicles — especially those with a gross vehicle weight (GVW) equal to or exceeding 3.5 tons — traveling on a highway is directly related to the pavement's service life, significantly influencing resource allocation and technical decision-making in road projects. According to the Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Structures [1], traffic volume is one of the primary parameters in the mechanistic-empirical pavement design process, being essential to ensure adequate structural performance over time. Therefore, obtaining accurate traffic volume estimates constitutes a strategic requirement for public administrators and civil engineers, particularly given the impact of these estimates on the durability and overall performance of road infrastructure.

In Brazil, the Traffic Studies Manual (IPR-723), published by the National Department of Transportation Infrastructure (DNIT) [2], establishes guidelines and methodologies for conducting traffic counts on highways, with an emphasis on vehicle classification, including heavy vehicles. This manual is fundamentally important for ensuring the collection of accurate and standardized data, thereby contributing to effective pavement planning and the formulation of public policies in the transportation sector. Accurate counting of heavy vehicles, for instance, enables more precise estimations of pavement damage, leading to more appropriate structural design and a more efficient highway life cycle. Thus, the traffic counting methodology established by DNIT aligns with the objective of this study, which seeks to integrate modern detection and tracking technologies to optimize road traffic counting and analysis, particularly in scenarios involving heavy vehicles.

With recent technological advances, computational methods are increasingly used to automate and improve traffic data collection. Convolutional Neural Networks (CNNs) are prominent in vehicle detection tasks using highway camera footage [3], [4]. Once objects are detected, vehicle counting is performed through tracking algorithms that associate bounding boxes over time to maintain each vehicle's identity [5]–[7], [9]. These algorithms predict future positions using spatial, velocity, and appearance cues, handling brief occlusions or positional changes [10], [11].

Camera positioning strongly affects detection reliability. Top-down views reduce occlusion but limit identification of features like silhouette and axle count. Side-view placements allow better morphological analysis but are more prone to occlusion, especially in heavy traffic. Studies confirm that camera angle involves a trade-off between minimizing occlusion and capturing key vehicle features [14], [16], [17].

To address the challenge of maintaining vehicle identity across frames, several tracking algorithms have been developed, including ByteTrack [9] and the SORT family of algorithms [5]. These algorithms differ in their robustness to occlusions, re-identification (Re-ID) capabilities, computational complexity, and tracking accuracy. Among them, BoT-SORT [13] and ByteTrack stand out by offering an effective balance between performance and ease of integration with modern object detection models, such as YOLOv8 (You Only Look Once, version 8).

Given this context, the present study aims to conduct a comparative evaluation of two tracking algorithms—BoT-SORT and ByteTrack—applied to both pre-trained and customized models for the task of counting heavy vehicles, as defined by the DNIT traffic manual [2]. The images used for the analysis were extracted from videos capturing the side-view silhouette of vehicles and integrated with the YOLOv8 detection model. The evaluation was conducted using metrics such as counting accuracy, false positive rate, and false negative rate, with the objective of assessing the impact of algorithmic choices in side-view scenarios on real highways.

Despite this progress, three gaps remain for highway-side traffic monitoring focused on heavy vehicles in Brazilian conditions: (i) most public benchmarks and reported deployments assume near top-down or urban intersection camera geometry rather than oblique side-view highway placements; (ii) heavy-vehicle taxonomies specific to DNIT classes are underrepresented in open datasets, limiting model transferability; and (iii) few studies quantify how downstream multi-object tracking affects *directional* counting accuracy when detections originate from a detector trained (or fine-tuned) for local vehicle classes. These limitations motivate the work presented here.

Unlike most traffic video studies that assume top-down or urban-intersection viewpoints and rely on generic vehicle classes (e.g., *car*, *truck*), this work targets the underexplored problem of *side-view highway monitoring of heavy vehicles* under Brazilian operating conditions and DNIT engineering practice. We curate and annotate a field dataset from BR-110 and train/evaluate a DNIT-aligned YOLOv8 detector (approximately 12,000 labeled images spanning 11 of the 32 DNIT vehicle classes), enabling domain-aware heavy-vehicle detection beyond COCO's coarse labels. Building on this, we present a fully modular monitoring pipeline in which frame-level detection feeds downstream multi-object tracking (BoT-SORT or ByteTrack) and a centroid-triggered directional counting stage; this design avoids hybrid-model ambiguity and allows independent substitution or tuning of components. We provide the first (to our knowledge) comparative study of BoT-SORT and ByteTrack in a lateral highway geometry with substantial foreground clutter, analyzing their impact on counting performance and error modes (duplicate IDs, occlusions, fueling-area interference). We also implement and validate a lightweight single-line virtual counting strategy suitable for estimating directional heavy-vehicle volumes when camera placement constrains the field of view. Finally, we discuss operational considerations for deployment in which roadside IP cameras stream video to centralized GPU servers, outlining scalability paths toward multi-site, real-time traffic monitoring and future expansion to the full 32-class DNIT taxonomy.

The remainder of this paper is organized as follows. Section II reviews related work on traffic monitoring, focusing on detection and tracking techniques for highway environments. Section III describes the proposed methodology, including the overall processing pipeline, data acquisition, detection model, tracking algorithms, counting strategy, and evaluation metrics. Section IV presents the experimental setup, performance analysis, and a discussion of the obtained results. Finally, Section V concludes the paper and outlines directions for future work.

II. State of the Art Review

Numerous studies have been conducted to facilitate object counting, ranging from people and vehicles to more complex scenarios such as dense crowds [8], animals in natural environments, drones in air traffic, and vessels in aquatic settings. These applications present additional challenges related to occlusion, scale variation, and unpredictable movement.

Regarding vehicle detection, several works have reported improvements in the performance of the YOLO algorithm. Sang et al. [19] proposed enhancements to YOLOv2 aimed at optimizing performance in urban environments, achieving a significant accuracy increase with a mean Average Precision (mAP) of 90.38% on the UA-DETRAC dataset, along with reduced inference time, rendering the approach suitable for real-time applications. Wang et al. [20] developed VV-YOLO, based on YOLOv4, which incorporates attention mechanisms to handle varying viewing angles, reaching an mAP of 91.2%—a 4.5% improvement over standard YOLOv4—and demonstrating greater robustness in occluded and oblique-angle environments. Liao et al. [21] focused on small vehicle detection in urban scenarios with SAM-YOLO, and more recently, Alif [22] analyzed YOLOv11, emphasizing its advantages in intelligent transportation applications.

In the domain of object tracking, techniques such as the Kalman Filter are widely employed to associate detections across consecutive frames, ensuring consistent counting over time. Baş et al. [23] incorporated optical flow into the Kalman Filter to improve object matching in occlusion scenarios, while Lin and Sun [25] proposed a method combining YOLO and coordinate tracking to address challenges such as duplicate counting and temporary detection failures. Balid et al. [24] explored complementary approaches to visual tracking, including magnetic sensors, for real-time counting and classification, highlighting the complexities of traffic environments.

Another critical factor influencing the quality of vehicle detection and tracking is camera positioning, particularly in scenarios involving heavy traffic and multiple lanes. Cameras mounted at high angles, such as top-down views, tend to minimize occlusion by reducing vehicle overlap; however, they limit the identification of specific vehicle features, such as silhouette and number of axles. Conversely, side-view cameras provide a more detailed analysis of vehicle morphology but are more susceptible to partial or complete occlusion.

Kanhere et al. [26] demonstrated that low-mounted cameras encounter significant challenges related to vehicle occlusion, complicating reliable tracking. To address this issue, Kim et al. [27] proposed an automatic camera calibration method capable of estimating object depth and detecting occluded regions within large-scale surveillance systems. Similarly, Kong et al. [16] emphasized the importance of strategic camera placement, especially in applications involving convolutional neural networks, to mitigate occlusion effects on vehicle detection. Collectively, these studies underscore the necessity of careful camera positioning and angle selection to balance vehicle visibility with the minimization of visual interference caused by object movement.

Recent research has enhanced object detection and tracking for intelligent transportation systems (ITS), often using YOLOv8 as a baseline. You et al. [28] combined an improved YOLOv8 with ByteTrack, boosting precision and recall in heavy traffic by refining association strategies. Zhou et al. [29] proposed a lightweight YOLOv8 variant with EfficientViT and GhostConv, reducing parameters by over 30% while retaining mAP@0.5 around 75% on UA-DETRAC. Desta and Jian [30] applied advanced augmentation and tuning, achieving mAP above 80% on BDD100K with real-time performance. Guo et al. [31] introduced FasterNet and CBAM attention with WIoU loss, surpassing 90% in precision and recall for small and occluded vehicles. Bakirci [32] tested YOLOv8-Nano in drone videos, reporting over 80% precision with low latency for edge devices, while Mudawi et al. [33] combined YOLOv8 with deep belief networks to reach nearly 95% accuracy on aerial vehicle classification. Although these works show YOLOv8's versatility, they focus on urban or aerial views. In contrast, our study targets the underexplored **side-view highway monitoring of heavy vehicles**, using a DNIT-custom YOLOv8 detector and a modular detection–tracking–counting pipeline (BoT-SORT and ByteTrack) tailored for directional counting on Brazilian highways.

In summary, despite progress from early Kalman-based trackers [5], [18] to recent YOLOv8- and tracker-augmented ITS systems [28]–[33], most studies rely on urban or elevated camera views that minimize occlusion but lack side-profile details relevant to heavy-vehicle analysis. While top-down perspectives ease tracking, they obscure vehicle length and axle cues critical to DNIT classifications, whereas roadside side-views capture these features but face higher occlusion rates, foreground clutter, and detection dropouts [14], [17]. To address this gap, our work focuses on **side-view highway monitoring of heavy vehicles** in Brazilian conditions, leveraging a DNIT-aligned YOLOv8 detector, comparing BoT-SORT and ByteTrack, and adopting a centroid-based directional counting approach suitable for real deployments.

III. Methodology

The methodology employed in this study was structured into three main stages: data acquisition, vehicle detection and tracking, and the counting process. The primary objective was to compare the performance of the BoT-SORT and ByteTrack tracking algorithms integrated with the YOLOv8 detection model, using actual highway traffic data collected during the analyzed period. This data was obtained through manual peer counting to provide a reliable and less error-prone reference, serving as the ground truth for the number of vehicles observed during the evaluation period.

Overall Processing Pipeline

Figure 1 summarizes the proposed traffic monitoring pipeline. Incoming video frames from an IP side-view camera are processed by a YOLOv8 detector (either COCO pre-trained or DNIT-customized). For each frame, YOLO outputs bounding boxes, class labels, and confidence scores. These detections are forwarded to one of two multi-object trackers—BoT-SORT or ByteTrack—which associate detections across time and maintain unique track IDs.

Directional counting is performed downstream: whenever the centroid of an active track crosses a virtual counting line placed at mid-frame, the corresponding directional counter is incremented. Direction is derived from the sign of the centroid's horizontal displacement between consecutive frames.

This modular staging (detection $\rightarrow$ tracking $\rightarrow$ counting) clarifies that the system is not a hybrid network but a loosely coupled processing chain. This design also facilitates independent replacement or tuning of detection and tracking components without retraining the entire system.

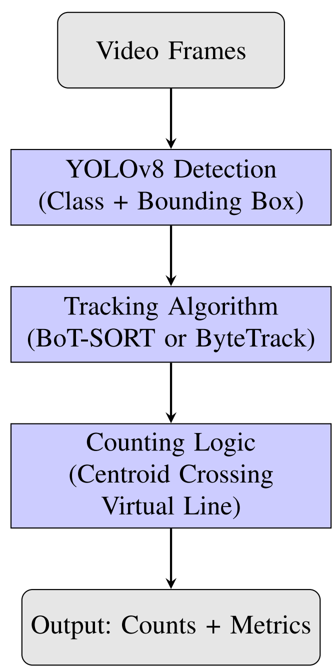

Figure 1. Overall processing pipeline: detection, tracking, and counting stages.

A. Data Acquisition

The images used for the analysis were captured on the BR-110 highway, located in the municipality of Olindina, in the state of Bahia, Brazil. The footage was recorded using a Jortan IP V380 camera mounted on a gas station structure at an approximate height of 3 meters, providing a lateral view of the roadway. The Jortan IP V380 is an IP-based surveillance camera that supports wireless connectivity and remote monitoring, offering a practical solution for outdoor video capture. The analyzed section of the highway consists of a single carriageway with two lanes, supporting traffic in both directions. The videos were recorded at 24 frames per second (FPS) with a resolution of 1280×720 pixels. The evaluated time period was on May 6th, 2024, from 6:00 a.m. to 12:15 p.m.

For this experimental phase, the video files were retrieved directly from the camera's internal memory card to ensure full-quality footage without potential interruptions or latency caused by network transmission. However, the long-term objective is to enable real-time streaming of video data over IP networks to centralized servers for processing, allowing scalable and continuous monitoring across multiple highway sites.

B. Vehicle Detection

For the detection task, the YOLO algorithm was employed due to its well-known capability to perform real-time object detection with high accuracy. Currently at version YOLOv11, the algorithm operates by dividing the input image into regions and simultaneously predicting object classes and bounding boxes in a single network pass. This design renders YOLO highly efficient for applications requiring fast processing, such as vehicle detection in traffic videos. The YOLOv8 family comprises several variants—YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x—differing in the number of layers, inference time, and parameter count. More recent iterations, including YOLOv11, introduce significant improvements in architecture, accuracy, and computational efficiency, establishing YOLO as a leading choice for modern automated road monitoring systems.

Among the YOLO family of models, and considering inference time on the hardware used (as shown in Figure 2), the YOLOv8l model was selected. In this study, two configurations of YOLOv8 were evaluated: (i) the default COCO pretrained model and (ii) a DNIT-customized model trained for heavy vehicle detection. The customized model was trained using a dataset containing approximately 12,000 images covering 11 of the 32 vehicle classes defined in the DNIT traffic manual. The dataset was split into training and validation subsets using a 90/10 ratio. All images were annotated following the YOLO format (class label and normalized bounding box coordinates).

To improve generalization, data augmentation was applied during training using the following parameters: brightness adjustment (`hsv_v`=0.4), image translation (`translate`=0.1), scaling (`scale`=0.5), and horizontal flipping (`fliplr`=0.5). Rotation, shear, perspective, and vertical flipping were disabled (`degrees`=0.0, `shear`=0.0, `perspective`=0.0, `flipud`=0.0). These augmentations aim to simulate variations in lighting and positioning commonly observed in roadside traffic scenes. Training was performed for 600 epochs. Table I reports the final metrics for both training and validation sets.

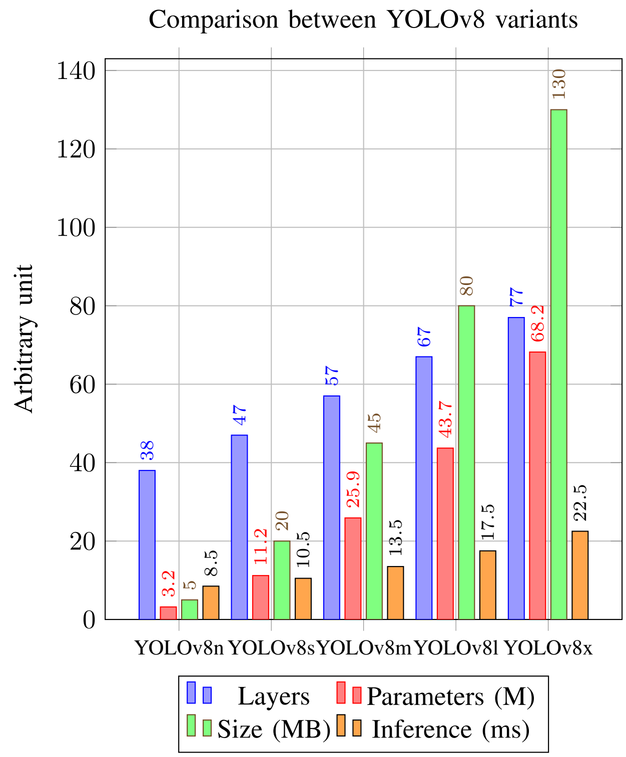

Figure 2. Comparison between YOLOv8 variants based on layers, parameters, size, and estimated inference time on RTX 2000 Ada Generation.

Table I
FINAL TRAINING AND VALIDATION METRICS FOR THE DNIT-CUSTOM YOLOV8 MODEL (EPOCH 600).

Metric	Train	Val
Box Loss	0.2035	0.2357
Cls Loss	0.1221	0.1556
DFL Loss	0.8104	0.8482
Precision	–	0.985
Recall	–	0.988
mAP@0.50	–	0.991
mAP@0.50:0.95	–	0.978

On the validation split, the DNIT-custom model achieved a precision of 0.985, recall of 0.988, mAP@0.50 of 0.991, and mAP@0.50:0.95 of 0.978, indicating strong detection capability for the classes included.

Although detection performance was high, comparative evaluation in this work focuses on directional counting rather than per-class metrics. This is because the custom model currently includes only 11 of the 32 DNIT classes, while the COCO-based model provides only coarse categories such as *truck* and *bus*.

The targeted classes are: Truck (2C), Tri-axle truck (3C), Tri-axle dual-directional truck (4CD), Bus (2CB), Tri-axle bus (3CB), Tractor truck with semi-trailer (2S3), Tri-axle tractor truck with semi-trailer (3S3), Tri-axle tractor truck with semi-trailer (3J3), Tri-axle tractor truck with semi-trailer (3J4), Tri-axle tractor truck with semi-trailer (3I3), and Articulated B-train (3D4).

C. Tracking Algorithms

The tracking algorithms evaluated in this work were ByteTrack and BoT-SORT. A key characteristic of ByteTrack is its use of both high- and low-confidence detections. Initially, detections are divided into two groups based on a confidence threshold: high confidence (e.g., > 0.6) and low confidence (e.g., between 0.1 and 0.6). The association between detections and existing tracks is then performed in two steps:

- In the first step, high-confidence detections are matched to active tracks using a strategy based on Intersection over Union (IoU) combined with the Hungarian algorithm [15]. The Hungarian algorithm is a classical combinatorial optimization method that efficiently solves assignment problems by minimizing the total cost of associations between two sets—in this case, detections and tracks.
- In the second step, unassociated tracks from the first step are reassigned using low-confidence detections, again employing the Hungarian algorithm.
- Finally, any remaining unmatched detections initiate new tracks.

Figure 3 illustrates this process through a layered diagram representing the ByteTrack workflow.

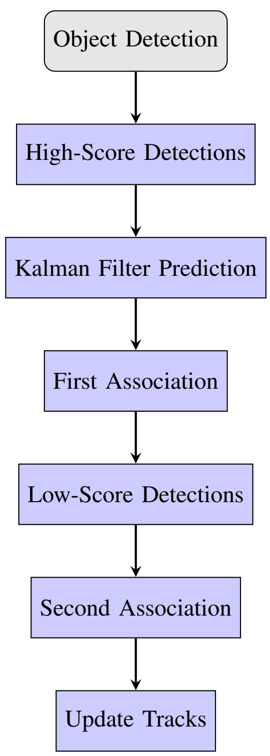

Figure 3. Layered diagram of ByteTrack execution

BoT-SORT, in contrast, employs a tracking mechanism based on Kalman Filter predictions [18], combined with associations performed via Intersection over Union (IoU). The Kalman Filter is a recursive algorithm designed to estimate the future state of an object based on previous measurements, even in the presence of noise, making it particularly effective for predicting object positions across successive video frames. Although this algorithm can be extended with re-identification (Re-ID) capabilities, in the present work it was utilized in its conventional form without additional Re-ID modules, as YOLO currently does not support this functionality. Consequently, the focus was placed exclusively on the accuracy of the tracked object count. Figure 4 illustrates the execution steps of BoT-SORT.

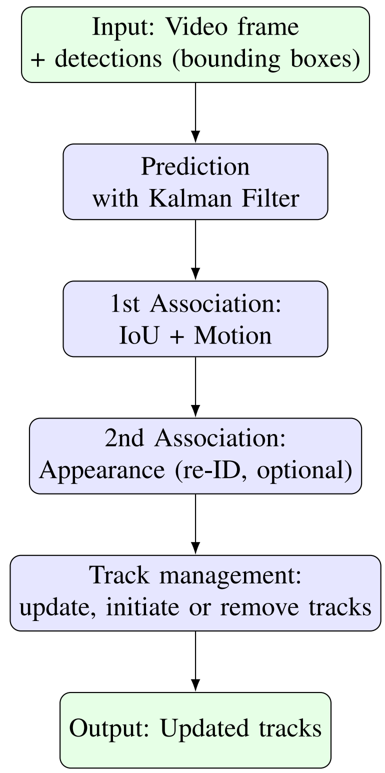

Figure 4. BoT-SORT algorithm operation diagram: Video processing, detection association and track management.

D. Counting

Vehicle counting was performed using the single virtual line technique, positioned at the horizontal center of the video frame. Each time the center of a vehicle's bounding box crossed this line, the counter was incremented, considering the direction of movement. The movement direction was determined by comparing the current X-coordinate of the object's center with its previous position:

- If the X-coordinate increased, the vehicle was moving from left to right.
- If the X-coordinate decreased, the vehicle was moving from right to left.

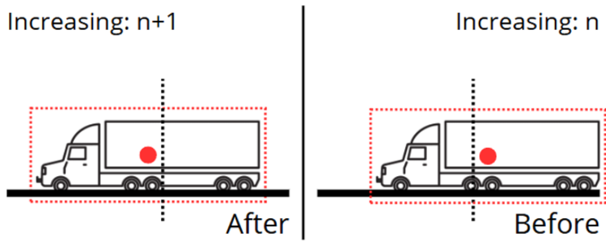

Figure 5. Camera position in relation to the highway

This approach was preferred over the double-line technique to minimize false negatives. In side-view contexts, temporary occlusions are frequent, and vehicles may cross only one of the virtual double-counting lines, leading to incorrect or incomplete counts. Therefore, the single-line method proved to be more suitable for this type of scenario. Figure 6 illustrates the single-line configuration in practice.

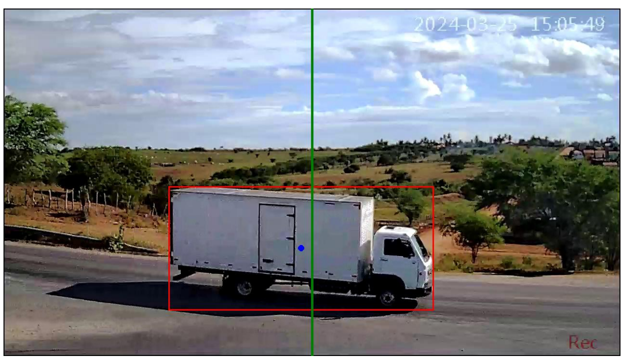

Figure 6. Example of the single-line application: the **counting line** is highlighted in green, the **detected object** (bounding box) is shown in red, and the **object centroid** (reference point for counting) appears in blue.

E. Evaluation Metrics

To assess counting performance, the following metrics were used, as proposed by the author of this study:

- **False positives (FP)**: vehicles that were incorrectly counted.

$$\text{FP} = \text{Total predictions} - TP \tag{1}$$

- **False negatives (FN)**: vehicles present in the scene that were not counted.

$$\text{FN} = \text{Total ground truths} - TP \tag{2}$$

Where TP represents **true positives**, i.e., correctly counted vehicles.

For a more detailed analysis of false positives and false negatives observed in the count, the **precision** and **recall** metrics were also considered, as they provide a deeper understanding of the trade-off between correct and incorrect detections.

Precision evaluates the proportion of correct detections (true positives) among all detections made by the model:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Recall measures the proportion of actual vehicles that were correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

These metrics enable the analysis of algorithmic performance from different perspectives:

- A model with high precision and low recall is conservative, detecting only when it is highly certain, but may fail to detect many vehicles.
- A model with high recall and low precision detects nearly all vehicles present but is prone to making numerous incorrect detections.

System performance is evaluated using event-based counting metrics derived from tracked object line-crossings, as these provide a realistic measure of traffic monitoring accuracy. Frame-level accuracy, given by $(\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$, is not meaningful in this context because the overwhelming number of background pixels inflates true negatives. Similarly, a simple relative counting error $|\text{Pred} - \text{GT}|/\text{GT}$ can be misleading, as sporadic false positives disproportionately affect the result and obscure tracker stability over time and across directions. For this reason, the evaluation relies on precision, recall, and F1-score computed from event-level TP, FP, and FN counts, which provide a more robust and interpretable assessment of the performance of the ByteTrack and BoT-SORT algorithms in the context of side-view traffic counting.

IV. Results

It is important to note that a per-class evaluation was not feasible in this study because the pre-trained YOLOv8 model used here relies on COCO classes, which only include generic categories such as *truck* and *bus*. These do not map directly to the detailed DNIT taxonomy used in manual traffic studies. Consequently, our analysis focuses on directional counting accuracy and overall detection/tracking performance. Ground truth values for heavy-vehicle flows were obtained through manual counting by trained annotators following DNIT guidelines.

After performing the counts for each of the tracking algorithms applied to the different detection models, the following total detections by direction were obtained:

Table II
Total Detections by Direction

Algorithm	Model	Increasing	Decreasing
BoT-SORT	Customized	180	137
BoT-SORT	Pre-trained	283	238
ByteTrack	Pre-trained	264	231
ByteTrack	Customized	179	132

Based on the number of detections for each case in Table II and considering the ground truth values for the evaluated period — 224 vehicles in the leftward direction and 164 in the rightward direction — the values of True Positives (TP), False Positives (FP), and False Negatives (FN) were calculated:

Table III
METRICS BY DIRECTION

Algorithm & Model	Direction	TP	FP	FN
BoT-SORT Custom	Increasing	157	23	67
BoT-SORT Custom	Decreasing	122	15	42
BoT-SORT Pre-trained	Increasing	162	121	62
BoT-SORT Pre-trained	Decreasing	122	116	42
ByteTrack Custom	Increasing	158	20	66
ByteTrack Custom	Decreasing	121	7	43
ByteTrack Pre-trained	Increasing	159	105	65
ByteTrack Pre-trained	Decreasing	125	106	39

Analyzing Table III, the TP and FN values for each direction tend to show no significant variation. For a more detailed analysis, the performance metrics Precision, Recall, and F1 Score were calculated:

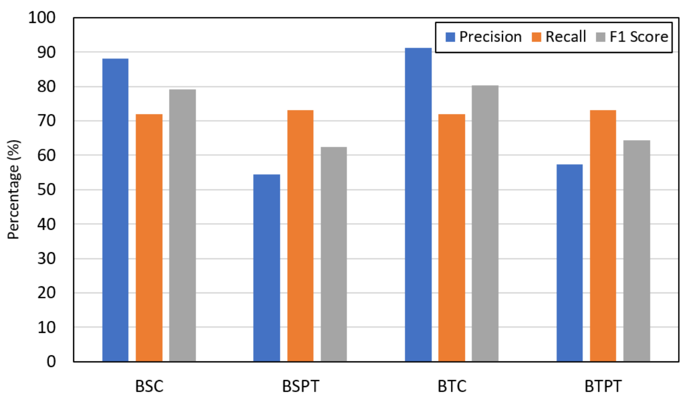

Figure 7. Comparison of Metrics by Algorithm and Model (Combined Directions). BSC = BoT-SORT Custom, BTC = ByteTrack Custom, BSPT = BoT-SORT Pre-trained, BTPT = ByteTrack Pre-trained

As shown in Figure 7, Two consistent trends emerge from the aggregated results. First, the DNIT-custom YOLOv8 detector substantially reduces false positives compared to the COCO pre-trained model: FP counts decrease from over 200 (COCO) to fewer than 40 (DNIT) under BoT-SORT, and from 211 to 27 under ByteTrack. This improvement drives a large precision gain—from approximately 55–57% to 88–91%—while recall remains stable near 72–73%. Second, comparing trackers under the same detector shows that ByteTrack achieves slightly higher precision than BoT-SORT (91.18% vs. 88.01% with DNIT; 57.37% vs. 54.51% with COCO), though recall is identical within rounding. Consequently, ByteTrack attains the best F1-score overall (80.4% with DNIT), but detector quality dominates performance differences in this experiment.

False positives are concentrated in a few error modes rather than distributed uniformly across time, which explains their strong influence on precision and why aggregate error metrics like relative error can misrepresent tracker behavior. Most spurious events occurred near the fuel station forecourt, where vehicles slow down and centroid jitter or partial detections lead to duplicate track IDs.

V. CONCLUSION

This paper evaluated a modular traffic monitoring pipeline for side-view highway video, combining YOLOv8-based detection with downstream multi-object tracking (BoT-SORT and ByteTrack) and a centroid-triggered directional counting strategy aligned with DNIT traffic engineering practices. Using footage collected on the BR-110 in Olindina (Bahia, Brazil), the proposed approach produced reliable directional heavy-vehicle counts under realistic roadside conditions.

The DNIT-custom YOLOv8 detector was trained on ∼12,000 annotated images from 11 of the 32 DNIT vehicle classes, achieving strong validation metrics (precision 0.985, recall 0.988, mAP@0.50 0.991, mAP@0.50:0.95 0.978). Per-class field evaluation was not performed because the custom model currently covers only a subset of DNIT classes, while the COCO pre-trained model provides only coarse heavy-vehicle labels (*truck*, *bus*). Thus, our field results focus on *directional count accuracy* rather than detailed class-level comparisons.

Ground truth volumes were obtained by *manual counting* following DNIT procedures over the 6 h 15 min observation period (06:00–12:15, 6 May 2024), enabling direct comparison with system-generated counts for both traffic directions.

The videos analyzed here were exported directly from the camera's on-board memory card in order to preserve full-quality footage and avoid connectivity artifacts during the experimental phase. In operational deployments, however, our objective is to stream video over IP networks to centralized processing servers, enabling continuous and scalable monitoring across distributed highway sites.

Practical deployment considerations: Centralized processing reduces maintenance overhead at roadside locations, simplifies uniform model updates, and allows resource sharing (GPU acceleration) across many cameras. End-to-end responsiveness depends on video encoding latency, available uplink bandwidth, and server throughput. Preliminary profiling suggests that near-real-time operation is achievable under moderate network conditions; a detailed scalability study remains future work.

Future work will (i) expand the DNIT dataset to include *all 32 vehicle classes* defined in the DNIT traffic manual, enabling full-spectrum traffic monitoring (heavy and light vehicles); (ii) retrain and benchmark the detector on the expanded taxonomy; (iii) perform large-scale runtime and bandwidth experiments under true network streaming conditions; and (iv) extend evaluation to additional roadway geometries, lighting and weather conditions, and multi-camera fusion scenarios.

ACKNOWLEDGEMENTS

The authors would like to thank the technicians and scholarship holders of the Paving Laboratory of the Federal University of Recôncavo da Bahia (PaveLab-UFRB) and the technical and financial support of the Institute of Transport Research (IPR) of DNIT through the research project linked to the Decentralized Execution Term (TED) No. 507/2023, in partnership with the UFRB. We also thank the FAPESB and the PIBIC/UFRB program for granting the scientific initiation scholarship, which made this research possible.

REFERENCES

[1] AASHTO, “Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Structures,” NCHRP Project 1-37A, Final Report, Transportation Research Board, 2004.

[2] Departamento Nacional de Infraestrutura de Transportes (DNIT), *Manual de Estudos de Tráfego*. Publicação IPR-723. Rio de Janeiro: DNIT, 2006. Disponível em: https://www.gov.br/dnit/pt-br/assuntos/planejamento-e-pesquisa/ipr/coletanea-de-manuais/vigentes/723_manual_estudos_trafego.pdf. Acesso em: 24 maio 2025.

[3] X. Hu, T. Tang, Y. Wang, and R. M. Haralick, “SINet: A Scale-insensitive Convolutional Neural Network for Fast Vehicle Detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3235–3246, 2019. DOI: 10.1109/TITS.2018.2879977.

[4] X. Xu, F. He, and S. Li, “Vehicle Detection Based on Improved Multitask Cascaded Convolutional Neural Network and Mixed Image Enhancement,” *IET Image Processing*, vol. 15, no. 7, pp. 1505–1515, 2021. DOI: 10.1049/iet-ipr.2020.1005.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 3464–3468, 2016.

[6] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 3645–3649, 2017.

[7] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” arXiv preprint arXiv:1603.00831, 2016.

[8] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, 2016.

[9] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, et al., “ByteTrack: Multi-object tracking by associating every detection box,” in *European Conference on Computer Vision*, pp. 1–21, 2022.

[10] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast online object tracking and segmentation: A unifying approach,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1328–1338, 2019.

[11] J. Luiten, A. Osep, P. Dendorfer, et al., “MOTChallenge: A benchmark for multi-object tracking,” arXiv preprint arXiv:2104.10829, 2021.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[13] J. A. Bewley, Z. Tian, and F. Zheng, “BoT-SORT: Robust Associations Multi-Pedestrian Tracker,” arXiv preprint arXiv:2206.14651, 2022.

[14] K. Kim, D. Lee and I. Essa, "Vehicle Detection and Tracking with Low-Angle Cameras," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 273–278. doi:10.1109/AVSS.2007.4425306.

[15] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[16] A. C. K. Kusuma, M. R. M. Said and M. Z. Rehman, "Camera-Based System for the Automatic Detection of Vehicle Axle Count and Speed Using Convolutional Neural Networks," *Journal of Infrastructure Systems*, vol. 29, no. 1, 2023. doi:10.1007/s13177-022-00325-1.

[17] C. Premebida, J. Carreira, J. Batista and U. Nunes, "Vehicle Detection with Occlusion Handling, Tracking, and OC-SVM Classification: A High Performance Vision-Based System," *Sensors*, vol. 18, no. 7, 2018, p. 2049. doi:10.3390/s18072049.

[18] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[19] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, "An Improved YOLOv2 for Vehicle Detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018. doi:10.3390/s18124272.

[20] Y. Wang, Y. Guan, H. Liu, L. Jin, X. Li, B. Guo, and Z. Zhang, "VV-YOLO: A Vehicle View Object Detection Model Based on Improved YOLOv4," *Sensors*, vol. 23, no. 7, p. 3385, 2023. doi:10.3390/s23073385.

[21] J. Liao, S. Jiang, M. Chen, and C. Sun, "SAM-YOLO: An Improved Small Object Detection Model for Vehicle Detection," *Journal of Intelligent Transportation Systems*, 2025. doi:10.1177/30504554251319452.

[22] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," *arXiv preprint arXiv:2410.22898*, 2024.

[23] E. Baş, A. M. Tekalp, and F. S. Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, Istanbul, Turkey, 2007, pp. 392–397.

[24] W. Balid, H. Tafish, and H. H. Refai, "Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3239–3249, 2017. doi:10.1109/TITS.2017.2741507.

[25] J.-P. Lin and M.-T. Sun, "A YOLO-Based Traffic Counting System," in *Proc. 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Taoyuan, Taiwan, 2018, pp. 82–87. doi:10.1109/TAAI.2018.00027.

[26] N. Kanhere, S. Pundlik, S. Birchfield, and W. Sarasua, "Vehicle Segmentation and Tracking in the Presence of Occlusions," Clemson University, 2008. [Online]. Available: https://cecas.clemson.edu/~stb/research/vehicle_tracking/

[27] S. Kim, H. Kim, and J. Paik, "Object Occlusion Detection Using Automatic Camera Calibration for a Wide-Area Video Surveillance System," *Sensors*, vol. 16, no. 7, p. 982, 2016. doi:10.3390/s16070982.

[28] Y. You, L. Wang, Z. Zhang, and H. Chen, “Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack,” *Electronics*, vol. 13, no. 15, p. 3033, 2024. DOI: 10.3390/electronics13153033.

[29] Y. Zhou, J. Zhao, and Q. Hu, “Based on Improved Lightweight YOLOv8 for Vehicle Detection,” *Advances in Computer Modeling and Simulation Research*, vol. 2, no. 1, pp. 72–82, 2024. Available: https://www.angus-publishing.com/index.php/acmsr/article/view/72.

[30] T. M. Desta and X. Jian, “Enhancing YOLOv8 for Vehicle Detection in Intelligent Traffic Management,” *Metallurgical and Materials Engineering*, vol. 31, no. 1, pp. 1423–1435, 2025. Available: https://www.metall-mater-eng.com/index.php/home/article/view/1423.

[31] C. Guo, R. Chen, and L. Sun, “Research on Vehicle Detection Based on Improved YOLOv8 Network,” *arXiv preprint arXiv:2501.00300*, 2024. Available: https://arxiv.org/abs/2501.00300.

[32] K. Bakirci, “Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems,” *Traitement du Signal*, vol. 41, no. 4, pp. 653–662, 2024. DOI: 10.18280/ts.410407.

[33] A. M. Mudawi, S. A. Osman, and Y. E. Osman, “Vehicle Detection and Classification via Y