Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.11 n.53, p. 193-200. May/June., 2025.
DOI: https://doi.org/10.5935/jetia. v11i53.1898

RESEARCH ARTICLE OPEN ACCESS

AI-POWERED SIMULTANEOUS MULTI-VEHICLE SPEED ESTIMATION FOR INTELLIGENT TRAFFIC MONITORING IN DEVELOPING REGIONS USING YOLOV7 AND DEEPSORT

Merrad Ahmed[1], Daoud Walid[2], Dalouli Aissa[3], Latrech Boubakeur[4] and Nouri Abdelkader Nabil[5]

[1,2,3,4,5] Laboratory of Computer Science & Applied Artificial Intelligence, Faculty of Exact Sciences & Computer Science, University of Djelfa PO Box 3117, Djelfa 17000, Algeria.

[1]https://orcid.org/0009-0003-4084-4291, [2]https://orcid.org/0009-0005-9609-6645, [3]https://orcid.org/0009-0005-7659-0769
[4]http://orcid.org/ 0009-0007-9367-3368, [5]https://orcid.org/0000-0003-1439-0209

Email: ahmed.merrad@univ-djelfa.dz , waliddaoud230@gmail.com , Dellouliaissa@gmail.com , b.latreche@univ-djelfa.dz, nabil.nouri@gmail.com

ARTICLE INFO

Article History
Received: May 19, 2025
Revised: June 20, 2025
Accepted: June 25, 2025
Published: June 30, 2025

Keywords:
Speed Estimation,
Vehicle Tracking,
YOLO,
DeepSORT,
Traffic Monitoring.

ABSTRACT

The development of intelligent, cost-effective vehicle speed monitoring systems is critical for enhancing road safety, improving traffic regulation, and enabling efficient law enforcement particularly in developing regions with limited infrastructure. This paper introduces a robust, AI-based framework for real-time speed estimation of multiple vehicles from monocular video streams. The proposed system integrates two advanced deep learning models -YOLOv7 for high-precision vehicle detection and DeepSORT for consistent multi-object tracking- ensuring accurate localization and identity preservation across frames. Speed estimation is performed by measuring the time it takes for each vehicle to travel a predefined distance between two virtual reference lines. The elapsed time is calculated based on frame count, and speed is derived using the basic motion formula. Experimental results show that the system achieves 100% detection and tracking accuracy, with an average speed estimation error of less than 3%, outperforming comparable methods in terms of efficiency and precision. The study also identifies and discusses key factors affecting estimation accuracy, such as frame rate variation, distance measurement error, and line placement precision. The approach's simplicity, accuracy, and use of open-source tools make it well-suited for deployment in resource-constrained environments. Future directions include bidirectional speed tracking, integration with vehicle classification systems, and the use of license plate dimensions for dynamic calibration—offering a scalable foundation for intelligent traffic surveillance.

I. INTRODUCTION

Efficient vehicular transportation forms the backbone of economic development and daily urban mobility, providing individuals with rapid access to services and facilitating the seamless movement of goods [1]. However, the rising volume of road traffic has led to a corresponding increase in traffic violations and road accidents [2], particularly in developing regions where traffic law enforcement remains inconsistent and infrastructure is often underdeveloped [3].

Speeding, in particular, is recognized as a primary contributor to severe road incidents, frequently resulting in injury or loss of life [4],[5].

Conventional methods of traffic monitoring, such as speed radars [6],[7], inductive loops [8],[9], and closed-circuit television (CCTV) [10],[11], offer limited scalability and often require costly installation and maintenance [12],[13]. Moreover, these systems typically struggle in scenarios involving multiple vehicles, varying lighting or weather conditions [14], and dynamic environments—challenges that are especially pronounced in resource-constrained settings [15]. As urbanization accelerates and vehicle density increases, there is a critical need for intelligent [16], automated solutions capable of real-time, multi-vehicle speed estimation that are both cost-effective and infrastructure-light [17].

Recent advancements in artificial intelligence (AI) and deep learning have enabled significant progress in computer vision-based traffic monitoring [18–20]. Object detection algorithms, particularly the YOLO (You Only Look Once) family [21–23], have demonstrated exceptional speed and accuracy in detecting vehicles in real-time video streams [24]. YOLOv12 [25], the latest iteration, builds on this foundation by enhancing feature extraction and localization capabilities, making it highly suitable for real-time vehicular detection in complex traffic scenes [26].

Equally important is the task of multi-object tracking, which ensures persistent identification and localization of vehicles across video frames. DeepSORT [27],[28], an extension of the Simple Online and Real-time Tracking (SORT) framework, integrates motion modeling via Kalman filtering with deep appearance features extracted using convolutional neural networks (CNNs) [29], enabling accurate and continuous tracking even under occlusions or abrupt motion changes [30]. The combination of YOLO-based detection and DeepSORT-based tracking offers a robust framework for real-time, multi-vehicle tracking under diverse conditions [31],[32].

Despite the promise of these technologies, current literature reveals several limitations: many solutions assume ideal conditions and perform poorly in scenarios with partial occlusions or low visibility [33],[34]; speed estimation accuracy often degrades with multiple targets due to tracking ambiguity [35]; and most existing systems are designed for deployment in developed regions and rely on expensive sensors such as radar or LiDAR [36], rendering them impractical in low-resource settings [37].

To address these challenges, this paper proposes an AI-powered framework that leverages YOLOv7 for real-time vehicle detection and DeepSORT for robust multi-object tracking, enabling simultaneous estimation of vehicle speeds in live video streams. The proposed system is designed with an emphasis on affordability, scalability, and adaptability to real-world constraints commonly encountered in developing regions—such as inconsistent lighting, suboptimal camera placements, and the absence of specialized hardware.

Moreover, the study examines the root causes of deviation between estimated and actual vehicle speeds, incorporating environmental, computational, and tracking factors into the analysis. These insights are critical for refining system performance and enhancing its applicability across diverse operational contexts.

The principal contributions of this work are as follows:

• A novel, low-cost framework for simultaneous detection, tracking, and speed estimation of multiple vehicles using only video feeds.

• Integration of YOLOv7 and DeepSORT, tailored to the constraints of developing regions, without reliance on additional sensors.

• Comprehensive evaluation and error analysis, highlighting factors that impact speed estimation accuracy under real-world conditions.

The remainder of the paper is organized as follows: Section 2 reviews relevant literature on video-based vehicle speed estimation and object tracking. Section 3 details the proposed methodology, including system architecture, detection, tracking, and speed computation techniques. Section 4 presents experimental results, performance evaluation, and error analysis. Finally, Section 5 concludes the study and outlines future research directions.

II. RELATED WORKS

Vehicle speed estimation has been the focus of extensive research in the domains of computer vision and intelligent transportation systems [38], particularly with the rise of AI-driven approaches [39],[40]. Numerous techniques have emerged, leveraging deep learning [41], geometric modeling [42], and image processing to enable real-time, multi-object vehicle tracking and speed measurement [43]. In this section, we review relevant literature, grouped by methodological themes, and highlight their contributions and limitations to position the novelty of our proposed approach.

Recent developments in object detection, particularly the YOLO (You Only Look Once) architecture and its derivatives, have significantly improved the accuracy and efficiency of vehicle detection in dynamic scenes [44]. When integrated with tracking algorithms such as DeepSORT, these methods allow for consistent multi-vehicle tracking in real-time video streams [45].

Nguyen et al. [46] employed YOLOv4 and DeepSORT for vehicle detection and tracking on highways, transforming pixel-based displacements into real-world measurements via static reference landmarks. To mitigate tracking errors caused by frame loss, a recycling mechanism was introduced, improving computational efficiency while maintaining multi-object speed estimation accuracy. Similarly, Luo et al. [47] designed a speed estimation system using YOLOv5s and DeepSORT, augmented by a Swin Transformer block to enhance performance under challenging surveillance conditions, eliminating the need for camera calibration.

In a related study, Cvijetic et al. [48] proposed an approach combining YOLO-based vehicle detection with a 1D convolutional neural network (1D-CNN) to estimate vehicle speeds by analyzing the changing bounding box area (CBBA) across frames. This method circumvents the need for explicit scene calibration, thereby offering flexibility for deployment in heterogeneous environments.

Several studies have adopted geometric reasoning to translate image-based observations into accurate speed estimations. Vahid Dastgerdi Vahid et al. [49] introduced a zone-based speed estimation method leveraging YOLOv8 for detection and projective geometry for road scene analysis. The approach defines speed computation zones using detected traffic cones as spatial references, enabling localized speed estimation with enhanced interpretability.

Kisingo et al. [50] addressed the challenges of multi-lane speed estimation by integrating inter-frame image processing with Mahalanobis distance-based matching. Their system detects speeding vehicles for traffic enforcement, yet relies heavily on consistent camera viewpoints and well-calibrated lanes, which limits generalization in unstructured settings.

In contrast to end-to-end deep learning approaches, some methods employ handcrafted features and traditional vision techniques for speed estimation. Bhatlawande et al. [51] presented a cost-efficient strategy based on a monocular camera system using FAST and FREAK feature descriptors, coupled with a voting-based classifier. While computationally lightweight, this method may struggle with occlusions and complex traffic environments.

Tayeb et al. [52] utilized Gaussian Mixture Models (GMM) for foreground segmentation and Kalman filtering for tracking, incorporating perspective transformation to estimate real-world speeds. Their framework emphasizes adaptability to dynamic lighting and cluttered backgrounds, yet remains dependent on accurate geometric modeling of the scene.

Alternative strategies leverage indirect cues such as shadows or headlights to infer motion. Lu et al. [53] bypassed conventional feature extraction by estimating speed based on vehicle shadow motion via projection histograms. While innovative, this approach is sensitive to lighting conditions and shadow clarity. For nighttime surveillance, Kim [54] developed a speed estimation method based on headlight detection using a moving mean algorithm, providing a lightweight solution for low-visibility environments but offering limited scalability.

Kumar et al. [55] tackled the issue of false positives by refining the region of interest (ROI) using a cropping mechanism optimized for pole-mounted cameras. While effective in reducing occlusions, the system's dependence on fixed-camera placement constrains its applicability in diverse traffic settings.

Several studies explored vehicle speed estimation in embedded or mobile environments, addressing challenges associated with platform motion and scene variability. Garcia-Aguilar et al. [56] proposed an onboard system using CNNs for detecting and tracking surrounding vehicles, followed by speed estimation via regression. Their system operates solely on monocular images without requiring LIDAR, making it a cost-effective solution for driver assistance systems.

In aerial surveillance contexts, Chen et al. [57] combined YOLO detection with depth features to re-identify vehicles across frames and applied exponential mapping to adjust for UAV altitude variations. The approach is particularly suited for dynamic aerial monitoring, yet computational demands and re-identification challenges limit its performance in congested scenes.

Although not directly related to road-based vehicle monitoring, several studies in adjacent domains highlight the versatility of speed estimation techniques. For instance, Tedesco et al. [58] utilized machine learning with inertial sensors to estimate human gait speed for clinical assessments. In military contexts, Biswas et al. [59] developed a UAV-based multi-object speed estimation framework using aerial imagery, while Chmielewski et al. [60] designed an optical ground speed estimator for miniature UAVs. Kamnardsiri et al. [61] extended vision-based tracking to sports, proposing a system for monitoring sprint performance during 100-meter races.

While substantial progress has been made in vehicle detection, tracking, and speed estimation, key limitations persist. Many state-of-the-art solutions assume ideal environmental conditions or require costly hardware (e.g., LIDAR, radar) [36]. Others lack scalability in multi-vehicle scenarios or perform poorly in complex, resource-constrained urban settings typical of developing regions [62]. Furthermore, the impact of environmental variables (e.g., lighting, camera angles) and tracking errors on speed estimation accuracy is often overlooked [38].

To bridge these gaps, the current study proposes a unified, real-time framework that combines the detection capabilities of YOLOv7 with the robust tracking of DeepSORT to estimate the speed of multiple vehicles from standard video feeds. Unlike prior methods, our approach is specifically optimized for low-cost deployment in developing regions, requiring no specialized sensors or calibration. Additionally, a comprehensive error analysis is conducted to assess and interpret the deviation between estimated and actual speeds under real-world conditions—an area that remains underexplored in the literature.

III. MATERIALS AND METHODS

The proposed approach for vehicle speed estimation is structured into three principal stages: (i) vehicle detection, (ii) multi-object tracking, and (iii) speed estimation through temporal frame analysis. The system is designed to be computationally efficient and hardware-agnostic, requiring only a low-cost monocular camera -such as a smartphone- and a basic processing unit. This design ensures accessibility and scalability in real-world applications, particularly within resource-constrained environments. An overview of the system architecture is presented in Figure 1.

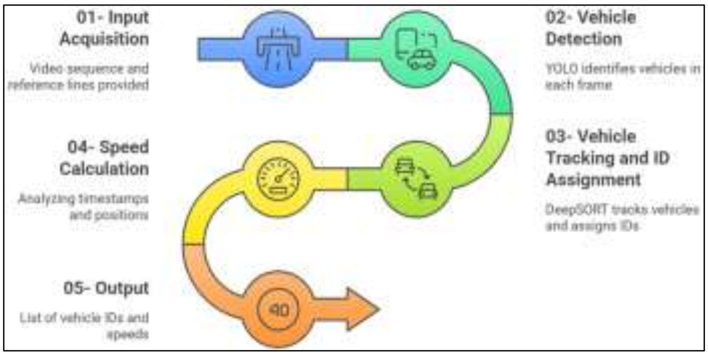

Figure 1: Flowchart of the Proposed Vehicle Speed Estimation System Using YOLOv7 and DeepSORT.
Source: Authors, (2025).

III.1 VEHICLE DETECTION

The initial phase focuses on robust and real-time vehicle detection. For this purpose, YOLOv7 (https://github.com/WongKinYiu/yolov7) a state-of-the-art convolutional neural network for object detection—is employed [43]. Trained on the COCO dataset, YOLOv7 can identify 80 object classes, including various types of vehicles (e.g., cars, buses, and trucks). Its real-time detection capabilities and high localization accuracy make it particularly suited for dynamic traffic environments.

Once a vehicle is detected, YOLOv7 assigns a bounding box based on class confidence scores (see Figure 2). This bounding box delineates the spatial coordinates of the vehicle in the image frame and serves as the input for the subsequent tracking stage.Analyzing and interpreting the results.

Figure 2: Real-time Vehicle Detection via YOLOv7.
Source: Authors, (2025).

III.2 VEHICLE TRACKING

Following detection, the system transitions into tracking the motion of the identified vehicles across successive video frames. The DeepSORT algorithm (https://github.com/nwojke/deep_sort) [28] is integrated for this purpose due to its ability to maintain object identities over time, even in the presence of partial occlusion or momentary disappearance from the camera view.

DeepSORT leverages both motion (via Kalman filtering) and appearance descriptors (via a deep association metric) to assign a unique ID to each tracked vehicle, thereby ensuring temporal consistency and robustness across multiple frames. The algorithm supports concurrent tracking of several vehicles, enabling scalable multi-object monitoring. A visual example of DeepSORT tracking is illustrated in Figure 3.

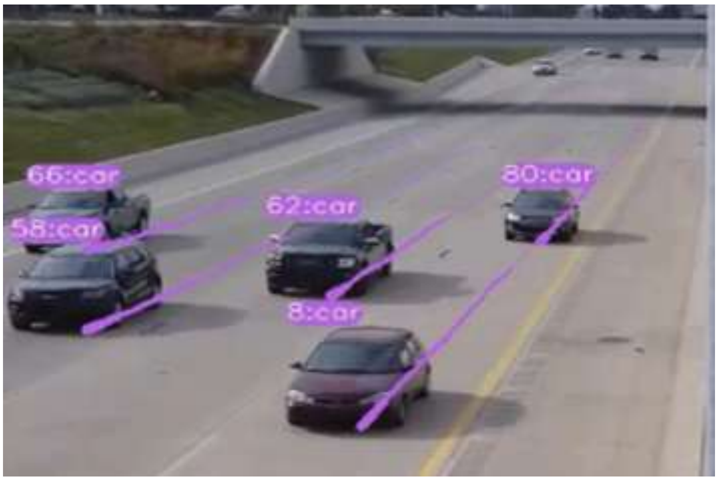

Figure 3: Real-time Multi-Vehicle Tracking Using DeepSORT.
Source: Authors, (2025).

III.3 SPEED ESTIMATION VIA FRAME-BASED TEMPORAL ANALYSIS

The final stage involves the quantitative estimation of vehicle speed using a frame-counting strategy between two predefined spatial landmarks: a starting line (L1) and an ending line (L2) (refer to Figure 4). This method does not require any intrinsic camera calibration or geometric transformation, simplifying deployment in uncontrolled outdoor environments.

The estimation process is performed as follows:

a) Frame Counting: Speed measurement is initiated when the vehicle's bounding box intersects the starting line ($L1$) and terminates when the same bounding box crosses the ending line ($L2$). The total number of frames (denoted as $nbrF$) required for this traversal is recorded.

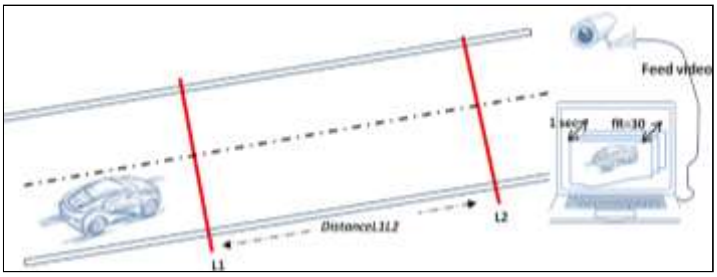

Figure 4: Visualization of Frame-Based Speed Estimation Using L1 and L2 as Reference Points.
Source: Authors, (2025).

b) Time Calculation: Once the frame count is found, the time is calculated by dividing the number of frames ($nbrF$) by the frame rate (fR), as specified in Equation (1). This ensures that the time measurement ($Time$) is accurate and truly reflects the vehicle's velocity.

$$Time = \frac{nbrF}{fR}\ (second) \quad (1)$$

c) Speed Calculation: The actual speed ($Speed_{m/s}$) is derived by dividing the known physical distance ($Distance(L1 - L2)$) between $L1$ and $L2$ by the computed time ($Time$), as shown in Equation (2).

$$Speed_{m/s} = \frac{Distance(L1 - L2)}{Time} \quad (2)$$

d) Unit Conversion: For practical interpretation, the speed is converted to kilometers per hour using the standard conversion factor ($1\ m/s = 3.6\ km/h$), as expressed in Equation (3)

$$Speed_{km/h} = Speed_{m/s} \times 3.6 \quad (3)$$

This lightweight yet accurate estimation framework makes the system suitable for real-time deployment in smart city infrastructures, especially in regions where expensive sensing technologies such as radar, LIDAR, or stereo cameras are impractical.

IV. RESULTS AND DISCUSSIONS

It This section presents a comprehensive evaluation of the proposed vehicle detection, tracking, and speed estimation system based on YOLOv7 and DeepSORT. The primary aim is to assess the system's effectiveness in real-world scenarios and its potential applicability in intelligent traffic monitoring, especially in developing regions where infrastructure may be limited and cost-efficiency is paramount.

To evaluate the performance of the proposed method, two video sequences were recorded in Djelfa, Algeria, using distinct vehicle types under varying speed conditions. To further demonstrate generalizability, additional publicly available video sequences depicting multi-vehicle traffic flows were included. All experiments were executed on the Google Colab platform [63], which provides a cloud-based Python environment equipped with essential libraries (e.g., NumPy [64], OpenCV [65], Matplotlib [66]) and optional GPU acceleration, facilitating efficient prototyping and rapid deployment.

The following subsections discuss the detection and tracking performance, evaluate the accuracy of the speed estimation algorithm, and provide a critical analysis of the factors influencing system performance.

IV.1 VEHICLE DETECTION AND TRACKING PERFORMANCE

Accurate vehicle detection and tracking are foundational to reliable speed estimation in video-based traffic monitoring systems. This study employs YOLOv7 for real-time object detection and DeepSORT for multi-object tracking, capitalizing on their respective strengths in recognition accuracy and temporal coherence.

Table 1 provides statistics on moving vehicles across several frames, showing both the real number of vehicles and the number detected and tracked. The results make obvious the effectiveness of the YOLOv7-DeepSORT combination in successfully detecting and tracking moving vehicles, even in complex scenes with multiple vehicles traveling at diverse speeds and directions. Furthermore, Figure 5 illustrates another frame, showcasing the true detection and tracking of all vehicles within the region of interest. These findings prove that our approach is highly efficient and reliable.

Compared to the prior approach by Kumar et al. [55], which reported an overall detection accuracy of 87.7%, the current system significantly improves detection robustness under real-world conditions, including occlusion, varying lighting, and diverse traffic behaviors.

Table 1: Detection and tracking results using the YOLOv7–DeepSORT framework.

Frame Number	Actual Vehicle Count	Detected and Tracked	Accuracy
121	12	12	**100%**
151	10	10	**100%**
170	6	6	**100%**
260	5	5	**100%**

Source: Authors, (2025).

Figure 5: Multi-Vehicle Detection and Tracking Using YOLO and DeepSORT.
Source: Authors, (2025).

IV.2 SPEED ESTIMATION RESULTS

Following the successful detection and tracking of road vehicles, this section evaluates the performance of the proposed system in estimating vehicular speeds under varying real-world conditions. The assessment was conducted using two different vehicle types—a DFSK minivan and a Toyota Yaris sedan—operating at distinct speed ranges. The objective was to examine the precision of the proposed speed estimation methodology by comparing estimated speeds with actual values reported by the onboard vehicle computer systems.

Table 2: Comparison of Real and Estimated Vehicle Speeds.

Video	Type vehicle	Actual speed	Estimated speed	Error
1	Vehicle 1 (DFSK)	60 km/h	62 km/h	3.33 %
2	Vehicle 2 (Yaris)	80km/h	79km/h	1.25 %

Source: Authors, (2025).

Two experiments were conducted using two different cars at varying speeds, as illustrated in Table 2. The results demonstrate the robustness of the proposed YOLOv7–DeepSORT-based framework for vehicle speed estimation. The observed error rates were consistently low, with a maximum absolute error of 3.33% and an average error of 2.29%. These findings highlight the effectiveness of the system in practical scenarios where high reliability and real-time performance are critical.

A comparative analysis with recent literature further underscores the merits of the proposed method. Nguyen et al. [46] reported an average error of 4.16% in their system, while Kisingo et al. [50] achieved a slightly better rate of 2.7%. In contrast, our approach attains comparable or improved accuracy despite using a straightforward, computationally efficient mechanism based on frame-counting and known physical distances.

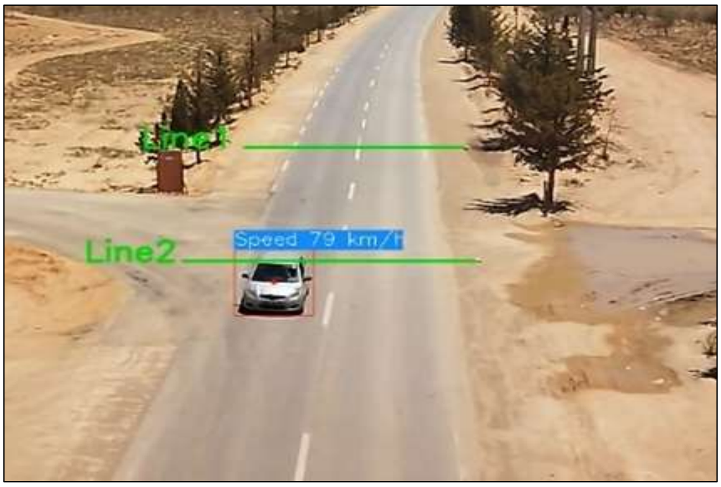

Figure 6: Detection, tracking and speed estimation of vehicle 1.
Source: Authors, (2025).

As illustrated in Figure 6, the frame captured from the video stream is structured into three key components that facilitate the speed estimation process. The first component consists of two predefined reference lines, which are strategically positioned to determine the number of frames a vehicle requires to traverse a known physical distance (30 meters in this study).

This frame count is then used to calculate the vehicle's speed based on the elapsed time and known distance. The second component represents the vehicle of interest—i.e., the object currently under analysis—whose movement between the two lines is tracked. The third component displays real-time metadata associated with the detected vehicle, including the estimated speed and tracking ID.

Furthermore, Figure 7 demonstrates the robustness of the proposed system in more complex scenes involving multiple vehicles. The framework accurately detects, tracks, and estimates the speed of several vehicles concurrently, validating its effectiveness in dynamic and high-traffic environments. This multi-object handling capability is essential for real-world deployment in urban scenarios where traffic congestion and vehicle diversity are common.

Figure 7: Simultaneous Detection, Tracking, and Speed Estimation of Multiple Vehicles.
Source: Authors, (2025).

IV.3 CHALLENGES AND LIMITATIONS IN ACHIEVING ACCURATE SPEED ESTIMATION

This section highlights the potential sources of error in the speed estimation process and outlines the challenges that must be addressed to enhance precision. A deeper understanding of these limitations allows for the formulation of strategies to improve the reliability and accuracy of the proposed system.

While the estimated speeds are close to the actual speeds with only minor errors, the methodology should theoretically result in zero or very negligible error, approaching 100% precision. This is because the method relies on the basic speed computation formula, using the traveled distance and elapsed time to find the average speed. Our scheme detects vehicles as they cross specified lines (points) in the scene, computes the number of frames (with 30 frames representing to 1 second), and uses a field-measured distance. Despite this, a variety of factors impact the results, and addressing these challenges is essential for achieving optimal accuracy. The key challenges are:

a) Inaccuracies in Field Distance Measurement: The precision of the distance measured between reference lines directly affects the speed estimation. Inaccuracies may arise from the measuring instruments used or human error during manual measurement, which in turn propagates into the speed computation.

b) Line Positioning via Image Editing Tools: The start and end lines used for calculating vehicle crossing times were manually placed using simple image editing software (e.g., Microsoft Paint). This introduces uncertainty in the exact pixel positioning of the reference lines. Given the sensitivity of the system, even a minor error (e.g., a one-pixel or one-centimeter discrepancy) can significantly impact the final speed estimate.

c) Vehicle Speedometer Inaccuracy: The actual speeds used for comparison are obtained from the vehicle's onboard speedometer. However, speedometers may not provide precise readings due to calibration issues, mechanical inconsistencies, or slight delays in measurement, contributing to the observed discrepancies.

d) Frame-Based Timing Resolution: Since video footage is composed of discrete frames (typically 24 to 40 FPS), there is a possibility that the exact moment a vehicle crosses the reference line is not captured. This leads to reliance on the nearest available frame for time calculation. Such rounding in frame detection introduces timing error, which affects the estimated speed, especially over short distances.

The challenges outlined above significantly influence the accuracy of the estimated speeds. To enhance the system's precision and approach theoretical accuracy, it is crucial to refine the distance measurement process, automate the calibration of reference lines with higher precision, account for vehicle instrumentation uncertainty, and improve temporal resolution. Addressing these factors comprehensively will contribute to more reliable and robust speed estimation results in real-world applications.

V. CONCLUSIONS

This study proposed a cost-effective and accessible approach for real-time vehicle speed estimation using freely available deep learning frameworks—YOLOv7 for object detection and DeepSORT for tracking. The system is well-suited for deployment in resource-constrained settings due to its reliance on open-source tools and its ability to operate efficiently without high-end hardware.

Accurate speed estimation is critically dependent on the robustness of vehicle detection and tracking. Once these tasks are reliably performed, computing the vehicle's speed using distance and elapsed time becomes straightforward. Experimental results demonstrated that the proposed system achieves 100% detection accuracy and delivers highly satisfactory speed estimation, with average errors below 3%.

However, certain factors significantly influence the precision of the estimated speeds. These include (a) the accuracy of field distance measurements, (b) the precise positioning of start and end reference lines, and (c) the limitations of video framing in capturing exact crossing moments. Addressing these challenges is essential to further improve the reliability and accuracy of the system.

In future work, this methodology can be extended to estimate vehicle speeds in both directions of traffic and incorporate vehicle classification based on legal speed limits, which may vary by type (e.g., trucks, passenger cars, motorcycles). Moreover, instead of relying on predefined reference lines, license plates can be utilized as scale-invariant reference objects, leveraging their known dimensions to calculate speed based on pixel displacement over time. This extension also opens the door to integrating driver identification modules, enabling the system to serve as a more comprehensive traffic monitoring and law enforcement solution—particularly valuable in developing regions where cost and infrastructure constraints are prevalent.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Methodology: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Investigation: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Discussion of results: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Writing – Original Draft: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Writing – Review and Editing: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Supervision: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

Approval of the final text: Merrad Ahmed, Daoud Walid, Dalouli Aissa, Latrech Boubakeur and Nouri Abdelkader Nabil.

VII. REFERENCES

[1] M. Shamsuddoha, M.A. Kashem, T. Nasir, A Review of Transportation 5.0: Advancing Sustainable Mobility Through Intelligent Technology and Renewable Energy, Future Transportation 5 (2025) 8. https://doi.org/10.3390/futuretransp5010008.

[2] R. Abdulrahman, M. Almoshaogeh, H. Haider, F. Alharbi, A. Jamal, Development and application of a risk analysis methodology for road traffic accidents, Alexandria Engineering Journal 111 (2025) 293–305. https://doi.org/10.1016/j.aej.2024.10.045.

[3] M. Fowler, A. Choudhury, Motor vehicle crashes and seat belt compliance among law enforcement officers: A systematic literature review, Journal of Safety and Sustainability 2 (2025) 11–21. https://doi.org/10.1016/j.jsasus.2025.02.004.

[4] Z. Tian, Investigating Impact of Speed on Traffic Safety using Collision Prediction Model, Adv Mat Res 779–780 (2013) 482–485. https://doi.org/10.4028/www.scientific.net/AMR.779-780.482.

[5] S. Ahmed, M.A. Hossain, S.K. Ray, M.M.I. Bhuiyan, S.R. Sabuj, A study on road accident prediction and contributing factors using explainable machine learning models: analysis and performance, Transp Res Interdiscip Perspect 19 (2023) 100814. https://doi.org/10.1016/j.trip.2023.100814.

[6] M. Naidoo, S. Paine, A.K. Mishra, M.Y.A. Gaffar, Reliable Traffic Monitoring Using Low-Cost Doppler Radar Units, (2025).

[7] X. Xu, B. Wu, Â.P. Teixeira, C.G. Soares, Ship Collision Risk Assessment and Visualization Using Radar Sequential Images for Intelligent Maritime Supervision, IEEE Transactions on Intelligent Transportation Systems (2025) 1–18. https://doi.org/10.1109/TITS.2025.3539011.

[8] F. Michael, F. Al-Turjman, M. Auwal, C. Altrjman, Traffic management system using different Internet of Things devices: literature review, in: Artificial Intelligence of Things (AIoT), Elsevier, 2025: pp. 47–53. https://doi.org/10.1016/B978-0-443-26482-5.00006-7.

[9] Y. Li, Q. Zhao, M. Wang, High-resolution traffic flow data from the urban traffic control system in Glasgow, Sci Data 12 (2025) 253. https://doi.org/10.1038/s41597-025-04494-y.

[10] Y. Iwao, Y. Yamamoto, H. Yaginuma, Y. Taniguchi, Traffic volume measurement using nonlinear count-lines, Mach Vis Appl 36 (2025) 52. https://doi.org/10.1007/s00138-025-01668-x.

[11] C. Panapiphat, E. Songkoh, S. Phonkaporn, K. Sirichunchuen, P. Unahalekhaka, Optimal of Placement for Battery Energy Storage System Installation Using Fuzzy Expert System in Thailand: A Case Study of Critical Closed-Circuit Television Positions, Energies (Basel) 18 (2025) 1328. https://doi.org/10.3390/en18061328.

[12] A. Mohanty, A.G. Mohapatra, S.K. Mohanty, Real-Time Traffic Monitoring with AI in Smart Cities, in: 2025: pp. 135–165. https://doi.org/10.1007/978-3-031-72959-1_7.

[13] K. Wu, J. Ding, J. Lin, G. Zheng, Y. Sun, J. Fang, T. Xu, Y. Zhu, B. Gu, Big-data empowered traffic signal control could reduce urban carbon emission, Nat Commun 16 (2025) 2013. https://doi.org/10.1038/s41467-025-56701-4.

[14] C. Szabo, B. Sims, T. Mcatee, R. Lodge, R. Hunjet, Self-Adaptive Software Systems in Contested and Resource-Constrained Environments: Overview and Challenges, IEEE Access 9 (2021) 10711–10728. https://doi.org/10.1109/ACCESS.2020.3043440.

[15] M. Bakirci, Enhancing vehicle detection in intelligent transportation systems via autonomous UAV platform and YOLOv8 integration, Appl Soft Comput 164 (2024) 112015. https://doi.org/10.1016/j.asoc.2024.112015.

[16] Z. Yan, L. Jiang, X. Huang, L. Zhang, X. Zhou, Intelligent urbanism with artificial intelligence in shaping tomorrow's smart cities: current developments, trends, and future directions, Journal of Cloud Computing 12 (2023) 179. https://doi.org/10.1186/s13677-023-00569-6.

[17] Y. Feng, J.A. Zhang, B. Cheng, X. He, J. Chen, Magnetic Sensor-Based Multi-Vehicle Data Association, IEEE Sens J 21 (2021) 24709–24719. https://doi.org/10.1109/JSEN.2021.3112161.

[18] T. Azfar, J. Li, H. Yu, R.L. Cheu, Y. Lv, R. Ke, Deep Learning-Based Computer Vision Methods for Complex Traffic Environments Perception: A Review, Data Science for Transportation 6 (2024) 1. https://doi.org/10.1007/s42421-023-00086-7.

[19] E. Dilek, M. Dener, Computer Vision Applications in Intelligent Transportation Systems: A Survey, Sensors 23 (2023) 2938. https://doi.org/10.3390/s23062938.

[20] M. Kamruzzaman, O. Alruwaili, AI-based computer vision using deep learning in 6G wireless networks, Computers and Electrical Engineering 102 (2022) 108233. https://doi.org/10.1016/j.compeleceng.2022.108233.

[21] T. Diwan, G. Anirudh, J. V. Tembhurne, Object detection using YOLO: challenges, architectural successors, datasets and applications, Multimed Tools Appl 82 (2023) 9243–9275. https://doi.org/10.1007/s11042-022-13644-y.

[22] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A Review of Yolo Algorithm Developments, Procedia Comput Sci 199 (2022) 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135.

[23] J. Lee, K. Hwang, YOLO with adaptive frame control for real-time object detection applications, Multimed Tools Appl 81 (2022) 36375–36396. https://doi.org/10.1007/s11042-021-11480-0.

[24] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016: pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.

[25] Y. Tian, Q. Ye, D. Doermann, YOLOv12: Attention-Centric Real-Time Object Detectors, (2025).

[26] S. Khokhar, D. Kedia, Integrating YOLOv8 and CSPBottleneck based CNN for enhanced license plate character recognition, J Real Time Image Process 21 (2024) 168. https://doi.org/10.1007/s11554-024-01537-2.

[27] Y. Zhou, X. Wu, Y. Li, H. Sun, D. Fan, Algorithm for surface flow velocity measurement in trunk canal based on improved YOLOv8 and DeepSORT, Eng Appl Artif Intell 148 (2025) 110344. https://doi.org/10.1016/j.engappai.2025.110344.

[28] H.-S. Sim, H.-C. Cho, Enhanced DeepSORT and StrongSORT for Multicattle Tracking With Optimized Detection and Re-Identification, IEEE Access 13 (2025) 19353–19364. https://doi.org/10.1109/ACCESS.2025.3535092.

[29] M. Adžemović, P. Tadić, A. Petrović, M. Nikolić, Beyond Kalman filters: deep learning-based filters for improved object tracking, Mach Vis Appl 36 (2025) 20. https://doi.org/10.1007/s00138-024-01644-x.

[30] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017: pp. 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962.

[31] L. Wang, C.T. Lam, K.L.E. Law, B. Ng, W. Ke, M. Im, Real-Time Traffic Monitoring and Status Detection with a Multi-vehicle Tracking System, in: 2022: pp. 13–25. https://doi.org/10.1007/978-3-030-97603-3_2.

[32] T. Zhonglin, M.N.A. Wahab, M.F. Akbar, A.S.A. Mohamed, M.H.M. Noor, B.A. Rosdi, SFFSORT Multi-Object Tracking by Shallow Feature Fusion for Vehicle Counting, IEEE Access 11 (2023) 76827–76841. https://doi.org/10.1109/ACCESS.2023.3297190.

[33] Z. Sun, G. Wei, W. Fu, M. Ye, K. Jiang, C. Liang, T. Zhu, T. He, M. Mukherjee, Multiple Pedestrian Tracking Under Occlusion: A Survey and Outlook, IEEE Transactions on Circuits and Systems for Video Technology 35 (2025) 1009–1027. https://doi.org/10.1109/TCSVT.2024.3481425.

[34] H. Zhang, R. Fang, S. Li, Q. Miao, X. Fan, J. Hu, S. Chan, Multi-Camera Multi-Vehicle Tracking Guided by Highway Overlapping FoVs, Mathematics 12 (2024) 1467. https://doi.org/10.3390/math12101467.

[35] M. Chen, R. Tharmarasa, T. Kirubarajan, S. Chomal, A Two-Step Multiframe Assignment Method for Multiple Extended Target Tracking With Azimuth Ambiguity Based on Pseudo Measurement Set, IEEE Trans Aerosp Electron Syst 61 (2025) 3589–3609. https://doi.org/10.1109/TAES.2024.3493054.

[36] I. Bilik, Comparative Analysis of Radar and Lidar Technologies for Automotive Applications, IEEE Intelligent Transportation Systems Magazine 15 (2023) 244–269. https://doi.org/10.1109/MITS.2022.3162886.

[37] S.I. Siam, H. Ahn, L. Liu, S. Alam, H. Shen, Z. Cao, N. Shroff, B. Krishnamachari, M. Srivastava, M. Zhang, Artificial Intelligence of Things: A Survey, ACM Trans Sens Netw 21 (2025) 1–75. https://doi.org/10.1145/3690639.

[38] D. Fernández Llorca, A. Hernández Martínez, I. García Daza, Vision-based vehicle speed estimation: A survey, IET Intelligent Transport Systems 15 (2021) 987–1005. https://doi.org/10.1049/itr2.12079.

[39] S. Talati, D. Vekaria, A. Kumari, S. Tanwar, An AI-driven object segmentation and speed control scheme for autonomous moving platforms, Computer Networks 186 (2021) 107783. https://doi.org/10.1016/j.comnet.2020.107783.

[40] S. Yaqoob, G. Morabito, S. Cafiso, G. Pappalardo, A. Ullah, AI-Driven Driver Behavior Assessment Through Vehicle and Health Monitoring for Safe Driving—

A Survey, IEEE Access 12 (2024) 48044–48056. https://doi.org/10.1109/ACCESS.2024.3383775.

[41] C. Kuai, B. Zhao, C. Si, L. Huang, Velocity Estimation for Vehicle-Mounted SAR Based on Deep-Learning Framework, IEEE Sens J 22 (2022) 22952–22962. https://doi.org/10.1109/JSEN.2022.3213903.

[42] H. Saedi, A. Abdi Kordani, H. Divandari, Improving safety in rural highways horizontal curves using geometric design consistency evaluation criteria based on operating speed modeling, Innovative Infrastructure Solutions 9 (2024) 134. https://doi.org/10.1007/s41062-024-01428-2.

[43] Y.-S. Lin, T.-Y. Chen, J.-J. Liaw, H.-H. Yang, C.-H. Hsieh, Vehicle Exhaust Estimation Using YOLOv7 and Support Vector Regression with Image Features, Information 16 (2025) 168. https://doi.org/10.3390/info16030168.

[44] L. Kang, Z. Lu, L. Meng, Z. Gao, YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection, Expert Syst Appl 237 (2024) 121209. https://doi.org/10.1016/j.eswa.2023.121209.

[45] M.I. Zaman, U.I. Bajwa, G. Saleem, R.H. Raza, A robust deep networks based multi-object multi-camera tracking system for city scale traffic, Multimed Tools Appl 83 (2023) 17163–17181. https://doi.org/10.1007/s11042-023-16243-7.

[46] P.H. Nguyen, M.B. Duy, An algorithm using YOLOv4 and DeepSORT for tracking vehicle speed on highway, Indonesian Journal of Electrical Engineering and Informatics (IJEEI) 10 (2022). https://doi.org/10.52549/ijeei.v10i1.3448.

[47] Z. Luo, Y. Bi, X. Yang, Y. Li, S. Yu, M. Wu, Q. Ye, Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification, Front Phys 12 (2024). https://doi.org/10.3389/fphy.2024.1371320.

[48] A. Cvijetić, S. Djukanović, A. Peruničić, Deep learning-based vehicle speed estimation using the YOLO detector and 1D-CNN, in: 2023 27th International Conference on Information Technology (IT), IEEE, 2023: pp. 1–4. https://doi.org/10.1109/IT57431.2023.10078518.

[49] M. Dastgerdi Vahid, A. Leus, V. Zuev, N. Vodichev, I. Kholodnyak, V. Efremov, D. Mangazeev, N. Pecherkin, Optimized zone-based vehicle speed estimation and classification, ITM Web of Conferences 59 (2024) 04002. https://doi.org/10.1051/itmconf/20245904002.

[50] E. Kisingo, N. Hamisi, H.U. Iddi, B.J. Maiseli, Multi-Vehicle Speed Estimation Algorithm Based on Real-Time Inter-Frame Tracking Technique, Tanzania Journal of Science 47 (2021) 1125–1137. https://doi.org/10.4314/tjs.v47i3.22.

[51] S. Bhatlawande, S. Shilaskar, A. Mahajan, M. Shete, Vision-Based Vehicle Speed Estimation, in: 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT), IEEE, 2024: pp. 1–5. https://doi.org/10.1109/ICDCOT61034.2024.10516237.

[52] A.A. Tayeb, R.W. Aldhaheri, M.S. Hanif, Vehicle Speed Estimation Using Gaussian Mixture Model and Kalman Filter, INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL 16 (2021). https://doi.org/10.15837/ijccc.2021.4.4211.

[53] S. Lu, Y. Wang, H. Song, A high accurate vehicle speed estimation method, Soft Comput 24 (2020) 1283–1291. https://doi.org/10.1007/s00500-019-03965-w.

[54] H. Kim, Vehicle detection and speed estimation for automated traffic surveillance systems at nighttime, Tehnički Vjesnik 26 (2019) 87–94. https://doi.org/10.17559/TV-20170827091448.

[55] T. Kumar, D.S. Kushwaha, An Efficient Approach for Detection and Speed Estimation of Moving Vehicles, Procedia Comput Sci 89 (2016) 726–731. https://doi.org/10.1016/j.procs.2016.06.045.

[56] I. García-Aguilar, J. García-González, E. Domínguez, E. López-Rubio, R.M. Luque-Baena, Real-Time Deep Learning Framework for Accurate Speed Estimation of Surrounding Vehicles in Autonomous Driving, Electronics (Basel) 13 (2024) 2790. https://doi.org/10.3390/electronics13142790.

[57] Y. Chen, D. Zhao, M.J. Er, Y. Zhuang, H. Hu, A novel vehicle tracking and speed estimation with varying UAV altitude and video resolution, Int J Remote Sens 42 (2021) 4441–4466. https://doi.org/10.1080/01431161.2021.1895449.

[58] S. Tedesco, C. Crowe, M. Sica, L. Kenny, B. O'Flynn, D.S. Mueller, S. Timmons, J. Barton, Gait speed estimation via inertial sensors and machine learning, Gait Posture 106 (2023) S201–S203. https://doi.org/10.1016/j.gaitpost.2023.07.243.

[59] D. Biswas, H. Su, C. Wang, A. Stevanovic, Speed Estimation of Multiple Moving Objects from a Moving UAV Platform, ISPRS Int J Geoinf 8 (2019) 259. https://doi.org/10.3390/ijgi8060259.

[60] P. Chmielewski, K. Sibilski, Ground Speed Optical Estimator for Miniature UAV, Sensors 21 (2021) 2754. https://doi.org/10.3390/s21082754.

[61] T. Kamnardsiri, S. Boripuntakul, C. Kaiket, Computer vision-based instantaneous speed tracking system for measuring the subtask speed in the 100-meter sprinter: Development and concurrent validity study, Heliyon 10 (2024). https://doi.org/10.1016/j.heliyon.2024.e24086.

[62] B. Yang, R. Chen, B. Li, C. Li, Multi-vehicle cooperative positioning based on edge-computed multidimensional scaling, China Communications 18 (2021) 53–63. https://doi.org/10.23919/JCC.2021.06.005.

[63] Google Colab, https://colab.research.google.com/ (accessed April 18, 2025).

[64] numpy, https://numpy.org/ (accessed April 18, 2025).

[65] opencv, https://opencv.org/ (accessed April 18, 2025).

[66] matplotlib, https://matplotlib.org/ (accessed April 18, 2025).