Real-Time Vehicle Detection and Tracking Using YOLOv8 and BoT-SORT for Traffic Surveillance

Mrs. Prachi Sodegaonkar [1], Dr.R.A.Khan[2]

[1]Department of Artificial Intelligence & Machine Learning, Deogiri Institute of Engineering & Management Studies, Chhatrapati Sambhajinagar, Maharashtra, India

[2]Department of Computer Science & Engineering, G.H.Raisoni International Skill Tech University, Pune, Maharashtra, India

---***---

Abstract - *With the rapid growth of urban traffic, there is an increasing need for efficient and automated monitoring systems in intelligent transportation applications. In this work, a real-time vehicle detection and tracking framework is presented using the YOLOv8 object detection model integrated with the BoT-SORT multi-object tracking algorithm. The system processes traffic surveillance video streams to detect vehicles and assign unique tracking IDs, ensuring continuity across consecutive frames. The YOLOv8 model provides reliable detection performance, while BoT-SORT enhances tracking by maintaining identity even under occlusion and motion variations. Experimental observations show that the proposed system can effectively detect and track multiple vehicles in real-world traffic scenarios. The approach is computationally efficient and suitable for practical traffic monitoring applications.*

Key Words: **YOLOv8, BoT-SORT, Vehicle Detection, Object Tracking, Traffic Surveillance, Deep Learning, Computer Vision, Intelligent Transportation Systems**

1. INTRODUCTION

The rapid increase in the number of vehicles due to urbanization has created significant challenges in traffic monitoring and management. Efficient surveillance is essential to ensure road safety and reduce congestion. Traditional methods rely on manual observation, which is not only time-consuming but also difficult to scale for large urban environments.

Recent advancements in computer vision and deep learning have enabled automated solutions for traffic analysis. Object detection models such as YOLO (You Only Look Once) have demonstrated high accuracy and real-time performance, making them suitable for applications in traffic surveillance. In addition, multi-object tracking techniques play a crucial role in maintaining the identity of detected vehicles across consecutive frames, allowing continuous monitoring of vehicle movement.

Vehicle detection alone is insufficient for understanding traffic dynamics, as it only provides information at a single frame level. To address this limitation, tracking algorithms such as BoT-SORT are employed to associate detections over time and assign unique identifiers to each vehicle. This enables consistent tracking even in the presence of occlusions and varying motion patterns.

In this paper, a real-time vehicle detection and tracking framework is developed using YOLOv8 and BoT-SORT. The system processes video streams and tracks vehicles across frames with consistent identity assignment. The approach is simple, efficient, and suitable for practical deployment in traffic surveillance scenarios.

The main contributions of this work are as follows:

- Implementation of a real-time vehicle detection system using YOLOv8
- Integration of BoT-SORT for robust multi-object tracking
- Development of a complete pipeline for traffic surveillance applications

The remainder of the paper is organized as follows: Section 2 presents the literature review, Section 3 describes the proposed methodology, Section 4 discusses the dataset, Section 5 explains implementation details, Section 6 presents results and discussion, and Section 7 concludes the paper.

2. LITERATURE REVIEW

Recent advancements in computer vision and deep learning have significantly improved the performance of vehicle detection and tracking systems in traffic surveillance applications. Various object detection models and tracking algorithms have been explored in the literature to address challenges such as occlusion, varying illumination, and real-time processing.

Early object detection approaches relied on region-based methods such as Faster R-CNN, which provided high accuracy but suffered from computational complexity, limiting real-time applicability [3]. To overcome these limitations, single-stage detectors such as YOLO were introduced, offering a balance between speed and accuracy for real-time object detection tasks [1]. Similarly, the SSD (Single Shot MultiBox Detector) model improved detection efficiency while maintaining competitive accuracy [2].

Recent improvements in the YOLO family, including YOLOv4 and YOLOv7, have further enhanced detection performance through architectural optimizations and training strategies [4], [5]. The latest version, YOLOv8, provides improved accuracy and efficiency, making it suitable for real-time traffic surveillance applications [18].

In the domain of vehicle detection, several studies have focused on applying deep learning models for parking detection and traffic monitoring. Amato et al. proposed a decentralized deep learning approach for parking occupancy detection using surveillance images [6]. Yin et al. developed a deep learning-based parking detection system capable of identifying parking spaces in real-time scenarios [7]. Similarly, Tang et al. utilized SSD for real-time parking detection, demonstrating improved detection speed [8].

Beyond detection, tracking algorithms are essential for maintaining object identity across frames. The SORT algorithm introduced a simple yet effective approach for real-time tracking based on Kalman filtering and data association [14]. DeepSORT extended this approach by incorporating appearance features for improved tracking performance under occlusion conditions [12]. More recently, ByteTrack demonstrated enhanced tracking accuracy by associating both high and low confidence detections, improving robustness in crowded scenes [13].

BoT-SORT, an extension of ByteTrack, further improves tracking performance by integrating motion and appearance cues, making it highly effective for multi-object tracking in complex environments [13]. These tracking methods enable continuous monitoring of vehicles and are widely used in intelligent traffic systems.

Vehicle behavior analysis has also gained attention in recent studies. Li et al. explored vehicle behavior modeling using tracking and machine learning techniques, highlighting the importance of temporal information in traffic analysis [11]. Mo et al. proposed event-based detection methods for traffic surveillance, emphasizing the role of temporal features in identifying complex traffic events [15].

Furthermore, several recent works have combined detection and tracking for traffic monitoring applications. Sharma et al. integrated YOLO-based detection with tracking algorithms for vehicle monitoring in urban environments [16]. Zhao et al. proposed attention-based detection methods to improve accuracy in traffic scenes [17]. Luz et al. developed a YOLO-based smart parking system for automated parking detection [10]. Kathait et al. presented a comprehensive review of deep learning approaches for parking detection, highlighting current challenges and future directions [18].

Machine learning techniques such as Random Forest have also been widely used for classification tasks in traffic analysis due to their robustness and interpretability [19]. Bishop's work on pattern recognition provides a strong theoretical foundation for machine learning-based approaches in computer vision applications [20].

Despite significant progress, challenges remain in achieving robust real-time performance under varying traffic conditions. The integration of efficient detection models such as YOLOv8 with advanced tracking algorithms like BoT-SORT provides a promising solution for real-time vehicle monitoring in intelligent transportation systems.

3. PROPOSED SYSTEM

Figure 1 illustrates a comprehensive framework for vehicle detection and tracking by integrating the YOLOv8 object detection algorithm with a multi-object tracking mechanism. The overall process is divided into three main components: detection, tracking, and output generation.

On the left side, the YOLOv8 algorithm is employed for object detection. The process begins with adopting publicly available datasets for training, followed by the development of an improved YOLOv8-based model. The model is trained using the MS COCO dataset and is capable of detecting vehicles in input video frames. The output of this stage consists of bounding boxes corresponding to detected vehicles.

In the central pipeline, the system processes the input video by performing object detection on each frame. The detected objects are then passed to the tracking module, where target tracking is performed. The system outputs the number of tracked vehicles based on consistent detections across frames.

On the right side, the tracking process is illustrated using a BoT-SORT-based approach. The tracking module extracts deep features from detected objects and utilizes both motion and appearance information to improve tracking accuracy. It applies motion prediction techniques such as the Kalman filter along with data association methods to associate detections across consecutive frames. This ensures that each vehicle is assigned a consistent identity even in the presence of occlusions and dynamic scene variations. Finally, the tracking states are updated, and the results are generated as output.

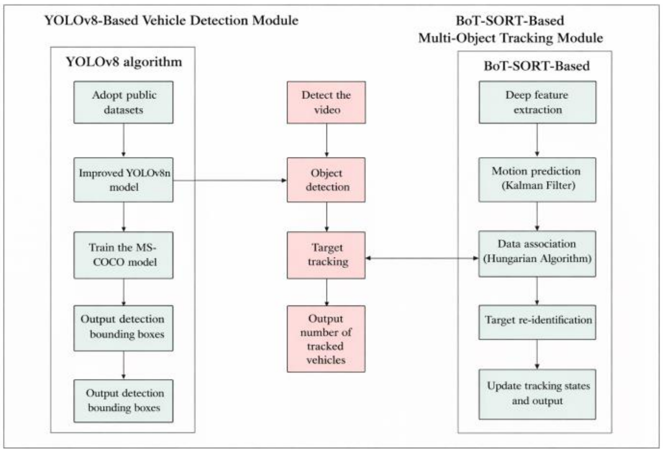

Fig -1: Integrated Vehicle Detection and Tracking Framework using YOLOv8 and Bot-SORT

Overall, the framework combines deep learning-based detection with robust tracking algorithms to achieve accurate and real-time vehicle monitoring in traffic surveillance scenarios.

The proposed system aims to perform real-time vehicle detection and tracking in traffic surveillance videos by integrating a deep learning-based object detection model with a robust multi-object tracking algorithm. The overall framework consists of sequential stages including frame extraction, vehicle detection, tracking, and trajectory analysis.

3.1 System Overview

Figure 2 illustrates the workflow of the proposed system, where the input video is processed frame-by-frame for vehicle detection using YOLOv8, followed by multi-object tracking using BoT-SORT to assign unique IDs and generate vehicle trajectories.

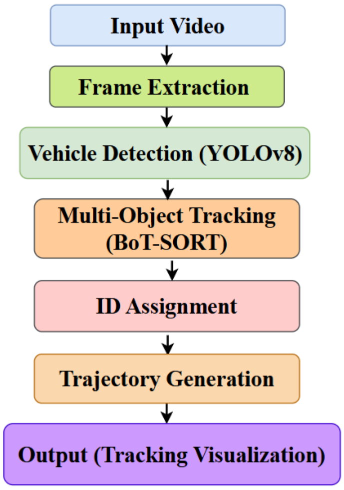

Fig -2: System architecture of the proposed vehicle detection and tracking framework

The system processes input video streams captured from traffic surveillance cameras. Each video is decomposed into individual frames, which are then analyzed using a deep learning model for vehicle detection. The detected vehicles are subsequently tracked across frames using a tracking algorithm to maintain identity consistency.

3.2 Frame Extraction

The input video is processed frame-by-frame using OpenCV. Each frame represents a snapshot of the traffic scene and is used as input for the detection module. This step ensures continuous processing of the video stream for real-time analysis.

3.3 Vehicle Detection using YOLOv8

Vehicle detection is performed using the YOLOv8 (You Only Look Once version 8) object detection model. YOLOv8 is a one-stage detector that processes the entire image in a single forward pass, enabling real-time performance.

The model is pre-trained on the COCO dataset and configured to detect relevant vehicle classes such as cars, buses, and trucks. Each detected vehicle is represented by a bounding box along with its spatial coordinates.

3.4 Multi-Object Tracking using BoT-SORT

To maintain temporal consistency across frames, the BoT-SORT tracking algorithm is employed. The tracker assigns a unique identifier (ID) to each detected vehicle and updates it across consecutive frames.

BoT-SORT utilizes motion and spatial information to associate detections between frames, ensuring that the same vehicle is consistently tracked even in the presence of occlusion or movement variations.

3.5 Trajectory Generation

The movement of each tracked vehicle is recorded across frames to generate trajectories. These trajectories represent the path followed by each vehicle within the scene and can be used for further traffic analysis.

Each vehicle observation can be represented as:

$$O = \{(t, id, x_t, y_t)\} \quad (1)$$

where t is the frame index, id is the tracking identifier, and $(x_t\, y_t)$ denotes the centroid position of the vehicle.

3.6 System Workflow

The overall workflow of the proposed system is summarized as follows:

- Input traffic video
- Frame extraction
- Vehicle detection using YOLOv8
- Multi-object tracking using BoT-SORT
- ID assignment and trajectory generation

4. EXPERIMENTAL SETUP

The experimental setup is designed to evaluate the performance of the proposed vehicle detection and tracking system in real-world traffic surveillance scenarios. The system is implemented using Python and executed on a standard computing environment.

4.1 Hardware and Software Configuration

The proposed system is implemented on a machine with the following configuration:

- Processor: Intel Core i5/i7
- RAM: 8 GB or higher
- Operating System: Windows 10
- Programming Language: Python 3.x
- Libraries: Ultralytics YOLOv8, OpenCV

The experiments are conducted using the Ultralytics implementation of YOLOv8 for object detection and BoT-SORT for multi-object tracking.

4.2 Dataset Description

The dataset used in this study consists of real-world traffic surveillance videos. The videos represent urban traffic environments with moderate vehicle density and varying motion patterns.

- Video format: AVI/MP4
- Resolution: 1280 × 720 pixels
- Frame rate: 25–30 FPS
- Scene type: Urban road traffic

Due to privacy constraints, the dataset is not publicly shared. However, publicly available traffic videos from online repositories can also be used for evaluation.

4.3 Implementation Details

The proposed system processes input video streams frame-by-frame. Each frame is passed through the YOLOv8 model to detect vehicles such as cars, buses, and trucks. The detected objects are then tracked across consecutive frames using the BoT-SORT algorithm, which assigns unique IDs to maintain temporal consistency.

Only relevant vehicle classes are considered to improve efficiency and reduce noise. The system operates in real-time and generates output in the form of annotated video frames with bounding boxes and tracking IDs.

4.4 Evaluation Criteria

The performance of the system is evaluated based on the following criteria:

- Detection accuracy (correct identification of vehicles)
- Tracking consistency (maintenance of unique IDs across frames)
- Real-time performance (processing speed and responsiveness)

The qualitative results are analyzed through visual inspection of detection and tracking outputs.

5. RESULTS AND DISCUSSION

The proposed system was tested on real-world traffic videos to evaluate its performance under practical conditions. The combination of YOLOv8 and BoT-SORT enables accurate detection and stable tracking of multiple vehicles across frames.

5.1 Detection and Tracking Performance

The output of the proposed system is illustrated in Fig. 3, where multiple vehicles are successfully detected and assigned unique tracking IDs. The system demonstrates the ability to handle multiple vehicles simultaneously while maintaining identity consistency across consecutive frames. It can be observed that vehicles moving in different directions and at varying speeds are correctly identified and tracked without significant loss of identity.

The results indicate that the YOLOv8 model effectively detects vehicles such as cars and trucks, while the BoT-SORT algorithm ensures robust tracking even under moderate traffic density. The assigned tracking IDs remain consistent, indicating stable tracking performance. Additionally, the system performs reliably even in slightly complex scenes, where multiple vehicles are present in close proximity, highlighting its capability for real-world traffic surveillance applications.

Fig -3: Output of the proposed system showing real-time vehicle detection and tracking with unique IDs assigned to each detected vehicle

5.2 Graphical Analysis

To further analyze system performance, three graphs were generated representing vehicle detection, tracking behavior, and processing speed across frames.

Vehicle Count Analysis

The variation in the number of detected vehicles across frames is shown in Chart 1.

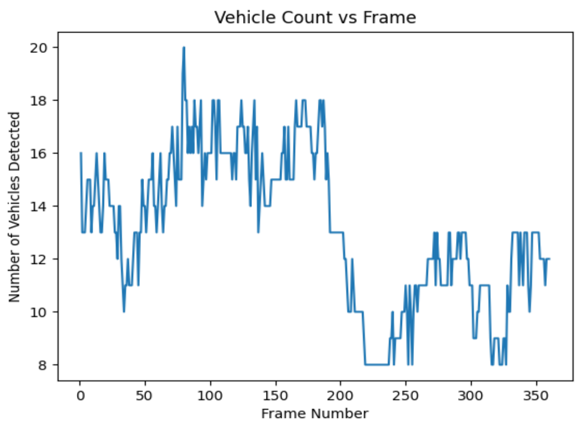

Chart -1: Number of detected vehicles versus frame number

The graph shows that the number of detected vehicles varies between 8 and 20 across frames, with an average of approximately 13 vehicles. This variation reflects real-world traffic conditions where vehicle density changes over time.

Tracking Consistency Analysis

The number of active tracking IDs across frames is illustrated in Chart 2.

The graph closely follows the vehicle count trend, indicating that each detected vehicle is successfully assigned and maintained with a unique ID. This demonstrates the effectiveness of the BoT-SORT tracking algorithm in maintaining identity consistency.

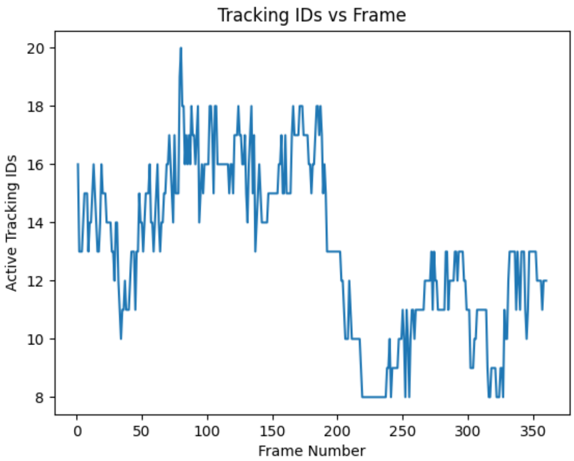

Chart -2: Number of active tracking IDs versus frame number

Processing Speed Analysis

The processing speed of the system in terms of frames per second (FPS) is shown in Chart 3.

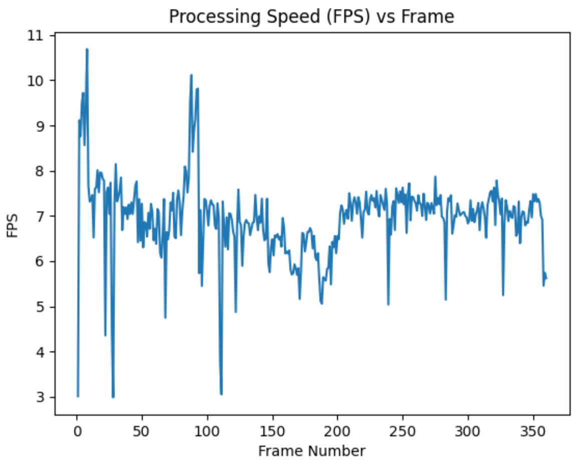

Chart -3: Processing speed (FPS) versus frame number

The system maintains an average processing speed of approximately 6.96 FPS, with minor fluctuations due to variations in scene complexity and object density. Occasional drops in FPS are observed during frames with higher vehicle density, which is expected in real-time processing systems.

5.3 Quantitative Performance Evaluation

A summary of the system performance is presented in Table 1. The results indicate that the system performs consistently across frames, accurately detecting and tracking vehicles while maintaining real-time processing capability. The close similarity between the average vehicle count and tracking IDs confirms that the tracking algorithm effectively preserves object identity. Furthermore, the system maintains a balanced performance across varying traffic conditions, demonstrating its robustness in real-world scenarios. The consistent relationship between detection and tracking results highlights the reliability of the overall framework.

Table -1: Performance Analysis of Proposed System

Metric	**Value**
Average Vehicle Count per Frame	13.14
Maximum Vehicles Detected	20
Minimum Vehicles Detected	8
Average Active Tracking IDs	13.14
Average Processing Speed (FPS)	6.96

5.4 Discussion

The results indicate that the system performs consistently in moderate traffic conditions. The tracking IDs remain stable across frames, showing that the tracking algorithm works effectively. Although some fluctuations in FPS are observed, the system still operates at near real-time speed.

The system is capable of handling moderate traffic density while maintaining stable tracking and acceptable processing speed. Although minor fluctuations in FPS are observed, the system remains suitable for near real-time applications. The approach can be further improved by optimizing hardware resources or using more lightweight models for faster inference.

Overall, the proposed system provides an effective solution for automated vehicle monitoring in traffic surveillance and can be extended for advanced applications such as traffic analysis and smart city management.

6. CONCLUSION

In this work, a real-time vehicle detection and tracking system has been developed using YOLOv8 and BoT-SORT. The system is capable of detecting multiple vehicles and maintaining consistent tracking across frames. The experimental results show that the approach performs well in real-world traffic scenarios with stable tracking and acceptable processing speed. The system can be useful for traffic monitoring and smart city applications. Future improvements can focus on increasing processing speed and extending the system for advanced applications such as behavior analysis and illegal parking detection.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, “SSD: Single Shot MultiBox Detector,” in Proc. European Conf. Computer Vision (ECCV), 2016.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2015.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020.

[5] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art,” 2022.

[6] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, “Deep learning for decentralized parking lot occupancy detection,” Expert Systems with Applications, vol. 72, pp. 327–334, 2017.

[7] X. Yin, X. Wang, and Z. Liu, “Deep learning-based parking detection system,” IEEE Access, vol. 7, pp. 123–134, 2019.

[8] Z. Tang, H. Wang, and Y. Zhang, “Real-time parking detection using SSD,” in Proc. IEEE Conf., 2020.

[9] R. Luz, J. Silva, and P. Rocha, “YOLO-based smart parking detection system,” 2024.

[10] S. Kathait, A. Singh, and R. Kumar, “Deep learning approaches for parking detection: A review,” 2025.

[11] Y. Li, X. Zhang, and H. Chen, “Vehicle behavior analysis using tracking and machine learning,” IEEE Access, vol. 9, pp. 45678–45689, 2021.

[12] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric (DeepSORT),” 2017.

[13] Y. Zhang, P. Sun, Y. Jiang, and D. Yuan, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” 2022.

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple Online and Realtime Tracking (SORT),” in Proc. IEEE Int. Conf. Image Processing, 2016.

[15] L. Mo, Q. Wang, and Y. Liu, “Mask-SpyNet for event detection in traffic surveillance,” 2022.

[16] Sharma, P. Gupta, and R. Verma, “YOLOv8 with DeepSORT for vehicle tracking in traffic surveillance,” 2023.

[17] Z. Zhao, X. Li, and Y. Chen, “Attention-based YOLO for traffic monitoring,” 2024.

[18] Ultralytics, “YOLOv8 Documentation,” 2023.

[19] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[20] M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[21] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer, 2009.