El Cezeri
Journal of Science and Engineering
ISSN: 2148-3736

Research Article

Real-Time Detection of Vehicle Queue States in Urban Traffic Using Deep Learning

Ahsen Battal[1a], Yunus Emre Avci[1b], Adem Tuncer[2c]
[1] Dept. of Computer Engineering, Institute of Graduate Studies, Yalova University, Yalova, Türkiye
[2] Dept. of Computer Engineering, Engineering Faculty, Yalova University, Yalova, Türkiye

battalahsen@gmail.com

DOI : 10.31202/ecjse.1755333
Received: 01.08.2025Accepted: 16.09.2025
How to cite this article:
Ahsen Battal, Yunus Emre Avci, Adem Tuncer, "Real-Time Detection of Vehicle Queue States in Urban Traffic Using Deep Learning", El-Cezeri Journal of Science and Engineering, Vol: 12, Iss: 3, (2025), pp.(356-364).
ORCID: [a]0000-0002-4824-5889; [b]0000-0003-3921-7162; [c]0000-0001-7305-1886.

Abstract : Traffic congestion and vehicle queue formation at signalized intersections represent critical challenges in modern urban transportation systems, requiring accurate real-time detection methods for effective traffic management. This study presents a deep learning-based approach for real-time vehicle queue state classification that integrates You Only Look Once (YOLO) object detection with Simple Online Real-time Tracking (SORT) algorithms using standard traffic camera footage. The proposed system performs multi-class vehicle classification, real-time vehicle tracking with unique ID assignment, and speed estimation through camera calibration techniques, achieving 16.42 FPS average processing speed across diverse video scenarios. A comprehensive queue state detection methodology is developed that categorizes traffic conditions into three categories: Heavy traffic, stable flow, and free flow based on the analysis of average speeds of the detected vehicles, excluding motorcycles and bicycles due to their distinct traffic behavior patterns. Experimental validation across several test datasets encompassing both high and low resolutions demonstrates robust vehicle detection performance across all vehicle classes. Speed estimation accuracy ranges from 89% to 99%, validated against vehicle counting and tracking in designated traffic lanes, providing essential data for queue analysis. The system achieves vehicle counting accuracy ranging from 78.57% to 100% across different scenarios. The system offers a cost-effective alternative to traditional sensor-based methods by utilizing existing traffic-surveillance infrastructure, making it suitable for widespread deployment in intelligent transportation systems. Results indicate the proposed approach successfully detects queue states in real-time conditions across diverse traffic scenarios, from heavy congestion to free flow conditions. This research advances computer vision-based traffic monitoring by demonstrating the practical effectiveness of integrated object detection and tracking algorithms, contributing to improved traffic flow optimization and congestion management.

Keywords : Deep learning, YOLO, SORT, traffic queue detection, speed detection, vehicle counting

1 Introduction

Traffic congestion has become a critical global challenge in urban areas, causing significant economic, environmental, and social impacts. The accurate measurement of vehicle queues at signalized intersections represents a fundamental challenge in modern Intelligent Transportation Systems (ITS), with direct implications for traffic flow optimization and congestion management. As urban traffic volumes continue to escalate globally, the demand for precise, cost-effective, and real-time traffic state detection systems has intensified significantly. As more people move to urban areas, the need for smart traffic solutions becomes even more urgent. Traditional queue detection methods including inductive loop sensors, manual observation, and basic computer vision algorithms suffer from significant limitations; high infrastructure, limited spatial coverage, poor real-time performance, and inadequate accuracy under varying environmental conditions. ITS are proving to be one of the best ways to fix these issues by tackling the root causes of traffic and finding solutions. Many studies show how effective ITS can be in detecting traffic jams. One of crucial elements in traffic management is the identification of queuing, which directly informs applications like the estimation of the level of service and the optimization of traffic signal control.

Recent advances in deep learning have shown significant promise for queue length estimation in traffic. A notable study [1] developed a hybrid Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to accurately estimate queue lengths at intersections, achieving significant error reduction compared to traditional methods. The CNN components excel at spatial feature extraction from traffic images, while LSTM networks capture temporal dependencies in traffic flow patterns. Additionally, LSTM was used to estimate short-term arrival models and long-term traffic demand trends for lane utilization rate forecasting. Compared to other studies, the developed integrated deep learning model demonstrated high performance in estimating queue lengths in individual lanes. However, it was inadequate for setting the duration, start time, and sequence of signal groups to within a few seconds or minutes for both isolated signalized intersections and area traffic. Traffic control and signal planning are known methods to mitigate traffic congestion and reduce delays. A major challenge in optimizing traffic signal scheduling is accurately predicting traffic conditions before the start of the next forecast cycle. In a study [2] utilizing real-time traffic data and forecasting cycles, LSTM was applied to forecast queue lengths for the upcoming cycle. Additionally, a sequential model-based optimization technique was implemented to prevent overfitting and to select optimal hyperparameters. Experiments using the traffic control system dataset aimed to estimate vehicle queue lengths only for straight movement. The study did not incorporate an adaptability optimization technique, focusing instead on fixed cycle times. Umair et al. [3] proposed a CNN-based approach for estimating vehicle queue length in urban traffic scenarios using low-resolution traffic videos. The queue length was estimated based on the total number of vehicles waiting at a signal, with stopped vehicles detected using Deep SORT-based object tracking. Due to the powerful and accurate CNN-based detection and monitoring, the estimated queue length using cameras was effective. The study conducted a comprehensive analysis of vehicle detection models, including YOLOv3, YOLOv4, YOLOv5, SSD, ResNet101, and InceptionV3, ultimately selecting YOLOv4 as the primary model due its superior accuracy and robustness.

Advanced sensor technologies have emerged as promising solutions for precise queue detection. A significant study [4] introduced a method using roadside LiDAR data, achieving an average accuracy of 98%. This method processes LiDAR data for real-time vehicle tracking and addresses issues like occlusion and package loss. However, several limitations were identified: the detection range of a single LiDAR sensor constraints queue length measurement, requiring additional sensors for longer queues, assumptions about vehicle speed and length could impact detection accuracy, and LiDAR performance may degrade under adverse weather conditions.

Probe vehicles, equipped with various technological tools to gather data from the road environment, are commonly used for calculating and estimating traffic capacity, density, and queue length on roads. These vehicles play a significant role in areas such as queue length and traffic volume estimation. In a study [5] focused on calculating traffic capacity and queue length, different Bayes-based approaches were developed using probe vehicles in each lane. These approaches, which estimate penetration and penetration rate (the ratio of probe vehicles to other vehicles on the highway), were used to calculate queue lengths for straight-going and right-turning lanes. The authors noted that the developed model achieved high prediction accuracy and could be utilized in traffic signal control. A more sophisticated nonparametric approach was developed that moves beyond traditional assumptions of random arrivals and parameter estimation. The main objective is to create straightforward, analytical, nonparametric models that estimate queue lengths at traffic signals on a cycle-by-cycle basis using partial queue observations from probe vehicles. A crucial feature is that this approach doesn't rely on assumptions about random arrivals or the need to estimate fundamental parameters such as market penetration rates or arrival rates. Its simplicity and comparable accuracy to more complex parametric models make it a valuable contribution to the field, particularly for real-time applications where primary parameters are difficult to obtain dynamically. However, its current limitations regarding oversaturated conditions and overflow queues highlight areas for future research [6].

Drone-based computer vision has shown particular promise for comprehensive traffic analysis. Zhou et al. [7] introduced a method for determining the tail profiles using high-resolution data from various sources. The study focused on three key components: signal status estimation, queue profile identification, and lane detection. The developed algorithms were validated using a real-world dataset collected by drones, with results indicating that the methodology effectively extracted tail profile information from the raw drone data. The pNEUMA dataset was employed in this validation process. Traditional computer vision approaches have also been explored, though with mixed results. Vector Auto Regression (VAR) model [8] was developed to estimate queue lengths in individual lanes for mixed traffic typical of developing countries. The model uses Passenger Car Units to account for different vehicle types and includes lane-changing behavior analysis. Using drone-collected data and time series analysis, the model achieved high accuracy (R-squared = 0.97, MAPE = 21.55%) by identifying arrival flow, discharge flow, and lane changes as key factors with a three-time lag dependency according to the authors. However, the model had limitations, including poor performance in zero-queue situations, occasional negative estimates, reduced accuracy for queues over 50 meters due to lane-changing effects, and signal timing proved unsuitable as an input variable. Integration of multiple data sources has emerged as a sophisticated approach to queue length estimation.

The recent increase in the deployment of License Plate Recognition (LPR) detectors has enabled the utilization of their data for more advanced applications, such as calculating queue lengths at intersections, inferring vehicular trajectories, and estimating overall traffic conditions and emissions [9]. Liu et al. [10] developed a Random Forest (RF) based real-time queue length estimation method utilizing Global Position System (GPS) and LPR data. GPS data provided vehicle stop positions, while the RF model was trained to predict these stopping positions using traffic flow features extracted from LPR data. The estimated stopping positions were also used to calculate the cyclic maximum length for each approach lane. Similarly, Zhan et al. [11] proposed a lane-based queue length estimation model utilizing LPR data. The model incorporates a Gauss-based interpolation method for each lane to reconstruct missing information for unrecognized or mismatched vehicles. A car tracking-based simulation is then applied to estimate real-time queue lengths for each lane, using vehicle arrival and departure information.

Advanced mathematical modeling approaches have been developed to leverage vehicle trajectory data. One study [12] focused on real-time queue length estimation at signalized intersections using vehicle trajectory data, employing shock wave theory to model the dynamic queue forming and dissipating processes without relying on signal timing, arrival patterns, or penetration rates. By identifying inflection points in vehicle trajectories, the method estimates stopping and discharging shock waves, providing robust queue length estimates under various conditions. However, it cannot estimate queue lengths without probe vehicles, and faces challenges when upstream disruptions affect traffic flow.

Al Okaishi et al. [13] proposed a queuing system consisting of two main steps: detecting queues using the square difference method to identify motion in target areas, followed by vehicle identification using the Single Shot Detector (SSD) algorithm when no movement is detected. The main limitation of this model is its instability; the queue length resets to zero with any detected motion at the front of the queue, necessitating a sufficient transition time to ensure continuous vehicle movement.

Conventional vehicle queue length detection methods typically depend on static sensors, manual counting, or basic algorithms that cannot provide the accuracy and real-time performance needed for contemporary traffic management systems. While existing approaches have shown promise, several limitations persist: sensor-based methods require expensive infrastructure and have limited coverage, existing computer vision approaches often lack real-time performance or comprehensive evaluation across diverse conditions, and most studies focus on either detection or tracking separately, without optimized integration.

This study addresses key gaps in the literature by introducing an accurate and cost-effective method for estimating traffic queue lengths using conventional traffic cameras. The proposed approach leverages a deep learning framework that combines YOLO for real-time vehicle detection with SORT for robust object tracking. By processing video data from existing camera infrastructure, the system enables real-time vehicle counting, speed estimation, and dynamic queue length measurement. This integrated solution provides essential insights for efficient traffic monitoring and intelligent traffic management. The results demonstrate the system's effectiveness and practical viability for real-world traffic management applications. The study begins with vehicle classification—categorizing vehicles into cars, motorcycles, trucks, buses, and bicycles—using labeled visual datasets, followed by queue state detection. The paper is organized as follows: Section 2 presents the methodologies for speed estimation and queue length detection. Section 3 details and discusses the experimental results. Finally, Section 4 concludes the paper by summarizing the key findings, highlighting their implications for traffic management, and proposing directions for future research.

2 Methodology

This study presents a comprehensive real-time queue detection system that integrates deep learning-based vehicle detection, multi-object tracking, and speed estimation algorithms. The proposed methodology consists of six main stages: video stream processing, frame extraction, vehicle detection and classification, lane-based tracking, speed estimation, and queue state analysis. The system utilizes YOLOv5 for object detection, SORT algorithm for vehicle tracking, and implements a rolling window approach for queue state determination based on estimated vehicle speeds.

The methodology begins with video stream processing and frame extraction, followed by lane boundary definition using predefined reference points. Subsequently, vehicles are detected and classified using the YOLOv5 model, with each vehicle assigned a unique identifier through the SORT tracking algorithm. The system calculates individual vehicles' speeds using camera calibration and Euclidean distance measurements, finally determining queue states through average speed analysis within a rolling window framework. Average speeds of 20 km/h or lower indicate heavy traffic; speeds between 20 km/h and 55 km/h (inclusive) represent stable flow; and speeds above 55 km/h indicate free flow conditions. The Figure 1 illustrates the flowchart of the proposed queue state detection system.

2.1 Vehicle Speed Detection

YOLOv5m model [14] for robust vehicle detection and classification in the study. The model was trained on a comprehensive dataset combining multiple sources to achieve optimal performance in various traffic conditions. In our previous study [15], an initial dataset was created using several images from two different datasets [16], [17], targeting vehicle class detection and lane-based counting. The dataset was subsequently augmented with additional images from another source [18]. The model successfully identified five distinct vehicle classes: car, motorcycle, truck, bus, and bicycle. Performance evaluation across three different training configurations is presented in Table 1 with optimal results achieved at 150 epochs, demonstrating the model's effectiveness in multi-class vehicle recognition under diverse traffic scenarios.

Following vehicle classification, the system implements lane-based vehicle tracking using the SORT algorithm [19]. SORT employs a Kalman filter for state estimation, using a constant velocity motion model, and associates detections to tracks via

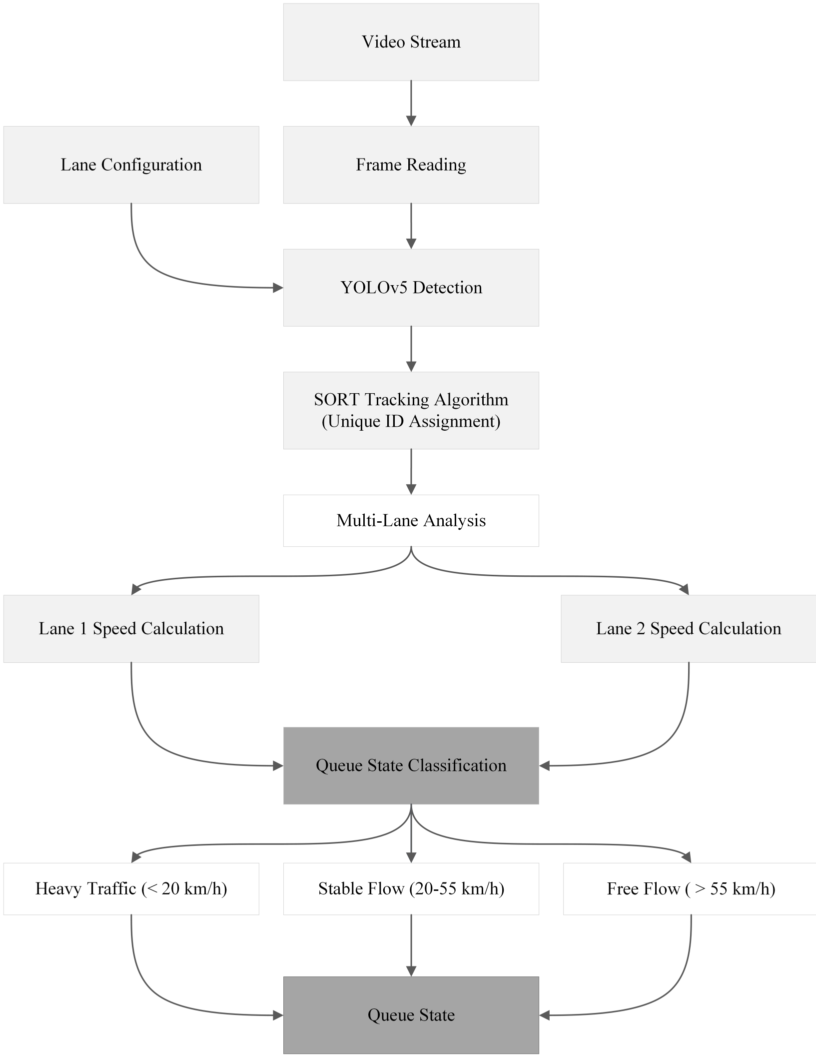

Figure 1: Flowchart of the proposed queue state detection system

Table 1: Performance results from the training dataset

Epoch	Precision	Recall	mAP
80	0.86	0.91	0.93
100	0.94	0.85	0.92
150	0.86	0.87	0.93

the Hungarian algorithm with an intersection over union (IoU) threshold. Each detected vehicle receives a unique identifier, enabling continuous tracking across frames.

Speed estimation plays a crucial role in queue state detection. As illustrated in Figure 2, several key parameters are extracted before speed calculation. These include the initial and final center point coordinates of the vehicle within the region of interest, the pixel-based length of the vehicle at its initial and final positions (with only the initial length used in the computation), and the timestamps of the vehicle's first and last detection in the video. Using these parameters, camera calibration—based on the method by [20]—is applied to convert pixel measurements into real-world distance, enabling the calculation of the actual distance traveled by the vehicle in meters. This calibrated distance, combined with the time difference, is then used to estimate the vehicle's speed.

$$k = \frac{L_{actual}}{L_{pixel}} \tag{1}$$

where k represents the camera calibration factor, L_{actual} is the real-world vehicle length (meters), and L_{pixels} is the pixel-based vehicle lengths. The system utilizes class-specific average vehicle lengths to enhance calibration accuracy across different

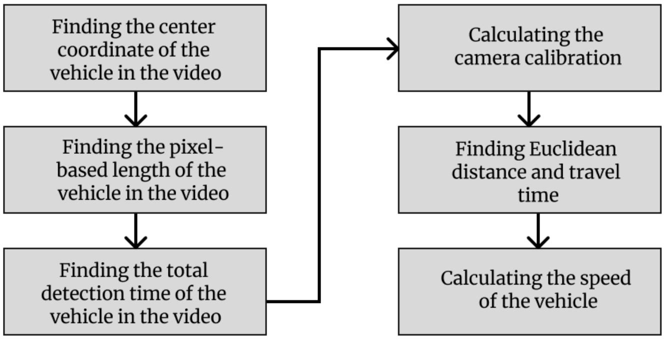

Figure 2: Vehicle speed detection diagram

vehicle types. In the study, the assumed average lengths were 4.5 meters for cars, 8.1 meters for trucks, 9.5 meters for buses [21].

$$d_{pixel} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

Once the camera calibration is computed, the initial and final center point coordinates are used to calculate the distance the vehicle travels throughout the video. The Euclidean formula in Equation (2) was used to find this length. Since the result of this calculation is in pixels, it is converted to meters by multiplying it by the camera calibration as given in Equation (3).

$$d_{meter} = k \times d_{pixel} \tag{3}$$

In the speed estimation phase, the time taken by the vehicle is as important as the distance taken. During this period, the total movement time is calculated as the difference between the timestamps of the vehicle's initial and final detections as given in Equation (4).

$$t = t_2 - t_1 \tag{4}$$

After computing all necessary parameters, the estimated speed is determined by dividing the distance traveled by the elapsed time. Since the initial speed is calculated in meters per second (m/s), it is converted to kilometers per hour (km/h) by multiplying by 3.6 as given in Equation (5).

$$v = (\frac{d_{meter}}{t} \times 3.6) \tag{5}$$

Table 2: Results on speed detection obtained from a dataset consisting of videos recorded by the authors

Video	Actual Speed (km/h)	Estimated Speed (km/h)	Accuracy
Vehicle-1	30	30.83	%97.23
Vehicle-2	50	46.51	%93.02
Vehicle-3	70	70.60	%99.14
Vehicle-4	100	89.11	%89.11

The speed estimations were validated using recorded videos in which the actual speeds of the vehicles were known. These estimated speeds, measured in kilometers per hour, are compared with the actual speeds in Table 2. For instance, a vehicle traveling at an actual speed of 30 km/h was estimated to be moving at 30.83 km/h. Additional tests demonstrated accuracy rates of 97.23%, 93.02%, 99.14%, and 89.11%, respectively.

2.2 Queue State Detection

The approximate speeds of each vehicle are calculated and recorded for queue state detection. The traffic state $Q(t)$ is determined based on the average speed of vehicles in the detection zone and is defined in Equation (6). If fewer than three vehicles are detected (or no vehicles are present), insufficient data is available for reliable queue state assessment, and the system defaults to classifying the traffic state as free flow.

$$Q(t) = \begin{cases} \text{Heavy traffic}, & \text{if } 0 \text{ km/h} \leq \bar{v}(t) \leq 20 \text{ km/h} \\ \text{Stable flow}, & \text{if } 20 \text{ km/h} < \bar{v}(t) \leq 55 \text{ km/h} \\ \text{Free flow}, & \text{if } 55 \text{ km/h} < \bar{v}(t) \end{cases} \tag{6}$$

where $\bar{v}(t)$ denotes the average speed of detected vehicles at time t.

After vehicle detection, counting, and speed estimation, the system proceeds to classify the traffic queue state. This process employs a rolling window approach, where the average speed of the three most recently detected four-wheeled vehicles is calculated as given in Equation (7) to determine the queue state. The window then shifts by excluding the earliest detected vehicle and incorporating the next one, calculating the average speed of the updated set (i.e., the second, third, and fourth vehicles) to reassess the queue condition as given in Equation (8).

$$\bar{v}(t) = \frac{1}{3}\sum_{i=1}^{3} v_i \tag{7}$$

$$\bar{v}(t+1) = \frac{1}{3}\sum_{i=2}^{4} v_i \tag{8}$$

Two-wheeled vehicles like motorcycles and bicycles are not included in traffic queue analysis for both theoretical and practical considerations. Unlike larger vehicles, these two-wheelers follow different speed and traffic flow patterns and use lanes differently. They can weave between stopped cars through lane-splitting and filtering, which means they don't follow the standard first-come, first-served queue rules that apply to cars and trucks. By leaving out motorcycles and bicycles from queue analysis, researchers can better measure the actual congestion levels that impact the main flow of traffic. The system identifies traffic queues by calculating the average speed of vehicles traveling in specific lanes and uses this data to assess queuing conditions. For all these reasons, two-wheeled vehicles were not included in the queue state analysis in the study.

3 Experimental Results

The comprehensive evaluation of the proposed traffic queue detection system was conducted across 11 video sequences, encompassing diverse traffic scenarios ranging from free-flowing conditions to severe congestion. The evaluation framework employs multiple performance metrics to assess system effectiveness across three primary dimensions: computational efficiency, detection accuracy, and queue classification performance.

Table 3: Dataset Specifications and Performance Metrics

Video	Resolution	Duration(s)	FPS	YOLO Detection(ms)	SORT Tracking(ms)
Video-1	1080x1920	127.1	14.36	44.47	0.63
Video-2	1080x1920	28.4	14.39	44.58	0.68
Video-3	480x848	43.6	21.81	41.34	0.40
Video-4	480x848	78.4	21.42	42.14	0.43
Video-5	480x848	76.9	21.70	41.62	0.40
Video-6	1080x1920	51.6	14.59	43.5	0.40
Video-7	1080x1920	74.8	14.56	43.24	0.38
Video-8	1080x1920	43.2	14.45	44.33	0.88
Video-9	1080x1920	84	13.99	45.88	0.53
Video-10	1080x1920	51	14.65	44.37	0.63
Video-11	1080x1920	45	14.78	45.92	0.52

As presented in Table 3, the experimental dataset consists of 11 videos with varying specifications to evaluate the robustness of the system under different recording conditions. The dataset includes both high-resolution videos (1080×1920 resolution) and low-resolution videos (480×848 resolution), with durations ranging from 28.4 to 127.1 seconds. Frame rates varied between 13.99 and 21.81 FPS, providing comprehensive coverage of typical traffic surveillance scenarios. The computational performance analysis reveals optimal resource allocation, with YOLOv5 detection requiring an average of 44.2 ms per frame, while SORT tracking operations consumed only 0.52 ms on average. This distribution demonstrates that 98.8% of computational time is dedicated to object detection, 1.2% to tracking operations, indicating highly efficient algorithm integration. The system successfully processed videos at varying frame rates while maintaining consistent detection performance.

Equations (9) and (10) are used to calculate the accuracy of the estimated number of vehicles and speed calculation in vehicle counting [22].

$$Error(\%) = \left(\frac{|Detected\ Number - Actual\ Number|}{Actual\ Number}\right) \times 100 \tag{9}$$

$$Accuracy(\%) = 100 - Error(\%) \tag{10}$$

The SORT algorithm successfully tracked a total of 441 vehicles across all video streams, with individual video performance ranging from 15 to 84 vehicles. The system demonstrated consistent tracking capabilities across varying traffic densities, with lane-specific vehicle counts providing granular insights into traffic distribution patterns. In particular, the left-right lane distribution analysis revealed traffic flow imbalances in several scenarios, such as video 1 and video 9, indicating asymmetric traffic patterns commonly observed in real-world scenarios. The results demonstrate high accuracy in vehicle counting, generally achieving more than 90%. According to the Table 4, 45 vehicles were detected, although only 42 vehicles actually passed through the area in video 8. The system may overestimate the number of vehicles present in reality, as shown in video 8. Possible reasons for this include occlusion, where a vehicle in the frame is temporarily blocked by another vehicle; when the occluded vehicle reappears, the system fails to recognize it as the same vehicle and instead detects it as a new one. Additionally, factors such as lighting conditions, viewing angles, and partial visibility may prevent the correct re-identification of previously detected vehicles. Such issues (ID reassignments and challenges in multi-object trajectory tracking) are commonly reported in the literature [23].

Table 4: Results regarding vehicle counting

Video	Number of Detected Vehicles		Number of Actual Vehicles		Accuracy (%)	
	Right	Left	Right	Left	Right	Left
Video-1	38	46	37	47	97.29	97.87
Video-2	6	9	6	9	100	100
Video-3	14	8	14	9	100	88.88
Video-4	16	11	19	14	84.21	78.57
Video-5	16	21	17	21	94.11	100
Video-6	20	24	19	24	94.70	100
Video-7	23	29	22	30	95.65	96.67
Video-8	20	25	17	25	82.35	100
Video-9	21	30	22	30	95.45	100
Video-10	15	20	15	20	100	100
Video-11	18	10	19	11	94.73	90.90

Traffic flow analysis results are illustrated in Figure 3 from 11 test videos processed using the YOLOv5+SORT algorithm. The histogram of overall vehicle speed distribution across all videos, with queue classification thresholds indicated that heavy traffic threshold at 20 km/h and a stable flow threshold at 55 km/h, showing a normal distribution with a mean speed of 62.5 km/h and a standard deviation of 22.51 km/h. The distribution demonstrates the system's capability to capture the full spectrum of urban traffic conditions. The comparative box plot analysis of speed distributions between the left lane and right lane illustrates typical traffic flow patterns where the left lane maintains higher average speeds. The queue state classification results are presented as a pie chart, demonstrating system accuracy with balanced detection across three categories: heavy traffic (13.6%, 3 of 22 lanes), stable flow (18.19%, 4 of 22 lanes), and free flow (68.2%, 15 of 22 lanes).

Table 5: Vehicle Speed Analysis and Queue Detection Results by Video

Video	Right Lane Vehicles	Left Lane Vehicles	Right Avg Speed (km/h)	Left Avg Speed (km/h)	Right Queue State	Left Queue State
Video-1	38	46	55.76	68.59	Free flow	Free flow
Video-2	6	9	18.42	18.9	Heavy traffic	Heavy traffic
Video-3	14	8	62.24	93.65	Free flow	Free flow
Video-4	16	11	61.15	84.73	Free flow	Free flow
Video-5	16	21	58.37	62.94	Free flow	Free flow
Video-6	20	24	65.67	84.33	Free flow	Free flow
Video-7	23	29	64.41	79.12	Free flow	Free flow
Video-8	20	25	55.15	72.27	Free flow	Free flow
Video-9	21	30	52.41	54.66	Stable flow	Stable flow
Video-10	15	20	72.87	85.63	Free flow	Free flow
Video-11	18	10	29.62	12.04	Stable flow	Heavy traffic

The comprehensive speed analysis and queue state detection results are summarized in Table 5, showcasing the system's capability to perform real-time queue state classification. The speed values given in the table were determined by taking the average speed of all vehicles detected on a lane basis in a video. The three-category classification system successfully categorized traffic states. In video 1, the SORT algorithm successfully tracked 46 vehicles in the left lane and 38 in the right lane, yielding average speeds of 55.76 km/h and 68.59 km/h, respectively. Based on the established threshold criteria (speed $>$ 55 km/h), the queue detection result was classified as free flow. Video 2 demonstrated clear congestion conditions with average speeds of 18.42 km/h (right lane) and 18.9 km/h (left lane), correctly classified as heavy traffic for both lanes. Video 11 showed mixed conditions with the left lane experiencing heavy traffic (12.04 km/h) while the right lane exhibited stable flow (29.62 km/h). Multiple videos (video 1, video 3, video 4, video 5, video 6, video 7, video 8, video 10) exhibited high-speed conditions with average speeds exceeding 55 km/h, correctly classified as free flow. Notable examples include video 3 with exceptionally high speeds (62.24 km/h right, 93.65 km/h left) and video 10 (72.87 km/h right, 85.63 km/h left). Video 9 demonstrated intermediate traffic conditions with speeds of 52.41 km/h (right) and 54.66 km/h (left), appropriately classified as stable flow for both lanes.

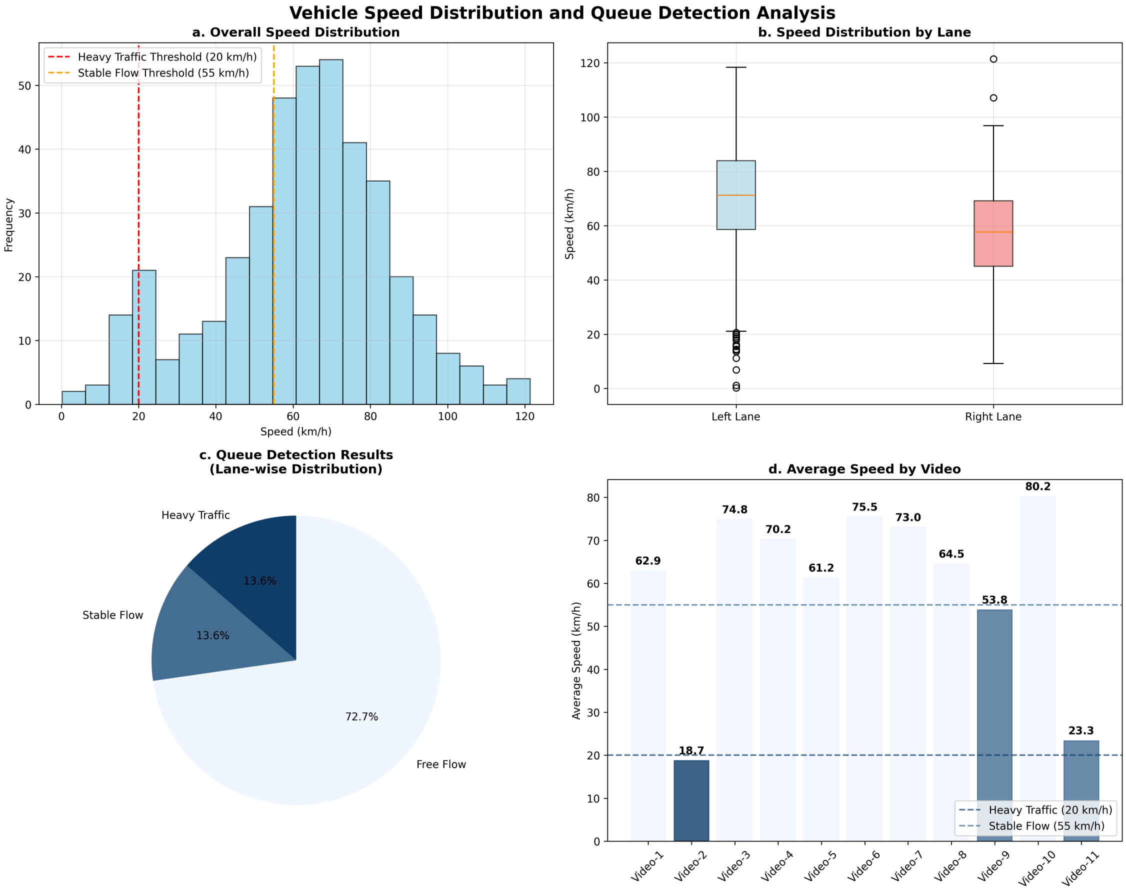

Figure 3: Vehicle Speed Distribution and Queue Detection Analysis

4 Conclusion

This study presents a validated, comprehensive real-time vehicle queue state detection system that integrates state-of-the-art deep learning methods with practical traffic management requirements. The proposed approach, which combines YOLOv5 for object detection with the SORT tracking algorithm, demonstrates exceptional performance across diverse traffic scenarios while maintaining the computational efficiency required for real-world deployment. The primary contributions of this work span several critical aspects of modern traffic monitoring systems. First, we introduce an integrated deep learning framework that seamlessly combines object detection and tracking components, specifically optimized for traffic queue analysis. Second, we propose a robust queue state classification methodology that accurately categorizes traffic conditions based on vehicle speed dynamics. Comprehensive experimental validation was conducted across multiple datasets, encompassing both high-resolution (1080×1920) and low-resolution (480×848) video streams, demonstrating the system's versatility and reliability under varying input conditions. The processing pipeline achieves efficient resource utilization: YOLOv5-based detection averages 44.2 ms per frame, while SORT tracking requires only 0.52 ms per frame. This results in a computational distribution where 98.8% of processing time is devoted to detection and just 1.2% to tracking, highlighting the efficiency of the overall algorithm design. Memory usage averaged 1050 MB across all test scenarios, confirming the system's feasibility for deployment in edge computing environments with limited computational resources. These characteristics make the proposed system well-suited for scalable, real-time traffic monitoring in urban and highway settings.

While the current implementation demonstrates robust performance across the evaluated scenarios, several areas for future studies remain. Although the dataset is comprehensive within its scope, it could be expanded to include adverse weather conditions, varying lighting environments, and diverse intersection geometries to further enhance the system's robustness and generalizability. Incorporating multiple camera viewpoints represents a promising direction, as it could enable more comprehensive spatial coverage and improve detection and tracking accuracy in complex, occlusion-prone traffic environments. Future studies should also focus on developing lightweight model architectures—such as through model pruning, quantization, or efficient network design—to reduce computational demands without compromising detection accuracy, thereby enabling deployment on lower-power edge devices. Additionally, integrating vehicle trajectory prediction and behavioral modeling could significantly enhance the system's intelligence, allowing it to anticipate traffic dynamics and support proactive traffic signal optimization. Such advancements would move the system beyond reactive monitoring toward adaptive, predictive traffic management.

Acknowledgments
This study was supported by the Research Fund of Yalova University (Project Number: 2023/AP/0002).

Authors' Contributions
In this study, AB, YEA, and AT proposed the study idea together. AB and YEA carried out the experimental studies, simulations, and wrote the initial draft. AT contributed to the study by supervising and interpreting. AB, YEA, and AT wrote the final version of the paper. All authors read and approved the final version of the paper.

Competing Interests
The authors declare that they have no conflict of interest.

References
[1] S. Lee, K. Xie, D. Ngoduy, and M. Keyvan-Ekbatani, "An advanced deep learning approach to real-time estimation of lane-based queue lengths at a signalized junction," *Transportation research part C: emerging technologies*, vol. 109, pp. 117–136, 2019.
[2] R. Rahman and S. Hasan, "Real-time signal queue length prediction using long short-term memory neural network," *Neural Computing and Applications*, vol. 33, pp. 3311–3324, 2021.
[3] M. Umair, M. U. Farooq, R. H. Raza, Q. Chen, and B. Abdulhai, "Efficient video-based vehicle queue length estimation using computer vision and deep learning for an urban traffic scenario," *Processes*, vol. 9, no. 10, p. 1786, 2021.
[4] J. Wu, H. Xu, Y. Zhang, Y. Tian, and X. Song, "Real-time queue length detection with roadside lidar data," *Sensors*, vol. 20, no. 8, p. 2342, 2020.
[5] Y. Zhao, J. Zheng, W. Wong, X. Wang, Y. Meng, and H. X. Liu, "Various methods for queue length and traffic volume estimation using probe vehicle trajectories," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 70–91, 2019.
[6] G. Comert, T. Amdeberhan, N. Begashaw, N. G. Medhin, and M. Chowdhury, "Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals: A combinatorial approach for nonparametric models," *Expert Systems with Applications*, vol. 252, p. 124076, 2024.
[7] Q. Zhou, R. Mohammadi, W. Zhao, K. Zhang, L. Zhang, Y. Wang, C. Roncoli, and S. Hu, "Queue profile identification at signalized intersections with high-resolution data from drones," in *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 1–6, IEEE, 2021.
[8] S. Jayatilleke, V. Wickramasinghe, and N. Amarasingha, "Introduction of a simple estimation method for lane-based queue lengths with lane-changing movements," *Journal of The Institution of Engineers (India): Series A*, vol. 104, no. 1, pp. 143–153, 2023.
[9] M. Y. Arafat, A. S. M. Khairuddin, U. Khairuddin, and R. Paramesran, "Systematic review on vehicular licence plate recognition framework in intelligent transport systems," *IET intelligent transport systems*, vol. 13, no. 5, pp. 745–755, 2019.
[10] D. Liu, C. An, M. Yasir, J. Lu, and J. Xia, "A machine learning based method for real-time queue length estimation using license plate recognition and gps trajectory data," *KSCE Journal of Civil Engineering*, vol. 26, no. 5, pp. 2408–2419, 2022.
[11] X. Zhan, R. Li, and S. V. Ukkusuri, "Lane-based real-time queue length estimation using license plate recognition data," *Transportation Research Part C: Emerging Technologies*, vol. 57, pp. 85–102, 2015.
[12] P. Pudasaini, A. Karimpour, and Y.-J. Wu, "Real-time queue length estimation for signalized intersections using single-channel advance detector data," *Transportation research record*, vol. 2677, no. 7, pp. 144–156, 2023.
[13] W. Al Okaishi, A. Zaarane, I. Slimani, I. Atouf, and M. Benrabh, "A vehicular queue length measurement system in real-time based on ssd network," *Transport and Telecommunication*, vol. 22, no. 1, pp. 29–38, 2021.
[14] Ultralytics, "Yolov5." https://github.com/ultralytics/yolov5?tab=readme-ov-file. (Accessed: 2023-02-10).
[15] A. Battal, Y. Avci, and A. Tuncer, "Vehicle detection and counting in traffic videos using deep learning," *ICENTE'23*, p. 272, 2023.
[16] M. Boneh, "Vehicle detection." https://github.com/MaryamBoneh/VehicleDetection/tree/main/Dataset. (Accessed: 2023-02-12).
[17] L. Soetanto, "Vehicle detection dataset." https://www.kaggle.com/datasets/lyensoetanto/vehicle-images-dataset. (Accessed: 2023-02-21).
[18] M. TÁRNOK, "5 vehichles for classification." https://www.kaggle.com/datasets/mrtontrnok/5-vehichles-for-multicategory-classification. (Accessed: 2024-06-06).
[19] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, Ieee, 2016.
[20] P. K. Thadagoppula and V. Upadhyaya, "Speed detection using image processing," in *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 11–16, IEEE, 2016.
[21] C.-J. Lin, S.-Y. Jeng, and H.-W. Lioa, "A real-time vehicle counting, speed estimation, and classification system based on virtual detection zone and yolo," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 1577614, 2021.
[22] A. M. Santos, C. J. Bastos-Filho, A. M. Maciel, and E. Lima, "Counting vehicle with high-precision in brazilian roads using yolov3 and deep sort," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 69–76, IEEE, 2020.
[23] Y. Zhao, X. Zhou, X. Xu, Z. Jiang, F. Cheng, J. Tang, and Y. Shen, "A novel vehicle tracking id switches algorithm for driving recording sensors," *Sensors*, vol. 20, no. 13, p. 3638, 2020.