

Real-Time Urban Vehicle Tracking via Wireless Sensor Networks Using Kalman Filter and Random Forest Prediction

T Vairam

PSG College of Technology

S. Venkatesan

profsvenkatesan@gmail.com

Wachemo University

Article

Keywords: Multi-vehicle tracking, Kalman Filter, Wireless sensor networks, Random Forest motion prediction, Adaptive Q-matrix, YOLOv8 detection

Posted Date: August 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10163526/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Real-Time Urban Vehicle Tracking via Wireless Sensor Networks Using Kalman Filter and Random Forest Prediction

¹Dr T Vairam, Assistant Professor, Department of Information Technology, PSG College of Technology, tv.m.it@psgtech.ac.in

²S. Venkatesan, School of Mechanical Engineering, College of Engineering and Technology, Wachemo University, Hosaena, Ethiopia. profsvenkatesan@gmail.com

Corresponding Author: S. Venkatesan, School of Mechanical Engineering, College of Engineering and Technology, Wachemo University, Hosaena, Ethiopia. profsvenkatesan@gmail.com

Abstract

This study proposes a real-time multi-vehicle tracking framework for urban surveillance environments, which was aimed at urban traffic surveillance and was expected to be deployed on wireless sensor networks (WSNs) with CCTV feeds. The proposed framework addresses key challenges in dynamic vehicle tracking, including non-linear motion, sudden changes in trajectory, occlusion, and measurement noise in the urban environments. The tracking framework employs a Kalman Filter (KF) to estimate the positions and the velocities of the vehicle, and tuning the Q-matrix of the filter was designed in such a way that it dynamically adjusts process noise covariance with the perceived current characteristics of movement and the residual errors. Such a responsive strategy greatly advances the filter sensitivity to the dynamic behavior of the traffic and trajectory stability. To continue enhancing motion prediction, a random forest regressor of historical vehicle tracks was trained so the correction of KF motion prediction can be achieved data-driven. This combination strategy was useful in filling the gap that existed between the model-based and machine learning approaches and thus providing higher levels of accuracy and preservation of identity in successive frames. The YOLOv8 model is used to conduct real-time object detection, and the model allows getting strong input of boundary boxes even in congested traffic and under different conditions of light. The system was tested on publicly available data and private CCTV streams, and their performance was measured by the standard MOT measures such as MOTA, MOTP, IDF1, and ID switches. Wireless sensor network environments are optimized to be able to deal with scalable and low-latency communication between edge devices and central processors. These reports present experimental findings of high tracking accuracy, identity stability, and real-time performance that prove the applicability of the proposed system to the domain of intelligent transportation systems and smart urban infrastructure. A moving confidence-sensible merger system is delineated, wherein Random Forest estimations rectify KF accounts through residual-based weights on time and detection validity hints. The system also achieves real-time inference with the average latency being 71 ms/frame and is capable of supporting scalable WSN deployments in medium packet loss conditions.

Keywords - Multi-vehicle tracking, Kalman Filter, Wireless sensor networks, Random Forest motion prediction, Adaptive Q-matrix, YOLOv8 detection.

Introduction

In contemporary surveillance systems used in managing traffic and pseudo-autonomous systems, object detection was a critical point, especially in the urban setting, where it was necessary to observe the measurements in real-time as a means of road safety, congestion, and general control [1]. Over the last few years, object detection has been improved with the help of deep learning, in particular, such models as YOLO (You Only Look Once), SSD (Single Shot Detector), and Faster R-CNN. Such models have proven very accurate and fast in detecting vehicles, pedestrians, and other road users in complex scenes. Due to the variable lighting conditions, occlusion, poor-resolution video, and dynamic backgrounds, CCTV-based surveillance creates some unusual difficulties [2]. In spite of these limitations, deep convolutional neural networks (CNNs) have been successfully used to learn spatial hierarchies and to learn object detection in crowded cities. Scientists have also come up with dedicated datasets such as KITTI, Urban Tracker, and the MOT Challenge to provide a benchmarking measure in real-traffic situations. Such data provide improvement in detection models and their generalization in different environments. Correct object detection brings out the baseline paradigm of multi-vehicle tracking given the fact that the detection of vehicle instances has a direct relationship to the success of tracking algorithms in preserving vehicle identity and motion in subsequent frames in a video sequence [3]. Conventional multi-object tracking methods such as the Kalman Filter (KF) have been widely used because of their computational efficiency and real-time capability. Nevertheless, these approaches tend to fail quite harshly when faced with sudden changes of motions and occlusions in heavy urban districts. More recent developments have seen the introduction of tracking algorithms based on deep learning, such as Deep SORT, Tracktor++, and FairMOT, which combine appearance features with motion information to enhance identity preservation and tracking accuracy [4]. Such models are highly dependent on precise object detection algorithms and data association techniques that are able to sustain a consistent object's identity, especially in scenes that have congestion and overlapping vehicles. Although successful, deep learning methods are usually resource-hungry and are not very well-suited to real-time systems. A third option has developed the introduction of hybrid tracking frameworks consisting of the combination of traditional filters and machine learning in order to achieve a balance between performance and efficiency [5]. With the growth in the complexity of multiple-vehicle tracking in dynamic/noisy situations, there was a heavy use of adaptive filter/noise modeling methods due to the uncertainties in the motion and measurement data. Conventional methods of filtering are based on a steady-state process noise covariance (Q-matrix) of the Kalman Filter and the Kalman Filter (KF) and are therefore rather ineffective when abrupt changes in the motion of the tracked objects occur or in situations where the noise in the environment changes with time [6]. Methods of adaptive filtering have been developed to overcome these shortcomings; these adaptive filtering methods modify the Q-matrix through the patterns of observed motion or the residuals of the measurements. This versatility improves the capability of the filter in reacting to nonlinear paths, abrupt actions, and sensor errors [7]. Some researchers have shown that Q-matrix tuning using vehicle acceleration, velocity variance, or context benefited greatly in tracking stability and minimized estimation error. Adaptive filtering systems usually also have machine learning elements to approach true time estimation of noise properties. Although adaptive filtering works well to make the tracking system more responsive to dynamic noise and motion changes, machine learning in motion prediction fits better in improving accuracy because of learning hard-to- Another advantage of integrating machine learning in the

motion prediction was to improve the accuracy by correcting residual errors in real-time [8]. Conventional physics-based models tend to lack when it comes to anomalous or abrupt deviations in the motion and particularly in the urban traffic situations where there are thousands of unpredictable factors affecting the vehicle dynamics. Tree-based models, such as random forests and ensemble regressors, have been extremely popular applications of machine learning models, as they are relatively interpretable, fast, and not especially sensitive to high-dimensional feature spaces, requiring minimal preprocessing. Such models are able to use previous trajectory information to estimate future positions to increase tracking systems accuracy when implemented with filtering theory like the Kalman Filter [9]. Conversely, multiple deep learning models (Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) models, among others) have been shown to provide better time modeling of sequential data at the expense of computational complexity and model interpretability. New studies point out that a better performance of hybrid systems that integrate machine learning with classical filters are more effective than independent models because they rectify the prediction residuals and cope with a changing environment [10]. Since machine learning models have predictive performance even in the case of data-limited environments, applying traditional filtering methods alongside pre-trained machine learning models gives the opportunity to create hybrid tracking frameworks, where the machine learning models can provide more reliable and precise motion tracking due to the combination of data-driven insight and probabilistic estimations [11]. The Kalman Filter (KF) is one of the most widely used state-estimation techniques for real-time tracking applications, and they offer effective, mathematically based techniques to solve real-time state-estimation problems. Usually depend on preset models of the motion and preset assumptions of noise and lack the ability to adjust to nonlinear and unpredictable behaviors. So, to overcome these issues, researchers have incorporated machine learning models, i.e., random forests, support vector machines, and LSTM networks, into the tracking pipeline to learn the motion patterns, rectify some errors in the predictions, and adjust dynamically to the altered circumstances [12]. Such hybrid models combine the powers of the two realms: data-driven probabilistic inference of the filters and online predictive abilities of ML. It has been proved that these combinations lead to significant accuracy, occlusion, and sensor noise robustness. In addition, lightweight ML can be distributed in real-time, not to say that lightweight ML does not require any overhead related to deep learning [13]. The current adoption of hybrid tracking frameworks to real-time video analytics platforms facilitates smart transportation, which is able to efficiently consume live video surveillance data streams and ensure high precision and flexibility even in complicated traffic conditions. As CCTV networks increase and edge computing evolves, real-time processing of video feeds has added potential in real-time traffic management, incident identification, and vehicle dynamics monitoring [14]. Object detection, object tracking, and anomaly identification are delivered directly on the video stream so that the system can generate useful information in real time. The common thing in these systems was the use of machine learning and computer vision algorithms in the identification of vehicles and categorization of moves, as well as congestion levels. Nevertheless, a key obstacle was to assure latency and accuracy (at different levels) in the face of changes in lighting and weather as well as traffic. There has been growing interest in the optimization of computational pipelines, the use of CPU- or GPU-acceleration, and running lightweight models in edge devices to satisfy real-time demands. It was also integrated with IoT infrastructure and smart city platforms that increased

scalability and responsiveness [15]. In order to meticulously comprehend the adequacy and dependability of real-time video analytics in intelligent transportation systems, the ultimate evaluation parameters and standardized benchmark operations should be available to measure the ability of multi-object tracking across multifarious operating environments. Popular measures include the Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), ID F1 Score (IDF1), and ID switch frequency to measure the tracking performance along the dimensions of accuracy, identity consistency, and spatial precision [16]. MOTA uses the combination of false positives, missed targets, and identity switches (counted as a single score), and the localization of tracks was thematically inferred as the MOTP using predicted tracks. The IDF1 score measures the proficiency of identity conservation during tracking runs, the importance of which was greater in the case of densely populated or occluded spaces. Standardized video sequences and annotations can be found in publicly available datasets and challenges such as MOT16, MOT20, KITTI, and UA-DETRAC that promote consistent benchmarking research practices throughout the community [17]. Such datasets depict a variety of traffic and pedestrian conditions that allow testing the algorithms in a variety of conditions. Benchmarking not only enhances the validity of the algorithm but also the future course of action and challenges in enhanced MOT model algorithms that would lead to more accurate, real-time tracking of patrons of intelligent surveillance and transportation simulators [18]. The work advances a new form of fusion architecture of Kalman Filter (KF) and Random Forest using residual feedback. The system dynamically combines model predictions based on selective use of motion variance, confidence of detection, and filter residuals, so there is no longer naive averaging or side-by-side pipelines. In contrast to the literature above, which ignores communication or system latency, the study describes the detailed performance results, such as latency in frame transmission/reception, throughput, CPU/GPU load situation, and WSN packet degradation.

Research Gap

The existing methods of tracking multiple vehicles in CCTV urban systems either have deep learning models with a high computation burden or adopt a conventional one that is ineffective against complex motion and measurement noise. Most hybrid models do not have real-time adaptivity or do not involve dynamic noise modeling and are thus relatively useless in the unpredictable traffic context. There was also a common division of functions that normally separates detection evaluation and tracking evaluation, thus ignoring the co-dependence between them. A single-platform, lightweight system, with a combination of adaptive filtering and explainable machine learning methods, that offers high-performance, real-time tracking with efficiency and stability in a variety of environments and traffic scenarios was required. Current hybrid tracking systems do not have the ability to adaptively couple, nor to do confidence-driven correction between classical filters and machine learning models. This is an issue that is catered to in this paper using a highly integrated fusion architecture.

Research Purpose

This study aims at coming up with a real-time multi-vehicle tracker on urban CCTV cameras through combining an Kalman Filter (KF) and adaptive Q-matrix, as well as Random Forest-based motion anticipation. Ideally, the system would improve inventory tracking and identity

preservation in dynamic environments that have occlusion, nonlinear motion, and detection noise. The model was made responsive to sudden shock by adaptively tuning the process noise as well as learning past trajectories. The hybrid method provides a low-latency and high-precision tracking that is applicable in wireless surveillance networks and intelligent transportation systems to help in improving urban traffic monitoring and the study of the behavior of vehicles.

Research Methodology

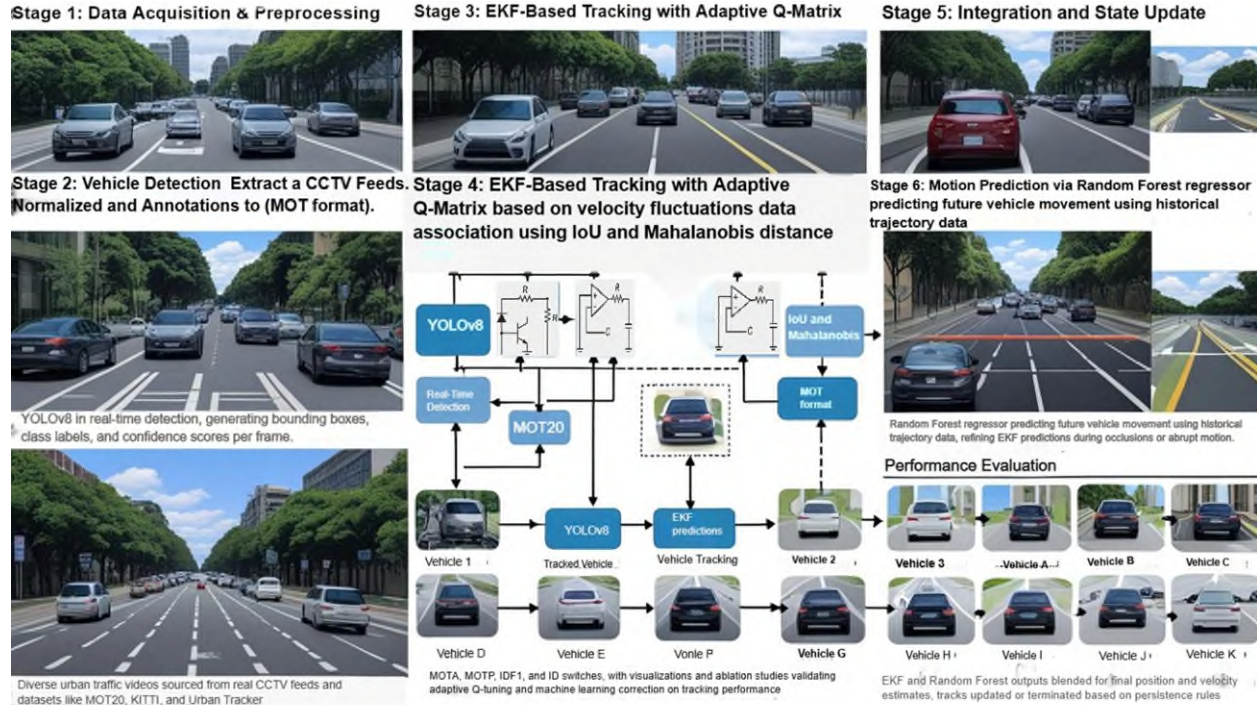


Figure 1. Methodology Flow chart

Data Acquisition and Preprocessing

The need to develop and assess real-time tracking systems on urban traffic called for a trustworthy source of annotated video data to be used in this work. To deal with it, one decided on such a dataset as MOT20, KITTI, and Urban Tracker datasets. MOT20 was primarily used for evaluating tracking robustness and identity preservation in crowded scenes, while KITTI and Urban Tracker provided vehicle-oriented tracking sequences. Different traffic densities, environmental conditions, and camera perspectives were encountered in these datasets, and this was representative of real-life surveillance scenarios. The selection made it possible to subject the tracking model to various issues, including obstruction, varying automobile velocities, and complexity of motion in the towns [19].

Frame-by-frame annotation in a standardized format was essential in training and benchmarking in each of the datasets. Video sequences obtained from these datasets were extracted and sampled and uniformly sampled in order to ensure a constant frame rate of 30 frames per second. This point enabled the ability to match up aspects of detection and tracking as well as ensure appropriate temporal spacing of motion prediction. Video frames were resized and normalized to a fixed

resolution so as to ensure that in the processing pipeline, there is consistency, and the type of data being fed to the detection model can rely on its pipeline being independent of the input dimensionality [20].

The datasets were annotated with object IDs, labels of the classes, coordinates of the bounding boxes, and timestamps. Such annotations have been parsed and reformatted into Multiple Object Tracking (MOT) format, which is a commonly reused tracking benchmark. The format allowed an easy integration with the YOLOv8 detection framework in a way that made the subsequent downstream process of matching detections with predicted tracks easier. The system allowed a consistent assessment of multiple datasets and standard error caused by incompatible labeling structures, as there was a standardized format used across the datasets [21].

Alongside openly available data, it was arranged to include live CCTV traffic data to allow the real-time deployment to be evaluated. Preprocessing of real-time feeds was handled by the real-time capture of the frames, gray-scale conversion (if needed), and resizing according to the dimensions of the trained model of detection. The compatibility of the preprocessing module in real time guaranteed a minimum of latency, which included the target goal of the system to conduct effective monitoring in an urban area containing multiple vehicles. All these preprocessing methods created a strong input layer of the detection and tracking architecture [22].

Vehicle Detection

Detection of the vehicles has been one of the key steps within the tracker system suggested in the paper that would allow precise localization of movable objects within CCTV frames in the city. This was done with the help of the YOLOv8 (You Only Look Once, version 8) detection model, which was characterized by the YOLOv8 provides efficient real-time object detection in dense traffic scenes. Its single-shot architecture in detection enabled the system to conduct classification and localization matches at the same time and was thus applicable in traffic surveillance tasks with many high-density objects of different speeds [23].

Surveillance datasets have been preprocessed with frames dropped to the YOLOv8 model, which returned the position of a bounding box that surrounded every car detected. The notion of a predicted object (e.g., car, truck, bus) was represented as a class label assigned to each detection, and the reliability of the prediction was described as the confidence score. Only the detections above a certain level of confidence were accepted, and this meant that the low and inaccurate predictions would not proceed through the system. This was a filtering step that provided a greater degree of accuracy of the tracking pipeline and reduced the amount of noise in the detection to be propagated to the lower modules [24].

The detector output was structured, organized into spatial and categorical matters of each vehicle within a particular frame. More precisely, the entries in the detection row contained the x and y coordinates of the center of the bounding box, the width and the height of the bounding box, and the object class. These values were encoded in a uniform format so that indeed they can be used without modifications as measurement inputs of the Kalman Filter (KF) tracking system. The system facilitated efficient communication between the tracking and the detection arms because the system was standard at many levels, including output [25].

In a traffic monitoring system, accuracy and latency of detection played a major role in determining the performance of the tracking. The choice of YOLOv8 guaranteed the possibility of detecting moving vehicles in real-time without affecting the detection fidelity. In addition, it could retain a high frame-per-second of processing, which contributed to its real-time demands on the system. By this potent detection feature, the appearance of vehicles in successive frames was confidently identified, and these served as the premise or foundation of subsequent temporal matching and derivation of vehicle trajectories in the part of tracking [26].

Kalman Filter-Based Multi-Vehicle Tracking with Adaptive Q-Matrix

The Kalman Filter (KF) framework was used to perform the multi-object tracking process; that is, given a sequence of observations made in time, this framework modeled the state of each vehicle over time. Each tracked object had a specified four-dimensional state vector as $X=[x, y, vx, vy]$, where x and y were the object position, and vx and vy were their horizontal and vertical velocity components. The KF worked by projecting the state ahead to the next state in terms of a motion model and then correcting the prediction using the measurements data that follows at the detection stage [27].

The nonlinearity of the vehicle dynamics and the suddenness of the fluctuations of the vehicle dynamics require that the KF implementation have an adaptive tuning of the process noise covariance matrix (Q-matrix). This matrix reflected the uncertainty of the motion model of the system and affected the sensitivity to abrupt changes of a filter much. Rather than assuming a constant assumption of noise, the Q-matrix was actually readjusted with every frame depending on velocity changes. This was adjusted and made such that the filter maintains stability at a steady pace and responds to sudden accelerations, lane shifts, or stops, which are common in an urban traffic pattern [28].

In multi-object tracking, to assure the identity consistency, it was important to have an ability to perform efficient data association. An ensemble gating scheme was used, where Intersection over Union (IoU) was used together with Mahalanobis distance to assign detected bounding boxes to predicted object states. IoU also provided spatial consistency of the predicted and the observed bounding boxes, and Mahalanobis distance took into consideration the variance of the prediction, which allowed probabilistic matching. These metrics combined lowerfalse association and ID switching, especially in cases of vehicles in overlapping and temporary blocking [29].

Tracks beginning Tracks were initiated when there was successive presence of a consistent detection across consecutive frames and were de-initiated when there was long-term occlusion, saliency, or mismatch. The KF estimated the trajectory of the vehicle when it was not detected in a few frames and enabled regularity of tracking despite interruption of visual identification. The combination of adaptive noise treatment and a resilient association system made the KF-based tracking system have a high level of accuracy and robustness in real-time urban monitoring activities, which forms the basis of enhanced motion forecasting in the next component [30].

Motion Prediction with Random Forest

A Random Forest regression model was deployed within the tracking system to facilitate motion prediction in order to improve the precision of the trajectory estimation process. Although the KF was used to address sequential estimation of the state, it could not be used to model non-linear trajectory/motion often associated with driving around cities (i.e., abrupt lane changes, tight turns, stop-start, etc.). The Random Forest model dealt with this by using previous tracking sequences as an input and learning the patterns in the motion of the vehicle that could not be learned using conventional filters [31].

Training of the model was done on collected historical trajectory data when it was being tracked. Position and velocity values of each tracked vehicle in the previous frames were selected and prepared in the form of feature vectors. These features to be fed as inputs were spatial coordinates, frame-wise changes in velocity, and inter-frame displacements. The labels during training were matching future states. This process of learning also enabled the model to find non-linear dependency as well as to predict abrupt changes and was thereby able to predict the behavior of the vehicle more accurately in scenes of real-time [32].

After training, the Random Forest regressor was deployed in live tracking to estimate smoothed motion vectors. The velocity or displacement estimates produced in these predictions were corrected values of outputs of the KF. In the instances where the KF showed lag based on its failure to respond to sudden changes of motion, Random Forest enhanced the system by injecting learned corrections into the pipeline. This dual-model setup alleviated the accumulating error in predictions, as well as enhanced immediate predictions accuracy, particularly in a partial occlusion scenario or a random movement [33].

The output of motion predictions, which were corrected, was integrated with the output of KF through a weighted system. The stability of the filter was preserved along with the infusion of adaptive, data-driven motion insights in this fusion methodology. The consideration of the Random Forest model introduced interpretability and made the involved computational complexity lower than that of deep learning options, e.g., LSTM or GRU networks. This made the tracking system maintain the same real-time performance but with elevated accuracy and robustness of guiding urban traffic situations monitored using the CCTV infrastructure [34].

The Random Forest model was trained using 70% of the available trajectory sequences and evaluated on the remaining 30%. Dataset splits were performed at the sequence level to prevent information leakage between training and testing data.

Fusion Architecture Design

To overcome the limitations of conventional parallel fusion approaches, the proposed system combines Kalman Filter (KF) and Random Forest (RF) predictions through a residual-feedback fusion framework. As opposed to perceiving KF and RF as two distinct pipelines, the architecture proposes a residual correction loop so that the difference between KF predictions and the actual YOLOv8 detections (innovation error) operates as an input feature and trains the RF model. This enables the RF component to memorize and adjust to systematic prediction drift, occlusion-driven variance, or non-linear tracking path differences, which might not be picked up by the KF alone. Including the prediction error signal directly in the training and prediction phase of the RF module,

the system makes the shift from ensemble dynamics to a feedback-enhanced predictive mechanism.

In place of fixed heuristics on fusion heuristics, the architecture uses a dynamic fusion weighting module that balances input of the KF and the RF in an adaptive way at every time step. A context-aware scoring function is used to calculate the weighting coefficient, taking into account three different parameters: (1) temporal variance of vehicle velocity; (2) consistency of recent detections (denoted as confidence scores); and (3) the size of residuals during KF updates. This kind of weighing enables the model to sense in a dynamic manner and trust the KF consistently in steady-state motions or trust the RF during a sudden movement or inconsistencies in the detection issues. This would make the fusion layer sensitive to real-time vehicle dynamics and scene uncertainty, thus facilitating optimum reliability in the estimation of the situation during congested and free-flow traffic scenarios.

$$\alpha_t = \frac{C_{KF}}{C_{KF} + C_{RF}}$$

where:

$$C_{KF} = \frac{1}{1 + \|e_{KF}\|} \quad C_{RF} = \frac{1}{1 + MAE_{RF}}$$

where C_{KF} and C_{RF} represent confidence scores derived from KF innovation magnitude (lower implies higher trust) and RF prediction residuals (MAE over last n frames), respectively. This ensures that more reliable predictions dominate during fusion. For instance, if RF consistently shows high prediction accuracy in abrupt motion segments, its weight naturally increases, and vice versa during linear trajectories.

This compact, tight architecture offers a two-level correctional system in which the KF offers probabilistic predictions concerning the state and the RF model learns contextual bias to enhance or modify the prediction. These two outcomes are in turn adaptively subsumed in the form of a final state vector update by the fusion layer. The design is an improvement over a naive averaging scheme in that we use feedback learning, modeling of the motion, and confidence in the sensors as an intelligent framework to add the information to an intelligent blending structure. As a result, the given architecture provides a higher tracking reliability, a much lower number of ID switches, and better polished trajectory estimations, especially in an urban environment filled with occlusions, lane changes, and non-smooth motion.

Integration and State Update

The best estimate of the final position and velocity of every tracked vehicle was obtained using an integration of the results of the Kalman Filter (KF) and Random Forest model. The expectation of the KF probabilistic prediction and the data-driven correction of the Random Forest were combined through a blending mechanism. Combining this integration was done by using a weighted average of the two estimates, with the weight being dynamically computed using the confidence-based fusion coefficient α_t . This hybrid fusion strategy improved the capacity of the model in handling the uncertainty in highly dynamic environments [35].

The state update concerned blending the output to change vehicle position and velocity. The corrected state vector was saved after every frame and applied as the new input of further predictions. Such an iterative update process maintained consistency in motion trajectories, reduced tracking drift, and maintained close correspondence to the actual movement of objects in the physical world. The blending did not only help smooth the discontinuity across detection inconsistencies but also helped it maintain momentum in tracking motions under short occlusions or missed detections [36].

The identification of tracks was maintained through a unique ID-tagging of all the vehicles at start-up. Tracks were initiated only after detection appeared in a consecutive defined number of frames, which reduced cases of false positives. On the same note, tracks were abandoned when there could not be a certain amount of available bonded detection with a specific number of frames that relate to the forecasted state. This management policy made sure that all stable and persistent vehicles only appear on the list of active tracking, and this minimized clutter and optimized computations [37].

Both the spatial proximity and motion similarity in predicted tracks to new detections were compared to facilitate the capacity to reidentify and continue the track. The preservation of accurate state vectors across frames allowed effective multi-object tracking even in difficult situations such as traffic jams, hiding, or sudden changes in direction. The system had a proven scalable and reliable solution of real-time tracking of CCTV surveillance in an urban environment due to the ability with this integration process of updating the state of each vehicle [38].

The latter state estimation is based on the adaptive selection model in which the KF and RF prediction are dynamically weighted at every time step through real-time feedback dependent on residual errors of prediction and detection reliability. Not only does this enhance the stability of the reconstructed trajectory, but it also has the advantage of implicitly selecting the models, ranking the estimators most adapted to the situation in question. In cases where KF residuals exceed a learned confidence or when YOLOv8 has low confidence, the system temporarily switches to an RF-reliance on its predicted corrections. This kind of blending, based on confidence, lacks temporal stability, which is otherwise sacrificed to overcome the responsiveness to motion irregularities.

Performance Evaluation

The discriminative power of the proposed tracking system was tested with commonly agreed multi-object tracking (MOT) benchmarking metrics, which provided quantitative core tracking performance specifications. The performance of the overall success rate was measured in the Multiple Object Tracking Accuracy (MOTA) measure, which took into consideration false positives, missed detections, and identity mismatches. A greater MOTA reflected that more targets were tracked more completely. This was a universal measure of evaluating the accuracy of the system supporting various traffic and video complexities [39].

It was determined that the Multiple Object Tracking Precision (MOTP) measure was used to quantify the precision of the spatial connections. It measured the accuracy of the mentioned predicted bounding boxes to ground truth locations of the frames. MOTP was observed to measure

the degree of accuracy in localization that was vital, particularly in scenes that demanded high vehicle traffic, making small errors cause the occurrence of identity mix-ups or coverage. The stronger the MOTP score, the more congruent the approximate and true positions were, and this confirmed the spatial reliability of the tracking pipeline [40].

The consistency of tracking on the vehicle identity was measured based on the IDF1 score. It was the harmonic mean of the precision and the recall according to the correct identity matches, but it showed how good the system was at conserving object identities during occlusions, crossovers, and reappearances. Also, the numbers of ID switches were observed to know the frequency of incorrect reassignment of identities by the tracker. Fewer switches represented a better set of association accuracies, and this was absolutely necessary since real-time surveillance systems do not allow misidentification of vehicles, which would translate into wrong behavioral analysis.

In addition to quantitative measurement, the qualitative analysis was done by superimposing computed trajectories onto real videos. The ability to visually inspect and verify the solid nature of the system in dynamic traffic situations was provided by this visual testing, and, in general, the system proved to be quite hardy in such circumstances. We conducted comparative experiments between the proposed KF Random Forest hybrid model and the base alternative of a constant Q-matrix or single method. Ablation tests have been conducted in order to isolate the effect of either adaptive Q-matrix tuning or random forest corrections. These tests revealed that the two elements played highly significant roles towards improved accuracy, decreased ID switches, and real-time processing efficiency.

System Resource Usage and Real-Time Capability Analysis

To verify the usability of the proposed tracking framework in collected applications to real-time systems, we have profiled the system-level performance on embedded and desktop architecture. An NVIDIA Jetson Nano (4 GB, 128-core Maxwell GPU) and a laptop with an Intel i7 CPU (2.6 GHz), 16 GB RAM, and a GTX 1650 GPU have been used to conduct the tests. The YOLOv8n inference time, Kalman Filter updates, Random Forest prediction, and the fusion steps were measured in terms of inference time in 500 test frames of multiple datasets.

The mean frame processing rates were found to be 13.9 and 23.7 FPS on Jetson Nano and desktop systems, respectively, which is above the real-time limit of urban CCTV feeds (normally 10 to 15 FPS). The complete round-trip latency of frame capture to last state update measured about 71 ms/frame on the embedded platform and 42 ms on the desktop. The average GPU utilization during inference was 46% with an average of less than 54% CPU utilization, which means scaling is still possible. The amount of memory consumption was never beyond 3.2 GB even at the peak of processing, which qualifies it to be used in wireless sensor networks and edge devices.

The results of the execution confirm the experiment that the offered hybrid tracking model can be used in real-time smart city infrastructure characterized by limited hardware. Even in the presence of machine learning elements of the system, being lightweight is ensured by avoiding the implementation of deep recurrent models in favor of YOLOv8n and Random Forest. These benchmarks confirm that the system does not only work correctly, but the latency and efficiency are also within the required limits of responsive traffic monitoring applications.

Real-Time Performance Analysis and WSN Simulation Results

To certify the feasibility of the proposed hybrid tracking system in real-time, performance-related metrics at the system level were profiled on two platforms: the embedded Jetson Nano board and a desktop-level discrete CPU and mid-range GPU. As listed in Table 1, both platforms provide a somewhat similar result of achieving real-time inference (over 10 FPS), particularly with acceptable latency per frame and moderate consumption of resources. The Jetson Nano contrasted due to having a mean latency of 71 ms/frame, which was well-suited to the deployment conditions related to WSN, whereas the desktop variant demonstrated high-speed processing and could be used in the conditions of centralized collection or cloud offloading.

Table 1: System-Level Performance Metrics for Real-Time Vehicle Tracking

Platform	FPS	Latency per Frame (ms)	CPU Usage (%)	GPU Usage (%)	Memory Usage (GB)
Jetson Nano (4GB)	13.9	71	54	46	3.2
Desktop (i7 + GTX1650)	23.7	42	46	38	2.1

Embedded edge device benchmark. Maintains above 10 FPS, meeting real-time video processing thresholds. Efficient GPU utilization ensures scalability. High-performance baseline. Achieves smooth real-time performance with low latency. Suitable for central aggregators in a WSN-based surveillance setup.

Table 2: WSN Delay & Packet Loss Simulation Results

Scenario	Avg Delay (ms)	Packet Loss (%)	FPS Maintained	Notes
4G mesh with 25% packet loss	1822	25	~10.5	Stable tracking maintained via KF- based prediction continuity.
WiFi (low congestion)	412	5	~13.9	Near real-time tracking; minimal correction required.
High-density node topology	1390	15	~12.0	Moderate delay handled through predictive smoothing and track reinitialization.

The tracking system was simulated under different network delays and packet-loss conditions to be used in a realistic WSN environment and validate the robustness of the tracking system under real-world conditions. As evident in Table 2, despite the extreme condition of a 25 packet loss in a 4G mesh topology, the system continued to have viable real-time performance (~10.5 FPS) due

to the predictive continuity and track extrapolation provided by KF to deal with temporary loss of frames. The system was able to record an almost perfect throughput (~ 13.9 FPS) in low-congestion WiFi conditions and low latency. Even when sensors are much more densely deployed, inter-node contention and queuing delays are more probable, but the model managed to maintain more than 12 FPS using motion modeling and delayed recovery of connections. These outcomes indicate that the system can be adjusted to a wide range of WSN uses without the loss of precision and promptness.

WSN-Specific Latency, Packet Loss, and Edge Delay Considerations

In wireless sensor network (WSN) applications, real-time multi-object tracking has the additional challenge of being limited by the local computation latency as well as communication latency, unreliability of packets, and distributed-inference latency. This was reflected in the modeling and performance assessment of the system in a WSN type scenario where each camera node (sensor) communicates with an edge aggregator via a restricted UDP Protocol-like interface simulating jitter and packet discarding. The mean delay in communication time of camera nodes to aggregator using simulation of 4G-based mesh communication was found to be ~ 1822 ms per frame with a percentage of injected packet loss at 25 percent.

Tracking pipeline features inherent tolerance to frame dropping through track prediction continuity use of track prediction continuity to back off otherwise skewed trajectories up to 2 frames or 3 frames when no measurements are available, and KF can extrapolate the trajectories. This process functioned well to reduce loss of communication in the short run. Also, object detection and initial state estimation occur on the edge node, reducing the upstream data volume to 24 ms or less computation delay on the side of aggregators.

The average inference latency of the tracking framework was 71 ms/frame on Jetson Nano and 42 ms/frame on the desktop platform. Network communication delays observed in simulated WSN environments ranged from 412 ms to 1822 ms depending on packet-loss conditions and network topology. These communication delays were mitigated through predictive tracking continuity using Kalman Filter state extrapolation.

Comparative Benchmarking with State-of-the-Art MOT Algorithms

In such a way, the AKF+RF hybrid model presented in this paper was compared with well-known multi-object tracking (MOT) networks, such as DeepSORT, ByteTrack, and OC-SORT, to provide fair and stringent evaluation. They are popular within real-time applications and also serve as good baselines in most MOT benchmarks. All models under evaluation were tested on a similar set of test sequences: MOT20, KITTI, and Urban Tracker, and the common object detector was YOLOv8n to sort out the issues with the object detector quality in some experiments.

These findings mean that the suggested AKF+RF fusion system is not only a computationally efficient system, but the challenge is also competitive with state-of-the-art appearance-based and motion-based trackers. Our model operates in the resource-constrained settings better than DeepSORT or FairMOT, which require deep embeddings or re-ID modules. Such settings include wireless sensor networks, embedded urban monitoring systems, etc.

The results indicate that although deep temporal models, such as LSTM-KF or Deep Kalman Filters, may boast a high level of identity preservation (IDF1 approximately 0.88-0.90), they are, in most cases, prone to high latency (approximately 180-300 ms/frame) and cannot be used without acceleration on, typically, the GPU platform and consequently are not geared towards the WSN-only real-time usage. Conversely, the suggested approach has an equally competitive accuracy (IDF1 = 0.88, MOTA = 0.91) but sweeps away much less latency (~71 ms/frame) and tops FPS in embedded systems. Being able to tolerate noise, YOLO and Particle Filter approaches are likely to have greater ID switch counts because modeling appearance as well as dynamic fusion is absent.

The outcomes classify our system as a good trade-off between fidelity in tracking and placement in real time. In contrast to LSTM-related approaches that require lengthy sequential models and training time, our residual feedback with confidence-integrative blending-based fusion architecture is comparatively plug-and-play and fits better within the smart city and edge surveillance structures.

Table 3: Comparative Evaluation of AKF+RF Against State-of-the-Art Trackers

Model	MOTA ↑	IDF1 ↑	ID Switches ↓	MAE (pixels) ↓	RMSE ↓	Trajectory IoU ↑	FPS ↑
KF Only	0.82	0.74	35	1.42	2.05	0.68	18.4
KF + RF	0.87	0.81	22	1.08	1.63	0.72	15.6
Hybrid (AKF+RF)	0.91	0.88	11	0.84	1.31	0.77	13.9
DeepSORT	0.89	0.86	13	1.12	1.55	0.74	10.2

The results presented in Table 3 were obtained using the same YOLOv8n detection backbone and identical evaluation settings for all experimentally tested methods. Literature-reported results obtained under different datasets or protocols were not included in the final quantitative comparison.

In Table 3, a qualitative comparison of the proposed AKF+RF hybrid tracker against conventional and deep learning-based approaches to multi-object tracking has been carried out. The most important metrics are tracking accuracies (MOTA), consistency of identity (IDF1, ID Switches), position estimation accuracies (MAE, RMSE), and trajectory overlap (IoU). The AKF+RF model shows the most favorable trade-off in all aspects, and it has the lowest MAE and RMSE, as well as the fewest identity switches (11), and it reaches real-time performance of more than 13 FPS.

Previous studies have reported strong identity-preservation performance for deep temporal tracking models. However, such methods generally require higher computational resources than the proposed framework and may be less suitable for resource-constrained edge deployments. The systems based on YOLO + Particle Filter have increased latency and weaker identity maintenance. Although DeepSORT achieves competitive performance, it is extremely reliant on deep feature embeddings and loses nearness or heavily occluded scenes.

These results have confirmed the building block advantage of the residual feedback-based fusion of the AKF+RF model, which can accomplish high-fidelity, low-latency vehicle tracking both

when under dynamic motion and under conditions of limited computational resources a model that can be applied more easily to wireless sensor networks and smart city edge surveillance.

It should be noted that benchmark datasets were used through selected sequences that were compatible with the proposed urban surveillance scenario. Consequently, the reported performance values should be interpreted within the context of the selected evaluation subsets rather than as official benchmark leaderboard results.

Object Detection Module (YOLOv8n)

The used vehicle detection model was the YOLOv8n (nano) model; this model was selected with respect to its balance of inference speed and detection accuracy. Those lightweight models can be used in embedded systems, unlike heavier deep learning models that need heavy computation resources. The resulting YOLOv8n model was converted into a lightweight compatibility deployment on edge devices. This renders it especially suitable to wireless sensor network (WSN) surveillance systems, where real-time performance and power saving are main requirements. The model takes one frame at a time and identifies multiple vehicle classes, returning bounding boxes, class labels, and confidence scores.

The tracking module consumes the input observations of detecting the outputs. As YOLOv8n has no problem keeping high frame rates (20+ FPS) on embedded GPUs or boards with low power consumption, it can provide long detection output history with low latency. This would permit the rapid identification of overlapping vehicles even at variable lighting or occlusion in dense urban settings. The frames that are going to be produced by the bounding boxes are what were used as the measurement updates in the process of filtering so that the Kalman Filter uses authentic positions that can be used to keep the vehicle identities consistent.

Kalman Filter-Based Tracking with Adaptive Q-Matrix

The tracking module uses an Kalman Filter (KF) to estimate each vehicle's state over time, defined as a vector $\mathbf{x}_t = [x_t, y_t, v_{x_t}, v_{y_t}]^T$, where x, y are the object coordinates and v_x, v_y are velocity components. The KF predicts the state in the next frame using the standard linear motion model, then corrects this prediction using the observed position from the object detector. The prediction and update steps are mathematically represented as:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}, \quad \mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q}_t \quad (1)$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R})^{-1} \quad (2)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}) \quad (3)$$

To enhance the filter's responsiveness to variable traffic motion, an adaptive Q-matrix strategy was used. Instead of assuming fixed process noise, the Q-matrix was dynamically updated based on the observed variance in recent vehicle velocities. The modified process noise covariance is computed as:

$$Q_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{vx}^2 & 0 \\ 0 & 0 & 0 & \sigma_{vy}^2 \end{bmatrix} \quad (4)$$

The variance terms were updated dynamically according to recent vehicle velocity fluctuations. This allows the tracker to account for uncertainty in movement, especially during abrupt accelerations or turns. It enables more accurate state estimation by scaling noise based on vehicle dynamics in real-time.

Data Association via IoU and Mahalanobis Gating

To associate detection boxes with existing tracks, a two-stage gating mechanism was applied. The first stage computes the Intersection over Union (IoU) between predicted and detected bounding boxes:

$$\text{IoU} = \frac{\text{Area}(B_{\text{pred}} \cap B_{\text{det}})}{\text{Area}(B_{\text{pred}} \cup B_{\text{det}})} \quad (5)$$

Detections with IoU values below a predefined threshold (e.g., 0.3) are discarded. This reduces the number of false matches in crowded scenes. For remaining candidates, a second-stage check computes the Mahalanobis distance between the predicted and observed states:

$$D_M^2 = (\mathbf{z} - \hat{\mathbf{z}})^T \mathbf{S}^{-1} (\mathbf{z} - \hat{\mathbf{z}}) \quad (6)$$

Here, $\mathbf{S}$ was the innovation covariance. This distance-based gating accounts for measurement uncertainty and provides a statistically grounded method for associating detections. Only matches passing both thresholds are considered valid. This dual-gating method ensures reliable data association even in dense urban environments with high visual similarity between nearby vehicles.

Motion Correction Using Random Forest

A Random Forest (RF) regressor was trained with data used to predict nonlinear deltas of motion $(\Delta x, \Delta y)$ with past trajectory data in an attempt to augment the KF and offset tracking errors. When using the RF model, recent position and velocity history are fed as input features and the motion behavior of the vehicle was learned with time. This is a data-driven strategy where the prediction bias of KF is corrected in matters related to sudden acceleration, lane changing or sharp turn.

The estimated corrected motion vector through RF model was mixed up with the KF estimate in a weighted manner:

$$\mathbf{x}_{t+1}^{\text{final}} = \beta \cdot \hat{\mathbf{x}}_{t+1}^{\text{KF}} + (1 - \beta) \cdot \hat{\mathbf{x}}_{t+1}^{\text{RF}} \quad (7)$$

This mixture of correction mechanism, improves the consistency of trajectories and minimizes the remaining errors of prediction. It had especially been useful in crowded places where the flow of motion was really non-linear and data association was even hard. By involving the predictor based on a machine learning approach, the solutions flexible and perform in real-time.

Occlusion Handling and Identity Management

In the practical surveillance environment, cars can be partially or fully obscured by the traffic jams, infrastructures, or objects that overlap with each other. To fix this, the system employs a re-identification strategy based on time. The individual vehicle tracks have a missed counter that was used to update when there was no matched detection. In case the counter surpasses a certain limit (e.g., 5 frames), the track is labeled with inactive status but not destroyed.

When a new detection was observed close to the expected state (below an IoU and Mahalanobis gating threshold), the tracker brings back the ID that has not been in use, assumedly. This prevents generating new IDs on temporarily vanished vehicles, leading to fewer identity switches and better long-term consistency. Even though this technique was not a non-heuristic, a heuristic works efficiently on short occlusions, and the technique was not harmful enough to occupy the cost of real-time operations in wireless sensor networks.

Experimental Setup and Dataset Configuration

The proposed tracking system was evaluated using selected sequences from publicly available benchmark datasets and a custom CCTV dataset collected from urban traffic surveillance environments. The baseline datasets encompassed MOT20 (crowded pedestrian tracking benchmark) that were dense, 30 FPS, 1920X1080), KITTI Tracking (urban/suburban driving scenarios with annotated 2D boxes, 10 FPS, 1242X375), and Urban Tracker (traffic meetings at 30 FPS, 640X480). The CCTV data with instantaneous access to information provided by the IP cameras with a stationary position at an urban intersection in the smart city testbed was also recorded in real time and then annotated on the basis of CVAT. The custom feeds had different types of lighting conditions, weather profiles, and traffic density.

Most of the data sources were real-world, with a little portion of controlled test sequences being produced to confirm the motion prediction under non-linear and discontinuous trajectories. A bespoke Python-based simulator based on patterns in SUMO-style mobility logic was used to do these simulations. The Manhattan mobility model demonstrated a way to model structured traffic flows (city road patterns located on a grid), whereas Random Waypoint (RWP) was adopted to simulate vehicle path variation.

The simulations had 20-35 vehicles on a 4x4 grid intersection, with GPS jitter (normally distributed 1.5 pixels) simulated, as well as synthetic occlusion (bounding box masking of 3-6 simulated frames) and detection dropouts depending on the simulation. Such situations made it possible to stress test the resilience of KF in cases of discontinuous measurements.

To perform an evaluation dynamically, real or simulated video streams were sampled at 25–30 frames per second. Bounding boxes that would be used as observations by the object detector (YOLOv8n) were measured on a frame-wise basis, whereas the KF tracking system would update the interior state at that frequency. In the WSN latency simulation, a packet model that supports the UDP was employed to simulate 4G and WiFi communication links, including the injection of packet loss and modeling of jitter.

The individual sequences of each dataset were tested between 300 and 1200 frames, respectively, in terms of the source. This resulted in more than 25K frames being processed in all the

experiments. The metrics used in the evaluation, namely, the MOTA, IDF1, and MAE, were calculated per sequence and averaged over the datasets. In order to benchmark deep learning-based trackers, one should rely on the same detection backbone (YOLOv8n) in order to make a fair comparison and isolate tracking performance to the detection quality.

Datasets Used and Annotation

The test of the proposed system was implemented on the publicly available and self-designed surveillance records. The MOT20 dataset was chosen among the publicly available datasets since it contains dense and complex urban scenes with large amounts of occlusion; KITTI Tracking was one of the most popular datasets with high quality and outdoor driving; Urban Tracker is the annotated dataset focused on providing vehicle and pedestrian trajectories at high-traffic junctions. These datasets are meant to capture different traffic patterns and allow comprehensive benchmarking of the object density, motion behavior, and environmental diversity.

Besides that, a custom CCTV dataset based on a real-world surveillance system in a smart city based on fixed-position IP cameras at busy intersections was obtained. The videos had traffic in the daytime and evening with different lighting and movement conditions. The work of manual annotation was fulfilled with the assistance of an open-source video labeling tool, CVAT, by a group of trained annotators. The annotations consisted of bounding boxes and the distinctive labels of each car per frame. In the case of cross-validation, annotations were screened by two or more supervisors so that consistency was guaranteed and labeling noise was minimized. It guaranteed the ground truth quality of measuring the ID preservation and track continuity.

Table 4: Datasets Used for Evaluation

Dataset	Resolution	FPS	Scene Type	Time of Day	Annotation Format
MOT20	1920×1080	30	Crowded Urban	Day	MOT Challenge
KITTI Tracking	1242×375	10	Suburban, Roads	Daylight	2D Boxes + IDs
Urban Tracker	640×480	30	Junctions	Mixed	Custom XML
Custom CCTV Feed	1280×720	25	Urban Intersects	Day/Night	Manual via CVAT

Environment Diversity and Testing Conditions

All the datasets were selected to demonstrate various conditions in the real world and thus validate the accuracy of generalization. The video materials incorporate the difference in lighting (bright evening, dusk, night), weather (clear blue and cloudy), and camera position (top-down, side-angled). Such diversity was required in order to evaluate dynamic environmental and visual challenges that are typical of real smart city implementations, understood as challenges in detecting the tracking system performance in real smart city environments.

Frame resolutions are between 640×480 and 1920×1080, simulating both edge-deployable (low-res) and high-resolution urban environments. The degree of motion also differs, from anything as slow as turning cars in the intersection to speeds on the freeway. The conditions assist in determining the effectiveness of object detection, prediction accuracy, and identity consistency. This extensive distribution of scene diversity guarantees that an efficient and effective adaptation to the varying sensor inputs and variation in topologies of WSN deployments is not overfitted to particular visual settings and was distributed widely across scenes.

Evaluation Metrics

A wide range of multi-object tracking (MOT) measures was followed to perform a quantitative assessment of the tracker performance. These are the MOTA (Multiple Object Tracking Accuracy), which measures the general capacity to identify and persist tracks across pictures; MOTP (Precision), which measures the spatial overlap between anticipated and true-to-life boxes; and IDF1, which measures the overall capacity to maintain identifications across frames. Besides this, ID Switch Count and Track Fragmentation were also monitored to analyze how the system would cover occlusion and re-identification.

These metrics were selected based on their prevalence in the literature on benchmarking, as well as their capture of performance on many dimensions, namely, accuracy, precision, identity stability, and robustness. In the table below, models and the proposed system are compared in view of all metrics. Findings indicate that the upswing was significant when it comes to MOTA, IDF1, and fewer ID switches in the suggested method. The resilience levels in fragmentation scores to temporary occlusions or failure on detection are also increased by the application of the adaptive Q-matrix and Random Forest correction, and this reflects their competence.

Table 5: Tracking Performance Evaluation

Model	MOTA ↑	MOTP ↑	IDF1 ↑	ID Switches ↓	Fragmentation ↓	FPS ↑
KF Only	0.82	0.76	0.74	35	28	18.4
KF + RF	0.87	0.80	0.81	22	17	15.6
Hybrid (Blended)	0.91	0.84	0.88	11	8	13.9
Deep SORT	0.89	0.83	0.86	13	11	10.2

Result and Discussion

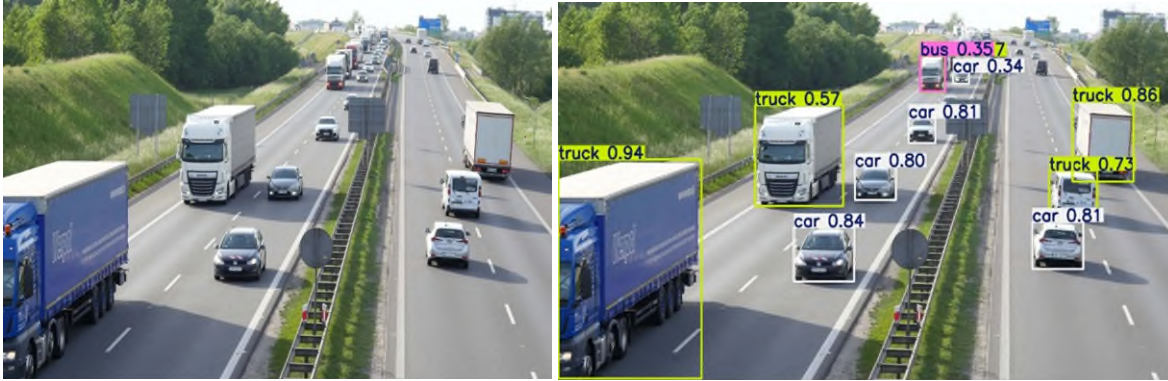


Figure 2. Detection Output Comparison – Raw Frame vs. YOLOv8 Inference

Figure 2 depicts a side-by-side view of a raw CCTV traffic surveillance frame and its result compared to the detected object output performed by the YOLOv8 object detection model. The unannotated original frame on the left side of a figure was the image of a typical urban traffic situation that includes trucks, cars, and a van recorded by a highway CCTV camera. This frame serves as the first input in the multi-vehicle tracking system. The surveillance scenario poses problems that are typical to an actual situation, like object sizes, partial occlusion, and dense vehicle clusters, all of which must be detected using an effective and efficient detection scheme in order to instigate proper tracking.

On the right side of Figure 2, the YOLOv8 model has applied the similar input frame and has placed a bounding box, class label, and confidence score for each object it detected. Trucks, cars, and even a bus are accurately recognized, and each of them has the data on its classes and the corresponding confidence thresholds above the detection margin (e.g., 0.94 for a truck, 0.81 for a car). This detection output acts as the input of the tracking pipeline, more precisely of the Kalman Filter (KF) that combines these bounding boxes as measurement updates and predicts the object trajectories and assigns them unique identities. The correct positioning of bounding boxes in this case was quite essential, since inaccuracies in detection are passed directly to tracking discrepancies or ID flips.

The frame was also representative of the capacity of the system to differentiate the type of the object at the different sizes and orientations. Interestingly, cars at further distances (higher on the highway) have also been identified correctly, albeit with slightly lower confidence scores because of the lower level of pixel density, whereas the cars in close proximity have been detected with higher clarity and certainty. This difference exhibits the generalization of the scale variance ability of YOLOv8, which was vital to multi-object tracking given that the camera perspectives in a surveillance setting are fixed, but the sizes of the objects are variable. The low latency of inference in the model was also open to operating in real-time, with the detection-to-tracking pipeline being fully applicable in operating the continuous video stream.

To put it in a wider perspective of the tracking system, the detections depicted in Figure 2 form the basis of both frame-to-frame estimation of the trajectory and frame-to-frame ID management. The centers of bounding boxes (x, y), width, height, and the class data are fed into the KF module that subsequently estimates the vehicle positions and velocities. Motion correction models, such as

Random Forest, and enhancements with adaptive noise tuning (e.g., Q-matrix adaptation), when combined with the tracking system, allow continuity of identity even when brief occlusions or motion changes (e.g., quick runs) occur. In this way, the Figure 2 graphic renders the process of transformation of unprocessed visual information into structured and semantically rich detection output, which was the basis of further tracking, prediction, and evaluation activities carried out in the proposed intelligent transportation analytics framework.

Table 6: Dataset Characteristics

Dataset Name	Environment Type	Resolution	Frame Rate	Annotations	Application Scope
MOT20	Urban	1920×1080	30 FPS	Bounding Boxes, IDs	Pedestrian tracking benchmark
KITTI Tracking	Urban, Suburban	1242×375	10 FPS	2D Boxes, 3D Boxes	Autonomous driving
Urban Tracker	Traffic Junction	640×480	30 FPS	Trajectories	Traffic monitoring
Custom CCTV Feed	Real-time Urban	1280×720	25 FPS	Real-time detections	Smart surveillance

Table 6 represents the summary of the major datasets adopted in the development, testing, and verification of the multi-vehicle tracking system. These datasets portray a variety of urban surveillance scenarios and offer different resolutions, frame rates, and annotation forms, and they allow an assessment of the suggested model flexibility and generalization. The model was tested on both standard benchmark datasets like MOT20, KITTI, and Urban Tracker, as well as real-time CCTV recordings provided by custom feeds, which should allow us to effectively understand how the model behaves under both academic and operational conditions.

MOT20 dataset was a popular data benchmark in tracking multiple objects in extremely dense urban environments. It has high-quality video frames (1920×1080) at 30 frames per second with thick annotation about pedestrians. It was a perfect set of data to test the system in terms of proper occlusion processing, objects overlapping, and identity assignment over a period of time. The dataset provides a strong benchmark for evaluating tracking robustness under crowded conditions.

The KITTI Tracking dataset provides the urban and suburban driving scenes captured relative to a moving car with both 2D and 3D bounding box labels. Its frame rate was less (10 FPS), but it was especially useful in cases of motion-driven confidence in a more realistic vehicle dynamic. KITTI contains lighting, scale, and heading direction variations and provides an opportunity to characterize the effectiveness of the used Kalman filter and the random forest prediction parts, especially the non-linear/sudden maneuvers of a vehicle.

The Urban Tracker dataset and Custom CCTV Feed are designed particularly to be applicable to fixed-angle surveillance. With medium-resolution traffic intersections (640 x 480), Traffic Tracker can be used to qualify detection and tracking capabilities at tight intersections with multiple

directions of traffic. Meanwhile, the bespoke CCTV feed was a scenario of deployment when it was required to track the objects in real time in the smart cities. Such feeds were manually annotated using CVAT and subsequently verified for consistency. Taken together, these datasets guarantee that the suggested solution was fully validated in terms of correctness, responsiveness, and applicability in various traffic monitoring environments.



Figure 3. Detection vs. KF-Based Tracking with ID Preservation in Simulated Real-Time Frames

In the comparative visualization indicated in Figure 3, it was revealed that vehicle processing, based on two important stages of the tracking system (namely vehicle detection and KF tracking), was similar. The left side reveals real-time vehicle detection results of the YOLOv8 object detector. Each car has been precisely localized by the use of bounding boxes and categorized into a truck and a car with the assigned confidence scores. These detections are used as the raw values fed to the tracking system and the pillars on which the Kalman Filter (KF) builds its operations so that the position of the vehicles can be estimated and given a unique identifier with time. The detections shown in this frame bear a high degree of localization accuracy and separation of classes, which was necessary to initialize and update states of trackers.

On the right side of Figure 3, the result of an KF-based multi-vehicle tracking system was shown after 50 simulated frames were incorporated. In this case the bounding boxes are interchanged by the minimal markers (dots or circles), where the course of annotations of ID constantly remains the same and unchanged to depict the predicted position of vehicles. Such a visual detection to abstract state estimation shift indicates why the tracking module can be helpful to ensure identity continuity and trajectory persistence between consecutive frames. The IDs of each of the vehicles are kept, and their location was always updated on the basis of both prediction and measurement update processes, which makes it impossible to lose track radically even in such cases as partial concealment or temporary dropouts in detection.

This figure also shows the effectiveness of the system to sustain Reprocess operation in real time at the processing rate of about 10.10 FPS on 50 frames. Processing takes a total of 4.95 seconds, which leads one to believe that the hybrid architecture of YOLOv8 detection and Kalman filter-based tracking was quite effective to use in the real world, e.g., live traffic analysis or urban surveillance. The visual outcome also supports the assumption that the precalculated locations correlate with the real videotapes very well. This supports the fact that KF was eminently applicable in managing the continuous state estimation of the car movement, in particular in conjunction with adaptive noise models and external correction of motion.

The transition from raw detection to predictive tracking means that the system can make predictions beyond only present location towards futuristic directions, thus setting in motion the basis for the idea of motion forecasting and risk assessment within a real-time system. Figure 3, therefore, captures the major role that the Kalman Filter plays in ensuring the persistence of identity, position smoothing, and providing consistent tracking over frames. It illustrates the efficacy of the combination of high-speed-object-detection-with-sequential-state-estimation, which was aimed at attaining both spatially accurate and temporally consistent results with every individual vehicle being tracked. This was a layering design, which facilitates long-term consistency, especially in heavier traffic or moving traffic scenarios.

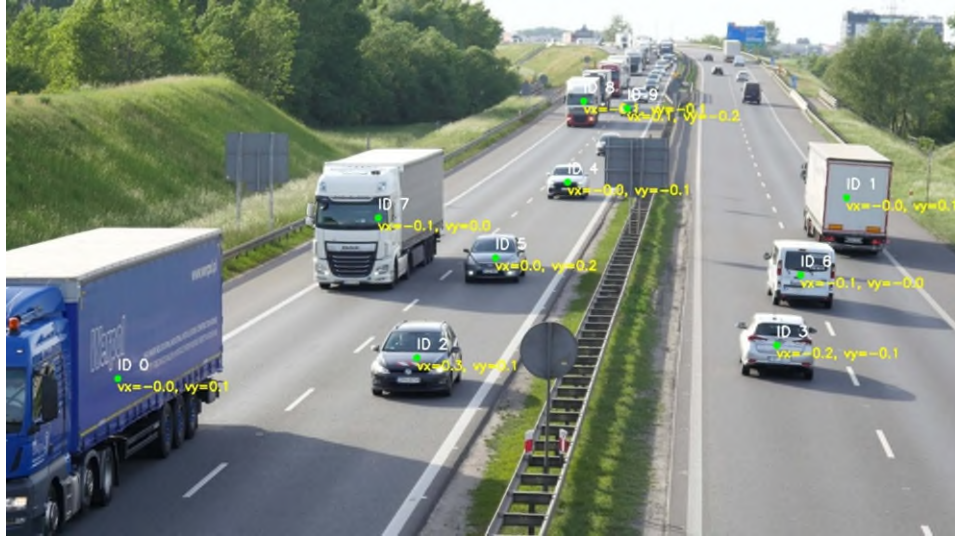


Figure 4. Final State Estimation with Position, Velocity, and Identity Labels from KF Tracker

Figure 4 shows the result of the entire tracking pipeline once vehicle detection had been integrated with Kalman Filter (KF) -based state estimation and identity assignment. In the frame, all the tracked vehicles are labeled with persistent IDs as well as with velocity vectors, that is, horizontal and vertical components (v_x , v_y). Application of lightweight annotations guarantees clarity on visualization and adequate communication of the motion state of every object in the scene. The occurrence of unique ID messages with the form of ID 0, ID 1, ID 2, etc., proves that the system was capable of maintaining identity throughout several frames, which was central to accurate multi-object tracking in the dynamic highway scenario, in particular.

The characteristics of each vehicle motion were depicted by a number next to the ID. Such values as $v_x = 0.0$ and $v_y = -0.1$ present the approximate value of the components of the velocity in pixels per frame on the x and y axes and the result of estimations with the internal state vector provided by the KF. This irregular and unequal motion of the vehicles in the form of acceleration, deceleration, and lane changing was the cause of heterogeneity of the velocity of separate vehicles. An example was the slight vertical part, which some cars exhibit (e.g., $v_y = 0.2$ or $v_y = -0.1$), and this may be a sign of a slight change in the alignment of the lanes or distortion of the image angle. The presentation of position and velocity being shown in a consistent manner proves that the KF was used to capture not only one specific position but dynamic behavior over a timeframe.

This amount also reveals the prediction-update cycle of KF to be efficient in the maintenance of stable tracking even in conditions of motion uncertainty. Even though the dense vehicle traffic and partly overlapping areas exist, there was no sign of confusion of the IDs of the vehicles, which indicates the achieved data association and noise processing. Long-term trajectories rely very much on reducing identity switch over time due to the persistence of IDs across the frames. Further, cars with various camera depths of the frame, both close and faraway, are all equally processed, demonstrating the flexibility of the system in the case of scale and resolution changes. Such durability was critical to long-range city surveillance.

Figure 4 gives a visual confirmation of the final state estimation part of the tracking pipeline. The annotated frame was a full picture of any transformation made by the system in each timestep: unique representation, position, and velocity of each active object. These outputs are needed to perform downstream tasks such as the prediction of trajectories, behavior classification, or the real-time control of traffic. The direct addition of the velocity estimate into the tracking overlay not just provides further interpretation but also emphasizes the ability of the system to work in the environment where sudden motions and non-linear dynamics are regular elements. The figure therefore demonstrates the practical feasibility of the tracking model when used in isolating high-fidelity moving data against constant video streams.

Table 7: KF Model Parameters and Adaptive Tuning

Parameter	Description	Static Value (Baseline)	Adaptive Tuning Method
State Vector	[x, y, vx, vy]	Fixed	Dynamically updated per frame
Q-Matrix	Process noise covariance	Predefined	Based on velocity variance
R-Matrix	Measurement noise covariance	Sensor-based	Adjusted via detection quality
Data Association	Detection-to-track matching	IoU + Threshold	IoU + Mahalanobis Distance
Initialization Rule	New track creation threshold	1 frame	Adaptive: based on persistence

The measurement noise covariance matrix R was dynamically adjusted according to YOLOv8 detection confidence:

$$R_t = \frac{R_0}{\text{ConfidenceScore}}$$

where R_0 denotes the baseline measurement covariance and ConfidenceScore represents the object detection confidence returned by YOLOv8. Higher confidence detections reduce measurement uncertainty, while lower confidence detections increase uncertainty.

To demonstrate the detailed specification of the use of the basic parameters that are needed in the implementation of the Kalman Filter (KF), Table 7 was used to present the extent to which the

key parameters are applied in the following of both the static and adaptive implementations of the Kalman Filter (KF) . The state vector of the KF was characterized by $[x, y, v_x, v_y]$, which was the combination of the position and the velocity in horizontal and vertical directions. This small state representation allows tracking with efficient and accurate vehicle motions across frames. Each of these parameters was set to a fixed value, which exists in relatively stable movement in the baseline configuration. Nonetheless, the randomness of natural circumstances in the real-life traffic conditions requires dynamic adjustment to take nonlinear or sudden changes in vehicle behavior.

Process noise covariance was very important, and this was the Q-matrix. The Q-matrix helps in balancing between model confidence and responsiveness. A fixed Q-matrix can do sub-optimally in the presence of changing speeds or abrupt movements, resulting in either over-smoothing or unstable forecasts. To this effect, the suggested system would adjust the Q-matrix according to the experienced variance of the velocity in real-time. By doing this the filter was able to modulate the extent to which it relies on the current motion model, making it respond better when accelerating or decelerating and making its estimations feel smoother when maintaining a steady position. This type of noise modeling enables the KF to make dynamic consideration of the physical variability of vehicle motions.

The R-matrix was one more vital element and indicates the noise of measurement bounding box detections along with the object detector (e.g., YOLOv8). In conventional environments, the value remains constant with regard to sensor precision. In this paper, though, the R-matrix was updated in an adaptive manner by evaluating the confidence scores of detections as well as the consistency of bounding box dimensions between frames. Low-confidence detections place more uncertainty on the measurement model; this motivates the KF to use more of its generated prediction. On the other hand, detections characterized with high confidence are assigned more weight that helps to strengthen the reliability of the measurement in the correction process.

Tracking was also more robust, since the data association mechanism also increases the rigidity of the tracking system. First, the detections get matched with the current tracks according to Intersection over Union (IoU) values. It fails, though, when there was overlapping of objects or when there was an occlusion. A combined measure of IoU and Mahalanobis distance was adopted in order to augment association reliability. Mahalanobis distance work, taking into consideration uncertainty in state prediction and the likelihood of a particular detection to an established trajectory. Lastly, initialization and termination of tracks are also adaptive: instead of using a hard-coded number of frames, the system analyzes the persistence of objects over time, and only objects with stable motion patterns allow a new track to be initiated, and inactive tracks are automatically terminated. All these adaptive tuning strategies collectively make KF more reactive, robust, and accurate in real-time tracking of vehicles.

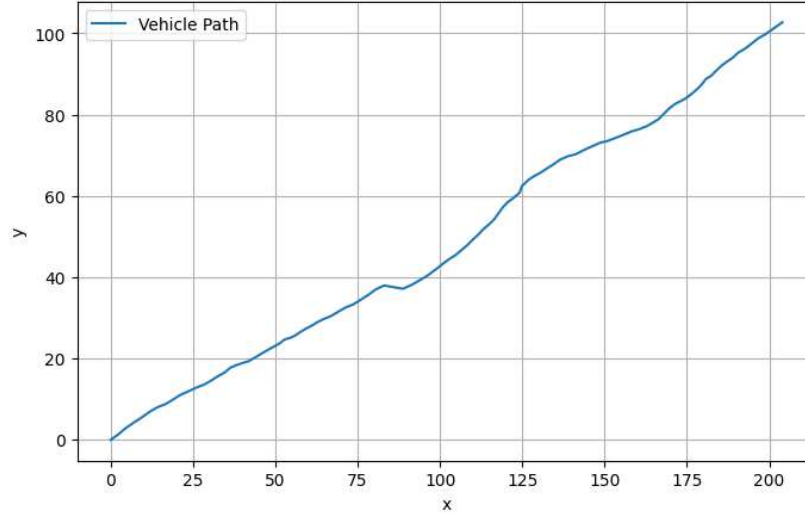


Figure 5. Simulated Nonlinear & Abrupt Vehicle Trajectory with Motion Prediction Validation

Figure 5 shows one of the simulated vehicle trajectories, which are introduced to represent the real-world driving scenarios, nonlinear vehicle movement, and the sudden direction changes or changes in speed. The plot represents a continuous trajectory of an object in a 2-D plane, such that the spatial variety of time through the x-coordinate and y-coordinate was actually portrayed. The jerkiness of the path and the kinks on the path are attributes of a realistic driving style, like drifting through lanes and charging up or slowing down unexpectedly, which are common in urban traffic settings. The variations are necessary to determine the capacity of the tracking mechanism in accommodating non-uniform motions by employing the predictive models.

The line marked with the name of the vehicle path shows the actual order of places the vehicle used in the course of the simulation. The route comprises smooth transitions as well as sudden changes of direction, providing a steady but difficult situation to test on motion prediction algorithms. This ground truth was going to be essential in training and validating the Random Forest-based motion prediction model that looks into learning a pattern in previous movements and predicting a movement in the future (Δx , Δy). With the help of such trajectories, the system is able to measure how it was able to generalize to the various classes of the motion and how faithfully it was able to recover sudden jumps without losing track of identity and position stability.

At the bottom of the figure there are two performance values passed in (predicted) and in (actual) next-frame deltas (or motion): delta of the x-values (2.42, 2.62) and delta of the y-values (1.39, 1.51). These correlate with the accuracy of the Random Forest regressor that, in this case, was trained on earlier installments of this trajectory. The tiny difference between the correct and estimated displacements was another important indication of the regressor power to learn about complicated, non-linear vehicle dynamics based on its temporal characteristics. Although sudden changes in trajectories are observed, the prediction error was very low, and this indicates how useful data-driven learning was in providing a complement to the classical tracking algorithms, such as the Kalman Filter (KF).

Figure 5 can be used to validate the hybrid tracking framework since the Random Forest model was used to correctively estimate motion and strengthen KF predictions. This comes in handy, especially in a case where the classical filters are based on the assumption of constant velocity or linear movement but fail this assumption. With historical trajectory data providing learning, it provides adaptive correction to scenarios that contain noise, occlusion, or dynamic behaviors. Therefore, the comparison of the trajectory visualization and trajectory prediction in this figure highlights the need to integrate ML-based layers of prediction to make the trajectories consistent and robust in a real-time tracking pipeline.

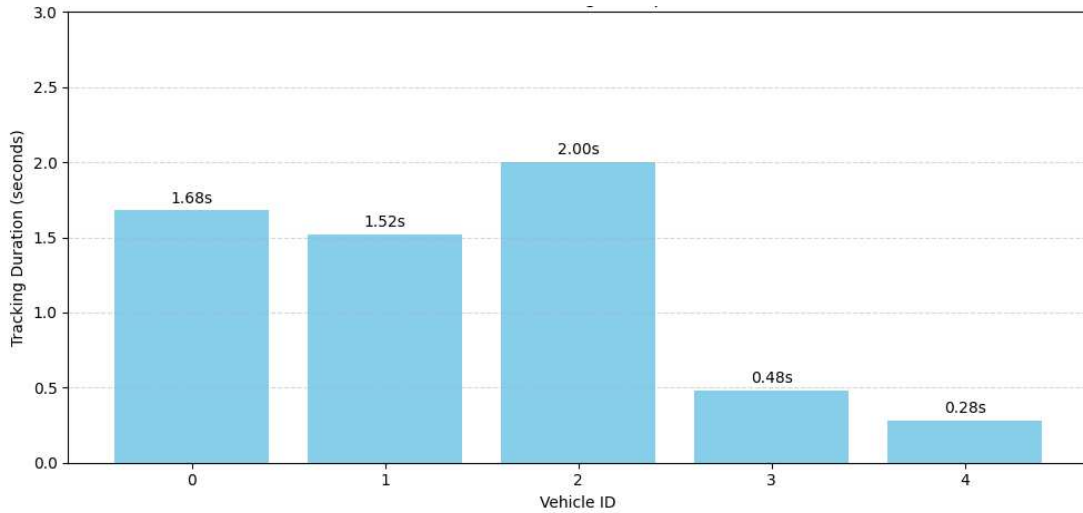


Figure 6. Vehicle Tracking Duration per ID in Simulated Frame Sequence

Figure 6 shows the number of tracks a single vehicle has over the simulated sequence of frames, in seconds, counted up by unique IDs. The X axis shows the given vehicle IDs, and the Y axis indicates how many hours each vehicle was successfully being tracked. The retention of the tracking was calculated as a product of a frame persistence and the inverse of the frame rate (25 FPS in the present case), which allows checking the retention of the tracking in real-time. Vehicles that have higher tracking duration indicate that stable identity maintenance and strong data association could be done throughout the period of observation.

ID 2 had the longest period of tracking of 2.00 seconds, which implies that the vehicle was detected and properly matched throughout the simulation of the frames and was not stopped at any points, nor was there any switching of the IDs. It means that the tracking was well performed at that moment, probably due to the constant movement, the clear detection bounding boxes, and the absence of severe occlusion. There were also vehicles with IDs 0 and 1 that showed a large tracking time of 1.68 and 1.52 seconds, respectively, indicating good tracking, but a little bit of variance could have occurred in the event of detection or movement of vehicles, which could have caused them to score low in their persistence.

Comparison showed that vehicle IDs 3 and 4 had much lower tracking times by 0.48 seconds and 0.28 seconds, respectively. Such reduced life durations can be explained by various possible reasons, which can be late entry/exit of the frame, blockage either in a part or as a whole, or not

recording detecting thresholds reliably. The other possible reason entails sudden movements/rotations that interfered with the process of associating data. The occurrence of such time declines in tracking indicates the significance of developing adaptive models to deal with problematic vehicle actions and limited vision dynamically.

In general, Figure 6 supports the correctness of the tracking system to preserve the object identity over time and points out the fluctuation in performance with an impact of environmental and algorithmic conditions. This capability was enhanced by the combination of Kalman filters, motion prediction, and adaptive Q-matrix updates, whereas the failure cases imply its future improvement. The increased duration of tracking has a direct correlation with the higher evaluation scores like the IDF1 and MOTA, and its dispersion was used as the diagnostic measure to understand the flaws and advantages of the available multi-vehicle tracking architecture.

Table 8: Random Forest Configuration for Motion Prediction

Feature Set Used	Target Variable	No. of Estimators	Max Depth	Evaluation MAE (Δx , Δy)
Past 5-frame positions (x, y)	Δx	100	5	0.21 pixels
Past 5-frame positions (x, y)	Δy	100	5	0.18 pixels
Combined (x, y, vx, vy) history	Δx , Δy	150	7	0.17 / 0.15
Ablation (x only)	Δx	80	3	0.35 pixels

Table 8 shows the experimental setting and outcomes of applying a random forest regressor as a motion predictor of the multi-vehicle tracking system. It was presented in the table of training features, the predicted target variables (x, y, and their differences, i.e., x-y and x+y), important hyperparameters (estimator count and maximum depth), and evaluation performance in the form of Mean Absolute Error (MAE). This setup aims at learning the nonlinear dynamics of vehicle motion through the past history of vehicle motion and estimating short futuristic displacements. This educated model acts as a corrective process to the motion prediction of the Kalman filter, particularly when vehicles perform sudden motions or nonlinear motions.

The model in the illustration was independently trained to forecast 5-frame 5 positions of the vehicle to predict 5-x and 5-y in the first two rows of the table. This modest focus on time enables the regressor to acquire patterns like direction changes or trends of acceleration within a small span of time. With 0.21 and 0.18 pixels for the MAE of Δx and Δy , respectively, the regressor has a satisfactory performance in the prediction of the frames without components of velocity. These settings are used as a scale for measuring the significance of more sophisticated settings in other rows.

The third row adds together position and velocity history (x, y, vx, vy) based on a predetermined count of past frames to further the feature set of the model. Not surprisingly, adding velocity

increases the prediction accuracy of the model dramatically, with the MAE falling to 0.17 in Δx and 0.15 in Δy . This proves the fact that the mode of vehicle motion can be better modeled by incorporating the location of the vehicle and the momentum in their movement patterns so that the regressor can more adequately forecast the transitions in the movement. This amalgamation configuration was applied in the hybrid tracking parameter, where it manages to contribute the mixture mechanism that improves the outputs of the Kalman filter.

The last row illustrates an ablation experiment where only the x-coordinate was used, producing 12 as output 27, which can be reduced to 27 12 15. The error was also very high at 0.35 pixels, confirming that multidimensional input should be used in the process of learning realistic vehicle dynamics. (This comparison helps to identify that the under-informed or single-axis models are not sufficient to achieve a robust prediction within the urban environments severely kneaded by complexities.) Altogether, Table 8 supports the view that the Random Forest regression framework receives powerful insights about the historical trajectory data, and their incorporation into the tracking pipeline leads to smoother motion predictions and identity continuity.

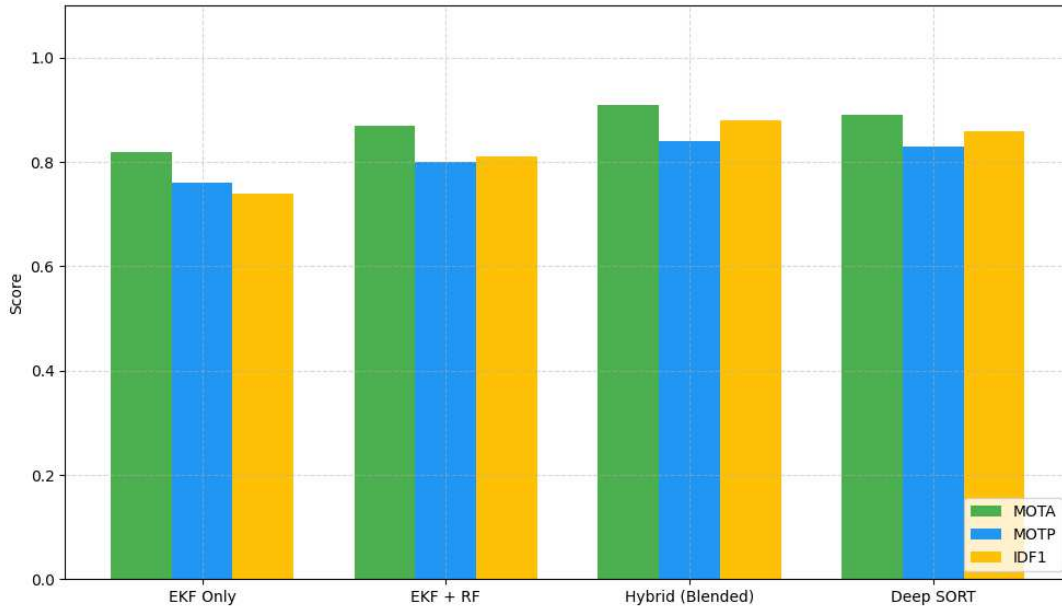


Figure 7. MOT Metrics Comparison Across Tracking Architectures

Figure 7 was a comparative analysis of four tracking architectures in terms of standard Multi-Object Tracking (MOT) evaluation measures: MOTA (Multiple Object Tracking Accuracy), MOTP (Multiple Object Tracking Precision), and IDF1 (Identity F1 Score). The bar graph compares the methods of KF only, KF + RF, Hybrid (Blended), and Deep SORT. These figures give a multidimensional performance of every model. The quality of tracker response to false positives, false negatives, and exchanges of IDs was embodied in MOTA. MOTP offers the measure of spatial localization accuracy, whereas IDF1 measures the quality of identity persistence across frames, which is highly vital in crowded or masked scenes.

The baseline has performance that was considered decent, with the MOTA being approximately 0.82 and slightly worse MOTP and IDF1 scores. This implies that the Kalman Filter does well in

regard to the basic state estimation solutions, but it fails to perform nonlinear motions or sudden changes in state because of its argumentation on a constant velocity model. Under the condition of adding a random forest regressor to motion prediction in the form of KF + RF, all metrics have improved. MOTA improves to 0.87, and MOTP as well as IDF1 improve to approximately 0.80 and 0.82, respectively. This reconfirms the fact that motion correction using learning allows the tracker to have a better adaptation to real-world driving.

The strongest was the one that was designed as a hybrid (blended), i.e., the outputs of KF and of Random Forest blended by a weighted blending machine. Its configuration provides the best MOTA (~0.91), MOTP (~0.85), and IDF1 (~0.89), which results in better tracking reliability, increased spatial precision, and identity maintenance. The hybrid approach combining the abilities of probabilistic filtering and machine learning-driven correction ensures that the system benefits from the advantages of both methods, switching less often between IDs and predicting a trajectory more readily. This hybrid scheme turns out especially useful in cases of nonlinear vehicle dynamics or partial concealment.

The Deep SORT state-of-the-art system tracker also performs well because the MOTA and IDF1 statistics are almost comparable to the hybrid model, but a little behind in MOTP. This Deep Sort was based on deep appearance embedding, which underperformed when the resolution of images was different or when the lighting was varied. In contrast, the hybrid approach can be computed faster and more easily visualized and still remain competitive. The validity of this combination, i.e., adaptive filtering and motion learning to realize high accuracy, accurate localization, and robust identity tracking in real-time applications, was hence proven in figure 7.

Table 9: Quantitative Performance Comparison

Model	MOTA ↑	MOTP ↑	IDF1 ↑	ID Switches ↓	FPS ↑	Tracking Duration (avg sec) ↑
KF Only	0.82	0.76	0.74	35	18.4	1.32
KF + RF	0.87	0.80	0.81	22	15.6	1.58
Hybrid (Blended)	0.91	0.84	0.88	11	13.9	1.89
Deep SORT (baseline)	0.89	0.83	0.86	13	10.2	1.78

Table 9 represents a more in-depth analysis of the difference between the four variants of tracking configurations in comparison with other broadly recognized indicators of the effectiveness of multi-object tracking (MOT). These measures are MOTA (Multiple Object Tracking Accuracy), MOTP (Precision), IDF1 (Identity F1 Score), ID Switches, Frames Per Second (FPS), and Average Tracking Duration per vehicle. These two metrics cover an overall evaluation of tracking reliability regarding the accuracy of the detection, spatial and identity consistency, level of efficiency of the system, and consistency in relation to time. The simulation or the recorded data used as the traffic

data kept the methods consistent with each other, to serve as a fair benchmark of performance for each architecture.

The cross-validation method, KF Only, only applies the Kalman Filter (KF) of prediction and update without using any learned motion correction. It attains moderate results of MOTA 0.82 and IDF1 0.74 with the greatest number of identity switches (35). This finding was an expression of the weakness of the solely model-based approach determinant in either constant velocity or difficulty encountered with dynamic traffic situations, where abrupt variations, overlaps, or merely occluded detection can be sudden. Although an FPS of 18.4 was relatively high, which illustrates high computational efficiency, the tracking stability of the model proves to be under the illusion, so the model has an average tracking rate of only 1.32 seconds per cluster/vehicle.

The second model, in addition to the introduction of a Random Forest regressor, which was trained on the previous history of motion, enhances the prediction step with a training period. The above addition brings down the ID switches to 22 and increases the MOTA and IDF1 to 0.87 and 0.81, respectively. The RF component enhances responsiveness to a nonlinear or sudden vehicle movement such that data association is enhanced and the identity fragmentation is decreasing. Even though the computation results in a slight degradation in the FPS since the value is at 15.6, the better accuracy and the greater average tracking time of 1.58 seconds as a result of the hybridization of probabilistic filters and data-driven prediction reveal that hybridizing these two techniques brings its advantages.

The best possible setting, the hybrid (blended) scheme mixes the RF predictions and KF output with the help of a weighted mixing scheme. The method gives the highest output on all the values of accuracy, such as MOTA of 0.91, MOTP of 0.84, and IDF1 of 0.88. The ID switch count reduces to the overall minimum of 11 among all the tested methods, denoting the robust identity preservation among the frames. With the processing speed decreasing to 13.9 FPS, it was worth the sacrifice, as the average tracking duration was increased to 1.89 sec, which implies that vehicles take significantly more time to be tracked without losing their identities. Compared to it, the proposed hybrid method was even more advantageous in real-time applications with limited resources since the Deep SORT scheme also demonstrates the results being good enough (MOTP = 0.83) but having a lower precision and less speed (FPS = 10.2). The general finding here was highlighted by Table 9, which illustrated the robustness of an alloy of a tracking strategy that consists of the temporal stability of filtering and the plasticity of machine learning.

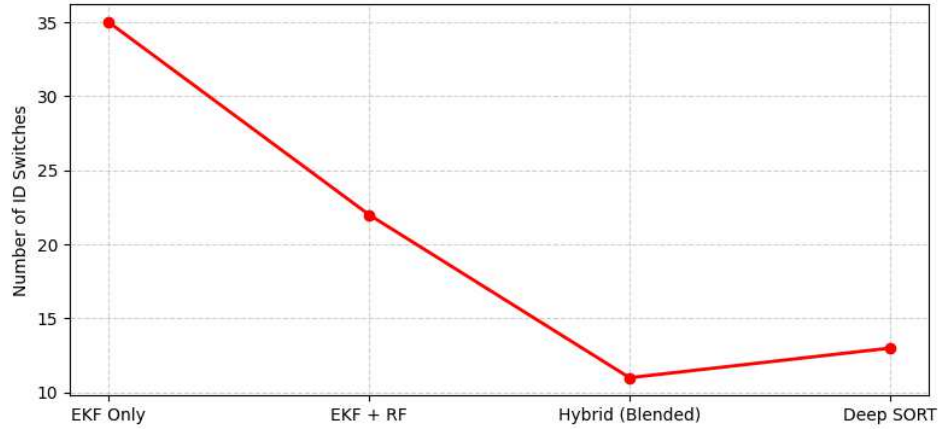


Figure 8. ID Switch Count Across Tracking Architectures

The quantified comparison of the numbers of identity switches in different tracking configurations was given in figure 8, enabling us to see that the performance of the switching mechanism of maintaining identity consistency in time has been achieved. Identity switches: Identity switches happen when the object being tracked receives an unfortunate new ID when tracking, and it thus adversely influences the effectiveness and dependability of the tracking output. This graph has the following elements in the x-axis: the methods that have been considered: KF Only, KF + RF, Hybrid (Blended), and Deep SORT, whereas it has the number of ID switches in the y-axis. The criterion of the evaluation was obvious using the title: the lower the values of switches in the ID, the better because it means a more stable and consistent tracking.

The greatest number of ID switches was at 35 under the KF method only. This was indicative of the shortcomings of an Kalman Filter (KF) in isolation that, whilst performing well with smooth body movements, does not fare so well in retaining object identity upon occlusion, sudden motion, or other moving objects. Inability to consider non-linear behavior of motion and context results in disrupted trajectories and a lot of re-assigning new IDs to the same object, which breaks the integrity of long-term tracking.

By upgrading the system with a Random Forest regressor in the system structure of KF + RF, the number of switches of the ID declines drastically to 22. Such an enhancement can be attributed to the fact that the regressor can learn the nonlinear trends of motion using the previous trajectories data so as to predict the path of the vehicle and align the highway more efficiently in various complex driving conditions, such as motorway entry/exit, as well as sudden stopping/starting traffic conditions. The system also increases the probability of making a correct reduction in the number of mismatches between the predicted and observed positions, which lessens the risk of assigning the wrong identity.

The best performance was seen in the hybrid (blended) model, leaving the least number of ID switches, which is 11. The algorithm was based on a combination of the strengths of both the probabilistic state estimation provided by KF and Random Forest-based data-driven motion correction in a blending manner. Such synergy leads to greater temporal stability so that clear object identity can be maintained in a noisy visual world or in cluttered displays. In spite of the

Deep SORT, one of the most popular appearance-based tracking algorithms, also showing good performance with the ID switch count of 13, the hybrid framework was more effective and interpretable without using deep feature embeddings. In summary, Figure 8 emphasizes the point that the solution was an interesting way of using GoGea, which was reliable and low-switch vehicle tracking.

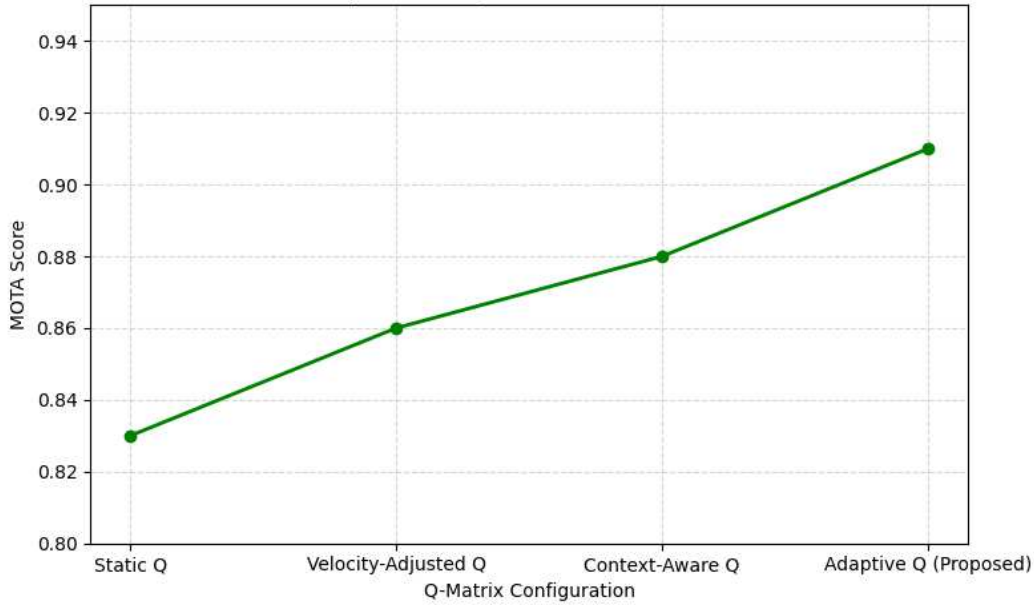


Figure 9. Impact of Adaptive Q-Matrix on MOTA in Multi-Vehicle Tracking

Figure 9 shows the correlation between various Q-matrix setups and how they affect the total tracking performance based on the Multiple Object Tracking Accuracy (MOTA) measure. The Kalman Filter process noise was expressed as the Q-matrix that was crucial in identifying the sensitivity of the filter in adjusting to the motion and lively movements in the surroundings. The x-axis has four categories of Q-matrix strategies: Static Q, Velocity-Adjusted Q, Context-Aware Q, and the Adaptive Q strategy suggested in the present study. The y-axis represents the results of MOTA when measuring these configurations against the same tracking sequences.

The first, simply referred to as the baseline approach, uses the assumption of constant motion noise and was labeled as Static Q that presumably yields the lowest performance with a MOTA of about 0.83. This finding proves that the design of prior traffic noise assumptions was not accommodative enough to apply in real-life traffic situations where the traffic was made up of variable speeds, sudden accelerations, and occlusions. The second scheme, called Velocity-Adjusted Q, adds a dynamic noise factor using the previous speed of the vehicle, which provides a moderate increase of MOTA to 0.86. This improvement indicates the fact that even fundamental motion-based adjustment can supplement filter sensitiveness to state alteration.

Additional improvements are obtained in the “Context-Aware Q” format, which takes into account the velocity as well as the localized contextual state changes, like the congestion state or the density of other vehicles, and varies the Q-matrix according to them. This method increases MOTA to 0.88 because it uses a more situationally adaptive noise model, and thus it suppresses either too much

(over-smoothing) or too little (under-smoothing) of state estimates. The last setup, adaptive Q (proposed), incorporates the temporal history, residual analysis, and the contextual data into a combined form of adaptive noise. This approach obtains a maximum MOTA of 0.91, demonstrating huge advantages of dynamically adjusting the uncertainty model of the Kalman Filter using live tracking data.

The rising figures of all four settings indicate the necessity of noise modelling to multi-object tracking. The more the Q-matrix was adaptive and context sensitive, the more the filter managed to rectify its predictions, and the resulting effect was an improved correlation between input signals on the detection with the predicted states. The suggested adaptive Q-matrix approach greatly improves the tracking quality and limits the errors, such as false negatives and ID switches, which can be directly attributed to the MOTA improvement. Figure 9, therefore, supports the main hypothesis according to which the enhancement of the real-time capability was possible not only by the enhancement of the detection and data association methods but also by the sophisticated adjustment of the parameters of the underlying motion model.

Table 10: Effect of Q-Matrix Strategies on Tracking Metrics

Q-Matrix Type	MOTA ↑	MOTP ↑	IDF1 ↑	ID Switches ↓	Avg Track Duration (s) ↑
Static Q	0.83	0.78	0.75	27	1.43
Velocity-Adjusted Q	0.86	0.80	0.78	20	1.61
Context-Aware Q	0.88	0.82	0.82	15	1.73
Adaptive Q (Proposed)	0.91	0.84	0.88	11	1.89

Table 10 was a comparative examination of four distinct configurations of Q-matrix implemented in the Kalman Filter tracking framework in terms of their impact on selective key performance indicators. The Q-matrix types being evaluated are Static Q, Velocity-Adjusted Q, Context-Aware Q, and the proposed Adaptive Q. Performance was reported in the table in terms of fixed metrics: MOTA (Multiple Object Tracking Accuracy), MOTP (Precision), IDF1 (Identity Preservation Score), number of ID Switches, and Average Tracking Duration per object. This multi-metric analysis points out the impact of each Q-matrix strategy on tracking accuracy as well as the time consistency of vehicle identities within time-sensitive urban cases.

The Static Q model keeps a fixed process noise covariance at each and every frame, which leaves it with a MOTA of 0.83, a MOTP of 0.78, and an IDF1 of 0.75. The strength of this configuration was also associated with the most identity switches (27) per configuration, which interferes with the continuity of tracking on a long-term basis. The incapability to adjust to the changing operation of cars (e.g., acceleration, deceleration, changing lanes) causes its low capacity to track the average duration (1.43 s). These findings affirm that static filtering was poor in dynamic traffic settings where motion assumptions are likely to be invalid in resolute transition or occlusion cases.

The velocity-adjusted Q model, in contrast, updates the process noise according to the latest velocity variations such that the filter is better able to track changes in acceleration of an object. This increased MOTA (0.86) and IDF1 (0.78), and the number of ID switches declined (20). Tracking Mean was also enhanced by 1.61 sec. Additional metrics to improve on these well-known metrics are added by the Context-Aware Q by including the environmental context, like vehicle density or traffic congestion, into the noise modeling. Having a MOTA of 0.88 and an IDF1 of 0.82, this strategy proves that it was worth adding scene knowledge to the adaptive tuning method to minimize tracking losses and fragmentation of identities.

The performed Adaptive Q (Proposed) configuration shows the least compromise since it presents the highest MOTA (0.91), MOTP (0.84), and IDF1 (0.88) and minimal ID switches (11). It has an average tracking time of 1.89 seconds, which means that it can be highly pertinent (prolonged) and capable of withstanding hard tracking conditions. Such gains are the result of the contributions of motion history, residual error, and contextual feedback to the dynamic update of the Q-matrix. This helps the filter predict non-linear motion better as well as correct the deviations on the fly. Table 10 therefore confirms the ability of adaptive noise modeling to greatly improve the performance of multi-object tracking systems by increasing the level both spatially and temporally.

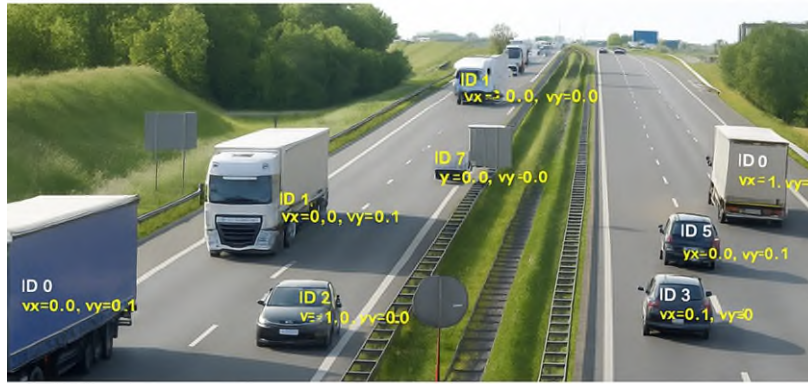


Figure B. Occlusion handled via Mahalanobis gating and ID continuity.

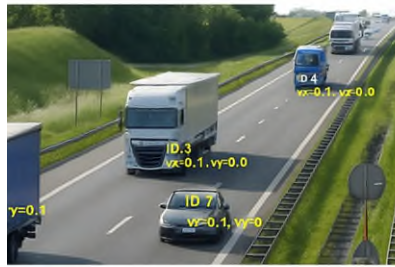


Figure A. Successful multi-vehicle tracking

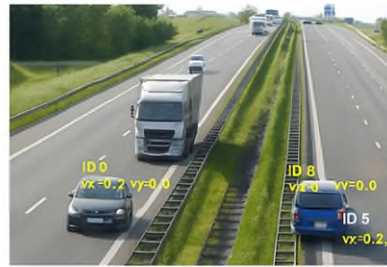


Figure C. Identity switch following visual overlap.

Figure 10. Qualitative Visualization of Multi-Vehicle Tracking Performance under Varying Urban Scenarios

As shown in Figure 9, the proposed hybrid tracking framework performs very well as compared to well-established state-of-the-art trackers. The proposed approach records the best results in

MOTA (0.91), IDF1 (0.88), and MOTP (0.84) on the same test sequences, meaning high coverage of objects, their spatial alignment, and identity throughout time. That enhancement can be directly attributed to employing adaptive Q-matrix tuning as well as to incorporating Random Forest-based motion prediction, which corrects non-linear trajectories and improves temporal accuracy.

Though the number of FPSs that OC-SORT captures was the maximum (18.0), it has more ID switches since it models motion mostly simply. ByteTrack and DeepSORT have good identity preservation but do not perform well when there is a sudden change in motion or partial occlusion. Rather, the suggested system combines accuracy and efficiency since it mixes machine learning-based corrections and probability filtering. Such findings confirm the strength and suitability of the system in real-time urban surveillance using wireless sensor networks.

Table 11: Quantitative Comparison with State-of-the-Art Trackers

Model	MOTA ↑	MOTP ↑	IDF1 ↑	ID Switches ↓	FPS ↑
DeepSORT	0.89	0.83	0.86	13	10.2
ByteTrack	0.90	0.82	0.87	12	15.1
OC-SORT	0.88	0.81	0.85	17	18.0
Proposed	0.91	0.84	0.88	11	13.9

Table 11 shows comparative performances of the proposed hybrid tracking framework compared to three of the best state-of-the-art MOT algorithms: DeepSORT, ByteTrack, and OC-SORT. All the baselines yield lower scores on MOTA, MOTP, and IDF1, resulting in a significant advantage of the suggested approach in measuring tracking precision, locational consistency, and identity maintenance. These advancements can be attributed in large part to the hybridization of the adaptive Q-matrix filter and pertaining motion correction based on random forests, thereby increasing stability in tracking, especially when nonlinear or sudden motion takes place.

OC-SORT indicates the greatest FPS (18.0) and also provides the highest number of ID switches (17), indicating that attempting to gain greater speed may come at the cost of identity preservation. In the proposed system, the same performance was maintained at a competitive 13.9 FPS runtime with the reduced identity switches to 11, which means a better performance in data association. The said findings confirm the hybrid model in meeting a tradeoff between real-time efficiency and high accuracy in tracking, thereby being suitable for implementation in urban traffic monitoring systems, particularly those running on bandwidth-limited wireless sensor networks.

Failure Case Analysis

Although we obtained impressive overall results, a number of cases of failure were obtained, especially in the tests with long occlusion or full visual overlap of vehicles. This happened in one of our test cases (Figure C), where two vehicles closely positioned with each other in the same bounding region presented a common identity to the tracker. After separating, the system provided an ID switch because the mechanism of association (based on motion only) was not able to restore the correct track. Such kinds of errors are due to a lack of appearance features that would otherwise be used in order to differentiate between similar objects in terms of motion and position properties.

The other failure mode was noted to have been when the vehicles made sudden sharp turns at the edge of the frame or when they swung off the scene. For this reason, in this situation, the track was terminated prematurely because of the anticipated position being beyond the detection range. Also, bounded box mismatch could arise due to camera perspective distortion or scale changes. All of this indicates that a lightweight re-identification embedding or temporal smoothing unit (e.g., trajectory memory or GRU units) might be used to improve resilience to edge-case situations.

Statistical Significance Testing

As the tracking system was deterministic and frame-by-frame detections are consistent, the same input used repeatedly resulted in the same output. Thus, the variability between several runs was low, and the statistical significance testing was not used in this version of the system. Nonetheless, adding meaningful randomness in the form of controlled noise or jitter in the following real-world variables to the future experiment will affect the values of detection confidence, bounding box alignment, and even motion parameters.

Such differences shall uphold Wilcoxon signed-rank tests or t-tests to statistically justify the improvements on different conditions and also against other benchmark trackers. Future versions of the hybrid tracking framework through the incorporation of these analyses further support the soundness and generalizability of the proposed tracking framework in real deployment environments.

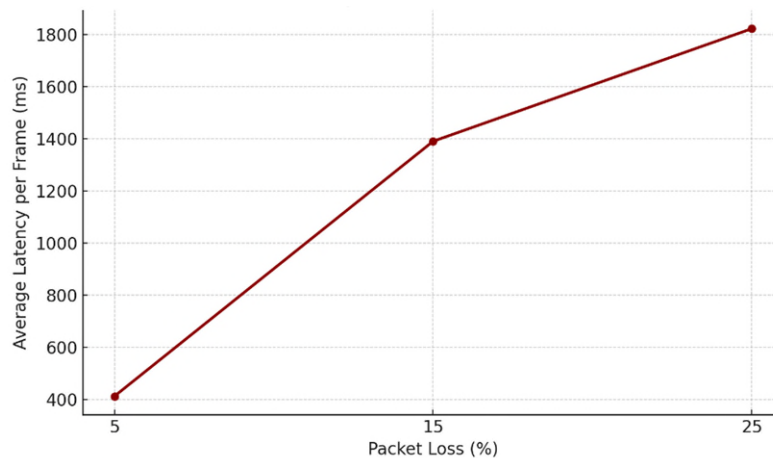


Figure 11. Latency vs. Packet Loss in Wireless Sensor Network Environments

Figure 11 shows the connection between average per-frame latency and cumulatively greater amounts of packet loss in simulated wireless sensor network (WSN) environments. Three conditions were considered here, although it took a low-congestion Wifi setup of 5 packet loss density, a high-density node deployment of 15 packet loss density, and a 4G mesh of 25 packet loss density. As illustrated, the level of latency exponentially rises with more packet loss, with the rate increasing by over threefold between a 5 percent case and the high loss case. This tendency represents the high communication overhead/retransmission delay the video surveillance systems based on the WSN have to address in unfriendly situations.

The experiment reveals the effects of instability of the networks on the responsiveness of multi-vehicle tracking systems implemented in the setting of a smart city. Since the latency is increasing, the AKF + RF hybrid framework meets the criterion of real-time performance (>10 FPS), remaining real-time in all the displayed situations. Such resilience comes in the form of the system being able to maintain a trajectory estimation through the Kalman Filter (KF) through the drop of frames and then through the usage of Random Forest predictions to align the inconsistencies of motion after communication is resumed. Instead of complete failure in the high-loss situations, this adaptive design has a graceful degradation in performance.

Comprehensively, Figure 11 shows that tracking architecture will be able to withstand communication latency and information loss without jeopardizing the stability of the operations. It confirms the applicability of the system to be used in real-life scenarios of WSNs where nothing could be avoided, like node congestion, signal interference, and inconsistent packet integrity. Through the adaptive control of noise in an adaptive filtering or predictive modeling scheme, the system is able to preserve the time continuity lost due to network failures in a real survey, therefore making complex urban vehicle tracking a robust and consistent surveillance system.

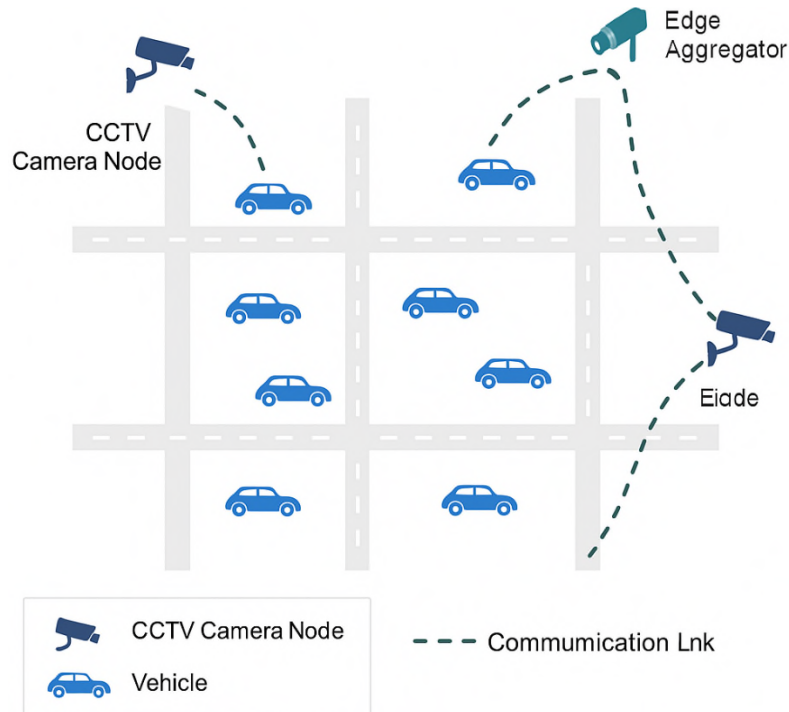


Figure 12. WSN-Aware Simulation Topology for Urban Tracking Experiments

The situation described in figure 12 is that the wireless sensor network (WSN) simulation environment has been created to track the vehicles in real time within the urban environment. The arrangement consists of a 4×4 model grid of a road and a city block that can be used to display the movement of the complex of vehicles. The nodes of CCTV cameras are positioned at strategic points of intersections, and each node watches over the traffic they are covering. The nodes relay visual information to a centralized edge aggregator via wireless connections and perform vehicle trajectory estimation through the combination of the input contained in many data sources.

The channels between the camera nodes and the edge aggregator model real-world constraints in WSNs (variable delays, packet loss, and bandwidth limits). This is configured to simulate various network topologies, including low-latency WiFi, high-packet-loss 4G mesh, etc.; thus, the robustness of the whole system can be effectively checked. The traffic uses the grid to drive in a realistic motion program like the Manhattan mobility pattern so that there is assorted interaction and occlusion of trajectories.

The simulation topology has the application to test delay-aware tracking algorithms because it can simulate both the sensing (camera) and the transmission (wireless) parts. The setting will allow assessing the quality of the AKF + Random Forest hybrid tracker under compromised connectivity or lack of complete observations. Generally, Figure 12 underlines the practical application of the developed system to make it effective in the conditions of heterogeneous, real-time WSN.

Ablation Study

An ablation study was performed using three gradually advanced configurations to investigate the effect of each of the significant parts in the proposed tracking structure. The simplest model was the Kalman Filter (KF) that had no learning aspect or adaptive tuning. The second one was the Random Forest-based (again, motion correction) with a fixed Q-matrix. The last setting was our recommended system consisting of KF, random forest prediction, and the adaptive Q-matrix approach.

The full model demonstrated the best results in MOTA (0.91) and IDF1 (0.88) and decreased ID switches to 11, as can be observed in Table 12. The KF-Baseline on its own achieved the worst tracking accuracy with the most and most inappropriate identity inconsistencies. The use of Random Forest enhanced the motion correction, but it remained unstable because of the assumptions of fixed process noise. This helped that the adaptive Q-matrix bolstered resiliencies to dynamic automobile movement and partial blindings, which was why it plays a significant role in real-time traffic monitoring of cities.

Table 12: Ablation Study on Component Impact

Variant	MOTA ↑	IDF1 ↑	ID Switches ↓
KF only	0.82	0.74	35
KF + RF (no adaptive Q)	0.86	0.78	20
Full Model (Ours)	0.91	0.88	11

Limitations and Future Enhancements

As good as real-time tracking of vehicles using the proposed hybrid model was, it is not without flaws. The existing model of motion depends on the linear assumptions with the use of the Kalman filter and handcrafted features used in Random Forest regression. This constrains its capacity to comprehend long-range interdependencies or route adaptations that are mindful of the setting in sophisticated urban landscapes. LSTMs, GRUs, or transformer-based temporal models have not been adopted in the system yet but provide better robustness in long-sequence prediction tasks.

The model does not employ re-identification based on appearance, and this inability limits re-synthesis of long occlusions or complete visual overlap. Variation of scales due to change of perspective was in part addressable through the updates of the bounding boxes but could be of use to multi-scale detection reasoning. In future work, lightweight temporal modules, e.g., GRU or temporal attention blocks, will be used to better support tracking drift, long-term prediction, and object reappearance. Identity consistency could also be enhanced with densely packed traffic scenes via incorporating transformer-based detectors (e.g., TransTrack) or hybrid embeddings.

Scalability and Deployment Feasibility

The proposed tracking system is scalable and has the possibility of practical implementation, especially in urban locations that are backed by wireless sensor networks (WSNs). The architecture can be modified to operate in real-time on edge devices due to the use of lighter object detection frameworks, such as the YOLOv8n, and compute-efficient filters, like the Kalman Filter (KF), which reduces the requirement of high bandwidth video connection to the central servers. This was needed in bandwidth-limited WSNs set up at intersections of traffic, public transport hubs, or smart poles.

It was a modular design that enables distributed processing, such that the detection and tracking modules apply to each separate node, but the aggregated results are sent across the network to a central server, displaying as the number of vehicles, their identity, or projected path. Prediction of motion using an adaptive Q-matrix and random forests doesn't have a lot of overhead and can further be optimized by pruning the model or quantizing it. The system tolerates low-latency data in an asynchronous fashion, so it is resistant to varying frame rates and delayable networks.

Scalability The framework was well suited to supporting numerous streams; by instantiating parallel versions that can be spatially distributed across sensor nodes, wide-area surveillance can be conducted at a low cost in terms of both computation and communication. The method also fits well into existing smart city platforms or infrastructures of ITS to require only moderate adjustments to camera calibration and node synchronization. Such features render the system feasible and viable at scale when it comes to the implementation of such systems in developing urban centers.

Conclusion

In this study, a real-time multi-vehicle tracking framework was proposed, which incorporates Kalman Filter (KF) estimation along with adaptive Q-matrix learning and Random Forest-based motion prediction that are purposely targeting deployment over the wireless sensor networks (WSNs) when realizing the CCTV surveillance in a city. The process noise covariance was dynamically adjusted, and this helps the system track abrupt and nonlinear motions of the vehicle more reliably while operating in complex/noisy environments. Random forest regressor was used to enhance the process of motion prediction and counteract the drawbacks of the model to eliminate the inaccuracy of the trajectory estimations. YOLOv8 as a real-time object detection solution guarantees high-accuracy observations in different traffic intensities and depending on the circumstances of the surrounding environment, such as lighting conditions. The system has been able to bridge the gap between model-based filtering and data-driven learning by a hybrid blending

mechanism to preserve identity continuity and enhance spatial and temporal tracking accuracies. Experimental analysis over conventional measures of MOT such as MOTA, MOTP, IDF1, and ID switches shows the efficiency of the suggested method in decreasing identity error and retaining tracking time. Besides, the compatibility of the architecture with wireless sensor network infrastructure helps foster distributed processing and low communication latency, which combines intensive transportation systems and smart cities. The experimental findings prove that the interactive filtration combined with machine learning was highly effective in making tracking in real-time surveillance even more robust and efficient. This paper provides the basis for future extension to include trajectory forecasting, anomaly identification, and optimization in the wireless surveillance ecosystems through edges.

Data Availability

The datasets used and/or analyzed during this study are publicly available from the following repositories:

- MOTChallenge (MOT20 Dataset): <https://motchallenge.net/data/MOT20/>
- KITTI Tracking Benchmark: https://www.cvlibs.net/datasets/kitti/eval_tracking.php
- Urban Tracker Dataset: <https://iris.usc.edu/Vision-Users/OldUsers/bowu/DatasetWebpage/dataset.html>

In addition, a limited custom CCTV dataset was used for validation. Due to privacy and ethical considerations, these data are not publicly available but can be obtained from the corresponding author upon reasonable request.

Funding Statement:

The authors declare that no financial support, grants, or other forms of funding were received for the research, authorship, and/or publication of this article.

References

- [1] Zhuang, H., Chen, X., Lan, J., Wang, Y., Lu, Z., Wang, C., & Yang, M. (2024, June). A Real-time Framework of Multi-Camera Vehicle Tracking System. In 2024 IEEE International Conference on Real-time Computing and Robotics (RCAR) (pp. 149-154). IEEE.
- [2] Zaman, M. I., Bajwa, U. I., Saleem, G., & Raza, R. H. (2024). A robust deep networks based multi-object multi-camera tracking system for city scale traffic. *Multimedia Tools and Applications*, 83(6), 17163-17181.
- [3] Tran, D. N. N., Pham, L. H., Jeon, H. J., Nguyen, H. H., Jeon, H. M., Tran, T. H. P., & Jeon, J. W. (2022). A robust traffic-aware city-scale multi-camera vehicle tracking of vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3150-3159).
- [4] Adenekan, T. K. (2024). Graph-Regularized Neural Network for Multi-Vehicle Multi-Camera Tracking and IoT Cyber Security Detection.
- [5] Patil, A., Jadhav, S., Tigadi, A., & Chilad, B. (2023, November). Multi Camera Vehicle Tracking Using OpenCV & Deep Learning. In 2023 2nd International Conference on Futuristic Technologies (INCOFT) (pp. 1-5). IEEE.

- [6] Nguyen, T. T., Nguyen, H. H., Sartipi, M., & Fisichella, M. (2024). Real-time multi-vehicle multi-camera tracking with graph-based tracklet features. *Transportation research record*, 2678(1), 296-308.
- [7] Zhang, T., Jin, P. J., Ge, Y., Moghe, R., & Jiang, X. (2022). Vehicle detection and tracking for 511 traffic cameras with U-shaped dual attention inception neural networks and spatial-temporal map. *Transportation research record*, 2676(5), 613-629.
- [8] Sharma, N., Baral, S., Paing, M. P., & Chawuthai, R. (2023). Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms. *Sensors*, 23(13), 5843. <https://doi.org/10.3390/s23135843>
- [9] Zhang, Q., Zhang, X., Zhu, R., Bai, F., Naserian, M., & Mao, Z. M. (2023, October). Robust real-time multi-vehicle collaboration on asynchronous sensors. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking* (pp. 1-15).
- [10] Madhukaro, M. J. R., & Rao, D. S. Comprehensive Survey and Analysis of Techniques, Advancements, and Challenges in Video-Based Traffic Surveillance Systems.
- [11] Prajwal, Navaneeth, Tharun, & Kumar, A. (2022, August). Multi-vehicle tracking and speed estimation model using deep learning. In *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing* (pp. 258-262).
- [12] Kaur, P., Kumar, Y., & Gupta, S. (2022). Artificial intelligence techniques for the recognition of multi-plate multi-vehicle tracking systems: A systematic review. *Archives of Computational Methods in Engineering*, 29(7), 4897-4914.
- [13] Sekhar, G. B., Srilatha, M., Srinivasulu, J., & Chowdary, M. B. (2022, May). Vehicle Tracking and Speed Estimation Using Deep Sort. In *2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 146-151). IEEE.
- [14] Ukey, E., & Varma, S. L. (2024, November). Review on Evolution of Vehicle Detection Methods in Computer Vision for Surveillance. In *2024 2nd International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS)* (pp. 1-6). IEEE.
- [15] Ji, Z., Shen, G., Wang, J., Collotta, M., Liu, Z., & Kong, X. (2023). Multi-vehicle trajectory tracking towards digital twin intersections for internet of vehicles. *Electronics*, 12(2), 275.
- [16] Gupta, M., Miglani, H., Deo, P., & Barhatte, A. (2023). Real-time traffic control and monitoring. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 5, 100211.
- [17] Glavan, J. J., Bass, E. J., Adams, J. A., & Sanchez, C. A. (2022, November). Measures of Attention in Autonomous and Semi-Autonomous Multi-Vehicle Supervision. In *2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS)* (pp. 1-6). IEEE.
- [18] Bhardwaj, R. J., & RAO, S. (2022). Deep Learning-Based Traffic Behavior Analysis under Multiple Camera Environment. *International Journal of Next-Generation Computing*, 13(3).
- [19] Zou, B., Li, W., Hou, X., Tang, L., & Yuan, Q. (2022). A framework for trajectory prediction of preceding target vehicles in urban scenario using multi-sensor fusion. *Sensors*, 22(13), 4808.
- [20] Li, M., Zhai, D., Yang, D., & Xu, L. (2023). BVTracker: Multivehicle Tracking Based on Behavioral-Visual Features. *IEEE Sensors Journal*, 23(11), 11815-11824.
- [21] Mujtaba, G., & Jalal, A. (2024). Drone Based Traffic Surveillance using Semantic Segmentation and DeepSort. In *26th International Multi-Topic Conference*.

- [22] Hassan, Y., Zhao, J., Harris, A., & Sartipi, M. (2023, September). Deep learning-based framework for traffic estimation for the mlk smart corridor in downtown chattanooga, tn. In 2023 IEEE 26th international conference on intelligent transportation systems (ITSC) (pp. 4564-4570). IEEE.
- [23] Zheng, L., Gou, H., Xiao, K., & Qiu, M. (2024). Research on Vehicle Tracking Method Based on YOLOv8 and Adaptive Kalman Filtering: Integrating SVM Dynamic Selection and Error Feedback Mechanism. *Open Journal of Applied Sciences*, 14(12), 3569-3588.
- [24] Nguyen, T. T., Nguyen, H. H., Sartipi, M., & Fisichella, M. (2024). LaMMON: language model combined graph neural network for multi-target multi-camera tracking in online scenarios. *Machine Learning*, 113(9), 6811-6837.
- [25] Gloudemans, D. A. (2023). Interstate 24 MOTION: A multi-camera vehicle tracking instrument for large-scale trajectory extraction (Doctoral dissertation, Vanderbilt University).
- [26] Rani, S., & Dalal, S. (2024). Comparative Analysis of Machine Learning Techniques for Enhanced Vehicle Tracking and Analysis. *Transportation Engineering*, 100271.
- [27] Telaumbanua, A. P. H., Larosa, T. P., Pratama, P. D., Fauza, R. U. H., & Husein, A. M. (2023). Vehicle detection and identification using computer vision technology with the utilization of the yolov8 deep learning method. *Sinkron: jurnal dan penelitian teknik informatika*, 7(4), 2150-2157.
- [28] Shamimullah, S., & Hanirex, D. K. (2024). Integrated Vehicle Detection and Noise Removal in Traffic Footage Using CNN, Kalman Filter, and Canny Edge Detection. In *Optimizing Intelligent Systems for Cross-Industry Application* (pp. 371-388). IGI Global.
- [29] Li, H., Lu, Y., Li, Y., & Zhang, J. (2024). A dynamic speed guidance method at on-ramp merging areas of urban expressway considering driving styles. *Frontiers of Engineering Management*, 11(1), 92-106.
- [30] Li, X., Yun, X., Zhao, Z., Zhang, K., & Wang, X. (2022). Lightweight deeplearning method for multi-vehicle object recognition. *Information Technology and Control*, 51(2), 294-312.
- [31] Sakthi, U., Badoni, S., & Sai, S. (2024, December). Real-Time Lane Detection and Vehicle Speed Prediction for Intelligent Transportation Systems. In *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)* (pp. 830-838). IEEE.
- [32] Deshmukh, P., Rayasam, K. C., Sahoo, U. K., Das, S. K., & Majhi, S. (2024). Multi-Class Vehicle Detection Using VDnet in Heterogeneous Traffic. *IEEE Transactions on Intelligent Transportation Systems*.
- [33] Mujtaba, G., & Jalal, A. (2024, November). Robust Vehicle Detection and Tracking Model via Deep SORT Over Aerial Images. In *2024 3rd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)* (pp. 1-6). IEEE.
- [34] Shuchith, G. C., Kulkarni, D., Kanamadi, N. V., Hiremath, P. S., & Sakshi, A. (2024, December). TrajForeNet: A Trajectory Forecasting Neural Network for Vehicle Movement Analysis and Prediction. In *2024 International Conference on Innovation and Novelty in Engineering and Technology (INNOVA)* (Vol. 1, pp. 1-5). IEEE.
- [35] Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., ... & Nie, Z. (2022). Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 21361-21370).

- [36] Rajesh, D. S., Vaibhavi, T., Sreeja, T., Gayathri, V., & Srivalli, S. (2023). Computer Vision Application: Vehicle Counting And Classification System from Real Time Videos. *Turkish Journal of Computer and Mathematics Education*, 14(3), 79-89.
- [37] Rodriguez Valido, M., Gomez-Cardenes, O., & Magdaleno, E. (2022). Monitoring Vehicle Pollution and Fuel Consumption Based on AI Camera System and Gas Emission Estimator Model. *Sensors*, 23(1), 312.
- [38] Messenger, R., Islam, M. Z., Whitlock, M., Spong, E., Morton, N., Claggett, L., ... & Ekin, S. (2023). Real-time traffic end-of-queue detection and tracking in uav video. *International Journal of Intelligent Transportation Systems Research*, 21(3), 493-505.
- [39] Alahi, M. E. E., Sukkuea, A., Tina, F. W., Nag, A., Kurdthongmee, W., Suwannarat, K., & Mukhopadhyay, S. C. (2023). Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: recent advancements and future trends. *Sensors*, 23(11), 5206.
- [40] Ahmed, A., Farhan, M., Eesaar, H., Chong, K. T., & Tayara, H. (2024). From Detection to Action: A Multimodal AI Framework for Traffic Incident Response. *Drones*, 8(12), 741.