International Journal of Basic and Applied Sciences, 14 (7) (2025) 348-361

SPC

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/4ktqvv23
Research paper

Comprehensive Review of Machine Learning Techniques for Vehicle Tracking in A Smart Environment

Seema Rani *, Sandeep Dalal

Department of Computer Science & Applications Maharshi Dayanand University,
Rohtak, Haryana, India
**Corresponding author E-mail: seema.rs.dcsa@mdurohtak.ac.in*

Received: August 20, 2025, Accepted: September 27, 2025, Published: November 14, 2025

Abstract

Real-time vehicle tracking has become very important for managing traffic, keeping an eye on safety, and letting cars find their own way, thanks to the fast growth of intelligent transportation systems (ITS) and urban traffic surveillance. Deep learning techniques have made a big difference in the field by making solutions that work in complicated, changing situations where traditional methods often fail. This work looks at all the latest deep learning-based vehicle tracking methods, including single-object tracking (SOT), multi-object tracking (MOT), and hybrid models that use both spatial and temporal data. In this review, we look at more than 90 studies from 2020 to 2024, focusing on different types of structures like CNNs, RNNs, and transformer-based models. The best tracking systems, such as DeepSORT, Byte Track, FairMOT, Trans Track, and CNN-LSTM hybrids, are tested for their accuracy, computational efficiency, ability to generalize datasets, and real-time performance. Problems like occlusion, identification switching, and tracking drift are discussed, especially when there is a lot of traffic in a city. Lastly, the study points out areas where more research is needed and suggests ways to go in the future to make vehicle tracking systems that are flexible, light, and aware of their surroundings so they work best in smart cities.

Keywords: *Deep Learning; Vehicle Tracking; Intelligent Transportation Systems; Multi-Object Tracking; Computer Vision; Smart Cities.*

1. Introduction

There needs to be a useful way to track cars as smart towns grow and more cars hit the roads around the world. Many important things depend on being able to keep an eye on cars, like police, fleet management, self-driving cars, and smart traffic lights. This is due to the fact that the traditional computer vision tools and features, which were developed manually to monitor cars, often struggle to function effectively in real-world scenarios. These tools cannot be effectively used on a larger scale, with varying car appearances, or under different sizes and lighting conditions. Because of these problems, they're not as useful. In the beginning, classical vision methods like background subtraction, frame differencing, and optical flow were used to determine moving vehicles. Thereafter, these were put together with custom features like HOG (Histogram of Orientated Gradients) and SIFT, as well as machine learning classifiers like SVMs and Random Forests, to make the identification more accurate. Motion prediction was usually done with Kalman filters or particle filters, which were good for simple traffic scenes but not so good for settings with many people or things in the way. These old-fashioned and machine learning-based methods were fast on computers and easy to understand, but they weren't reliable in the real world. After 2012, deep learning became more popular, which made automatic feature learning possible through CNNs and RNNs. The result was a big change from tracking vehicles by hand to tracking them based on data. Deep learning has revolutionised car tracking by simplifying the extraction of desired traits with reduced model time. Convolutional Neural Networks (CNNs) like ResNet, EfficientNet, and MobileNet are now used by many people to get the visual features they need for real-time and embedded systems. Moving picture predictors (mostly Kalman filters) and data association algorithms (like the Hungarian algorithm) help tracking systems like DeepSORT and ByteTrack make sure the cars don't move from one frame to the next. These new mixed methods use Long Short-Term Memory (LSTM) networks or transformer-based models to make time make more sense and cut down on the number of identification changes that happen over long runs. Problems still include picture complexity, persistent occlusions, tracking drift, and the speed of processing. This phenomenon is especially true when devices that aren't very powerful are used in real time. This research looks at all the deep learning-based ways to keep track of cars that came out between 2020 and 2024. This research focuses on tools that enable the tracking of both individual items and groups of items. We look closely at the YOLOv8 with DeepSORT, the ByteTrack, the FairMOT, and trackers that use transformers as some of the best examples [1]. We discuss their pros and cons and how to use them in real city traffic. Figure 1 shows several ways to keep track of a car.

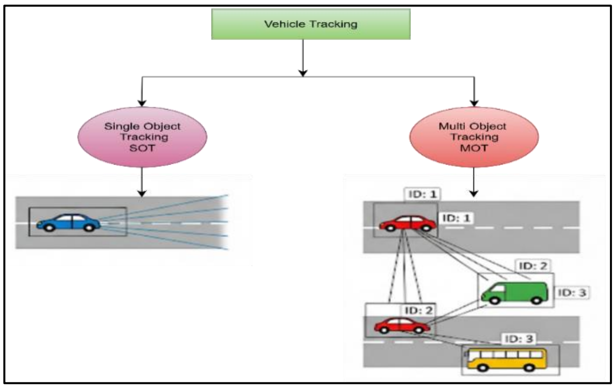

Fig. 1: Vehicle Tracking.

The rest of the paper is organised as follows: Section 1 introduces the purpose, objectives, and extent of vehicle tracking research, while Section 2 reviews relevant literature using both traditional and modern approaches. Section 3 outlines the methodology, including datasets, evaluation metrics, and the experimental environment. Section 4 discusses the fundamentals and traditional methods of vehicle tracking, whereas Section 5 discusses machine learning-based methods. Section 6 introduces deep learning-based techniques, while Section 7 emphasises hybrid approaches that integrate many techniques. Section 8 addresses the primary issues and challenges in vehicle monitoring, while Section 9 provides a comparative analysis of several systems, evaluating their efficacy, benefits, and drawbacks. Section 10 concludes the paper and discusses prospects with an emphasis on context-aware, lightweight, multi-modal tracking systems for intelligent mobility.

2. Related Work

Single Object Tracking (SOT) and Multiple Object Tracking (MOT), both used to track cars, have seen significant advancements in deep learning. CNN (Convolutional Neural Network) designs like ResNet, EfficientNet, and MobileNet make images look good while using as few computations as possible, which is important for real-time intelligent transportation systems (ITS) [2]. When tracking systems like DeepSORT and ByteTrack use motion models (like the Kalman filter) and data link algorithms (like the Hungarian algorithm), they do a great job of making sure that all video frames show the same car [3]. Using deep look descriptions, DeepSORT makes teeth less likely to become stuck together. LSTM-based models effectively capture long-term temporal links and interactions, preventing identity loss over time [4]. Transformer-based and mixed models have recently shown better accuracy in predicting complex relationships between time and environment. This makes tracking more accurate and stable in tough urban settings [5]. Things are still moving slowly because of issues like frequent occlusions, changes in scale, complicated urban settings, and strict real-time computer limits. Several recent studies address various facets of vehicle tracking and accident detection:

1) Josephinshermila et al. (2023discussed how to create and utilise a complex tracking and alarm system for monitoring car parameters, including accident detection and oversight of various vehicle metrics. The technology keeps the data logs on a safe digital memory card and in the cloud, and it keeps an eye on the vehicle's parameters all the time. It also actively looks for accidents that happen without warning and can quickly study and report any accidents to emergency services. A black box in the car would record information about how the driver was going, like speed, acceleration, stopping, steering, and airbag deployment before, during, and after a crash. Researchers have found that the suggested way of finding accidents is more accurate than RFID, SVM, CNN, and RNN methods [5].
2) Pardeshi et.al (2023) A collision site warning system is made by this system, which uses Arduino, detectors, cellular, and a global positioning system (GPS). When something goes wrong, the warning goes off on its own. The details of the accident, with the car number and the name of the person who was hurt, will be sent to the nearest police office and hospital with a Google Maps link. Once we have the information, the treatment facility staff will find the person or staff member who is closest to the scene of the accident so that they can get there quickly. This will give medical staff more time to help those who were hurt, lower the number of accidents that end in death, and shorten the time that traffic is blocked [6].
3) Chan et.al (2023) The study examined a collection of serious incidents involving highly automated vehicles, although the dataset is relatively small due to current strict regulations on these vehicles worldwide. Both illustrative and deductive statistics are presented regarding the frequency of incidents as influenced by such contextual factors as the topography, time frame, perceptibility, atmospheric state, automated vehicle velocity, automation system condition, incident category, and road bend. The main discovery is that the proportion of the incident tally with the automation system condition being activated to that with it being deactivated (i.e., manual operation in and of itself) is significantly reduced at a high velocity (74 km per hour or higher) of the automobile compared to at a low velocity. Present-day automated systems are significantly more technologically, lawfully, economically, and monetarily advantageous and beneficial to individuals for moving on conveyances at accelerated rates [7].
4) Yan et al. (2023) When railway systems move into a new era of advanced automation, safety and stability become more and more important. But there are still not many smart and effective ways to avoid train accidents, especially proactive ways to avoid accidents. This article explained a way to keep trains from wrecking by using the reinforcement learning framework and different types of data to make dynamic methods to avoid train wrecks. To show how well dynamic deterrent methods work, three indicators were created [8].
5) Beck et al. (2023) provided a way to look at accidents involving Automated Vehicles (AVs) by handling raw data from devices on board. The conduit turns sensor data into visual results that crash site investigators can look at without having to know how the car's sensor system works. By looking at real-life AV accidents in California and using fake data from the CARLA simulator, the pipeline's value is shown. Data visualisation and analysis demonstrate significant enhancements in accident investigations through the use of advanced sensing and thinking systems in self-driving cars [9].
6) Karthik et al. (2023). The rapid development of technology has greatly changed our lives, making things easier and more convenient. Still, even though we live in a time of growth, the number of car accidents keeps going up, which is mostly because emergency services take too long to arrive. Our project came up with a complete answer to this important problem. When a car accident happens, the MEMS sensor picks up the hit and sends the information to an Arduino unit, which then processes it. When this is set up, an alert message with the exact position is sent through the GSM module to either the police control room or the rescue team. Once the cops

have this information, they can quickly use the GPS module to find the scene of the accident. So, as soon as the spot is confirmed, action can be taken right away. The project successfully alerts both emergency services and the affected family members, allowing quick help through the sharing of locations in the event of an accident [10].

7) Das & Kumar,(2023) Cyberattacks can happen to connected autonomous cars (CAVs) because the communication system doesn't have enough security measures. Real-time intrusion detection is needed to avoid deadly collisions. The writers offered a way to find intrusions using Logical Analysis of Data (LAD), a method that takes a set of historical CAN messages and pulls out rules for telling the difference between normal and abnormal behaviour in CAVs. The suggested way gets a higher F1 score than the others because it can be detected in less than 47.5 seconds on a Raspberry Pi. Real-time intrusion monitoring is what keeps CAVs safe [11].
8) Liao et al., (2023) A key part of traffic accident management (TIM), which is an important job for traffic management agencies, is finding crashes quickly and correctly. Effective TIM strategies not only lessen the bad effects of one-time events, but they also make transportation more resilient, passengers happier, and the general quality of service better. Few studies have looked at how accidents affect the environment [12]. Thus far, the majority of studies have concentrated on identifying accidents and enhancing road safety.
9) Kumar et al. (2022). This article discussed an Internet of Things (IoT) system for Android phones that can automatically find and sort car accidents into eight different groups. By transmitting accident information to rescue services and affected families, the system aims to reduce the number of fatalities. They use a method that combines different sensors with simple techniques and a stacking approach based on logistic regression to bring together machine learning classifiers like Decision Tree, Naïve Bayes, and Random Forest. A score of 0.95 on the F1 test shows that the stacked classifier is more accurate than single classifiers [13].
10) Attar et.al (2022) "Real traffic surveillance video requires continuous monitoring to observe and respond appropriately in the event of catastrophic events. However, it is time-consuming and error-prone to regularly monitor them under human supervision. Therefore, a deep learning approach for the automatic identification and placement of traffic accidents has been suggested by framing the issue as anomaly detection. Using a spatio-temporal encoder and a sequence-to-sequence extended short-term memory encoder, the technique uses a single-category classification strategy to describe the spatial and temporal representations in the video. Using the prototype on actual video traffic monitoring datasets has shown significant results in terms of both number and quality, as shown in Table 1 [14].

Table 1: Summary of Prior Reviews, Key Studies, and Research Gaps in Deep Learning–Based Vehicle Tracking

Reference & Years	Type	Focus / Scope	Key Contributions / Focus Areas	Identified Gaps / Limitations
(2024) [15]	Broad Survey	Vision-based Single Object Tracking and Multi-Object Tracking, multi-modal integration	Comprehensive analysis of over 230 studies on object and vehicle tracking	Limited vehicle-centric approach; lack of quick deployment context
Sharma et al., 2022 [16].	Survey	ML, DL, and classical methods for tracking objects	Coverage of SOT and MOT in both traditional and deep learning methods	Doesn't go into enough depth about transformers or the problems they can cause in real time
Wang et al., 2023[17].	Systematic Review	Tracking a vehicle in traffic CCTV using CNN (Convolutional Neural Network) and mixed methods	tracking methods based on CNN and hybrids; detection-tracking pipelines	Recent lightweight CNN and transformer-based trackers are missing.
Haque et al., 2021 [18]	Narrative Review	Tracking systems for multiple objects, such as DeepSORT	DeepSORT and video analytics are important things to keep focused on	It's not very broad; it doesn't cover mixed and bidirectional tracking.
Ma et al., 2024 [19]	Review	Tracking models based on transformers	CNN-RNN and transformer trackers side by side	Specific to architecture, no full category
Deep Reinforcement Learning Survey (2020) [20]	Review	DRL in tracking vehicles through a wireless sensor network (WSN)	Focus on reinforcement learning methods that are useful for certain situations	Out of date compared to CNN and transformer improvements
Josephinshermila et al., 2023 [21]	Experimental Study	Accident detection and tracking of vehicles	Real-time tracking of vehicle parameters and an accident detection device	System design was the main focus, and comparisons with DL tracking models were restricted.
Pardeshi et al., 2023 [22]	Experimental Study	Arduino and GPS are used in this collision site warning device.	An automated method for alerting people to accidents that works with emergency services	focused on hardware; minimal deep learning integration
Chan et al., 2023 [23]	Statistical Analysis	Analysis of a dataset of automated car accidents	Frequency of incidents linked to speed and automation triggering	The dataset is small and only includes AV crashes.
Yan et al., 2023 [24]	Method Development	Learning through reinforcement to stop train accidents	Dynamic ways to stop accidents using the RL system [4]	designed to work with train systems
Beck et al., 2023 [25]	Experimental Study	Visualization of AV crash data	Processing pipeline for sensor data for crash analysis [5]	Rather than monitoring the efficiency of algorithms, concentrate on their presentation.
Karthik et al., 2023 [26]	Experimental Study	MEMS sensor-based accident notice system	Real-time accident detection and alerting through the use of GSM and GPS modules [6]	Limited to an alert system; no deep learning monitoring component
Das & Kumar, 2023 [26]	Security Study	Finding intrusions in self-driving cars that are related	Find CAN bus intrusions in real time with logical data analysis [7]	Focus on security, not tracking per second
Liao et al., 2023 [27]	Review	Traffic accident control and crash spotting	It is stressed that TIM methods and their impact on the world are important [8].	A lower priority focused on systems for tracking vehicles.
Kumar et al., 2022 [28]	Experimental Study	A method for classifying accidents based on IoT	Using stacked machine learning models and sensor fusion to sort accidents [9]	Instead of tracking, pay attention to identifying crashes.
Attar et al., 2022 [29]	Experimental Study	Finding strange things in traffic camera videos	LSTM and spatio-temporal encoders are used in a deep learning method [10].	Always focus as much on constant tracking as you do on finding oddities.
This Study (2025)	Systematic Survey	Vehicle tracking based on deep learning (SOT, MOT, Hybrid)	Detailed classification and comparison, with transformers included	It focuses on ITS limitations in real time and the newest deep learning models.

3. Methodology

The method used in this work is a thorough study, which brings together changes in car tracking from 2024 to 2025. The time range was chosen to show both the rise of standard vision-based methods in the early 2010s and the rapid development of deep learning and mixed models in the last few years.

3.1. Literature sources and search strategy

We found relevant studies on Scopus, IEEE Xplore, ScienceDirect, SpringerLink, and Google Scholar by searching with the terms "intelligent transportation systems", "multi-object tracking", "YOLO detection", and "vehicle tracking". These libraries were picked because they have a lot of peer-reviewed studies in engineering and computer science.

3.2. Categorization of studies

To make the research easier to understand, the chosen studies were put into three main groups based on how they were done: Traditional methods: Early methods like the Kalman filter, visual flow, and background removal depend on motion modelling and features that are made by hand.ML techniques, like Support Vector Machines (SVM) and Random Forests, rely on handcrafted feature extraction and pattern detection. They are classification-based methods that were first used. Modern techniques in deep learning (DL) include the YOLO series, DeepSORT, ByteTrack, FairMOT, and transformer-based designs, which combine tracking and recognition in full pipelines. This classification shows how tracking has improved over time by becoming more reliable, scalable, and able to be used in real time. It does this by showing the evolutionary shift from feature engineering to feature learning.

3.3. Analytical perspective

Both qualitative and numeric points of view are emphasised in the study. Table 7 shows a summary of the performance measures that were used in the original studies. Qualitative analysis looks at trade-offs like accuracy vs. speed, stability vs. computing demand, and real-time usability.
By doing things this way, the review doesn't claim to be structured in the PRISMA sense. Instead, it's a thorough summary that puts scientific progress in its proper context and points out unresolved problems in car tracking. Figure 2 shows evolution tracking techniques.

3.4. Evolution of vehicle tracking technologies (2010–2025)

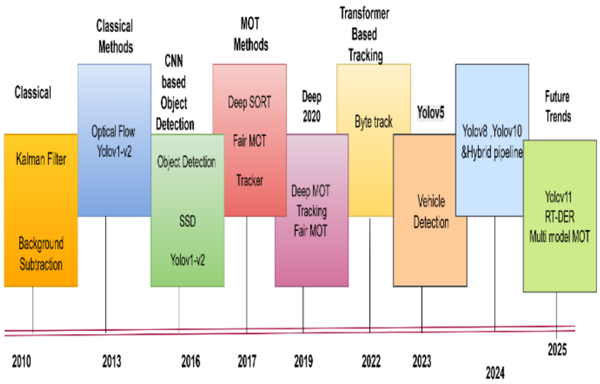

Fig. 2: Evolution of Vehicle Tracking Techniques.

2010 – Classical Methods Kalman filters and background subtraction were the primary algorithms employed in the initial vehicle tracking systems. These algorithms were rule-based and lightweight. These methods were computationally efficient and easy to implement; however, they were not as robust and encountered challenges when faced with occlusion, illumination variations, and complex motion patterns.
2013 – Optical Flow Integration Tracking was improved by optical flow techniques (e.g., Lucas–Kanade), which enabled motion-based object tracking without the necessity of explicit object detection in each frame. Despite the fact that these innovations enhanced the efficiency of the tracking process, the methods continued to encounter challenges in environments that were either congested or swiftly changing.
2016 – CNN-Based Object Detection. The introduction of YOLOv1–v2 and SSD (Single Shot Detector) marked the transition to deep learning-based object detection as the foundation of tracking. These models were capable of directly identifying vehicles from unprocessed images, which increased their robustness in a diverse range of environments. Nevertheless, monitoring was frequently implemented as an independent procedure after detection during this phase.
2017 Multi-Object Tracking (MOT) Methodologies. To incorporate detection and monitoring into unified pipelines, frameworks such as DeepSORT, FairMOT, and Tracktor were created. Particularly in multi-vehicle scenarios, they implemented motion models in conjunction with appearance embeddings to improve the consistency of identities and the accuracy of monitoring systems.
2019: The Era of Deep MOT Tracking. Deep learning has become the predominant method of tracking, as models like FairMOT have incorporated detection and re-identification (Re-ID) within a single network. This significantly improved MOTA (Multiple Object Tracking Accuracy) and reduced ID switches, thereby facilitating robust performance in challenging datasets such as UA-DETRAC and KITTI.
2023 – YOLOv5 for Vehicle Detection YOLOv5 facilitated real-time vehicle tracking by facilitating faster inference rates, which simplified object detection, even for embedded systems with GPU support. The integration of prominent trackers, including Byte Track and DeepSORT, was made possible by its modular architecture.
2024 Transformer-based Tracking and Hybrid Pipelines Transformers have had a substantial influence on the tracking domain, as evidenced by architectures like ByteTrack and multi-modal fusion pipelines. By implementing contextual attention mechanisms and utilising multi-source sensor data, YOLOv8, YOLOv10, and hybrid detection–tracking systems began to dominate benchmarks.

2025 – Future Trends (RT-DETR & YOLOv11) Recent advancements concentrate on RT-DETR (Real-Time Detection Transformer), YOLOv11, and multi-modal MOT (Multi-Object Tracking). This project advances the development of end-to-end real-time tracking that is capable of managing a wide range of weather, illumination, and traffic conditions with minimal latency by incorporating visual, LiDAR, and radar data streams.

4. Vehicle Tracking and Its Approaches

Vehicle tracking is the process of constantly finding, identifying, and locating moving vehicles over a series of video frames. It is an important part of real-time tracking, driverless guidance, monitoring of traffic violations, and clever transportation data [1]. The whole process includes finding cars in separate frames, putting together detections over time, keeping unique identifiers, and guessing where the car is even when it's partly or fully hidden. There are three main types of tracking methods: classical tracking, tracking based on machine learning, and tracking based on deep learning [29]. Figure 3 shows a complete classification of car tracking methods, showing how they have changed over time from motion-based algorithms that were built by hand to modern systems that are powered by deep learning.

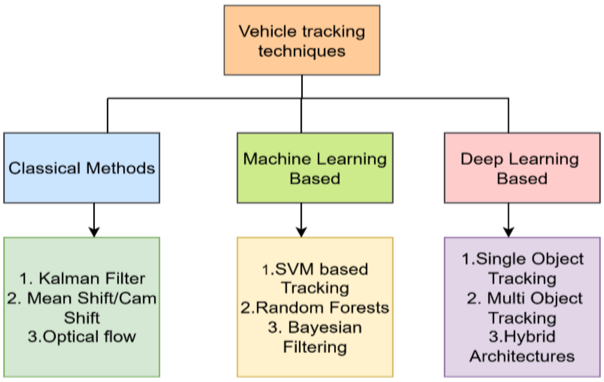

Fig. 3: Vehicle Tracking Techniques.

4.1. Traditional tracking methods

Traditional tracking methods mostly use mathematical models and traits that were made by hand to figure out where and how a car is moving.

4.2. Kalman filter

This is a recursive linear predictor called the Kalman filter. It is based on the idea of Gaussian noise and is often used to keep track of items. It predicts a car's future speed and location by analysing its past movements and adding more data. Because it works quickly and doesn't get thrown off by measurement noise, it is the best choice for real-time tracking. But it only works well for straight lines or Gaussian curves [30].

4.3. Mean shift algorithm

The mean-shift algorithm is a repetitive, non-parametric way to determine the peak of a probability density function. Colour histograms are often used in this process. It does a competent job of identifying cars with similar features, but it has trouble with differences in size and partial barriers. CamShift (Continuously Adaptive Mean-Shift) improves this by changing the size of the search window in a way that improves it for targets that are different sizes in the picture [29-30].

$$y_{k+1} = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

4.4. Optical flow

Lucas and Kanade figure out pixel-level motion between two frames by looking at the colour changes. The Lucas–Kanade method uses a small area to see small, steady motion patterns. This technique proves that it can be used to quickly track down car traits. But it is prone to big shifts and changes in light [31].

$$I_x u + I_y v + I_t = 0 \tag{2}$$

4.5. Background subtraction

This method analyses moving objects by modelling a stable background and detecting unusual changes over time. It works well in controlled settings, like highway security, and is proficient at using computers efficiently. But it has trouble with backgrounds that change over time, like trees that sway and shadows, and it has to keep adapting to new lighting [30-31].

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)\,p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{3}$$

5. Machine Learning-Based Tracking

Trackers that use machine learning have features and models that were intentionally made to tell the difference between cars and other things in the background or to identify trends of movement. Deep learning has gotten all the attention lately, but these methods have made it possible to track things very well.

5.1. Support vector machine (SVM) tracking

Support Vector Machines (SVMs) are supervised models that use features like Histogram of Orientated Gradients (HOG) or Local Binary Patterns (LBP) to tell the difference between cars and other objects or the background. Support Vector Machines (SVMs) are often used with online learning in tracking to address changes in look-over tracking cycles [32]. This is how the SVM decision function is usually written:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{4}$$

5.2. Random forest tracking

Random Forests use a variety of decision trees to help with classification or regression. In car tracking, they are taught to tell the difference between places with and without vehicles. This technique helps them make a good re-identification in hectic settings. They are more likely to generalise because they work together, but this takes more work to compute [32-33]. Usually, the mean of the results from each tree is the expected class.

$$\hat{y} = \mathrm{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{5}$$

5.3. Bayesian filtering

Bayesian filters, which include particle filters, are used to get a rough idea of how likely it is that a vehicle's state will change over time. They need more computing power because they use data from more than one state, but they are better than Kalman filters at handling nonlinear motion and non-Gaussian noise [16]. Table 2 shows machine learning applications for vehicle tracking.

$$p(x_k \mid z_{1:k}) = \frac{p(z_k|x_k) \int p(x_k|x_{k-1})\, p(x_{k-1}|z_{1:k-1})\, dx_{k-1}}{p(z_k|z_{1:k-1})} \tag{6}$$

Table 2: Machine Learning Applications in Vehicle Tracking and Data Analysis

References	Machine Learning Algorithm	Application in Vehicle Tracking	Effectiveness
[2]	Anomaly Detection	Real-time detection of strange driving behaviours, such as quick changes in direction or hard stopping. - Early detection of possible problems with tools. — Active security steps to stop people from trying to get in without permission.	Very good at finding oddities and deviations from the norm. To avoid fake results, the needs must be carefully changed.
[3]	Route Optimization	Moves goods in the most efficient way possible based on current traffic, weather, and past trends. Cuts down on fuel use, upkeep costs, and journey time.	It can make routes much more efficient, especially in areas with a lot of people. Needs to have access to up-to-date, correct traffic info.
[4]	Predictive Maintenance	Look at old information about how vehicles were used and how well they worked to guess when machines might break down. Scheduling preventative repair ahead of time can help cut costs and downtime.	Shown to be very good at reducing unplanned repair events. Needs big samples and constant model training.
[5]	Analysis of Driver Behaviour	Finds dangerous driving habits, like driving while drunk, fast, or tired. Offers tailored driving training to improve behaviour. Lowers the number of accidents and makes people more careful when they drive.	Helpful for finding dangerous driving habits. We need ways for drivers to give feedback and clear descriptions of dangerous behaviours.
[6]	Classification	Sorts different types of vehicles into groups so that traffic can be managed better and there is less congestion. - Finds stolen cars by using real-time position tracking and info from the past. - Sorts and groups driving behaviours, like being violent and being eco-friendly.	Very good when trained on big data sets with unique labels. Need to keep updating their models in order to be effective.

6. Deep Learning-Based Tracking

Deep learning has changed the field of car tracking by making it easier to learn both spatial features (from CNNs) and temporal motion patterns (from RNNs, LSTMs, or transformers). Deep detectors can be roughly put into the following groups:

6.1. Single object tracking (SOT)

The system is equipped with the location (bounding box) of a target object in the initial frame of a video sequence in Single Object Tracking (SOT). The objective is to maintain the same object in subsequent frames, regardless of variations in scale, illumination, occlusion, and background detritus [32]. In contrast to multiple object monitoring (MOT), which entails the simultaneous monitoring of multiple objects, SOT focuses on a single target. This streamlines the identity management process; however, it remains challenging to implement in practical applications. By acquiring robust feature representations that can accommodate intricate appearance changes, SOT methods that are based on deep learning have significantly improved their performance. The following are a few of the most notable deep learning-based SOT techniques, shown below in Figure 4:

$$s(z, x) = f_\theta(z) \star f_\theta(x) \tag{7}$$

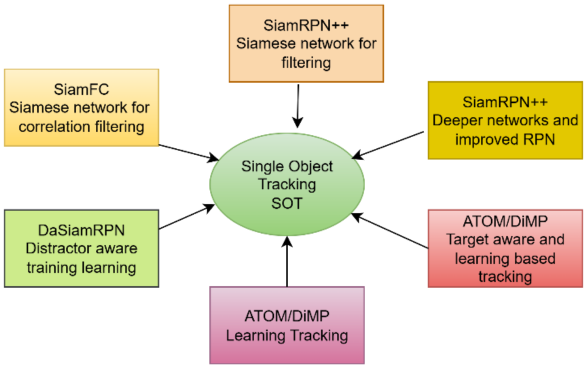

Fig. 4: Single Object Tracking.

6.2. SiamFC (siamese fully convolutional network

Based on deep learning, it is one of the first SOT systems. In this context, tracking is defined as a similarity learning problem. A Siamese network compares a target template from the first frame with possible regions in the current frame. The network's similarity score creates the map's highest point, which displays the goal's position. You can train this method from the beginning to the end, and it requires minimal computer power. But because it is static, it might not be very good at long-term tracking [32-33].

6.3. SiamRPN++

Adds a Region Proposal Network (RPN) to the Siamese design to improve it. This lets it compare traits and come up with good suggestions for bounding boxes, which makes localisation more accurate. SiamRPN++ also uses thicker backbones, like ResNet, and depth-wise cross-correlation to improve feature matching. This procedure makes tracking more accurate in complex scenes.

6.4. SiamRPN

(Distributor-Aware) SiamRPN enhances its own performance by directly addressing the issue of distractors, which are objects that resemble the target goal. It uses a distractor-aware feature that lets the tracker tell the difference between the real goal and similar items during online changes and training. Because of this, the system is much more reliable when there are occlusions and busy backgrounds, improving it for use in the real world [34].

6.5. ATOM

(Accurate Tracking by Overlap Maximisation) uses a different method by splitting target classification and bounding box estimation into two separate sections. Online learning helps the classification module quickly adjust to changes in the target's look. The bounding box estimate module, on the other hand, is taught offline to guess Intersection-over-Union (IoU) scores. This mix lets ATOM achieve very accurate results, even when there is a lot of size difference. A similar method called Discriminative Model Prediction (DiMP) makes ATOM better by teaching the whole model prediction process, which improves it at adapting to things it hasn't seen yet [35-36].

7. Multiple Object Tracking (MOT)

One of the main ways that smart cars have been watched is with Multiple Object Tracking (MOT), which is also used for Intelligent Transportation Systems (ITS). In these cases, you need to keep an eye on more than one car at the same time to do things like figure out how traffic moves, get rid of traffic jams, and find accidents. It is the structure that most MOT methods that are based on deep learning use to track by sensing. Here, an object tracker first looks through each picture for all possible targets. Figure 5 illustrates multi-object tracking techniques [36].

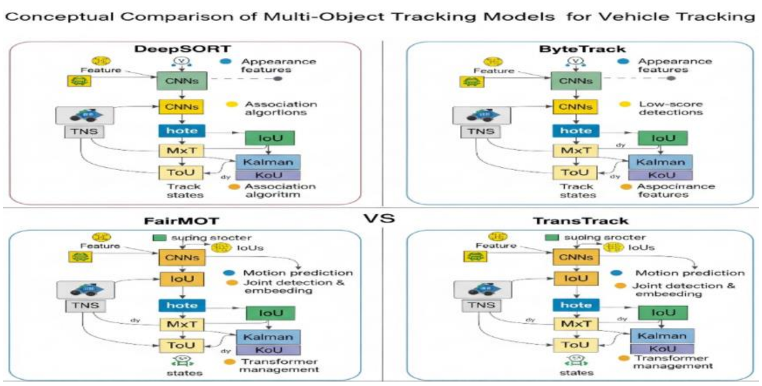

Fig. 5: Multi-Object Tracking.

A data connection tool then links these detections over time to ensure that object IDs stay the same. Modern MOT systems, such as DeepSORT, FairMOT, and CenterTrack, combine features of look (like re-identification embeddings) with signs of motion (like Kalman filtering or visual flow). This makes it easier to see through objects, miss places of interest, and cross paths. Fast models like YOLO, Faster R-CNN, or CenterNet aid the detection step in this system. Graph-based optimisation, or the Hungarian method, is used in the link step to make sure that detections are lined up across frames. Sometimes it's challenging for MOT to achieve the right balance between being very accurate and working quickly when there is a lot of traffic. This is because the presence of cars obstructing each other, rapid lane changes, and similar appearances can all lead to identification switches or tracking loss [38].

$$\widehat{X_t} = \backslash\text{undersetX} \in \mathcal{C}_t \arg\max \sum_{i=1}^{N_t} S(T_i, x_i) \quad (8)$$

Table 3: Comparative Analysis of Multiple Object Tracking (MOT) Algorithms Highlighting Their Key Features, Strengths, Weaknesses, and Suitable Application Domains

Algorithm	Key Features	Strengths	Weaknesses	Best Use Cases
SORT (Simple Online and Real-time Tracking)	Uses the Kalman Filter to predict movements and the Hungarian Algorithm to connect data.	Very quick, light, and simple to use	Heavy occlusion and similar object motion don't work, and there are no appearance traits.	Easy, real-time jobs that don't get in the way
DeepSORT	It adds deep CNN look embeddings to SORT.	More stable when blocked; fewer ID switches	CNN makes it slower than SORT, and it needs high-quality embeddings.	Traffic and monitoring areas where protecting identities is very important.
ByteTrack	Keeps both high-confidence and low-confidence link finds	Fixes missed detections and works well in the real world	Depends on the strength of the detector; sensitive to noise in the signals	Vehicle tracking when there are obstacles, motion fuzz, and changing light
FairMOT	ReID and joint identification in the same network	Fast and accurate at the same time; works in real time; fewer ID changes	It's hard to train, and you need good samples.	Smart transportation, tracking of pedestrians and vehicles
OC-SORT (Occlusion-Aware SORT)	A Motion model that works better and handles occlusions	Tracking is better when an object is in the way without using a lot of CNN.	Still not as reliable as trackers that are based on looks.	Crowded traffic scenes with many blockages

DeepSORT added detailed look features to the tracking process, building on the base of Simple Online and Real-time Tracking (SORT). The Kalman filter is mostly used to predict motion in SORT, while the Hungarian method is used to connect data. On the other hand, it often doesn't work when many targets move in the same way, like when they are filled or stopped. DeepSORT gets around this problem by using a deep convolutional neural network (CNN) to create unique-looking embeddings for each item that is recognised. When these embeddings are mixed with motion guesses, they make it easier to connect things correctly between frames. In settings that change, this lowers the number of identifying changes and makes tracking more stable over time.

ByteTrack is an easy but very useful way to improve the tracking-by-detection approach. Normal methods get rid of low-confidence detections when they try to connect data, but ByteTrack keeps them and tries to connect them with existing tracks. This way lets the tracker get back on track after temporary detection fails, which happens a lot in tough situations like partial occlusion, motion blur, or changing lighting. ByteTrack improves memory without losing accuracy by using both high- and low-confidence detections. This makes it especially useful for real-world car tracking situations where detection confidence can change due to environmental factors.

FairMOT addresses the issue of mismatched recognition and re-identification (ReID) in many multi-object tracking systems. The goals for optimisation and speed weren't always in sync in the past because object recognition and look feature extraction were done by two different networks. FairMOT answers this problem by building a single network that can predict both ReID embeddings and object bounding boxes at the same time. This unified method makes sure that both accuracy in recognition and look discrimination are optimised at the same time, which makes it easier to keep people's identities safe. It works well for smart transportation systems that need to be fast and accurate, too, because its design makes real-time processing easy [38-39].

Trans Track is a new version of multi-object tracking models that use designs based on transformers. Trans Track uses attention processes to model temporal relationships between object instances across frames. This is different from traditional methods that directly link data by matching motion and appearance. The network can see long-distance relationships and follow objects through complicated situations with obstacles, thick groups, and sudden moves, thanks to this global context modelling. Trans Track gets rid of the need for separate stages of recognition and matching, making it a more complete and reliable option for real-world car tracking needs.

Table 4: Various Techniques Used for Vehicle Tracking

References	Algorithm	Technique Used	Application	Limitations	Future Scope
[9]	Convolutional Neural Networks (CNNs)	Deep Learning	Vehicle classification (cars vs. trucks) and automatic vehicle recognition and tracking are both used in traffic control.	Accuracy is poor in busy places (with a lot of people and shadows). Taking care of occlusions (partially hidden cars) is hard.	Make CNN structures that can handle bad lighting and weather. CNNs and object recognition algorithms (YOLOv5) should be used together for better tracking.
[10]	Object Detection Algorithms (YOLOv5, Faster R-CNN)	Deep Learning	car tracking and discovery in real time; automated study of crash areas based on car identification.	It can be hard to pick out small or faraway cars in low-resolution video. Not being able to drive around cars that are covered or overlapping.	Use learnt models on bigger datasets to make them more useful in real life. Fix the centre box so that it can handle cars that are crossing better.
[11]	Optical Flow and Motion Analysis	Traditional Computer Vision Techniques	Analysing patterns of movements to figure out speed and keep track of cars. Looking at sudden changes in motion, like sudden stops, to find crashes.	Can tell when the lighting changes or when the camera moves. Having trouble figuring out what actions are being done on purpose besides the emergency brakes.	When you mix deep learning with visual flow, you get more accurate motion analysis. Make accident recognition systems that are aware of the area around the crash and the way traffic flows.
[12]	LiDAR (Light Detection and Ranging)	Sensor-based Techniques	Being able to correctly find and name 3D objects so that they can be followed when there isn't much light. The three-dimensional models of the cars that were there were used to put together the crash scene.	Webcams and LiDAR devices are not the same price. Short range and the chance of being blocked by tall items.	Make LiDAR devices with better precision and a longer range that are cheaper. Look into sensor fusion methods that use camera data to learn more about the scene.

8. Hybrid Architecture

Mixed methods take different ideas and put them together to strengthen them. CNN–LSTM systems, for example, use Long Short-Term Memory (LSTM) networks to describe changes in time and convolutional neural networks to figure out things about space [21]. Transformer-based trackers have also been made to track better when there are obstacles and changes in visibility [22]. These trackers record the overall links between time and place.

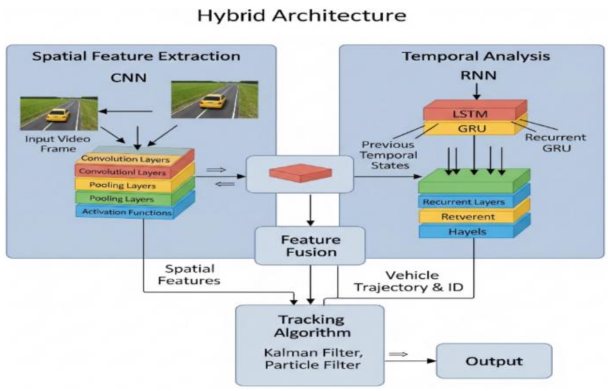

Fig. 6: Hybrid Architecture for Vehicle Tracking.

CNN-BiLSTM now has bidirectional LSTM layers added to it to make this method more useful. These layers look at patterns both forwards and backwards. This lets the tracker combine data from past and future frames to make a stronger tracking system that understands its surroundings better. Transformer-based models, like Trans Track and Track Former, work best when there are obstructions, complicated motion patterns, or many things that are working with each other. This happens because they use self-attention systems that take in the world context and long-range relationships between sequences. Combining temporal reasoning with spatial feature extraction makes hybrid systems better at tracking and more resilient in real-world settings that change quickly [40].

Table 5: Comparative Summary of Popular Deep Learning-Based Vehicle Tracking Models

Model	Technique(s)	Key Features	Advantages	Limitations	Best Used For
DeepSORT	MOT	CNN (appearance encoding) + Kalman filter + Hungarian matching	Tracking that works, low ID switches, and a flexible design	challenges with detections with low confidence	Urban monitoring, walking, and driving
ByteTrack	MOT	Uses detections with low confidence in association	High accuracy in tracking and better object memory	Needs a strong tracker (like YOLOv5) to work well.	Congestion and busy situations
FairMOT	MOT	Finding and re-identifying things together in a single shot	Real-time training and competition MOTA scores from start to finish	A little less accurate because the joints were optimised.	Mixed object tracking and smart cities
TransTrack	MOT	encoder-decoder based on transformers for tracking time	No need to match the way they look, and they work well for long-term blockage.	A lot of memory use and slow reasoning	Scenes with a lot of occlusions
TrackFormer	MOT	DETR-style design with forecast based on sets	Strong against fake hits, and no hand-tuned link	Needs a lot of training data, slow	Smart driving based on vision
SiamRPN++	SOT	Siamese networks with a suggestion network for the area	Strong tracking from frame to frame and quick reasoning	Can't deal with various cars or changes in look	Watching with UAVs and following license plates
CNN-LSTM	Hybrid (SOT/MOT)	It uses CNN to model look and LSTM to model movements.	It records trends in time and is good for prediction tracking.	It's slower and harder to scale to big goals.	Predicting the flow of vehicles
YOLOv8+DeepSORT	Hybrid (MOT)	Real-time recognition and appearance/motion-based connection in one	Real-time efficiency that is correct, scalable, and light	Performance changes based on how strong the device is	Smart parking and roadside cameras
TransT	SOT	Matching features based on transformers	Takes care of flexible movements and changes in look	High cost of computation	Activity tracking from above or on the street

9. Challenges and Open Issues

Despite extensive efforts, tracking vehicles still faces certain challenges. Figure 7 shows some of these problems, including occlusion, identity switching, and changes in appearance and lighting. Classical methods, such as the Kalman filter, are good at using computers, but they are fragile and have trouble with these complicated real-world situations. Traditional machine learning models are more balanced, but they can't handle data with a lot of variation very well. On the other hand, new deep learning models are better at dealing with these problems because they are more accurate and stable. However, this performance comes at the cost of high computational demands and the need for enormous datasets for training. The main unresolved problem in the field is this trade-off between accuracy and efficiency. Such performance shows how important it is to do more study to find tracking solutions that are light, strong, and protect privacy for real-time edge device deployment in smart cities.

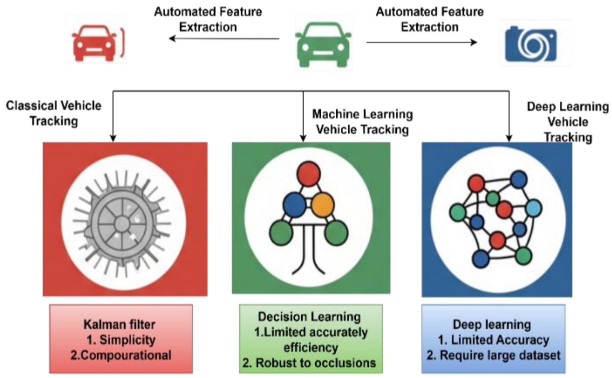

Fig. 7: A Visual Representation of the Key Challenges in Multi-Object Vehicle Tracking, Including Occlusion, Identity Switching, and Inconsistent Feature Extraction Under Diverse Environmental Conditions.

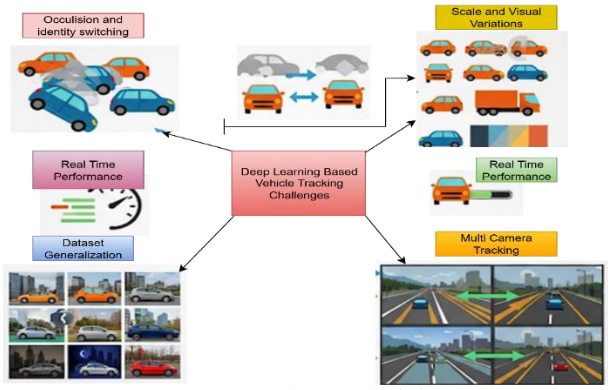

Fig. 8: Deep Learning-Based Challenges.

In invisible environments, models also encounter challenges with dataset generalization. Despite demonstrating satisfactory performance on benchmarks, they are unable to accomplish comparable results. Additionally, the complex nature of multi-camera surveillance in smart cities necessitates the implementation of effective global association techniques and robust RFID models. Table 6 shows the challenges of existing techniques.

Table 6: Challenges of an Existing Technique

Problem	Existing techniques	Limitations	Suggested research directions
Occlusion and changing identities	DeepSORT appearance embeddings and Kalman + IoU gates	When long occlusions happen, short-term look cues don't work, so identity switches happen.	Combine long-term temporal models (transformer/RNN), multi-frame re-ID, and a data link that takes occlusions into account.
Spreading out the domain or shifting the dataset	Pretraining: strategies for domain adaptation	Performance drops in many places and cameras; not enough labelled data	Domain adaptation that isn't monitored or is self-supervised, synthetic-to-real transfer, and test-time adaptation
Latency and FLOPs for real-time rollout	Mobile Net's light backbones, trimming, and quantization	Loss of accuracy after compression; transformers are often too heavy	Focus on creating lightweight models and hardware-aware NAS that work well.
Cross-view tracking and multiple cameras	Global graph matching, re-ID plus correction	Needs camera alignment and synchronization; re-ID drift between views	Global association frameworks, multi-camera embedding alignment, and cross-view learning with little supervision
Weather and lighting dependability	Multispectral devices and adding to the data	Poor performance of detectors in rain, fog, and at night	Multiple sensing types (camera, radar, and LiDAR), strong feature learning, and domain randomness
Small or faraway cars	Scale-aware detectors and high-resolution pictures	When things are small, they have low IoU and bad appearance traits.	Scale-aware anchor-free detectors, super-resolution preprocessing, and contrastive learning for small things
Label noise and the cost of annotation	Learning with some supervision and fake data	It costs a lot to label multiple object tracks, and noisy boxes make tracking harder.	Active learning, datasets based on simulations, pre-training that is not supervised, and weak oversight

10. Comparative Analysis

Classical, machine learning (ML), and deep learning (DL)–based vehicle tracking approaches were systematically contrasted across performance metrics, datasets, and computational requirements in the surveyed literature to emphasize the trade-offs. Table 7 shows different methods of tracking.

Table 7: Performance Metrics for Key Deep Learning-Based Multi-Object Tracking Models. Note That Values Are as Reported in the Respective Original Studies. Direct Numerical Comparisons Should Be Interpreted with Caution Due To Variations in Datasets and Evaluation Protocols. Missing FPS Values Often Indicate That This Information Was Not Explicitly Detailed or Varies with Hardware

Method	Year	Paradigm	Detector / Back-bone	MOTA (%)	IDF1 (%)	HOTA (%)	FPS	Notes
DeepSORT	2017	MOT(Multi object tracking)	Faster-R-CNN	~61	—	—	Real-time	Classic baseline using both Kalman and look descriptions.
Byte Track	2021	MOT	YOLOX	~80	~77	—	—	Uses detections with low confidence to boost memory.
FairMOT	2021	MOT	Joint detection + RfID	73.7	72.3	59.3	—	Finds the right mix between name protection and discovery.
Trans Track Transformer	2021	MOT	CNN + Transformer	75.2	63.5	54.1	—	Attention processes are used for end-to-end tracking.
Track Former Transformer	2021	MOT	Pure Transformer	74.1	68.0	57.3	—	Combines tracking and recognition into a single system.
YOLOv8 + DeepSORT	2023	MOT	YOLOv8 + DeepSORT	57.6	73.2	—	—	Tracking works a lot better when recognition is improved.
Byte Track UA-DETRAC	2023	MOT	YOLOv8 + Byte Track	57.4	75.1	—	—	Keeps a high level of name protection and memory.
BoT-SORT	2022	MOT	YOLOX + BoT association	80.5	80.2	65.0	—	At the cutting edge in MOTA and IDF1 on MOT17.
MOT_FCG++ Hybrid	2024	MOT	CNN + Hybrid features	80.4	81.3	76.1	—	The best total success comes from using more than one planning strategy together.

Table 7 shows that there isn't a single model that does better in every way. ByteTrack, for instance, achieves high MOTA and IDF1 by using detections with low confidence. It works well in places with a lot of traffic because of this, but it might need more computer power. FairMOT finds the right balance between awareness and identification. Such an approach leads to competitive results but lower HOTA scores. It's better for attention-based models like TransTrack and TrackFormer to overcome hurdles, but their FPS numbers aren't always shown because they take a long time to calculate. When it comes to MOTA and IDF1, BoT-SORT works the best, but it is harder to use because it uses complicated association methods. The most balanced approach is a hybrid one, like MOT_FCG++ (2024). Better and more accurate, they use CNNs along with mixed features, but it's still not clear how they will work on edge devices. There are costs to getting things right, staying stable, and moving quickly. They also show how important it is to make more models that are very accurate and can be quickly generated for ITS systems that work in real time.

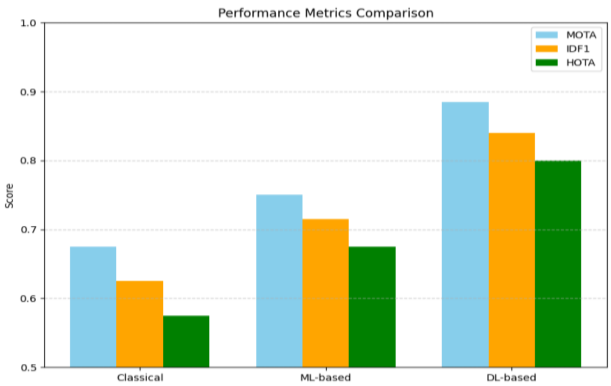

Fig. 9: Performance Metrics Comparisons.

9.2. Dataset usage

Vehicle tracking research employs a diverse array of benchmark datasets, each of which presents its own set of challenges shown in Table 8.

Table 8: Vehicle Tracking Datasets

Dataset	Year	Size	Data Type	Key Features	Challenges
UA-DETRAC	2018	A video that spans 10 hours	RGB	Multiple weather conditions, dense traffic	Varying illumination, high occlusion
KITTI	2013	Video+ 15k images	RGB + LiDAR	Stereo vision, 3D labels, and urban driving	Scenes that are restricted to the night or mists
MOT17/MOT20	2017 / 2020	8 sequences / 14 sequences	RGB	MOT evaluation benchmark	Mixed objects, not specific to any particular vehicle
BDD100K	2018	One hundred thousand videos	RGB	Diverse location, illumination, and weather conditions	Noise associated with annotations in certain frames

CityFlow	2019	3+ hours of multi-view video	RGB	Urban monitoring with multiple cameras	Calibration and synchronization of the camera
HighD	2019	110,000 trajectory paths	Drone video	Highway scenes from a bird's-eye perspective	No data on adverse weather or low-light conditions

9.3. Performance comparison: classical vs ML vs DL

The performance evaluation of classical, ML-based, and DL-based vehicle tracking methods illustrates a clear trade-off between accuracy, speed, and computational requirements. In difficult conditions, such as congested traffic, varying illumination, and occlusion, deep learning-based trackers consistently outperform their classical and ML-based counterparts in critical metrics such as HOTA, IDF1, and MOTA, thereby demonstrating superior robustness. However, this precision is at the cost of hardware demand and performance. Classical methods are capable of obtaining 60–120 FPS on standard CPUs, whereas DL-based trackers rarely surpass 15 FPS without GPU acceleration. ML-based methodologies are located between these two extremes, offering a balanced combination of speed and accuracy for mid-range hardware. The efficacy of DL-based methods is significantly improved by GPU acceleration, which enables them to operate in near-real time. This is demonstrated in Figure 10, which compares CPU and GPU FPS for each approach category.

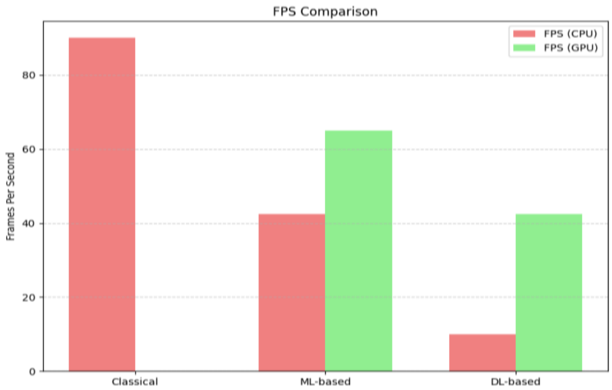

Fig. 10: Comparisons of FPS (CPU), FPS(GPU).

Table 9: Summary of Key Surveys in Vehicle Tracking for Autonomous Driving

Domain	Systematic Review	Dataset & Preprocessing	Technological Advancement	Result Analysis	Challenges & Future Direction	Contribution
√	√	√	√	×	√	Addresses environmental variability.
×	√	×	√	√	×	Integrates multi-sensor data effectively.
√	×	√	×	√	√	Creates structures for predictive maintenance.
√	√	×	√	×	√	Improves computational efficiency for real-time processing.
×	√	√	√	√	×	Improves the accuracy of object recognition when there are obstructions.
√	×	×	×	×	√	Analyses road user behaviour under dynamic conditions.
√	√	√	√	√	√	Proposes scalable architectures for autonomous systems.

11. Conclusion

This paper provided a thorough systematic review of vehicle tracking techniques based on deep learning, with a particular emphasis on the advancements that occurred between 2020 and 2024. We categorized approaches into single object tracking (SOT), multiple object tracking (MOT), and hybrid architectures, and analysed key models such as DeepSORT, Byte Track, FairMOT, and transformer-based trackers. Our results suggest that deep learning has significantly improved the robustness, adaptability, and accuracy of tracking. Nevertheless, deployment is still impeded by persistent issues, including real-time performance limitations, poor generalization to unseen environments, and occlusion handling challenges. Real-time applications are presently dominated by models that strike a balance between robust feature association and efficient detection. However, there is still room for improvement.

Future Scope

Future research should concentrate on the development of tracking systems that are both lightweight and potent and that can operate efficiently on edge devices without sacrificing accuracy. The integration of context-aware and multi-modal data sources, including radar, LiDAR, and traffic signal information, can enhance robustness in challenging environments. To guarantee that generalisation is seamless across a variety of environments and camera configurations, it is necessary to implement improved domain adaptation techniques. In addition, the deployment of large-scale smart cities will necessitate advancements in cross-view monitoring and multi-camera deployments. Beyond these technical areas, it is critical for future work to address the ethical and privacy implications of data collection. Developing privacy-preserving methods, such as federated learning, will be essential for ensuring responsible deployment in public spaces. By addressing these areas, we will not only overcome current limitations but also facilitate the development of fully autonomous, more efficient, and safer transportation systems.

References

[1] P. Josephinshermila, S. Sharon Priya, K. Malarvizhi, R. hegde, S. Gokul Pran, and B. Veerasamy, “Accident detection using automotive smart black-box based,” *Meas. Sensors*, vol. 27, no. March, p. 100721, 2023, https://doi.org/10.1016/j.measen.2023.100721.

[2] V. R. Patil and S. S. Pardeshi, “Materials Today : Proceedings Mechanism for accident detection , prevention and reporting system,” *Mater. Today Proc.*, vol. 72, pp. 1975–1980, 2023, https://doi.org/10.1016/j.matpr.2022.11.215.

[3] K. Victor and Y. Chan, “ScienceDirect Descriptive and inferential statistics of serious accidents involving considerably automated vehicles — a necessity of smart cities,” *Procedia Comput. Sci.*, vol. 219, pp. 856–863, 2023, https://doi.org/10.1016/j.procs.2023.01.360.

[4] D. Yan, K. Li, Q. Zhu, and Y. Liu, “A railway accident prevention method based on reinforcement learning – Active preventive strategy by multi-modal data,” *Reliab. Eng. Syst. Saf.*, vol. 234, no. February, p. 109136, 2023, https://doi.org/10.1016/j.ress.2023.109136.

[5] J. Beck, R. Arvin, S. Lee, A. Khattak, and S. Chakraborty, “Automated vehicle data pipeline for accident reconstruction: New insights from LiDAR, camera, and radar data,” *Accid. Anal. Prev.*, vol. 180, no. December 2022, p. 106923, 2023, https://doi.org/10.1016/j.aap.2022.106923.

[6] M. Karthik, L. Sreevidya, K. Vinodha, M. Thangaraj, G. Hemalatha, and T. V. Sena, “Automatic messaging system by detecting the road accidents for vehicle applications,” *Mater. Today Proc.*, vol. 80, pp. 3124–3128, 2023, https://doi.org/10.1016/j.matpr.2021.07.177.

[7] A. Kumar and T. K. Das, “CAVIDS: Real time intrusion detection system for connected autonomous vehicles using logical analysis of data,” *Veh. Commun.*, vol. 43, p. 100652, 2023, https://doi.org/10.1016/j.vehcom.2023.100652

[8] X. Liao, G. Wu, L. Yang, and M. J. Barth, “A Real-World Data-Driven approach for estimating environmental impacts of traffic accidents,” *Transp. Res. Part D Transp. Environ.*, vol. 117, no. September 2022, p. 103664, 2023, https://doi.org/10.1016/j.trd.2023.103664.

[9] N. Kumar, D. Lohani, and D. Acharya, “Vehicle accident sub-classification modeling using stacked generalization: A multisensor fusion approach,” *Futur. Gener. Comput. Syst.*, vol. 133, pp. 39–52, 2022, https://doi.org/10.1016/j.future.2022.03.005.

[10] K. Pawar and V. Attar, “Deep learning-based detection and localization of road accidents from traffic surveillance videos,” *ICT Express*, vol. 8, no. 3, pp. 379–387, 2022, https://doi.org/10.1016/j.icte.2021.11.004.

[11] A Review of Machine Learning and IoT in Smart Transportation, Mdpi, Apr. 10AD, https://doi.org/10.3390/fi11040094.

[12] A. C. Phan, T. N. Trieu, and T. C. Phan, “Driver drowsiness detection and smart alerting using deep learning and IoT,” Internet of Things (Netherlands), vol. 22, no. January, 2023, https://doi.org/10.1016/j.iot.2023.100705.

[13] J. P. A. Yaacoub, H. N. Noura, and O. Salman, “Security of federated learning with IoT systems: Issues, limitations, challenges, and solutions,” Internet Things Cyber-Physical Syst., vol. 3, no. January, pp. 155–179, 2023, https://doi.org/10.1016/j.iotcps.2023.04.001.

[14] V. Raosaheb Patil and S. Suresh Pardeshi, “Mechanism for accident detection, prevention and reporting system,” Mater. Today, Proc., vol. 72, pp. 1975–1980, 2023, https://doi.org/10.1016/j.matpr.2022.11.215.

[15] J. Biegańska-Banaś, P. Banaś, M. Zięba, J. K. Gierowski, and J. Trzebiński, “PTSD malingering detection in damage claim cases: Diagnostic accuracy in cases of personal injury as a result of motor vehicle accidents,” Int. J. Law Psychiatry, vol. 88, no. July 2022, pp. 27–32, 2023, https://doi.org/10.1016/j.ijlp.2023.101885.

[16] B. Li, L. Tan, F. Wang, and L. Liu, “A railway intrusion detection method based on decomposition and semi-supervised learning for accident protection,” Accid. Anal. Prev., vol. 189, no. May, p. 107124, 2023, https://doi.org/10.1016/j.aap.2023.107124.

[17] J. Choi and S. J. Lee, “RNN-based integrated system for real-time sensor fault detection and fault-informed accident diagnosis in nuclear power plant accidents,” Nucl. Eng. Technol., vol. 55, no. 3, pp. 814–826, 2023, https://doi.org/10.1016/j.net.2022.10.035

[18] I. Lashkov, R. Yuan, and G. Zhang, “Machine learning-based vehicle detection and tracking based on headlight extraction and GMM clustering under low illumination conditions,” *Expert Syst. Appl.*, vol. 267, p. 126240, Apr. 2025, https://doi.org/10.1016/j.eswa.2024.126240.

[19] A. Parekh, “Comparative Analysis of YOLOv8 and RT-DETR for Real-Time Object Detection in Advanced Driver Assistance Systems,” *Electron. Thesis Diss. Repos.*, Mar. 2025, [Online]. Available: https://ir.lib.uwo.ca/etd/10693. https://doi.org/10.1007/978-3-031-94962-3_3.

[20] A. H. B, M. P. M. M, U. Verma, and R. M. Pai, “Vehicle Re-Identification and Tracking: Algorithmic Approach, Challenges and Future Directions,” *IEEE Open J. Intell. Transp. Syst.*, pp. 1–1, 2025, https://doi.org/10.1109/OJITS.2025.3538037

[21] P. Chen, T. Xin, C. Kong, S. Zhao, X. Zheng, and J. Wang, “Comparative analysis of vibration and noise reduction effect of different tracks on u-shaped bridges,” *KSCE J. Civ. Eng.*, vol. 29, no. 2, p. 100044, Feb. 2025, https://doi.org/10.1016/j.kscej.2024.100044.

[22] G. Li, X. Wang, S. Wu, D. li, L. Ma, and W. Ding, “Dynamic response analysis of vehicle-track coupled system at subgrade-bridge transition zone in seasonal frozen region under multi-source excitation,” *Soil Dyn. Earthq. Eng.*, vol. 190, p. 109216, Mar. 2025, https://doi.org/10.1016/j.soildyn.2025.109216.

[23] S. Rani and S. Dalal, “Comparative analysis of machine learning techniques for enhanced vehicle tracking and analysis,” *Transp. Eng.*, vol. 18, p. 100271, Dec. 2024, https://doi.org/10.1016/j.treng.2024.100271

[24] P. Azevedo and V. Santos, “Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices,” *Robot. Auton. Syst.*, vol. 171, p. 104558, Jan. 2024, https://doi.org/10.1016/j.robot.2023.104558

[25] A. Geetha, M. Al Rabbani Alif, M. Hussain, and P. Allen, “Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy,” *Vehicles*, vol. 6, Aug. 2024, https://doi.org/10.3390/vehicles6030065

[26] A. Geetha, *Comparing YOLOv5 Variants for Vehicle Detection: A Performance Analysis*. 2024.

[27] A. Artuñedo, M. Moreno-Gonzalez, and J. Villagra, “Lateral control for autonomous vehicles: A comparative evaluation,” *Annu. Rev. Control*, vol. 57, p. 100910, Jan. 2024, https://doi.org/10.1016/j.arcontrol.2023.100910.

[28] L. Fei and B. Han, “Multi-Object Multi-Camera Tracking Based on Deep Learning for Intelligent Transportation: A Review,” *Sensors*, vol. 23, p. 3852, Apr. 2023, https://doi.org/10.3390/s23083852.

[29] “Multiple Moving Vehicles Tracking Algorithm with Attention Mechanism and Motion Model.” Accessed: Apr. 01, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/13/1/242. https://doi.org/10.3390/electronics13010242.

[30] V. Biradar and K. C. Gull, “Multiple Object Detection and Tracking Using Deep Learning Framework with Non-Maximum Suppression. | EBSCOhost.” Accessed: Apr. 01, 2025. [Online]. Available: https://openurl.ebsco.com/EPDB%3Agcd%3A4%3A26920788/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A182062467&crl=c&link_origin=scholar.google.com.

[31] “Object Detection and Tracking Algorithms for Vehicle Counting: A Comparative Analysis | Data Science for Transportation.” Accessed: Apr. 01, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s42421-020-00025-w.

[32] A. S. Bhadoriya, V. Vegamoor, and S. Rathinam, “Vehicle Detection and Tracking Using Thermal Cameras in Adverse Visibility Conditions,” *Sensors*, vol. 22, no. 12, Art. no. 12, Jun. 2022, https://doi.org/10.3390/s22124567.

[33] J. Liu, Y. Xie, Y. Zhang, and H. Li, “Vehicle Flow Detection and Tracking Based on an Improved YOLOv8n and ByteTrack Framework. | EBSCOhost.” Accessed: Apr. 01, 2025. [Online]. Available: https://openurl.ebsco.com/EPDB%3Agcd%3A7%3A16706935/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A182473236&crl=c&link_origin=scholar.google.com.

[34] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, “Vehicle Tracking and Speed Estimation From Roadside Lidar,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 5597–5608, 2020, https://doi.org/10.1109/JSTARS.2020.3024921.

[35] M. Al Rabbani Alif, *YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems*. 2024.

[36] S. S. P. Naresh, R. Talasu, D. S. S. P. Venkat, S. K. Korada, and B. K. Mohanta, “Real Time Vehicle Tracking using YOLO Algorithm,” in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jul. 2023, pp. 1–5. https://doi.org/10.1109/ICCCNT56998.2023.10307265.

[37] S. Javadi, M. Rameez, M. Dahl, and M. I. Pettersson, “Vehicle Classification Based on Multiple Fuzzy C-Means Clustering Using Dimensions and Speed Features,” *Procedia Comput. Sci.*, vol. 126, pp. 1344–1350, Jan. 2018, https://doi.org/10.1016/j.procs.2018.08.085

[38] M. Alipour Sormoli, M. Dianati, S. Mozaffari, and R. Woodman, “Optical Flow Based Detection and Tracking of Moving Objects for Autonomous Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12578–12590, Sep. 2024, https://doi.org/10.1109/TITS.2024.3382495

[39] M. H. Ashraf, F. Jabeen, H. Alghamdi, M. S. Zia, and M. S. Almutairi, “HVD-Net: A Hybrid Vehicle Detection Network for Vision-Based Vehicle Tracking and Speed Estimation,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 8, p. 101657, Sep. 2023, https://doi.org/10.1016/j.jksuci.2023.101657

[40] Y. Gao, S. Wang, and H. K.-H. So, “A Reconfigurable Architecture for Real-time Event-based Multi-Object Tracking,” *ACM Trans Reconfigurable Technol Syst*, vol. 16, no. 4, p. 58:1-58:26, Sep. 2023, https://doi.org/10.1145/3593587.

[41] X. Wang, Z. Sun, A. Chehri, G. Jeon, and Y. Song, “Deep learning and multi-modal fusion for real-time multi-object tracking: Algorithms, challenges, datasets, and comparative study,” *Inf. Fusion*, vol. 105, p. 102247, May 2024, https://doi.org/10.1016/j.inffus.2024.102247.