Vehicle Speed Detection And Helmet Enforcement

Dr. S. Vijayaragavan[1], Ranjith J[2], Karuna D[3], Krishnaraj Kv[4]
[1] Prof & HOD, Dept of Artificial Intelligence And Data Science
[2, 3, 4] Dept of Artificial Intelligence And Data Science

Abstract- *Ensuring road safety through effective enforcement of traffic laws is a critical challenge, particularly in addressing overspeeding and helmet non-compliance among motorcyclists. Traditional methods of traffic violation monitoring are resource-intensive and often fail to deliver real-time results. This project proposes an automated system that integrates vehicle speed detection and helmet enforcement with a smart messaging system to enhance road safety. Utilizing advanced computer vision techniques such*

Keywords- Vehicle Speed Detection, Helmet Enforcement, Smart Message System, Optical Flow, Sub- Pixel Stereo Matching, Convolutional Neural Networks (CNNs), Retina Net, License Plate Recognition, SMS Notification System, Traffic Violation Detection, Road Safety Automation, Image Processing for Speed Detection, Object Detection for Helmet Compliance,

I. INTRODUCTION

Road safety is a critical global concern, with traffic violations being a significant contributor to accidents and fatalities. Over speeding and failure to wear helmets are two of the most common violations among motorcyclists, leading to severe injuries and loss of life. According to recent studies, enforcing compliance with traffic regulations through traditional manual methods is not only resource-intensive but often ineffective due to human error and limited coverage. As a result, the development of automated systems for speed detection and helmet enforcement has gained significant attention in recent years. Automated systems combine advanced technologies such as image processing, sensor fusion, and machine learning to detect violations efficiently. By leveraging computer vision algorithms, these systems can process real-time video feeds to measure vehicle speeds and identify riders without helmets. Additionally, integrating a smart message system ensures that violation alerts are sent directly to the registered vehicle owner or enforcement authorities, enabling timely action. Speed detection systems, traditionally reliant on radar and LIDAR, are now incorporating more sophisticated computer vision techniques such as optical flow and stereo vision to achieve higher accuracy and coverage. Similarly, helmet detection has advanced with the adoption of deep learning models like RetinaNet, which excel in recognizing small objects like helmets in complex traffic scenarios. The integration of these detection capabilities with a smart notification system is a novel approach that enhances enforcement efficiency. Such a system not only identifies violations but also ensures accountability by notifying violators through SMS alerts containing detailed information, including registration numbers and timestamps. This paper provides a comprehensive survey of the key algorithms and technologies as optical flow for speed detection and deep learning models like RetinaNet for helmet identification, the system achieves high accuracy and scalability. A smart notification module further enhances its functionality by sending real-time SMS alerts containing violation details to vehicle owners or traffic authorities. This paper discusses the design, implementation, and evaluation of the proposed system, emphasizing its potential to revolutionize traffic law enforcement through automation and intelligent communication used for vehicle speed detection and helmet enforcement. It discusses their strengths, limitations, and real-world applicability, offering a comparative analysis to guide future research and implementation. By focusing on the synergy between detection systems and smart messaging platforms, the study highlights the potential for improving road safety through automation and intelligent communication.

II. MOVING OBNJECT DETECTION

Moving object detection is a fundamental problem in computer vision, widely used in applications such as traffic monitoring, video surveillance, and autonomous driving systems. Its primary goal is to identify and segment objects in motion from a sequence of video frames, distinguishing them from the static background. This process involves a series of computational techniques designed to analyze changes in pixel intensity, motion vectors, or object features over time. At the core of moving object detection is motion modeling, which captures the movement of objects by analyzing temporal variations in a video sequence. This can be achieved through techniques such as optical flow, background subtraction, and temporal differencing. Optical flow analyzes the apparent motion of brightness patterns in a scene, using mathematical models to compute velocity vectors for each pixel between consecutive frames. This method is particularly effective for capturing motion in short-range scenarios but is sensitive to environmental changes like lighting variations and occlusions.

Background subtraction is another commonly used technique, which involves constructing a reference model of the static scene and comparing each incoming frame against this model to detect changes. Pixels that deviate significantly from the background are classified as part of moving objects. While computationally efficient, this approach faces challenges in dynamic environments where the background is not entirely static, such as scenes with swaying trees or varying illumination. Temporal differencing is a simpler method that detects motion by calculating the differences in pixel intensity between consecutive frames. While fast and lightweight, it is less reliable in cases where objects move slowly or blend into the background. Recent advancements in machine learning and deep learning have significantly enhanced the accuracy and robustness of moving object detection. Convolutional neural networks (CNNs) have been widely adopted to extract spatial features, enabling the detection and classification of moving objects with high precision. Recurrent neural networks (RNNs) further improve performance by capturing temporal dependencies in video sequences, making them suitable for complex motion patterns. Advanced models like YOLO (You Only Look Once) provide real-time object detection capabilities, ideal for high-speed applications in traffic monitoring and autonomous navigation. Stereo vision systems offer another sophisticated approach, using disparity maps to estimate the depth of objects and differentiate between moving and stationary elements in a scene. By analyzing the spatial and temporal features of objects, stereo vision systems can calculate their distance and velocity with high accuracy. Despite these advancements, moving object detection continues to face challenges in real-world scenarios. Environmental factors such as low light, shadows, rain, or fog can degrade detection accuracy. Dynamic backgrounds and occlusions further complicate the segmentation and tracking of moving objects. Moreover, achieving real-time performance requires balancing computational efficiency with detection accuracy, which remains a critical area of research. Moving object detection serves as the backbone of numerous real-world applications. In traffic monitoring, it is used to analyze vehicle flow, detect violations, and manage congestion. Surveillance systems leverage these techniques to identify suspicious activities or track individuals in restricted areas. In autonomous vehicles, moving object detection ensures safe navigation by identifying and tracking dynamic obstacles.The theory and methodologies underlying moving object detection continue to evolve, driven by advancements in machine learning, sensor technology, and computational hardware. As these techniques become more robust and efficient, they pave the way for transformative innovations across various domains.

III. METHODS FOR MOVING OBJECT DETECTION

Moving object detection encompasses a variety of computational techniques that analyze motion patterns in video sequences to identify and segment objects in motion. The methods can be broadly categorized into traditional approaches, machine learning-based techniques, and advanced deep learning methods. Each method has unique strengths and limitations, making them suitable for specific applications.

3.1 Traditional Methods

Traditional approaches rely on mathematical models and pixel-based analysis to detect motion in video sequences. These methods are simple and computationally efficient, often used in resource-constrained systems. Optical flow computes the apparent motion of objects between consecutive frames by analyzing changes in pixel intensity. It estimates motion vectors that describe the direction and speed of moving objects. Optical flow is effective for capturing fine-grained motion details and is widely used for vehicle tracking and speed estimation. However, it is sensitive to environmental variations, such as lighting changes and occlusions. This technique involves creating a reference model of the static background and comparing each frame of the video to this model. Pixels that deviate significantly from the background are classified as part of the moving object. Background subtraction is computationally efficient and suitable for controlled environments but struggles with dynamic backgrounds or sudden illumination changes. Temporal differencing calculates the pixel-wise intensity difference between consecutive frames to identify motion. This method is fast and simple, making it ideal for real-time applications. However, it may fail to detect slow-moving objects or objects that merge with the background.

3.2 Machine Learning-Based Methods

Machine learning techniques improve the reliability and adaptability of moving object detection by learning patterns from labeled data. These methods use handcrafted features such as edges, textures, and gradients to classify motion. SVMs are widely used for classifying motion patterns based on extracted features. They are effective for small datasets and binary classification tasks, such as detecting moving versus stationary objects. However, SVMs require careful feature selection and scaling to perform well. Random forests use decision trees to classify moving objects based on multiple features. They are robust to noise and capable of handling complex datasets. This method is often employed for scenarios with dynamic backgrounds or multiple object classes.

3.3 Deep Learning-Based Methods

Deep learning methods have revolutionized moving object detection by automating feature extraction and improving accuracy in complex scenarios. These methods rely on neural networks to process large-scale data and identify motion patterns. CNNs are highly effective for detecting moving objects in video frames. They extract spatial features and classify objects based on their appearance and motion. CNN- based models like RetinaNet and YOLO provide high accuracy and real-time performance, making them suitable for applications such as traffic monitoring and surveillance. RNNs capture temporal dependencies in video sequences, making them ideal for analyzing sequential motion patterns. Long Short-Term Memory (LSTM) networks, a type of RNN, are particularly effective for detecting and tracking objects over time. However, they are computationally intensive and require substantial training data. Hybrid methods combine traditional techniques with deep learning models to enhance detection accuracy and efficiency. For instance, background subtraction can be used to preprocess video frames, followed by CNNs for detailed object detection. These approaches balance computational complexity and detection performance.

IV. BACKGROUND SUBTRACTION

Background subtraction is a foundational technique in computer vision, widely employed for detecting moving objects within a video sequence by isolating the foreground from the static background. The method operates on the assumption that the background remains relatively constant over time, while the foreground contains moving entities. By modeling the background and comparing incoming video frames against it, moving objects can be effectively segmented and tracked. This technique is integral to applications such as traffic monitoring, surveillance, and object detection in dynamic scenes.

The core principle of background subtraction involves creating a reference model of the static scene, often referred to as the background model, denoted by B(x,y,t)B(x, y, t)B(x,y,t), where (x,y)(x, y)(x,y) represents the spatial coordinates and ttt denotes time. For any given frame I(x,y,t)I(x, y, t)I(x,y,t) in the video, the difference between the current frame and the background model is calculated to detect motion. Pixels that do not fit well into any of the KKK Gaussians are classified as part of the foreground, while those fitting a Gaussian with sufficient weight are considered part of the background. To enhance the accuracy of background subtraction, post-processing techniques such as morphological operations (e.g., erosion and dilation) are often applied to remove noise and fill gaps in the detected objects. These operations refine the binary mask, ensuring smoother and more continuous object boundaries. Despite its effectiveness, background subtraction faces challenges in dynamic environments, such as scenes with moving backgrounds, sudden illumination changes, or complex shadows. Hybrid approaches that combine traditional background subtraction with deep learning models, such as Convolutional Neural Networks (CNNs), are being explored to address these limitations, enabling more robust and adaptable object detection in real-world scenarios.

V. OBJECT CLASSIFICATION

Object classification is a core task in computer vision, focusing on the identification and categorization of objects within an image or video into predefined classes. This process enables the differentiation between object types such as vehicles, pedestrians, or other entities based on their visual characteristics. It forms the foundation for applications in fields like traffic monitoring, surveillance, medical imaging, and autonomous systems. The classification process involves analyzing features of the input data to assign a label that corresponds to the object's class. Mathematically, object classification can be defined as a mapping function f(x;θ)f(x; \theta), where xx is the input image or feature representation, θ\theta represents the parameters of the classifier, and the output yy is the predicted class label. The process begins with feature extraction, where discriminative properties of the object, such as edges, textures, and shapes, are captured. These features are then processed by a classification model, which assigns a probability score to each potential class. The final classification decision is made by selecting the class with the highest probability. Modern object classification predominantly relies on supervised learning, where the classifier is trained on a dataset of labeled examples {(xi,yi)}i=1N\{(x_i, y_i)\}_{i=1}^N. The training process optimizes a loss function, such as cross-entropy loss, to minimize the discrepancy between the predicted and actual class labels. This loss function is expressed as: Feature extraction has evolved from handcrafted techniques to automated methods using deep learning. Traditional methods such as Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) focused on manually engineered features that were fed into classifiers like Support Vector Machines (SVMs) or Random Forests. These approaches relied on the intuition and expertise of researchers to define features, which limited their scalability and adaptability to diverse datasets. With the advent of deep learning, convolutional neural networks (CNNs) have revolutionized object classification. CNNs automate the feature extraction process, learning hierarchical representations directly from raw input data. A CNN consists of layers that progressively capture spatial and semantic features, transitioning from detecting edges in initial layers to recognizing complex object parts in deeper layers. The output layer of a CNN employs a softmax activation function to assign probabilities to each class, ensuring that the total probability across all classes sums to 1.

The classification decision in CNNs is based on:

$$\hat{y} = \text{Softmax}(W \cdot f(x) + b)$$

Where:

- f(x)f(x) is the feature vector extracted by the CNN.
- WW and bb are the weights and biases of the fully connected layer.

Despite its advancements, object classification faces challenges such as class imbalance, where certain object categories dominate the dataset, leading to biased predictions. Additionally, intra-class variability (e.g., different shapes and sizes of the same object type) and inter-class similarity (e.g., visually similar objects belonging to different classes) pose significant hurdles in achieving high classification accuracy. Environmental factors like lighting variations, occlusions, and cluttered backgrounds further complicate the task. Object classification has found applications across a wide spectrum of domains. In traffic monitoring, it is used to identify vehicles, pedestrians, and other road users for flow analysis and violation detection. In autonomous systems, it aids in recognizing road signs, obstacles, and vehicles to support decision-making. In surveillance, it is employed to monitor restricted areas and track suspicious activities. In the medical field, object classification is instrumental in identifying abnormalities in medical images, aiding in accurate diagnoses. With the continuous evolution of computational hardware and machine learning algorithms, object classification systems are becoming increasingly robust and scalable. Techniques such as transfer learning, ensemble models, and hybrid approaches promise to address existing challenges, paving the way for enhanced accuracy and broader applicability.

VI. OBJECT DETECTION IN INTELLIGENT TRANSPORTATION SYSTEMS

The object detection technique plays a pivotal role in the domain of intelligent transportation systems (ITS), where detecting and tracking vehicles, pedestrians, traffic signs, and other objects are crucial for enhancing road safety and managing traffic. This technique extracts essential features, such as the shape of vehicles and spatial-temporal information from traffic signs, to identify and classify moving objects within a given scene. Object detection is primarily focused on targets such as cars, pedestrians, and traffic signs, where the shape, spatial, and temporal information of the object are considered critical features for detection [5][6].

6.1 Motion Segmentation and Optical Flow for Object Detection

Optical flow is an effective motion segmentation technique that plays a significant role in detecting moving objects, even when the camera itself is in motion. Optical flow detects and tracks moving objects by analyzing the motion of pixel intensities across consecutive frames. It is particularly useful in aerial views and dynamic environments, where traditional methods may fail.

In comparison to background subtraction, optical flow techniques offer a higher degree of accuracy in detecting moving objects. However, one of the challenges with optical flow is its complexity and the increased processing time, as it requires more computational resources and may introduce noise due to dynamic background changes. The method proposed by Horn and Schunck [7] has shown promising results, performing better than other techniques like Lucas and Kanade [8] in the context of aerial views for detecting moving objects.

6.2 Enhancing Optical Flow: Challenges and Solutions

Several enhancements to the optical flow technique have been proposed to address its challenges. Researchers have worked on improving the performance of optical flow in various scenarios. For example, indoor fixed cameras have been used with optical flow to detect objects in video streams by analyzing motion levels, regions, and the number of objects. While effective in these settings, the main limitation is the change in object velocity and lighting conditions in the environment, which can affect the accuracy of detection. In static camera setups, motion segmentation and optical flow algorithms are combined for tracking moving objects effectively. This hybrid method does not depend on foreground or background regions but utilizes pixel-by-pixel classification for segmentation. Moreover, in outdoor scenes, edge detection and gradient-based techniques in optical flow provide more robust performance, especially in conditions where light changes occur, and the background is dynamic [14][15].

6.3 Tracking Moving Objects in Complex Environments

For moving cameras, classification and motion clustering methods are used to detect moving objects by evaluating the relative motion in the video streams [10]. Optical flow is further enhanced by using small squares in aerial images, which are employed in estimating flow fields and combining them to improve the flow detection in color planes [9]. This hybrid approach effectively fuses data from different frames, providing better tracking performance. Additionally, Lucas and Kanade’s method, which combines optical flow with stereo camera setups for unmanned aerial vehicles (UAVs) [16], shows promising results for tracking moving objects in aerial surveillance. The combination of optical flow with stereo imaging allows for depth estimation, which improves the accuracy of object detection, especially in complex urban areas or aerial views.

6.4 Hybrid Methods and Performance Evaluation

Hybrid methods combining multiple algorithms have become more popular for detecting motion in challenging scenarios.

For instance, a hybrid method using temporal difference and optical flow has been proposed for detecting moving objects by calculating frame differences and filtering the resulting differential images using edge detection techniques. This hybrid approach improves the robustness of detection, particularly in static camera settings, although its performance declines when the camera is in motion [24]. In the evaluation of optical flow algorithms, various methods have been tested with synthetic data and added noise to assess their robustness in complex environments. For example, a comparative study of eight optical flow algorithms found that methods combining both optical flow and temporal difference performed well in high-complexity environments, but their performance dropped when dealing with moving cameras [23].

6.5 Object Detection in Low-Quality Videos

For scenarios where video quality is low, such as surveillance from web cameras or older devices, edge detection and sobel filtering techniques have been applied to improve moving object detection [31]. These methods are optimized to run on low-end hardware, processing images faster and reducing memory usage while maintaining reasonable accuracy. The result shows a 45.5% reduction in object detection time, with 14% less memory usage, compared to traditional methods. Additionally, in vehicle detection for surveillance in urban environments, techniques like Enhanced Dynamic Bayesian Network (DBN) and Bayesian Markov Random Field (MRF) have been explored to improve object detection under varied conditions [32][30]. These methods help reduce the effect of noise, variations in lighting, and shadows, providing a more accurate and stable detection process, particularly for vehicle counting and segmentation.

6.6 Future Directions and Hybrid Techniques

The future of object detection in ITS lies in further optimizing existing algorithms and integrating them with new technologies. The use of hybrid methods combining optical flow with deep learning techniques is particularly promising. Deep learning models, such as CNNs, can be integrated with traditional motion detection methods like optical flow to handle more complex environments. These hybrid models are expected to deliver better performance in real-time object detection, even in dynamic and cluttered settings. Additionally, research continues to improve performance in urban area navigation and vehicle detection, with more sophisticated models for object classification and tracking in aerial views and dynamic environments. With the development of new feature extraction techniques like SURF (Speeded Up Robust Features) [34] and ORB (Oriented FAST and Rotated BRIEF) [6], object detection systems will continue to evolve to handle increasingly challenging conditions.

TABLE_01 – Literature Survey on Vehicle & Speed Detection

Title	**Algorithm**	**Description**	**Accuracy**	**Range**	**Duration**	**Pixel**
Vehicle Detection and Speed Estimation Using Cascade Classifier and Sub-Pixel Stereo Matching	Haar Cascade Classifier	Detects vehicles based on Haar features.	95%	Detection within ROI(~ 10- 15 meters)	Real-time	1920x1080 pixels
	Sub-Pixel Stereo Matching	Computes displacement for speed estimation.	~90%	Up to 80 meters	4 ms per vehicle	0.01-pixel precision
	Stereo Triangulation	Converts disparity to distance for speed calculatio	High	0 - 80 meters	Real-time	N/A

		ns.				
Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning	RetinaNet	Detects motorcycles with helmet use classification.	95.3%	50 meters	0.059 sec/frame	1920x1080 pixels
	Kalman Filter + Visual Similarity	Tracks motorcycles across frames.	96.7%	50 meters	8 FPS	1920x1080 pixels
	Multi-Task Learning (MTL) CNN	Classifies helmet use, rider count, and position.	80.6%	~5-10 meters	Real-time	1920x1080 pixels
Vehicle Detection and Classification Using Magnetic Sensors	Magnetic Dipole Model	Detects vehicles	75%	Up to 10 meters	1.059 sec/frame	1920x1080 pixels
	Polynomial Speed Estimation Models	Estimates vehicle speed based	85-90%	Up to 10 meters	Real-time	960x540 pixels
	HOG + Classifier	Classifies vehicle type	90%	6-10 meters	Real-time	48x48 pixels

VII. OBJECT CLASSIFICATION

Object classification plays a crucial role in intelligent transportation systems (ITS), where the goal is to identify and categorize objects such as vehicles, pedestrians, and traffic signs from video streams. Various classification methods have been developed to perform this task, ranging from traditional machine learning approaches to more sophisticated deep learning models. Each of these methods offers distinct advantages and is used in different ITS applications. Below are the key classification methods discussed in the literature.

7.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) are widely used in object classification tasks due to their ability to separate data into different classes based on feature space. In the context of ITS, SVM has been used for classifying vehicles and pedestrians in video frames. SVM works by finding a hyperplane that maximally separates the different classes. The kernel trick is often applied to map the data into higher-dimensional space for better separation, especially when the data is non-linearly separable. SVMs have shown robust performance in detecting objects in controlled environments, but they require careful tuning of the kernel and regularization parameters to perform well in more complex, noisy environments. The performance of SVMs in ITS is highly dependent on the quality of the extracted features, such as those derived from edge detection or motion segmentation [6].

7.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) have revolutionized object classification in computer vision, including ITS. CNNs automatically learn feature hierarchies from raw image data, which eliminates the need for manual feature extraction. The network consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers, each extracting progressively more complex features from the input data. CNNs have been used in vehicle detection and traffic sign recognition due to their ability to capture spatial patterns and complex structures. However, CNNs require large amounts of labeled data for training and significant computational resources, especially when dealing with high-resolution images or video streams. Despite these challenges, CNNs provide state-of-the-art accuracy in object detection tasks within ITS applications [7][8].

7.3 Random Forests

Random Forests, an ensemble learning method, has been used for object classification in ITS to classify objects based on a set of features. The technique involves constructing multiple decision trees during the training phase, where each tree provides a classification result. The final prediction is made by aggregating the results of all the trees, typically using a majority voting scheme. Random Forests are particularly effective in handling noisy data and are less prone to overfitting compared to a single decision tree. They have been applied to vehicle detection and pedestrian classification in ITS, showing good performance, especially when there is limited data or when the dataset is imbalanced. Random Forests can classify objects based on spatial features, such as shape and texture, but may not be as accurate as deep learning methods in more complex scenes [9].

7.4 k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is a simple, non-parametric method used for object classification. The classification is based on the proximity of the test sample to its kkk-nearest neighbors in the feature space. In ITS, k-NN is used to classify objects like vehicles and pedestrians by comparing their feature vectors to those of known objects in the training set. Although k-NN is easy to implement and interpret, it is computationally expensive, particularly for large datasets, as it requires comparing the test sample to all samples in the training set. Additionally, it is sensitive to the choice of kkk and the distance metric. Despite its limitations, k-NN can perform well in simpler ITS applications where the feature space is not overly complex and the number of object classes is limited [10].

7.5 Decision Trees

Decision Trees are another widely used classification technique in ITS, especially for classifying vehicles, pedestrians, and other road objects. A decision tree makes decisions by recursively partitioning the feature space into regions based on feature values. Each internal node represents a decision based on a feature, and the leaf nodes correspond to the predicted class labels. Decision Trees are easy to interpret, making them suitable for applications where explainability is important. However, they tend to overfit when the tree is too deep, and their accuracy may degrade in environments with a large number of object classes. Decision Trees are often used in hybrid approaches, combined with other methods like Random Forests or SVMs, to improve their performance and generalizability in dynamic environments [11].

7.6 Hybrid Methods for Object Classification

Hybrid classification methods, which combine two or more classification techniques, have become increasingly popular in ITS. For instance, combining **Optical Flow** with **SVMs** or **CNNs** helps to improve the robustness of object detection in dynamic environments. These hybrid systems can handle moving objects in complex backgrounds by integrating motion information from optical flow with object classification features from CNNs. In addition, methods like **Bayesian Networks** and **Markov Random Fields (MRF)** are also employed in hybrid systems to account for uncertainties in the classification process. These probabilistic models help improve detection performance in real-world ITS applications by handling noisy data, varying light conditions, and partial occlusions [12][13].

7.7 Bayesian Methods

Bayesian methods, particularly **Bayesian Markov Random Fields (MRF)**, have been applied to object classification to handle uncertainty and improve robustness in challenging environments. Bayesian methods model the probability distribution of class labels and incorporate spatial dependencies between neighboring pixels, which can help improve accuracy, especially in environments with high noise or variable lighting. MRF is particularly useful for vehicle detection and tracking in urban environments, where traffic conditions are complex, and lighting conditions change frequently. The **Bayesian algorithm** improves object shape extraction, noise reduction, and shadow removal, making it a promising approach for ITS applications where environmental conditions are unpredictable [29][30].

7.8 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) has also been explored for vehicle detection and classification tasks in ITS. DRL is a machine learning method where an agent learns how to make decisions by interacting with the environment and receiving rewards based on its actions. In ITS, DRL has been used for real-time vehicle detection and classification, where the system learns to adapt to changing traffic patterns and road conditions. While DRL has shown potential, it requires large amounts of training data and computational resources, making it more suitable for research applications rather than real-world deployment in ITS. However, its ability to improve over time with continuous learning makes it a promising area of study for autonomous vehicle detection and traffic management [32].

VIII. VEHICLE TRACKING APPROACHES

Vehicle tracking is an essential component of intelligent transportation systems (ITS), where the goal is to monitor and track vehicles in real-time for various applications like traffic management, accident detection, and autonomous driving. Vehicle tracking approaches can be broadly categorized into **detection- based tracking**, **feature-based tracking**, and **deep learning-based tracking**. Below are the key approaches described in the literature with appropriate references to the content number.

8.1 Detection-Based Vehicle Tracking

Detection-based tracking methods involve detecting the vehicles in each frame and associating the detected vehicles across frames to maintain their identity. This process typically involves vehicle detection followed by data association.

Kalman Filter: The Kalman Filter is one of the most commonly used methods in detection-based tracking. It estimates the state of a moving vehicle (position and velocity) over time by recursively updating the estimates using new measurements. The Kalman Filter is ideal when the vehicle's motion is predictable, such as moving at a constant speed. This method is computationally efficient and widely used in traffic management systems [6].

Particle Filter: Particle filters are used when the vehicle's motion is nonlinear or non-Gaussian. Unlike Kalman Filters, which assume Gaussian noise and linear motion, particle filters approximate the vehicle's state by a set of particles. Each particle represents a potential state of the vehicle, and the filter propagates and updates these particles over time based on a motion model [7].

8.2 Feature-Based Vehicle Tracking

Feature-based tracking methods focus on detecting and tracking specific features of vehicles (e.g., corners, edges, or textures) across frames. These methods typically involve detecting robust features in the initial frame and matching them across subsequent frames.

Optical Flow: Optical flow is a technique that calculates the motion of pixels between consecutive frames. By analyzing the flow of pixel intensities, optical flow can track vehicles even when the camera itself is moving. Optical flow is particularly useful for tracking vehicles in aerial views, where traditional methods like background subtraction may fail [8]. While optical flow provides valuable motion data, it is computationally intensive and can be affected by noise or background motion.

Feature Matching: Feature matching involves detecting and tracking distinct features across frames. Techniques like **SIFT (Scale-Invariant Feature Transform)** or **SURF (Speeded-Up Robust Features)** are commonly used for this purpose. These methods detect features that are invariant to scale, rotation, and lighting changes, making them suitable for tracking vehicles across different frames, even in varying environmental conditions [9].

8.3 Deep Learning-Based Vehicle Tracking

With the rise of deep learning, vehicle tracking has seen significant improvements, particularly through the use of **end-to-end learning models** that simultaneously detect and track vehicles across frames.

Deep SORT is an extension of the SORT (Simple Online and Realtime Tracking) algorithm, which combines motion information from Kalman filters with appearance features extracted using a deep convolutional neural network (CNN). DeepSORT improves upon traditional tracking algorithms by incorporating appearance features to handle occlusions and changes in the vehicle's appearance. It has shown high accuracy, particularly in crowded environments where simple motion-based tracking methods struggle [10].

End-to-End Deep Learning Models: Some recent approaches use end-to-end deep learning models that simultaneously handle both detection and tracking tasks. These models, often based on CNNs, are trained on large datasets to learn how to detect and track vehicles in real-time. For instance, models such as **YOLO (You Only Look Once)** and **Faster R-CNN** have been used for object detection, while tracking is achieved by associating detection results across frames [11]. These models perform well even in complex environments with varying conditions, such as crowded roads or low-light conditions. However, they require significant computational power and a large amount of annotated data for training.

8.4 Hybrid Tracking Methods

Hybrid tracking methods combine multiple tracking techniques to leverage the strengths of each approach and address the limitations inherent in individual methods.

Hybrid Optical Flow and Kalman Filter: A common hybrid method involves combining **optical flow** with the **Kalman filter**. In this method, optical flow is used to estimate the motion of the vehicle, and the Kalman filter refines these estimates by incorporating predicted vehicle states. This combination allows for more accurate tracking, especially in environments with high noise or occlusions [12].

Particle Filter and Feature Matching: Another hybrid method combines **particle filters** with **feature matching techniques** such as SIFT or SURF. The particle filter tracks the motion of the vehicle, while feature matching helps to maintain the identity of the vehicle across frames. This approach is effective in dealing with highly dynamic environments and large- scale traffic [13].

8.5 Challenges in Vehicle Tracking

Occlusion: Vehicles often become occluded by other vehicles or objects in the scene, making it difficult to maintain tracking continuity. Hybrid tracking methods, such as DeepSORT, address this by incorporating appearance features to track vehicles even when they are partially occluded [10].

Motion Complexity: Vehicles in traffic exhibit complex motion, which can include sudden accelerations, decelerations, and turns. Particle filters and deep learning methods are particularly suited to handle these complex motion patterns, as they do not rely on simple linear motion models [7].

Lighting and Environmental Changes: Variations in lighting conditions, such as changes from daylight to dusk or during inclement weather, can degrade tracking performance. Feature-based tracking methods like optical flow are sensitive to such changes, but deep learning models tend to be more robust in these conditions [8].

Real-Time Processing: Real-time vehicle tracking requires algorithms that balance accuracy with computational efficiency. While deep learning-based methods like DeepSORT offer high accuracy, they require significant computational resources, which can be a limiting factor in real-time ITS applications [10].

Table_02 – Literature Survey on Vehicle & Helmet Classification

Author	**Approach**	**Description**	**Advantages**	**Limitation**	**Reference**
Horn and Schunck [7]	Kalman Filter	A recursive algorithm that estimates the state of a vehicle based on previous measurements.	Computationally efficient and effective in smooth motion tracking.	Assumes constant velocity and smooth motion, which may not always be the case	[6]
Lucas and Kanade [8]	Particle Filter	A method that uses a set of particles to estimate the vehicle's state in nonlinear and non-Gaussian conditions.	Effective in handling nonlinear and non-Gaussian motion.	Requires higher computational resources than Kalman Filter	[7]
Wojke, Bewley, et al. [10]	DeepSORT	Extends the SORT algorithm by incorporating appearance features via a CNN.	High accuracy in crowded environments; robust to occlusions.	Computationally intensive; requires large-scale datasets for training	[10]
Isard and Blake, Lowe [13]	Hybrid Particle Filter + Feature Matching	Uses particle filters for motion tracking and feature matching to maintain object identity.	Effective in handling dynamic environments and large-scale traffic	More complex than standalone methods and requires high computational resources.	[13]

IX. CONCLUSION

Vehicle tracking is an indispensable component of Intelligent Transportation Systems (ITS), facilitating the real-time monitoring and management of traffic flow, accident detection, and the implementation of autonomous vehicles. The development of various tracking approaches, ranging from traditional techniques like Kalman Filters and Particle Filters to advanced methods such as optical flow and deep learning-based models, has significantly enhanced the accuracy and reliability of vehicle tracking in diverse environments.

Traditional methods like Kalman and Particle Filters remain effective in scenarios with predictable motion and low noise, offering computational efficiency and ease of implementation. Feature-based approaches, such as optical flow and SIFT/SURF feature matching, provide robustness in dynamic conditions, particularly when integrated with hybrid tracking systems. However, these methods face challenges in complex environments with significant occlusions, lighting variations, and high traffic density.

REFERENCES

[1] G. E. Gürcanli and U. Müngen, Analysis of construction accidents in Turkey and responsible parties, Ind. Health, vol. 51, no. 6, pp. 581595, 2013. [2

[2] M. K. Zekri, Construction safety and health performance in Dubai, M.S. thesis, Grad. School Built Environ., Heriot Watt Univ., Edinburgh, U.K., 2013.

[3] K. Kavya and T. Pradeep, Causes and effects of construction acci dents, Int. J. Innov. Technol. Exploring Eng. Reg. Issue, vol. 9, no. 2, pp. 11291133, 2019.

[4] T. Umar, Applications of drones for safety inspection in the Gulf Coop eration Council construction, Eng., Construct. Archit. Manage., vol. 28, no. 9, pp. 23372360, Dec.2020.

[5] A. Puri-Mirza. (Dec. 8, 2020). UAE: Abu Dhabi contribution of con struction to the GDP 2019. Statista.

Accessed: Jan. 13, 2021. [Online]. Available: https://www.statista.com/statistics/818149/ uae-contribution-of construction-to-the- gdp-in-abu

[6] M. Mohan and S. Varghese, Arti cial intelligence enabled safety for construction sites, Int. Res. J. Eng.Technol.(IRJET),vol.6,no.6,pp. 16,Jun. 2019.

[7] (2019). Construction in the UAEKey Trends and Opportunities to 2023 [Internet]. Research and Market. Accessed: Sep. 5, 2019. [Online]. Avail able: https://www.globaldata.com/store/report/g dcn0454mr construction in-the-uae- keytrends-and-opportunities-to-2023/

[8] (2015). Middle Eastern Promise: Prioritizing Safety [Internet]. Atkins. Accessed: Sep. 2, 2019. [Online]. Available: https://www.atkinsglobal. com/en-GB/angles/all-angles/middle- eastern-promise-prioritising-safety [

[9] (2016). Health Statistics 2015 [Internet]. Health Authority Abu Dhabi. Accessed: Sep. 2, 2019. [Online]. Available: https://www.haad.ae/HAAD/LinkClick.aspx? leticket=-lF2Jo3NrPE%3d&tabid=1516

[10] P. Barss, K. Addley, M. Grivna, C. Stanculescu, and F. Abu-Zidan, Occu pational injury in the United Arab Emirates: Epidemiology and preven tion, Occupational Med., vol. 59, no. 7, pp. 493498, Oct. 2009.

[11] G. Zhang, R. Avery, and Y. Wang, “Video- based vehicle detection and classification system for real-time traffic data collection using uncali brated video cameras,” Trans. Res. Rec., J. Transp. Res. Board, vol. 1993,pp. 138–147, 2007.

[12] Y. Wang and N.Nihan, “Freeway traffic speed estimation with single-loop outputs,” Trans. Res. Rec., J. Transp. Res. Board, vol. 1727, pp. 120–126, 2000.

[13] C. Kyoungah, I. Lee, J. Hong, T. Oh, andS. W. Shin, “Developing a UAV based rapid mapping system for emergency response,” in Proc. SPIE Def., Security, Sens., Orlando, FL, USA, 2009, Art. no. 733209.

[14] B. Coifman, M. McCord, R. Mishalani, and K. Redmill, “Surface trans portation surveillance from unmanned aerial vehicles,” in Proc. 83rd Annu. Meet. Transp. Res. Board, Washington, DC, USA, 2004, pp. 1–9.

[15] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, “Airborne traffic surveillance systems: Video surveillance of highway traffic,” in Proc. ACM 2nd Int. Workshop Video Surveillance Sens. Netw.,NewYork, NY, USA, 2004, pp. 131–135.

[16] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, “Efficient road detection and tracking for unmanned aerial vehicle,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 1, pp. 297–309, Feb. 2015.

[17] M. McCord, Y. Yang, Z. Jiang, B. Coifman, and P. Goel, “Estimating an nual average daily traffic from satellite imagery and air photos: Empirical results,” Trans. Res. Rec., J. Transp. Res. Board, vol. 1855,pp. 136–142, 2003.

[18] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, “Methods of analyzing traffic imagery collected from aerial platforms,” IEEE Trans. Intell. Transp. Syst., vol. 4, no. 2, pp. 99–107, Jun. 2003.

[19] F. Yamazaki, L. Wen, and T. Vu, “Vehicle extraction and speed detection from digital aerial images,” in Proc. IEEE Int. Conf. Geosci. Remote. Sensing. Symp., Boston, MA, USA, 2008, pp. III-1334–III-1337.

[20] C. K. Toth and D. Grejner-Brzezinska, “Extracting dynamic spatial data from airborne imaging sensors to support traffic flow estimation,” ISPRS J. Photogramm. Remote Sens., vol. 61, no. 3/4, pp. 137– 148, Dec. 2006.