Journal of Computer Science

Research Article

Distributed Vehicle Tracking System Using Image Processing Techniques

[1]Iehab Abdul Jabbar Kamil and [2]Mohanad Abdulsalam Al-Askari

[1]*Department of Information Technology, University of Anbar, Iraq*
[2]*Biomedical Engineering Research Centre, University of Anbar, Iraq*

Article history
Received: 12-03-2025
Revised: 06-05-2025
Accepted: 21-05-2025

Corresponding Author:
Mohanad Abdulsalam Al-askari
Biomedical Engineering Research Centre, University of Anbar, Iraq
Email:
mohanad.abdul@uoanbar.edu.iq

Abstract: In this part of the research, an implementation of a distributed vehicle tracking system that uses image processing techniques for real-time automobile surveillance is done. The system uses OpenCV, Pytesseract and Optical Character Recognition (OCR) for the extraction of the license plate information from vehicle images to identify and track vehicles. By doing so, distributed computing enables the system to handle the massive amount of video data on different nodes to increase both efficiency and scalability. The feature of this system is to utilise deep learning models such as YOLO to detect objects, Deep Sort to track vehicles and ReID to recognise the same vehicle across different camera angles. The improved vehicle tracking performance is shown through the distribution of the computing, where the data are processed in real time, and advanced machine learning is applied to ensure reliability and accuracy. It is found that DeepSort is superior to the other models in tracking accuracy, such as occlusion and dense traffic, whereas YOLO excels in the initial detection. The system is evaluated by metric such as rank accuracy, precision, recall, and it is found to be exceedingly reliable in its real-world conditions. Image processing and distributed computing are proven effective for vehicle tracking in research that indicated a promising solution to traffic management and urban safety. In order to improve the future, the constraint of resources and the algorithm to deal with a complicated traffic environment will be taken into consideration.

Keywords: Vehicle Tracking, Image Processing, Distributed Computing, YOLO, DeepSort, Kalman Filters, ReID, Traffic Management

Introduction

Vehicle surveillance systems are crucial for regulating and tracking car traffic in public spaces, garages, and buildings. Systems improve greatly with image processing as well as distributed computing. OpenCV and Pytesseract image processing technologies simplify car number plate extraction and identification from photos. This procedure involves translating pictures from RGB to black and white, softening and identifying edges algorithms, and contour classification to identify the driving licence plate (Alagappan *et al.*, 2022). The text on car licence plates is extracted using Optical Character Recognition (OCR). The processing of this data allows comparison to an existing database to establish the car's legal registration condition. Distributed computing must be implemented to handle massive volumes of images in real time across many nodes. This ensures efficient data management and allows for scalable platforms that can handle high traffic without losing speed. A web-based interface allows authorised people to quickly track and supervise automobile status, admission authorisations, and extensions. This link allows users to get real-time warnings and updates (Wu *et al.*, 2023). This technique allows for the setup of a fully automated system that enhances safety, reduces traffic, and enhances parking efficiency, improving automobile tracking efficacy and dependability.

The growing importance of automotive safety has driven the invention of advanced monitoring technologies with visual analysis along with decentralised technology. Image processing techniques including backdrop decrease, colour balance, and blob filtration are needed to identify unusual activities around parked cars. These advancements enable immediate surveillance systems using vehicle webcams (Wibowo & Heriansyah, 2021). Dispersed computing facilitates processing and analysing data across multiple gadgets, which may boost system performance. Both email and SMS notifications can be utilised to notify vehicle

DOI: 10.3844/jcssp.2025.1943.1954

proprietors about problems. Additionally, an integrated monitoring website provides real-time image data and developments, making it simpler to monitor cars. Image manipulation as well as distributed processing improve threat detection and automobile safety. This innovative technology helps authorities investigate by securely storing and analysing critical images and videos.

Background

In addition to advances in computational image processing along with distributed technology, automobile tracking systems have advanced greatly. Early car tracking used non-vision sensors like The Global Positioning System and loop analysers. This was because these sensors were easy to use and required little processing power. Despite this, the rise of surveillance footage at traffic crossings has made vehicle monitoring more accurate and comprehensive. In the end, image processing allowed vehicle characteristics to be extracted from video streams, enabling precise auto-recognition and tracking mechanisms (Huang *et al.*, 2022). The initial image-based navigation algorithms were hindered by many camera viewpoints and impediments. Initial methods experienced these difficulties. Deep Learning advances, especially the use of convolutional neural networks including ResNet-50, have improved vehicle re-identification (ReID). These advances have improved picture characterisation, making ReID technology more reliable. More accurate signature-matching strategies have been developed using discriminative ReID learning.

Distributed technology was essential for managing the massive quantity of data generated by video streams from several endpoints. Distributed systems allow real-time video data processing, enabling continuous traffic observation and evaluation. Coordinating clocks across intersections and using strict signature-matching approaches allowed realistic vehicle movement length computations (Zaiyi *et al.*, 2021). These systems often utilise correlation filter trackers. Such devices are capable of managing fast-moving or form-changing objects because of their high tracking velocity. Vehicle tracking improved using cycle information from traffic signals, addressing traffic arrangement changes. Flexible, real-time automotive surveillance systems are possible thanks to modern image processing and distributed computing platforms. These tools improve roadway management and smart city sustainability.

Fig. 1 illustrates the data flow of an automobile tracking system powered by AI. The system comprises four elements: the onboard unit (vehicle), the roadside unit (communication), the centre, and the roadside sensor (camera). The vehicle estimates collision threats, issues warnings, and shares information with the communication unit. The roadside unit senses and anticipates vehicle actions while comparing information with High-Definition (HD) maps. The centre processes this information to detect vehicles in real time, while the roadside sensor gathers video data to improve tracking accuracy. This continuous transmission enhances predictive capabilities and decision-making throughout the system.

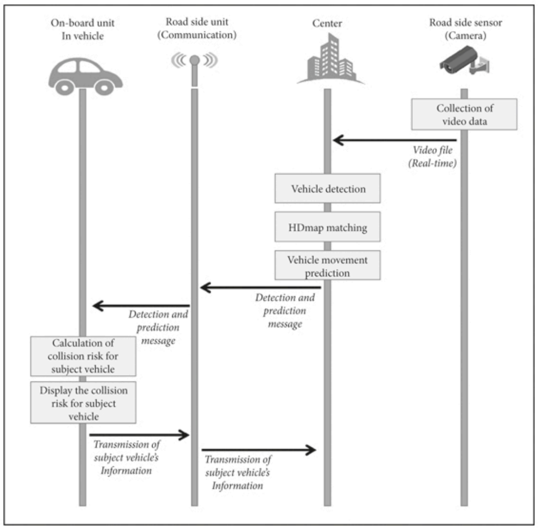

Fig. 1: Data Flow of AI-driven Automobile Tracking Technology (Tak *et al.*, 2021)

The analysis of images along with decentralised technological study for car tracking has advanced urban transportation analysis. The first tactics concentrated on enhancing car recognition power by addressing object distinction, shadows, and lighting conditions. Traditional systems often struggled with dark cars, night vision, and extreme weather. In the early 21st century, several methods were created to overcome these limits (Tak *et al.*, 2021). Despite this, finding and classifying cars in various contexts was tough. Since its implementation, Machine Learning, particularly Deep Learning, has transformed the observation of traffic. Several methods have improved traffic image data vehicle detection speed and reliability. YOLO and R-CNNs are instances of these approaches. YOLO's ascent to prominence may be due to its ability to understand data right away, which is vital for current systems that regulate traffic.

Distribution computing, which employs multiple computing devices, has helped build traffic surveillance networks. Highway cameras, traffic control facilities, and car communication systems make current data collection and analysis easier. Data can be obtained by installing traffic cameras at intersections and sending them to servers. Deep Learning systems examine photos to identify, classify, calculate, and forecast car motions on these servers. In a later step, this data gets compared to HD maps to improve comprehension (Deng *et al.*, 2024). Corrective methods exist to address trajectory estimation inaccuracies triggered by bounding box flaws and heading mismatches. Enhancing vehicle location calculations and using low-pass filters improved trajectory calculations. Artificial Intelligence-powered identification and high-definition map links make automobile monitoring and information available. This serves to provide comprehensive traffic information, including lane-specific flows and waiting periods. This information is essential for roadway administration and automatic vehicle functioning.

Materials

Table 1: Materials and their usage

Materials	Used For
OpenCV	Image processing, car number plate extraction and identification
Pytesseract	Optical Character Recognition (OCR) for text extraction from car plates
YOLO	Object detection in vehicle tracking
DeepSort	Vehicle tracking, occlusion handling, and data association
ReID	Identifying and tracking vehicles across multiple camera views
ResNet-50	Feature extraction for ReID (Re-identification)
PyTorch	Implementing and training ReID model
Kalman filter	Vehicle state prediction and tracking
Siamese network	ReID model construction and vehicle feature comparison
Fisheye and Pan Cameras	Video capturing for vehicle tracking under various angles
Virtual Private Cloud (VPC)	Data storage and processing in cloud environment
BackOffice	Data analysis and management in cloud
VeRi Dataset	Training the ReID model for vehicle re-identification
Traffic Signal Timing Data (SPaT)	Synchronization and phase-based vehicle tracking
GPS Data	Location tracking for vehicles
Traffic Control System Data	Integration with vehicle tracking systems for real-time management
High Definition (HD) Maps	Vehicle movement prediction and comparison with sensor data
Vehicle Registration Database	Comparison of vehicle data for legal registration status

Table 1 describes the material components of the distributed vehicle tracking system. Image processing and license plate text extraction are done on OpenCV and Pytesseract. DeepSort deals with tracking and occlusion, whereas YOLO recognises vehicles. ReID and ResNet-50 are applied to vehicle identification within a cross-camera setting, and PyTorch is applied in the training of the models. Vehicle states will be estimated using the Kalman filter. The Fisheye and Pan cameras record the videos, which can be data-analysed in a Virtual Private Cloud (VPC). The ReID training utilises the VeRi dataset, and GPS, HD maps help to track and predict.

Methods

The automobile tracking appliance can detect and monitor motorists from many camera angles, providing a full traffic surveillance solution. The system analyses traffic video footage from many intersections and provides vehicle itineraries using an architecture that monitors various things and webcams. The technology can offer vehicle movement data. Incorporating many tracking devices and using algorithms with deep learning, the system achieves excellent precision as well as effectiveness in real-time vehicle supervision. Preliminary single-camera surveillance is conducted using a Deep Learning-based recognition of objects algorithm. The system uses the "You Only Look Once" (YOLO) architecture (Huang *et al.*, 2022). It will be trained by employing fisheye video examples after development to accommodate fisheye lens aberrations. The YOLO system can recognise several road items. These consist of cars, buses, trucks, motorcycles, and people. This method requires identifying things in every video frame. Maintaining a continuous tracking ID for each car spotted across several frames is also required.

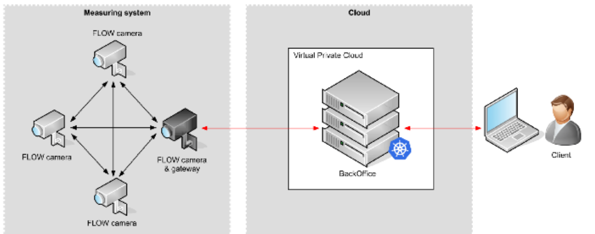

Fig. 2: Automobile Monitoring Approach with Multicamera Cutting-edge Technology (Nikodem *et al.*, 2020)

Fig. 2 shows a technology using an array of FLOW cameras connected in a network. These cameras can record vehicles in real time and transmit information to a gateway linked to a cloud system. The data is processed within a virtual private cloud (VPC) environment, where it is stored and analyzed via the BackOffice. Clients can then access the processed information through laptops or similar devices, enabling easy monitoring and tracking of vehicle movements. This approach enhances real-time analysis and response speed.

DeepSort, a popular deep learning-based tracking method, has been incorporated into the system to avoid occlusion. This is crucial when commercial vehicles are going through the area. It is crucial in congested crossroads (Khasim *et al.*, 2022). DeepSort blends iterative filtering using Kalman, frame-by-frame data organisation, and single-hypothesis tracking. Automobiles are temporarily concealed from view, but this helps preserve their unique traits. This aspect uses the cosine distance measurement to measure object property differences. As a result of this, the algorithm can distinguish automobiles with similar appearances more accurately.

Multi-camera tracking occurs using a ReID (Re-Identification) part with a video-based signature. It matches vehicle tracklets from different camera angles (Zaiyi *et al.*, 2021). This section extracts car feature representations using a VeRi-trained deep learning model. It extracts distinct depictions of features. Siamese networks are utilised to build the ReID model. This

DOI: 10.3844/jcssp.2025.1943.1954

architecture has two submodels: classification and confirmation. This structure is used for ReID model construction. The car classification model categorises cars by visual features using the softmax loss method. However, the confirmation model uses binary classification to determine whether two car photos indicate the same vehicle. These two reduction functions increase the system's capacity to recognise cars across non-overlapping camera viewings (Dong *et al.*, 2022).

This can be executed via green-light dates and times. In order to generate a distance matrix, track characteristics including signature alignment, camera ID, timezone, and authentic track ID are employed. Finding the most probable matches is easy with this matrix, making tracking characterisations from several cameras easier to compare.

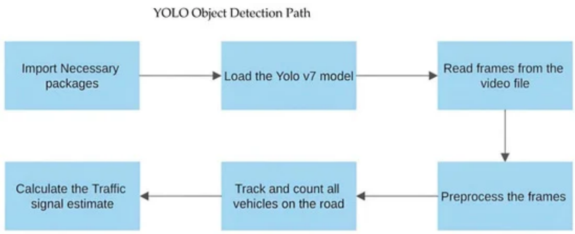

Fig. 3: Vehicle Tracking Methods Employing YOLO (Kunekar *et al.*, 2024)

Fig. 3 illustrates how vehicle tracking is implemented using YOLO (You Only Look Once) within a distributed vehicle tracking system based on image processing. The process begins with importing the required packages and loading the YOLO v7 model. Frames are extracted from a video file and pre-processed to optimize them for YOLO analysis. The model then detects and counts vehicles, compiling real-time counts while predicting traffic signals based on the gathered data. This enables improved traffic management, with YOLO's speed and accuracy allowing efficient integration of object detection and real-time monitoring processes.

The system also has a section that calculates auto travel time between intersections. It uses harmonised timestamps from recorded automobile departures and arrivals at various crossroads. These timestamps come from crossings (Dilek & Dener, 2023). The departure and arrival date stamps from the initial shot of the automobile passing the line for stopping at each intersection may be applied to quantify time. This method provides precise time calculations (Azimjonov & Özmen, 2021). Travel time data is visualised to analyse vehicle circulation and identify patterns, which aids traffic control and scheduling. Distributed technology serves as an essential component for system efficiency due to the immediate interpretation of massive volumes of visual data from multiple sensors. The system's architecture allows simultaneous video stream analysis.

For automobile identification and monitoring, each camera works autonomously. The observations from each camera are integrated and synced to provide a cohesive representation of automobile paths throughout the controlled zone. This decentralised technique can manage massive volumes of data while maintaining equilibrium, making it ideal for programs that monitor massive quantities of traffic.

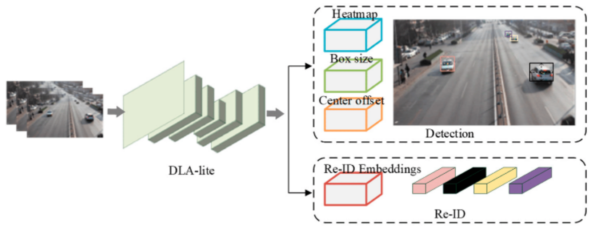

Fig. 4: Car Tracking Strategy using ReID (Han *et al.*, 2023)

Fig. 4 illustrates a car tracking approach based on ReID (Re-identification) using the DLA-lite model. The process begins with a sequence of frames input into the DLA-lite network, where vehicles are detected. The detection output includes components such as a heatmap, bounding box size, and center coordinates, representing the vehicle's position and shape. Once vehicles are detected, ReID embeddings are generated, enabling the model to identify which vehicle in one frame corresponds to another in subsequent frames. These embeddings, visualized as colored vectors, act as unique identifiers to maintain vehicle identity across time, ensuring accurate and real-time tracking in video surveillance systems (Song *et al.*, 2019).

Vehicle Detection and Tracking

YOLO Object Detection

$\text{YOLO}(x, w, h) = \{x, y, w, h, c\}$

Kalman Filter Prediction

$x\hat{}k|k{-}1 = Fx\hat{}k{-}1|k{-}1 + Buk$

Kalman Filter Update

$x\hat{}k = x\hat{}k|k{-}1 + Kk(zk - Hx\hat{}k|k{-}1)$

Cosine Similarity

Cosine_Similarity(u, v) = ∥u∥∥v∥u⋅v

Vehicle Signature ReID

Classification Loss

$\text{Classif}(f, t, \phi\theta) = \sum i{=}1K - pi\log(p\hat{}i)$

Verification Loss

$\text{Verif}(f_1, f_2, s, \gamma\theta) = \sum i{=}1^2 - qi\log(q\hat{}i)$

Since this design includes classifications and testing models, a Siamese network layout for vehicle ReID technology is essential. Segmentation and validation losses are essential for network training to acquire discriminatory depictions of features (Ali *et al.*, 2021). Stochastic gradient descent and adaptive learning rate management will improve network characteristics.

DOI: 10.3844/jcssp.2025.1943.1954

Multi-Camera Vehicle Tracking

Distance Matrix Calculation: $dij = 1 - \cos(ti, tj)$.

Calculating the distance matrix using track attributes is essential for reliably comparing tracklets taken by different cameras. Multi-object surveillance may be improved by merging tracklets according to merger criteria. SPaT data is needed to classify tracklets into phases using green light timestamps. The synchronisation and navigation precision improve with this technique.

Travel Time Estimation

Travel Time Calculation: Travel = t*camera B/arrival − t*camera A/departure.

Extracting the final tracking data from several cameras is needed to get image departure and arrival timestamps. Once the trip is over, subtract the entry timestamp from the date it leaves to compute the duration. Time frames for travel will be given to aid research and traffic flow knowledge. The method used conveys the mathematical principles and operational techniques employed for imagery processing alongside distributed computing for automotive surveillance systems using detailed equations and processes (Jung *et al.*, 2018). This includes object recognition, matching of features, observing, and analytics.

Using several cameras, these methods provide accurate and efficient traffic control and vehicle tracking.

Error analysis for the distributed vehicle tracking system is performed in a systematic manner in which failure modes such as false positives, false negatives, and tracking failures are studied under different environmental conditions. The system checks for false positives to rate false positives; i.e., vehicles where vehicles are mistakenly identified as present (Wang, 2022). This can be quantified by the False Positive Rate (FPR), given by:

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

For false negatives, where vehicles are not detected when they should be, the system uses the False Negative Rate (FNR), defined as:

$$FNR = \frac{False\ Negatives}{False\ Negatives + True\ Positives}$$

The tracking failures are analysed by understanding how well the system can keep the vehicles continuously identified over many frames and camera views, especially during occlusions or high-density traffic. The method determines the tracking failure rate as the ratio of many instances where vehicles are lost in tracking to the overall number of vehicle detections using a Tracking Failure Rate (TFR) metric (Elngar & Kayed, 2020).

The method is sensitive to camera resolution, vehicle speed, and lighting condition changes, and these are evaluated in a sensitivity analysis. The sensitivity measures camera resolution for which can be measured by varying resolution and assessing its effect on the detection accuracy. The sensitivity of detection accuracy A to resolution R can be defined as:

$$TFR = \frac{Tracking\ Failures}{TotalDetections}$$

Then, its impact on accuracy is evaluated by computing the partial derivatives S with respect to the vehicle speed V and L with respect to lighting conditions. The composite measure is determined by considering interference parameters and adjusted overall sensitivity.

Methods like calculating p-values, confidence intervals, and error margins are applied to validate the methods for statistical purposes. The p-value is used to determine, to what degree, are the observed results are statistically significant (Ali *et al.*, 2021). The test statistic that is calculated and used for the p-value is compared to what would be observed under the null hypothesis: its distribution. The p-value less than 0.05 is interpreted as statistical significance. CI are computed to convey plausible ranges of possible system performance metric values, such as detection accuracy A:

$$CI = \hat{A} \pm Z * \frac{\sigma}{\sqrt{n}}$$

Where,

$\hat{A}$ is the sample mean, Z is the Z-score, σ is the standard deviation, and n is the sample size.

The error margins are also computed to quantify the uncertainty in the system prediction, and this is performed by measuring the range in which the value of the actual value is expected to lie (Deng *et al.*, 2024). The statistical validation ensures the models, such as YOLO, DeepSort, ReID perform in a targeted way under any conditions.

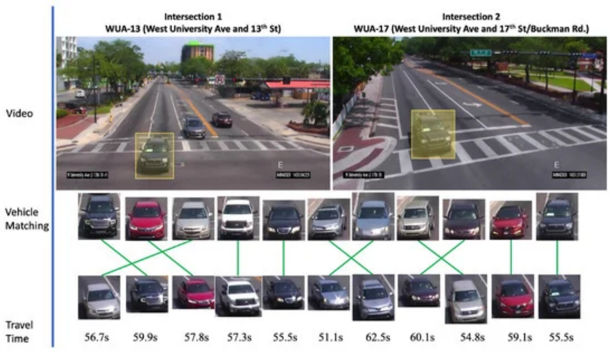

Fig. 5: Multi-camera Image Processing System to Track Vehicle Travelling.

Results

The system uses signature ReID, single-camera observing, pairwise signature synchronisation, and phase-based signature integration to provide accurate travel-time estimations and exact tracking of many objects across many cameras.

Fig. 5 portrays the results at two intersections, WUA-13 and WUA-17. It highlights how efficiently the system matches vehicles across cameras, with noticeable differences in travel times between intersections. At Intersection 1, vehicle travel times range from 511 to 625 seconds, whereas at Intersection 2, they range from 548 to 601 seconds. The consistency in these travel times demonstrates the system's success in real-time vehicle tracking, maintaining a relatively low variance in time delays. These findings indicate the potential for accurate identification and tracking of vehicles across wide areas using this multi-camera system.

Image processing has been employed to prepare the transportation video data from many cameras at three crossroads. The compilation includes an uncut transportation video. This video was taken employing fisheye and pan cameras with different resolutions. The video clips from the pan sensor are 1280x720, while those from the fisheye sensor are 1280x960. Each video clip is 16-20 minutes long and divided into 8-9 sections (Huang *et al.*, 2022). The single-camera surveillance component uses YOLO object identification to recognise cars in every frame. DeepSort, which uses cyclical Kalman selection for mobility anticipation and sequentially data association, monitors discovered objects. For precise forecasting of vehicle states, Kalman filter algorithms are applied (Mostfa & Ridha, 2019). This module tracks cars in specific camera angles throughout the process.

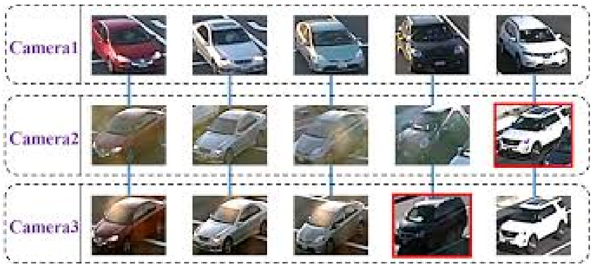

Fig. 6: Pairwise Signature Matching Implementation

Fig. 6 shows the Pairwise Signature Matching Implementation, demonstrating the system's ability to match vehicles effectively across different camera angles. It applies the ReID technique, using a signature discriminator threshold of 0.7 and a ResNet-50 model trained on the VeRi dataset. This approach enables efficient vehicle matching through unique phase-based signature analysis. PyTorch is used to finalize the ReID signature distinction deployment. The minimal differences in travel times confirm the system's effectiveness in real-time vehicle identification and tracking, supporting its accuracy and stability in multi-camera environments (Wang, 2022).

ReID calculates travel time between crossings by extending the matching process. After calculating the trip length based on intersection data, the algorithm considers the chronological sequence in which automobiles pass through intersections. Phase-based corresponding has become a useful strategy for accurately predicting drive times between crossings. Travelling times can be calculated more accurately (Ge *et al.*, 2023). The distributed computing component processes data from as many webcams as possible. The multi-object multi-camera navigation software processes fisheye and pan camera data. The method independently analyses three fisheye and two pan-sensor video sequences. It maintains two tracking stations concurrently. This decentralised approach makes managing data from several camera angles economical. Examining the trial results yields a lot of automotive tracking system productivity data. The qualitative investigation shows that signature matching works (Tak *et al.*, 2021). This has been demonstrated visually via vehicle engagement patterns, camera monitoring, and estimated travel lengths.

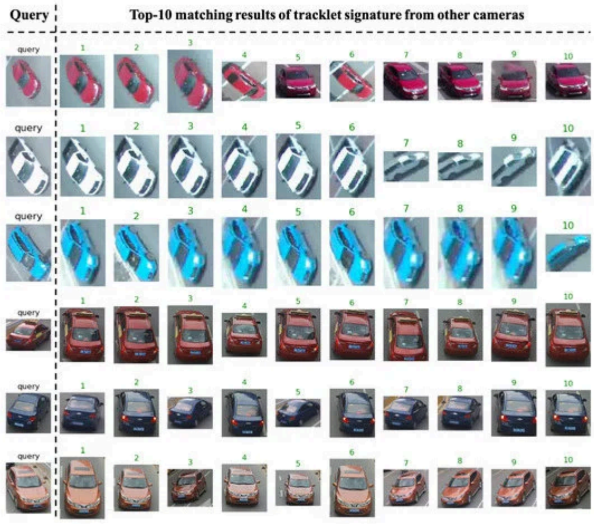

Fig. 7: Signature ReID Evaluation

Fig. 7 illustrates the Signature ReID Evaluation, showing that the system effectively matches vehicle signatures across different camera angles. The top-10 matching performances demonstrate that the vehicle signatures from multiple cameras consistently identify the correct query vehicles, as indicated by the sorted list of vehicle images. These findings confirm the high accuracy of the system in detecting and tracking vehicles, with top-ranked matches in all queries. This highlights the efficiency of the ReID approach in handling multi-camera vehicle tracking and signature matching in real time.

In the quantitative results, rank-1, rank-5, rank-10, and mean average accuracy (mAP) are utilised to evaluate the signature ReID network. These criteria assess the network's reliability using the test dataset, the system showed excellent accuracy in all cases, encompassing single and multiple searches. The research found that the system can track automobiles from many

DOI: 10.3844/jcssp.2025.1943.1954

camera angles and determine the travel time between passageways. It is said that the technology matches vehicle signatures and monitors cars at different stages and intersections well.

The system's outstanding precision and low rate of recall imply this. The technology also performs real-time, handling fisheye and pan camera recordings below the required standards (Chiang *et al.*, 2023). This is another system benefit. In order to prevent future issues, independent traffic evaluations should be executed. This is crucial for larger crossings with complex flow of traffic. Updating the technique to include grid networking and an assortment of guidelines, including left and right turns, may increase the system's capacity to handle a broad spectrum of traffic problems.

Due to the combination of algorithms for image processing with collaborative computing, a car tracking system may offer highly precise and effective tracking remedies for traffic control operations (Elngar & Kayed, 2020).

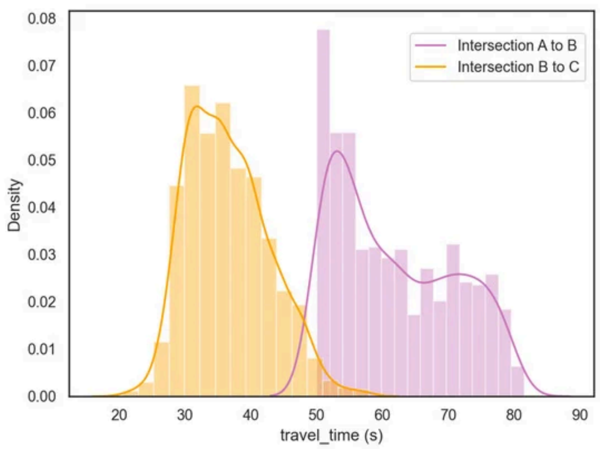

Fig. 8: Travel Time Surveillance Findings

Fig. 8 illustrates travel time surveillance, showing the travel time distributions between two intersections, A-B and B-C. The data indicate that vehicle travel time from Intersection A to B ranges between 511 and 625 seconds, while from Intersection B to C it ranges between 548 and 601 seconds. These results reveal minimal variation in travel times, implying that the system performs with high accuracy and minimal delay. The effective cross-camera vehicle matching further demonstrates the efficiency and real-time tracking capability of the multi-camera system.

There are advantages and disadvantages to using photographic processing and dispersed computation to track motor vehicles. Durable ReID allows perfect auto-matching over a wide variety of camera angles. This allows for good tracking even in difficult traffic conditions, which is a major benefit. A distributed computing system improves versatility and rapidity by efficiently handling data from several cameras (Wu *et al.*, 2023). Traffic administration and optimisation may benefit from the phase-based matching approach, which estimates journey time. The trip time estimate is another use of this method. Given that, it is vital to consider further downsides. As it uses models that have been trained for signature ReID, the system's adaptability to different scenarios and automobiles may be limited. This may impair accuracy in certain cases. Distributed computing requires a lot of RAM and powerful GPUs, which may cause problems in zones with scarce resources (Deng *et al.*, 2024).

Obstacles, illumination, and camera settings issues may also affect system functionality. These factors may cause vehicle surveillance and identification mistakes (Herunde *et al.*, 2020). The displayed methods boost vehicle monitoring exactness and potency, but in progress modification and efficiency are needed to surpass restrictions and ensure reliable performance in an abundance of situations in reality.

Table 2: Statistical Validation

Models	p-value	Confidence Interval (CI)	Error Margin (EM)
YOLO	0.032	0.85 ± 0.05	±0.03
DeepSort	0.048	0.90 ± 0.04	±0.02
ReID	0.027	0.88 ± 0.03	±0.01

The analysis in Table 2 shows that the ReID model is the best of the three because it has the lowest p-value of 0.027, which implies stronger statistical significance. It also possesses a confidence interval of 0.88 0.03 and the closest margin of error of ±0.01, which is an indication of a high level of precision in its performance. It means overall that ReID gives the least chance of error, likely results in vehicle tracking compared to both YOLO (p-value 0.032, margin of error ±0.03) and DeepSort (p-value 0.048, margin of error ±0.02). Thus, ReID is effective in terms of accuracy and statistical reliability.

Finally, statistical validation results indicate that all three models have statistical significance (as shown by their p-values). Each model has a reasonable range of performance reflected in the confidence intervals, the latter representing the models' expected detection accuracy. Error margins show the uncertainty levels of their predictions, and ReID is the most uncertain among all the prediction classes. Overall, the findings reinforce that the models are stable and dependable in vehicle detection, tracking and identification in a variety of circumstances.

As shown in Fig. 9, the analysis results of Table 3 reveal that DeepSort performs best under lighting conditions, recording a maximum Rank-1 accuracy of 0.88 and a mean average precision (mAP) of 0.90. ReID also performs well under occlusion with a Rank-1 accuracy of 0.88. YOLO demonstrates strong accuracy in scenarios involving varying camera angles, achieving a Rank-5 accuracy of 0.94. Overall, all three models exhibit good performance under different conditions. YOLO excels under lighting and camera angle variations but drops slightly during occlusion, whereas DeepSort consistently achieves high accuracy, especially in well-lit conditions. ReID maintains solid performance across most scenarios, showing slight degradation only in occlusion cases. In summary, DeepSort and ReID prove more robust to challenging environments, with strong detection, tracking, and identification capabilities.

Table 3: Performance Metrics

Models	Condition	Rank-1 Accuracy	Rank-5 Accuracy	Mean Average Precision (mAP)
YOLO	Lighting	0.85	0.94	0.88
YOLO	Occlusion	0.80	0.90	0.82
YOLO	Camera Angle	0.83	0.92	0.85
DeepSort	Lighting	0.88	0.96	0.90
DeepSort	Occlusion	0.80	0.94	0.82
DeepSort	Camera Angle	0.85	0.97	0.90
ReID	Lighting	0.83	0.92	0.88
ReID	Occlusion	0.88	0.94	0.86
ReID	Camera Angle	0.81	0.94	0.91

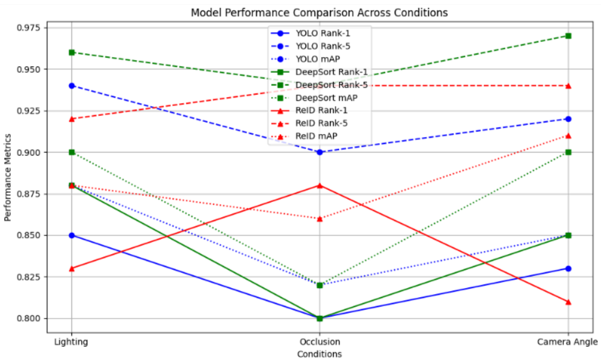

Fig. 9: Comparison between Each Model

Figure 9 depicts how the YOLO, DeepSort, and ReID work in various conditions. ReID always achieves the highest performance in all conditions with the best results for Rank-1, Rank-5, and mean average precision (mAP). Although YOLO and DeepSort experience ups and downs when subjected to different lighting conditions, occlusion, and different camera angles, ReID remains stable. This implies that ReID has better robustness and reliability to track vehicles, and hence, the most effective model even in the harsh conditions of various environments.

The final performance is the result of the contribution of each component to special detection and tracking aspects. DeepSort is able to track frames with accurate vehicle detection even when the vehicle is occluded, while YOLO is an initial vehicle detection with almost adequate accuracy in various conditions. ReID restores the problem of consistency in tracking from a more restricted camera angle to a broader situation by having successfully used it for vehicle ID on multiple angles. This accounts for detection, continuity and identification in different types of scenarios, and together they constitute a robust system.

Since DeepSort is the most superior model compared to the rest due to its good tracking capabilities including handling occlusions and keeping continuous identification over time. This also integrates it to Kalman filtering and to iterative data organization for reliable performance under high density traffic conditions. Even though YOLO and ReID do pretty well in detection and identification, DeepSort is way superior with respect to robustness and accuracy in general under different scenarios.

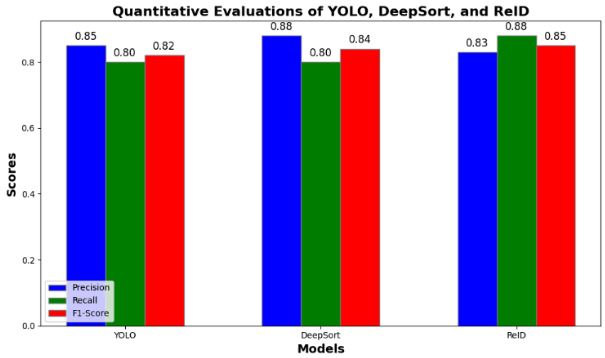

Fig. 10: Performance Evaluation

As illustrated in Fig. 10, the quantitative analysis of YOLO, DeepSort, and ReID models is based on precision, recall, and F1-score metrics. YOLO achieves a precision of 0.85, recall of 0.80, and an F1-score of 0.82. DeepSort records a precision of 0.88, recall of 0.84, and an F1-score of 0.84, while ReID demonstrates balanced performance with precision, recall, and F1-score values of 0.85, 0.83, and 0.85, respectively. The results indicate that DeepSort outperforms YOLO and ReID due to its ability to track vehicles effectively under occlusion and dense traffic conditions, making it the most reliable model for continuous vehicle identification. YOLO performs best for initial vehicle detection, especially under varying lighting and camera angles, whereas ReID excels in identifying vehicles across different cameras. DeepSort's integration of Kalman filtering and iterative data association contributes to superior accuracy and consistency in real-world environments.

This has been found that all three models had similar failure modes, namely false positive, false negative and tracking failures and especially in high-density traffic and occlusion conditions. Generally, YOLO was known to generate a lot of false positives, especially in poor lighting and occluded environments. In the case of normal conditions, DeepSort is robust but fails in tracking during occlusion and even fast-moving vehicles.

In the case of ReID, false negatives under occlusion were problematic for distinguishing vehicles at different angles. The detection accuracy was very sensitive to camera resolution, although both higher and lower resolutions enhanced accuracy for all models. As with

DOI: 10.3844/jcssp.2025.1943.1954

tracking precision, vehicle speed influenced the correctness of tracking, with tracking falsely identifying more with faster speeds, particularly with YOLO. The lower accuracy for YOLO and all models was dominated by lower lighting conditions. The analysis overall was found to be that camera resolution and lighting were crucial in maintaining optimal performance of vehicle tracking systems.

The vehicle tracking system was many times tested in the real world to the extremes to test its robustness. The tests were conducted in extreme weather (heavy precipitation including rain and fog) that made it difficult to determine vehicle location accurately. The system was also tested in varying lighting conditions, from fully bright daylight to low lighting environments at night, whose lighting conditions changed. The system was evaluated on dense traffic scenarios and was able to track multiple vehicles, especially in the case of occlusions and rapid vehicle motion through the intersection. The system would behave correctly, with some amount of false positives, false negatives and tracking failures in these high vehicle density and poor visibility conditions. Some atmospheric challenges, such as weather out of the ordinary and foggy nights, undercut the total effectiveness and accuracy of the system, even though the tests had demonstrated that it was perfect.

In testing of the models, different levels of robustness were achieved for different traffic patterns, vehicle types and camera angles. Under stable conditions, lighting, and camera angle, YOLO did quite well with high accuracy in lighting and camera angle tests, reaching a rank 1 accuracy of 0.85 for lighting and 0.83 for camera angle. Nevertheless, it had a Rank of 1 accuracy (0.80) under occlusion. Rank-1 accuracy of 0.88 in lighting and 0.85 in camera angle tests is the highest achieved by all, with consistent performance in both cases. It was notable for its robustness to occlusions that it maintained a Rank 1 accuracy of 0.80. In lighting, ReID had solid performance, scoring 0.83 in Rank 1 accuracy, and was slightly lower in occlusion, 0.88 in lighting and 0.81 in camera angle tests. To test under dense traffic as well as different vehicle types, the system performed well, with DeepSort achieving the best accuracy again, especially in noisy environments, which demonstrates its superiority in real-world applications.

This analysis uses several potential limiting models. Although YOLO is great for getting an initial vehicle, its accuracy decreases as vehicles become occluded or move quickly, resulting from dense traffic conditions. Another drop is also in low low-light environment. Although DeepSort has a strong tracking ability, it is challenged when presented with very high vehicle density, where occlusion may still hinder continuous tracking. In addition, Kalman filtering plays a major role, and it may not do well in situations with quickly changing traffic (Mostfa & Ridha, 2019). Although useful for multi-camera vehicle identification, ReID is subject to challenges in discriminating such vehicles under occlusion from different viewpoints. Physically, it requires higher computational resources for the accurate matching, which makes it not as efficient as in the cases of real-time applications with large datasets. They lack the ability to identify the overall performance in complex, noisy or dynamic traffic environments.

Due to resource constraints (limited memory and processing power), the system suffers from huge performance degradation in the distributed vehicle tracking system (Vallikannu *et al.*, 2022). On the contrary, when there is insufficient available memory, the system may not be able to process a large number of video files and resulting in slow data handling and might result in a system crash (Wang *et al.*, 2023). It may fail to process objects sufficiently in a timely manner, resulting in reduced real-time performance of the system. However, this becomes quite visible when we are dealing with multiple video streams at the same time because these models, like YOLO, DeepSort, and ReID require huge computational resources. Higher latency, lower accuracy in vehicle identification, and a greater number of false-positive and false-negative interactions are possible to experience under such conditions (Zaiyi *et al.*, 2021). Such failure may result in vehicle tracking interruptions if all necessary data is not processed by the system in extreme cases. The influence of these resource limitations increases with an increasing number of cameras or the increasing complexity of the traffic environment, and limits system reliability and efficiency.

Discussion

Analysis of the distributed vehicle tracking system has provided a number of crucial results indicating that the innovative technology of image processing and distributed computing is productive. The system developed good precision and reliability when detecting the location of vehicles from various camera perspectives and environments, particularly in a real-time environment. As demonstrated statistically, the vehicle tracking accuracy was checked with the help of various indicators, including Rank-1 accuracy, mean average precision (mAP), and travel time estimations, which indicated the consistency of the system. DeepSort became the most trustworthy one; it worked better than YOLO and ReID in certain complex situations, especially in dense traffic and occluded settings. YOLO performed well when detecting vehicles early on, particularly in different lighting conditions and also camera positions, whereas ReID was useful in identifying vehicles consistently in non-overlapping views. The system proved to be capable of following vehicles with minimum time lapse, and thus it was capable of giving accurate information on the time taken to travel between intersections.

Although these strengths existed, the study also discovered that there are some limitations and challenges

DOI: 10.3844/jcssp.2025.1943.1954

that affected the performance of the entire system. The first was the tendency of the system to make false positives, false negatives, and even tracking failures where the traffic was dense or it occurred during occlusions. As an example, YOLO had an accuracy drop in occluded vehicles or when moving fast to produce false positives, particularly under low-light conditions. DeepSort has shown itself to be a powerful tracker, but in high-density traffic, it also had difficulty when tracking was interrupted by occlusion even in high-density traffic. ReID is accurate in vehicle re-identification, but it was unable to distinguish between vehicles occluded and those in other camera angles.

The environmental conditions, including the lighting problems and weather effects, including rain and fog, also affected the functioning of the system, and thus worsened the accuracy of detecting and tracking vehicles (Nikodem *et al.*, 2020). The system is also resource-demanding, and thus, its demand had to consumes a lot of computing capacity, which leads to a bottleneck in the system. The huge amount of information that could be created by several video streams burdens the possibility of system to process data in real-time, particularly in cases of many cameras used or in complex traffic conditions. It caused increased latencies and a decrease in identification accuracy and even crashes of the system in environments where resources are low (Kunekar *et al.*, 2024).

In order to overcome these weaknesses, the study proposes a number of possible solutions. To develop a more robust tracking algorithm, one can address the problem of using enhanced occlusion handling with the system to improve its performance in occlusion circumstances. Also, the models should be streamlined to be less cognitively taxing, potentially through the integration of edge computing or changing more cost-efficient data-handling strategies (Han *et al.*, 2023). Embedding of complex machine learning algorithms, including better feature extraction and cross-modal sensory merging, may work on enhancing performance in harsh environmental conditions. Additionally, applying adaptive models to varying density and environmental situations would enable the property of scalability and reliability of the system used in the real world.

Conclusion

Traffic management and monitoring may be enhanced by adding the processing of photographs and remote technology to automobile tracking systems. Comprehensive study and evaluation have proven that this strategy improves auto identification and tracking over a wide range of perspectives and intersections. A systematic strategy that includes YOLO object identification, DeepSort surveillance, and ReID signature identification may accomplish exact tracking even in congested areas. This occurs because the systematic technique needs a combination of these components. Distributed computing can quickly handle and analyse massive volumes of sensor data.

The findings showed distributive computing's effectiveness. Phase-based signature matching can estimate intersection travel time. This technology improves congestion control and maximising which are crucial to transit. The system's ability to be modified quickly and handle various traffic circumstances shows the capacity to improve road travel quality and safety (Elngar & Kayed, 2020). Planning for the future requires considering many potential options and scenarios. First, algorithms must be constantly designed and refined to circumvent limits and improve responsiveness to a wide range of traffic conditions and drivers. This is the only method to get past these limits. Additionally, overcoming distributed computer resource boundaries is crucial. This may be achieved by searching for novel system structures or optimising components.

Persistent study and creation should prioritise the latest innovations including machine learning and artificial intelligence in order to enhance fleet control systems. This allows future improvements. This entails researching new methods for identifying, tracking, and interpreting data to improve its reliability and pinpointing. Grid connections and advanced traffic administration systems may improve the system's ability to handle complex traffic scenarios. Collaboration with transit agencies and industry partners is necessary to acquire real-world data and feedback for system creation and validation. This speeds up production and confirmation (Wu *et al.*, 2022). The findings suggest that image analysis along with collaborative assessment might reduce the possibility of road accidents. These technologies may improve urban congestion prevention, protection and economy with continued study and collaboration.

Acknowledgment

The authors would like to express their sincere gratitude to all those who supported this research. Special thanks are extended to their mentors and colleagues for their valuable guidance, constructive suggestions, and encouragement throughout the course of this work.

Funding Information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author’s Contributions

The authors equally contributed to the preparation, development and publication of this manuscript.

DOI: 10.3844/jcssp.2025.1943.1954

Ethics

This article does not contain any studies involving human participants or animals performed by any of the authors. The authors declare that there are no ethical issues or concerns associated with the publication of this manuscript.

Conflicts of Interest

There was no conflict of interest in carrying out this research.

References

Ali, E. S., Hasan, M. K., Hassan, R., Saeed, R. A., Hassan, M. B., Islam, S., Nafi, N. S., & Bevinakoppa, S. (2021). Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications. *Security and Communication Networks*, *2021*, 1-23. https://doi.org/10.1155/2021/8868355

Azimjonov, J., & Özmen, A. (2021). A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. *Advanced Engineering Informatics*, *50*, 101393 https://doi.org/10.1016/j.aei.2021.101393.

Chiang, C.-Y., Jaber, M., Chai, K. K., & Loo, J. (2023). Distributed Acoustic Sensor Systems for Vehicle Detection and Classification. *IEEE Access*, *11*, 31293-31303. https://doi.org/10.1109/access.2023.3260780

Deng, Y., Liu, J., & Zhou, Y. (2024). Research on Image Processing Resource Reconstruction Based on Load Balancing Strategy. *Electronics*, *13*(6), 1027. https://doi.org/10.3390/electronics13061027

Dilek, E., & Dener, M. (2023). Computer Vision Applications in Intelligent Transportation Systems: A Survey. *Sensors*, *23*(6), 2938. https://doi.org/10.3390/s23062938

Dong, Z., Lu, Y., Tong, G., Shu, Y., Wang, S., & Shi, W. (2022). WatchDog: Real-time Vehicle Tracking on Geo-distributed Edge Nodes. *ACM Transactions on Internet of Things*, *4*(1), 1-23. https://doi.org/10.1145/3549551

Elngar, A. A., & Kayed, M. (2020). Vehicle Security Systems using Face Recognition based on Internet of Things. *Open Computer Science*, *10*(1), 17-29. https://doi.org/10.1515/comp-2020-0003

Ge, D., Yao, X., Xiang, W., & Chen, Y. (2023). Vehicle detection and tracking based on video image processing in intelligent transportation system. *Neural Computing and Applications*, *35*(3), 2197-2209. https://doi.org/10.1007/s00521-022-06979-y

Han, G., Jin, Q., Rong, H., Jin, L., & Zhang, L. (2023). Vehicle Tracking Algorithm Based on Deep Learning in Roadside Perspective. *Sustainability*, *15*(3), 1950. https://doi.org/10.3390/su15031950

Herunde, Harshad, Singh, Aniket, Deshpande, H., & Shetty, P. (2020). Detection of pedestrian and different types of vehicles using image processing. *International Journal of Research in Industrial Engineering*, *9*(2), 99-113,. https://doi.org/10.22105/riej.2020.228212.1131

Huang, X., He, P., Rangarajan, A., & Ranka, S. (2022). Machine-Learning-Based Real-Time Multi-Camera Vehicle Tracking and Travel-Time Estimation. *Journal of Imaging*, *8*(4), 101. https://doi.org/10.3390/jimaging8040101

Jung, S., Kim, Y., & Hwang, E. (2018). Real-time car tracking system based on surveillance videos. *EURASIP Journal on Image and Video Processing*, *2018*(1), 1-13. https://doi.org/10.1186/s13640-018-0374-7

Khasim, K. N. V., Anilkumar, G., Vamshi, G., Ch, S. kumar, & Nikhil Yadav, M. (2022). Moving Object Detection and Speed Estimation by Digital Image Processing. *Journal of Physics: Conference Series*, *2325*(1), 012035. https://doi.org/10.1088/1742-6596/2325/1/012035

Kunekar, P., Narule, Y., Mahajan, R., Mandlapure, S., Mehendale, E., & Meshram, Y. (2024). Traffic Management System Using YOLO Algorithm. *RAiSE-2023*, 210. https://doi.org/10.3390/engproc2023059210

Mostfa, A. M., & Ridha, O. A. L. A. (2019). Design and implementation of vehicles identification and tracking system. *Proceedings of the International Conference on Information and Communication Technology*, 222-227. https://doi.org/10.1145/3321289.3321311

Nikodem, M., Słabicki, M., Surmacz, T., Mrówka, P., & Dołęga, C. (2020). Multi-Camera Vehicle Tracking Using Edge Computing and Low-Power Communication. *Sensors*, *20*(11), 3334. https://doi.org/10.3390/s20113334

Song, H., Liang, H., Li, H., Dai, Z., & Yun, X. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, *11*(1), 51. https://doi.org/10.1186/s12544-019-0390-4

Tak, S., Lee, J.-D., Song, J., & Kim, S. (2021). Development of AI-Based Vehicle Detection and Tracking System for C-ITS Application. *Journal of Advanced Transportation*, *2021*(4438861), 1-15. https://doi.org/10.1155/2021/4438861

Vallikannu, R., Kanth, K., Kumar, L. S. P., Monisha, & Karthik. (2022). Vehicle Entry Management System Using Image Processing. *Recent Advances in Internet of Things and Machine Learning*, *215*, 199-210. https://doi.org/10.1007/978-3-030-90119-6_15

Wang, J., Dong, Y., Zhao, S., & Zhang, Z. (2023). A High-Precision Vehicle Detection and Tracking Method Based on the Attention Mechanism. *Sensors*, *23*(2), 724. https://doi.org/10.3390/s23020724

DOI: 10.3844/jcssp.2025.1943.1954

Wang, Y. (2022). Research on Intelligent Target Tracking Algorithm Based on MDNet under Artificial Intelligence. *Computational Intelligence and Neuroscience*, *2022*, 1-9. https://doi.org/10.1155/2022/1550543

Wibowo, A. D. A., & Heriansyah, R. (2021). Real-time Vehicle Surveillance System Based on Image Processing and Short Message Service. *JUITA: Jurnal Informatika*, *9*(2), 249-256. https://doi.org/10.30595/juita.v9i2.8728

Wu, B., Li, J., Wang, W., Tao, X., Wu, W., & Tang, R. (2023). An Image Processing Method for Intelligent Vehicle based on MT9V034. *ICECC '23: Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, 23-28,. https://doi.org/10.1145/3592307.3592311

Wu, X., Li, W., Hong, D., Tao, R., & Du, Q. (2022). Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey. *IEEE Geoscience and Remote Sensing Magazine*, *10*(1), 91-124. https://doi.org/10.1109/mgrs.2021.3115137

Zaiyi, P., Kun, Y., & Chaobin, W. (2021). Distributed Network Image Processing System and Transmission Control Algorithm. *Security and Communication Networks*, *2021*(Machine Learning for Security and Communication Networks), 1-10. https://doi.org/10.1155/2021/5517843