Prototyping a graph-based approach for efficient vehicle tracking in a cluster controlled CCTV surveillance systems

Apurva NARAYAN [1], Charumathi PARIVALLAL[1], Olirva MAHESWARAN [1], Sarath CHANDRAN KALATHILPARAMBIL RAMACHANDRAN [2]

[1] Department of Computer Science and Engineering,

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, Tamil Nadu, India

[2] Department of Computer Science and Engineering,

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, Tamil Nadu, India

apurva2110547@ssn.edu.in, charumathi2110213@ssn.edu.in, olirva2110544@ssn.edu.in, sarathchandran@ssn.edu.in

Abstract: In this work, an advanced multimodal video summarization system is developed for a cluster-controlled traffic surveillance environment. Addressing the growing challenges of vehicle density and traffic complexity, the system employs state-of-the-art deep learning techniques for real-time vehicle detection, tracking, and route analysis. Utilizing algorithms like YOLO for object detection, the system ensures an accurate and efficient monitoring across multiple surveillance nodes. The optical character recognition (OCR) enables a detailed number plate recognition, while the multimodal data fusion enhances the robustness of vehicle tracking in dynamic conditions. Supported by data pipelines and frameworks, the system processes extensive CCTV footage to generate concise video summaries, optimizing the surveillance operations. The architecture offers scalable and adaptive solutions, aiming to improve the traffic management and emergency response by providing actionable insights through seamless integration of visual, audio, and textual data. This innovation has the potential to set new benchmarks in the intelligent traffic surveillance systems, with significant implications for public safety and urban mobility.

Keywords: Multimodal Video Summarization, Traffic Surveillance System, CCTV Footage Analysis, Real-time Monitoring, Urban Mobility.

1. Introduction

The increasing vehicle density in India demands a more efficient and secure monitoring system. The traditional manual CCTV analysis is time-consuming, often taking days to extract insights. With the CCTV market projected to grow to 22.35%, an efficient data management is crucial, as vast amounts of footage remain unwatched or irrelevant without any advanced filtering techniques. Currently, only one in three non-residential users engage in live monitoring, highlighting the need for automated surveillance solutions.

This work addresses the challenges of the urban traffic management by reducing the reliance on the manual monitoring and providing real-time, actionable insights. An automated surveillance can enhance public safety, reduce congestion, and improve traffic flow, benefiting the urban planners, law enforcement, and emergency response teams.

The existing systems suffer from storage and bandwidth limitations, an inefficient manual processing, and an inability to provide real-time tracking. This work proposes an automated, multimodal traffic surveillance system for efficient vehicle tracking, route mapping, and video summarization, overcoming these limitations and improving the urban traffic monitoring.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive literature review outlining recent advancements and limitations in intelligent surveillance. Section 3 presents the proposed system architecture. Section 4 details the implementation methodology, including hardware setup, video processing, data storage, and path reconstruction techniques. Section 5 evaluates the system through experimental results and performance comparisons. Section 6

https://doi.org/10.33436/v35i2y202508

discusses the strengths and limitations of the approach, while Section 7 concludes the paper with final remarks and directions for future research.

2. Literature survey

The recent advancements in multi-camera tracking, traffic detection, and intelligent transportation systems (ITS) have significantly improved the accuracy, scalability, and robustness. A notable contribution is the space-time-view (STV) hypergraph approach, which models the higher-order relationships among tracklets, overcoming occlusions and ambiguities by identifying dense sub-hypergraphs representing consistent trajectories (Wen et al., 2019). Similarly, a cross-view tracking method enhances the multi-human 3D pose estimation by integrating the geometric affinity measurements, improving the accuracy and scalability (Zhang et al., 2024).

The Deep learning has greatly advanced multiple object tracking (MOT), reducing the identity switching through the attention mechanisms and improved affinity learning strategies (Park et al., 2021). In ITS, the vehicle detection and counting have benefited from the deep learning models, with an enhanced RefineDet algorithm outperforming Faster R-CNN and YOLO for real-time traffic monitoring in congested conditions (Vikram et al., 2024). The visual odometry (VO) for multi-camera setups has also improved the robustness in the low-light and texture-deficient environments through the direct sparse tracking and sliding window optimization (Liu et al., 2018).

Now, beyond the visual data, the acoustic signal-based traffic detection employs the spectral subtraction and triangular wave analysis for improved noise reduction and vehicle separation, enabling the high-accuracy real-time detection in congested areas (Ma et al., 2022). Anomaly detection in road traffic has evolved through supervised, unsupervised, and semi-supervised learning, with model-based and classification-based methods offering real-time insights (Fei & Han, 2023).

The edge computing infrastructures in ITS have been optimized via three-tier models that manage the bandwidth and energy consumption efficiently. Algorithms like YLLO and BATS selectively process video frames and adapt transmission based on the available bandwidth, improving the resource utilization (Kumar, Pal & Kant, 2022). Despite these advances, challenges such as privacy concerns and high initial deployment costs persist, necessitating further research in the adaptive learning models and data fusion strategies (Zhang et al., 2018).

The CNN-based traffic monitoring frameworks integrating YOLOv5 and Deep-SORT have improved the vehicle trajectory mapping and intersection analysis, leveraging the homographic projection techniques for a precise spatial-temporal mapping (Pi et al., 2022). The hybrid approaches in anomaly detection, incorporating the trajectory-based modeling and neural networks, have enhanced the detection of deviations in traffic patterns (Santhosh, Dogra & Roy, 2020).

The multi-camera tracking and the natural language-based vehicle retrieval have also advanced, with pseudo-labeling and Circle Loss improving the cross-modal alignment and the retrieval accuracy (Vikram et al., 2024). A robust retrieval framework integrating textual and visual feature extraction reduces the reliance on annotated datasets while improving scalability and precision (Ngo et al., 2023). These advancements position the multi-camera tracking and intelligent surveillance systems as essential tools for the urban traffic management, anomaly detection, and real-time monitoring in the densely populated regions.

Despite these considerable advancements, many of the reviewed systems are siloed – specializing in object detection, tracking, or anomaly analysis – without combining these capabilities into a unified and operational pipeline. Most also assume a consistent visual input stream and do not integrate querying, path inference, and video summarization under a modular framework.

The proposed system builds on these state-of-the-art components by integrating them into a scalable, cluster-controlled traffic surveillance architecture. Using YOLOv5s for detection, EasyOCR for license plate recognition, and a graph-based backend for spatio-temporal modeling, the method provides vehicle-centric summaries and a path prediction through a combination of deterministic and learning-based algorithms. This integrated approach justifies the present contribution from the earlier task-specific efforts.

2.1. Gaps identified

Despite the significant advancements mentioned above, several gaps persist in the field of intelligent traffic surveillance:

1. **Real-Time Adaptability:** Current systems often lack the capability to adapt to rapidly changing conditions, such as variations in lighting, weather, or sudden traffic changes. This limits their effectiveness in real-time applications, where dynamic adjustments based on real-time feedback are crucial.
2. **Integrated Systems:** While research has progressed in individual areas like object tracking, anomaly detection, and vehicle counting, there is a lack of systems that integrate these capabilities into a unified framework.
3. **Scalability for Large-Scale Environments:** Many existing systems face challenges when scaling up for large environments. Efficient deployment of deep learning models particularly in optimizing computational resources and bandwidth for real-time performance.
4. **Diverse and Comprehensive Datasets:** There is a noticeable shortage of datasets that cover a broad range of real-world conditions, such as diverse weather scenarios, geographic settings, and varying traffic densities. The existing datasets and evaluation metrics may not fully reflect real-world complexities, limiting the generalizability of the current methods.

3. System architecture

Figure 1 illustrates a comprehensive surveillance system for processing and analyzing the CCTV footage. A query validation module ensures the input accuracy, followed by data retrieval and correlation. The core processing layer handles the image and video processing, integrating the external data via API calls. The structured storage enables an efficient retrieval, consolidating data into a centralized database for vehicle tracking and monitoring.

The system integrates the computer vision and geospatial data for real-time vehicle tracking, pattern detection, and automated enforcement. The YOLO-based models detect vehicles, capturing the bounding boxes, types, and license plates. By maintaining the position history, it estimates the speed and movement patterns.

Each camera feed is geotagged, synchronizing vehicle appearances across multiple feeds. A robust matching algorithm links the vehicles using license plates and attributes, enabling to composite the video generation with synchronized map visuals. The system overcomes challenges like occlusions and angle variations through license plate recognition and attribute-based matching. The timestamp synchronization ensures the cohesive tracking.

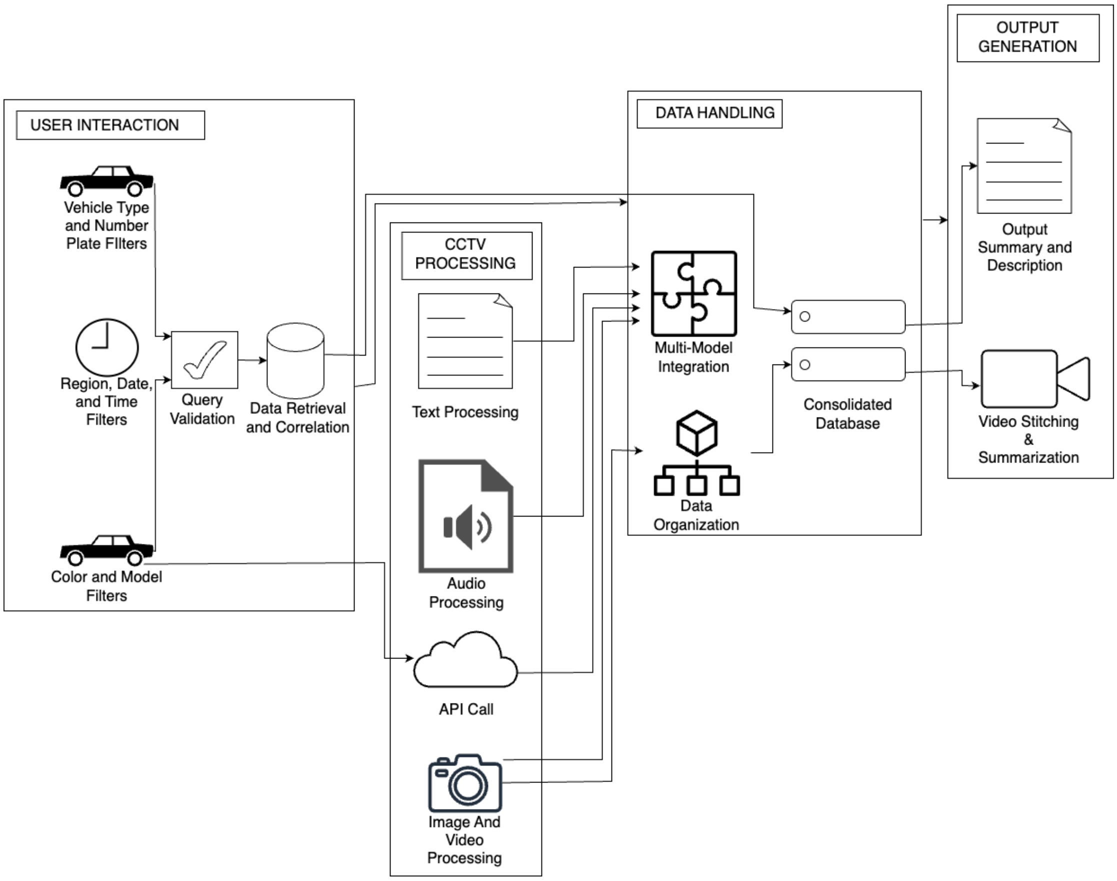


Figure 1. System design (Own research)

This solution has applications in traffic management, congestion analysis, law enforcement, and forensic investigations. It can also aid environmental monitoring by analyzing vehicle flow and emissions. Future enhancements include advanced deep learning models for re-identification, real-time edge processing, and cloud scalability, transforming the multi-camera surveillance into an intelligent, real-time tracking framework.

4. Implementation and methodology

4.1. Hardware module setup and CCTV clustering

The cluster prototype consists of Raspberry Pi 4 (8GB and 4GB RAM) with Nightvision cameras. Two cameras are placed at different locations to simulate a cluster, with a total of three clusters.

The system employs a manual clustering to organize the CCTV cameras into logical groups, optimizing the video management and retrieval. The CCTV Clustering Blueprint Module groups cameras based on proximity, traffic conditions, or monitored areas, forming controlled environments for efficient tracking and analysis. To enhance the efficiency, the Cluster Monitoring and Management Module oversees the data flow and performance. The load balancing distributes processing tasks evenly, preventing overload and ensuring smooth operations. The cameras are divided into predefined clusters, each with a dedicated monitoring and storage pipeline. This strategy improves video handling, query efficiency, and targeted analysis, making it crucial for the large-scale surveillance.

4.2. Video processing and data management

The system employs a modular pipeline for real-time video analysis and vehicle tracking. The input footage from the CCTV clusters is processed frame-by-frame using an optimized object detection module based on YOLOv5s, tailored for a low-latency performance on edge devices. The detected vehicles are filtered by class, and license plates are recognized through an integrated OCR unit that handles diverse visual conditions via internal preprocessing enhancements.

Post-processing, relevant vehicle metadata – including type, speed, timestamp, and plate number – are stored in a structured, queryable format. To support scalable analysis across multiple cameras, the data are transformed from camera-centric to vehicle-centric representations, mapping each unique license plate to a chronological series of sightings across all cluster nodes.

4.3. Graph-based data storage

This module structures the vehicle movement data as a graph as illustrated in Figure 2 for efficient storage and visualization using Pyvis. The graph is represented by triplets:

1. **Node 1**: CCTV location
2. **Node 2**: Vehicle
3. **Edge**: Timestamp of vehicle capture at CCTV

This structure enables an intuitive tracking and querying of the vehicle movements. Using Pyvis, an interactive graph is generated where vehicles are represented as blue nodes, the CCTV cameras as red nodes, and the edges contain the timestamp data, visualizing the connections between them as in Figure 3.

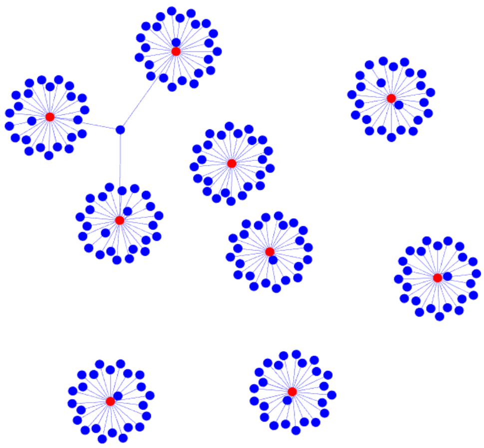


Figure 2. Visual representation of the graph-based storage system (own research)

Figure 3. A detailed view of the graph-based storage approach (own research)

4.4. Query handling

This module enables the efficient retrieval of the vehicle movement data based on user-defined queries, allowing tracking of a vehicle's path using the CCTV records within a specified timeframe. The users can input a vehicle's number plate ID and a start and end time to filter the relevant CCTV records. The system checks for the vehicle in the data and retrieves all CCTV locations where it was detected during the specified period.

The timestamps are processed as datetime objects for accurate comparisons. If the vehicle appears at multiple CCTV locations, the module returns a list of the locations and timestamps. If no records are found, the system notifies the user.

4.5. Path-finding and missing data handling

This module determines the possible vehicle paths when data is missing by employing multiple techniques to infer the most probable route. It constructs a graph of CCTV nodes using the

NetworkX and applies the Depth-First Search (DFS) to explore all potential paths between the known locations.

When the distance data is available, Dijkstra's Algorithm is used to compute the shortest path between the detected points, considering the edge weights for distance or travel time. It also identifies alternative routes with the same total cost.

For a more accurate path estimation, the module integrates the Graph Neural Networks (GNNs) using PyTorch Geometric. The GNN model learns the node relationships and infers the missing paths based on the node embeddings, capturing real-world variations beyond the traditional algorithms.

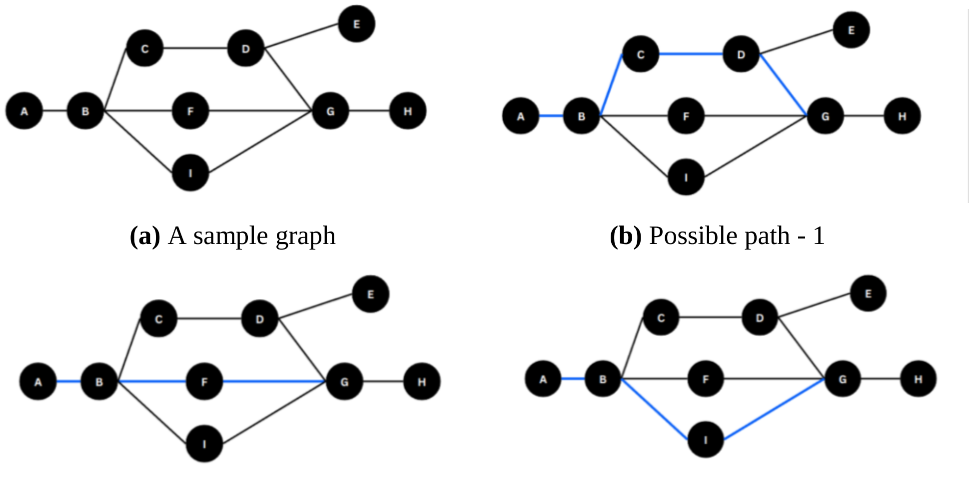
5. Experimentation

This section assesses the effectiveness of the system in reconstructing vehicle paths under incomplete surveillance data using three techniques: DFS, Modified Dijkstra's Algorithm, and Graph Neural Networks (GNNs). The experiments simulate a real-world urban scenario using data generated from a three-cluster CCTV prototype setup. Each cluster contains two night-vision cameras connected to Raspberry Pi 4 units, which log detections with timestamped metadata, including license plates and location tags. To enable the accurate vehicle re-identification across non-overlapping camera feeds, the system transforms camera-wise records into vehicle-centric timelines. Each unique vehicle identifier is mapped to its spatiotemporal sightings across all clusters.

A test case involving vehicle ID KA-05-AB-1234, observed at nodes CCTVA, CCTVB, and CCTVG between 01:45:00 and 17:40:00 on 2024-04-13, was used to evaluate the route reconstruction. During this period, the vehicle was not detected by the intermediate cameras, simulating the realistic surveillance blind spots due to occlusion, limited frame coverage, or temporary signal loss.

5.1. Graph representation

The road network is modeled as a graph where each CCTV camera is represented as a node. Roads between cameras are represented as edges. A sample graph is given in Figure 4 a.



(a) A sample graph **(b)** Possible path - 1

(c) Possible path - 2 **(d)** Possible path - 3

Figure 4. Graph representation and possible vehicle paths

For this experiment, the vehicle *KA-05-AB-1234* was detected at three key nodes between the given timeframe (2024-04-13 01:45:00 to 2024-04-13 17:40:00) CCTVA, CCTVB and CCTVG. Since only partial data were available, the vehicle could have taken one of the three potential paths from CCTVB to CCTVG:

Path 1 (Figure 4b): CCTVB —> CCTVC —> CCTVD —> CCTVG

Path 2 (Figure 4c): CCTVB —> CCTVF —> CCTVG

Path 3 (Figure 4d): CCTVB —> CCTVI —> CCTVG

5.2. Methodologies evaluated

To reconstruct the possible paths, the following three approaches were used:

5.2.1. Graph traversal and pathfinding using DFS

Depth-First Search (DFS) is a brute-force approach that explores all the possible paths exhaustively. The algorithm traverses the graph in a depth-first manner until all paths between the source and the destination nodes are discovered.

Possible Paths using DFS:

1. CCTVB —> CCTVC —> CCTVD —> CCTVG
2. CCTVB —> CCTVF —> CCTVG
3. CCTVB —> CCTVI —> CCTVG

It is implemented using NetworkX to recursively explore all possible paths between the source and the destination nodes. It successfully identifies all feasible paths but becomes computationally prohibitive as the number of nodes increases. DFS, while exhaustive, is computationally impractical for large graphs due to its O(V!) time complexity, making it unsuitable for the real-time vehicle tracking applications.

5.2.2. Shortest path identification using modified Dijkstra's Algorithm

Dijkstra's Algorithm is a well-known shortest-path algorithm that finds the minimal-cost path between two nodes in a weighted graph. A modified version was implemented to account for some multiple shortest paths instead of a single optimal one. The algorithm is optimized using a min-heap priority queue, resulting in a sub-second performance even in moderately large graphs.

All Paths Using Dijkstra's Algorithm:

1. CCTVB —> CCTVC —> CCTVD —> CCTVG
2. CCTVB —> CCTVF —> CCTVG
3. CCTVB —> CCTVI —> CCTVG

The modified Dijkstra's Algorithm exhibits $O((V + E) \log V)$ time complexity, which is significantly more efficient than the DFS and scales well for large graphs.

5.2.3. Learning-based approach using Graph Neural Networks (GNNs)

The GNNs are machine learning models that can learn graph representations and predict possible paths based on trained patterns. A pretrained GNN model was used to infer the missing vehicle paths.

All Possible Paths from CCTVB to CCTVG (Using GNNs):

1. CCTVB —> CCTVC —> CCTVD —> CCTVG
2. CCTVB —> CCTVF —> CCTVG
3. CCTVB —> CCTVI —> CCTVG

While the GNNs offer rapid inference with a theoretical time complexity ranging from O(1) to O(log V), they require substantial computational resources for training and frequent updates to maintain the accuracy. The high cost of retraining makes them impractical for this dynamic use case.

5.3. Performance comparison

A detailed comparison of the three approaches is provided in Table 1, Table 2 and Table 3, highlighting the differences in the performance, the time taken and the methodology used respectively.

Table 1. Performance Comparison of Different Approaches

Approach	Time Complexity	Space Complexity	Latency (Large graphs)
Modified Dijkstra	$O((V + E) \log V)$	$O(V + E + P)$	Moderate (scales well)
DFS (All Paths)	$O(V!)$	$(V + E)$	Very High (impractical for large graphs)
GNN (Pretrained)	$(1) – O(\log V)$	$(V + E + model)$	Low (fast inference, high training cost)

Where V represents the number of vertices, E represents the number of edges, and P is the number of the shortest paths.

Table 2. Time Comparison for Different Approaches

Approach	Time Taken
Modified Dijkstra	0.5 seconds
DFS (All Paths)	2.4 seconds
GNN (Pretrained)	0.2 seconds

Table 3. Comparison of Different Vehicle Tracking Methods

Aspect	Present Method	Optimized Vehicle Recovery: Leveraging AI and CCTV for Effective Tracking (Priya et al., 2025)	Deep Learning Based Traffic Surveillance System For Missing and Suspicious Car Detection (Kadambari et al, 2020)	Multi Camera Vehicle Tracking Using OpenCV & Deep Learning (Patil et al., 2023)
Features Used	Vehicle model, color, type, geolocation, and license plate information extracted through OCR	Vehicle model, color, texture, and number plate provided by the car owner for tracking within CCTV footage	Size, font, color of the license plate	Vehicle color, texture and shape
Modules Used	YOLO-based vehicle detection for object identification, EasyOCR for license plate recognition, graph-based tracking for movement inference, and DFS, Dijkstra, and GNN for missing data prediction	User-provided details are matched against extracted CCTV footage, followed by frame processing and movement analysis	YOLOv3 for object detection, pix2pix GAN for image enhancement, Tesseract-OCR for license plate recognition, and an error correction module for improving OCR accuracy	YOLOv3 for vehicle detection, object tracking using Kalman filters, vehicle Re-ID for multi-camera tracking, and speed estimation through frame-to-frame analysis

Output	An annotated video with highlighted vehicles and a structured visualization of vehicle movement using graph-based tracking	A reconstructed route map depicting the movement of the missing vehicle based on CCTV analysis	Successful identification of stolen or suspicious vehicles with enhanced license plate recognition	Comprehensive vehicle tracking across multiple cameras, including speed estimation, traffic congestion analysis, and predictive modeling for traffic flow insights

6. Discussions

The proposed cluster-controlled traffic surveillance system integrates key components—object detection, license plate recognition, vehicle path reconstruction, and interactive querying—into a unified and scalable framework. One of its primary strengths is the modular pipeline architecture, which allows real-time monitoring and summarization across multiple CCTV clusters with minimal latency. The use of the YOLOv5s for vehicle detection ensures a high accuracy with efficient inference speeds on the edge devices. In combination with the OCR-based license plate recognition and graph-based storage, the system achieves an effective vehicle re-identification across the non-overlapping cameras.

Another key strength lies in the hybrid path reconstruction module, which combines traditional search algorithms like the Depth-First Search (DFS) and Dijkstra's algorithm with the machine learning-driven Graph Neural Networks (GNNs). This hybrid design offers flexibility in balancing the inference speed and path accuracy depending on the application context and the computational budget.

However, the system also presents certain limitations. First, the current implementation relies heavily on the license plate recognition for vehicle matching, which can be unreliable under poor lighting, motion blur, or occlusion. Incorporating additional features such as vehicle color, shape, or manufacturer information could enhance robustness. Second, while the GNNs offer a superior inference performance in path prediction, their training process is computationally expensive and may not adapt well to real-time environmental changes without frequent retraining. This restricts their applicability in highly dynamic or large-scale deployments.

Moreover, the current testbed involves a limited number of CCTV nodes and controlled scenarios. Scaling this to a real-world urban environment will require addressing challenges related to bandwidth, synchronization across large networks, and privacy compliance. The system also assumes timestamp consistency and correct geotagging, which may not always be reliable in live deployments.

Future work may involve expanding the system's dataset coverage to diverse traffic and weather conditions, integrating unsupervised re-identification methods, and leveraging the federated learning to enable privacy-preserving model updates across the distributed clusters. Additional efforts can also focus on refining video summarization outputs and automating anomaly detection through real-time traffic behavior modeling.

7. Conclusion

The Graph-Based Vehicle Tracking System efficiently detects, tracks, and visualizes vehicle trajectories across multiple video feeds. By integrating the computer vision, GPS mapping, and real-time visualization, it enhances the CCTV-based surveillance, addressing challenges like vehicle re-identification and synchronized tracking. The key findings are (i) DFS is computationally infeasible due to its factorial time complexity; (ii) GNNs offer fast inference but require costly retraining, limiting practicality; (iii) Modified Dijkstra's Algorithm achieves the best balance between efficiency and completeness, making it the most suitable for real-time vehicle tracking and missing path reconstruction.

REFERENCES

Fei, L. & Han, B. (2023) Multi-Object Multi-Camera Trackig Based on Deep Learning for Intelligent Transportation: A Review. *Sensors.* 23(8), 3852. doi:10.3390/s23083852.

Kadambari, K.V. & Nimmalapudi, V.V. (2020) Deep Learning Based Traffic Surveillance System For Missing and Suspicious Car Detection. To be publishing in *Computer Vision and Pattern Recognition*. arXiv [Preprint]. https://doi.org/10.48550/arXiv.2007.08783. [Accessed 11 June 2025].

Kumar, P. P., Pal, A. & Kant, K. (2022) Resource Efficient Edge Computing Infrastructure for Video Surveillance. In *IEEE Transactions on Sustainable Computing*. 7(4), 774-785. doi: https://doi.org/10.1109/TSUSC.2021.3064245.

Liu, P., Geppert, M., Heng, L., Sattler, T., Geiger, A. & Pollefeys, M. (2018) Towards Robust Visual Odometry with a Multi-Camera System. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1-5 October 2018, Madrid, Spain*. pp. 1154-1161. doi: 10.1109/IROS.2018.8593561.

Ma, Q., Ma, L., Liu, F. & Sun, D. (2022) A Novel Driving Noise Analysis Method for On-Road Traffic Detection. *Sensors*. 22(11), 4230. doi:10.3390/s22114230.

Ngo, B. H., Nguyen, D. T, Do-Tran, N.-T., Thien, P. H. H., An, M.-H., Nguyen, T.-N., Hoang, L. N., Nguyen, V. D. & Dinh, V. (2023) Comprehensive Visual Features and Pseudo Labeling for Robust Natural Language-Based Vehicle Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern RecognitionWorkshops (CVPR).* pp. 5409-5418.

Park, Y., Dang, L. M., Lee, S., Han, D. & Moon, H. (2021) Multiple Object Tracking in Deep Learning Approaches: A Survey. *Electronics.* 10(19), 2406. doi:10.3390/electronics10192406.

Patil, A., Jadhav, S., Tigadi, A. & Chilad, B. (2023) Multi Camera Vehicle Tracking Using OpenCV & Deep Learning. In *2nd International Conference on Futuristic Technologies (INCOFT), 24-26 November 2023, Belagavi, Karnataka, India*. pp.1-5. doi:10.1109/INCOFT60753.2023.10425239.

Pi, Y., Duffield, N., Behzadan, A. H. & Lomax, T. (2022) Visual Recognition for Urban Traffic Data Retrieval and Analysis in Major Events Using Convolutional Neural Networks. *Computational Urban Science*. 2(2), 1-16. doi:10.1007/s4376202100031w.

Priya, R. L., Sengupta, S., Wadhwani, G., Phatak, T., Thakare, T. & Karotra, T. (2025) Optimized vehicle recovery: Leveraging AI and CCTV for effective tracking. *Communications on Applied Nonlinear Analysis*. 32(5s), 607–621. doi: 10.52783/cana.v32.3200

Santhosh, K. K., Dogra, D. P. & Roy, P. P. (2020) Anomaly detection in road traffic using visual surveillance: a survey. *arXiv* [Preprint]. https://doi.org/10.48550/arXiv.1901.08292 [Accessed 11 June 2025].

Vikram, A., Akshya, J., Ahmad, S., Rubini, L. J., Kadry, S. & Kim, J. (2024) Deep Learning Based Vehicle Detection and Counting System for Intelligent Transportation. *Computer Systems Science and Engineering*. 48(1), 115-130. doi:10.32604/csse.2023.037928.

Wen, L., Lei, Z., Chang, M.-C., Qi, H. & Lyu, S. (2019) Multi-Camera Multi-Target Tracking with Space-Time-View Hypergraph. *International Journal of Computer Vision*. 122(2), 313-333. doi:10.1007/s11263-016-0943-0.

Zhang, H., Fang, R., Li, S., Miao, Q., Fan, X., Hu, J. & Chan, S. (2024) Multi-Camera Multi-Vehicle Tracking Guided by Highway Overlapping FoVs. *Mathematics*. 12(10), 1467. doi:10.3390/math12101467.

Zhang, Y., Song, B., Du, X. & Guizani, M. (2018) Vehicle Tracking Using Surveillance With Multimodal Data Fusion. *IEEE Transactions on Intelligent Transportation Systems*. 19(7), 2353-2361. doi:10.1109/TITS.2017.2787101.

***

Apurva NARAYAN is an undergraduate student in Computer Science Engineering at Sri Sivasubramaniya Nadar College of Engineering, with research interests in artificial intelligence and computer vision. She is an IndiaAI Fellowship recipient and a recipient of the Millenium Fellowship by United Nations Academic Impact. As part of her summer internship at the Institute of Space Technology, she worked on hyperspectral imagery and prediction models. Her work was presented at the IEEE India Geoscience and Remote Sensing Symposium. She has co-authored two papers, Classification of Disaster Images Using Transfer Learning Techniques and Dynamic State Fuzzy Markov Model and Its Applications in Mobile Navigation through Prapsa Graphs. Her ongoing work emphasizes accessible and socially impactful technological solutions.

***

Charumathi PARIVALLAL is an undergraduate student in Computer Science at Sri Sivasubramaniya Nadar College of Engineering, specializing in software development with a strong focus on building accessible solutions. Her research interests include multimodal learning, knowledge graphs, rough decision-making, and large language models (LLMs). She is a recipient of the prestigious Google Women Engineers (WE) Scholarship and the Grace Hopper Celebration Scholarship. She has co-authored two research publications: SSN-MLRG at Text to Picto 2024: A BERT-Based Approach for Mapping French Sentences to Pictogram Terms (ImageCLEF Lab at CLEF 2024) & Rough Decision Making of Facial Expression Detection (Communications in Mathematics and Applications, Vol. 15, No. 1, 2024).

***

Olirva MAHESWARAN is an undergraduate student in Computer Science Engineering undergraduate at Sri Sivasubramaniya Nadar College of Engineering. Her primary area of interest is software development.

***

Sarath CHANDRAN KALATHILPARAMBIL RAMACHANDRAN is an Assistant Professor in the Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India. He received his Ph.D. degree from Anna University, Chennai, India. His research interests include Reconfigurable Computing, Energy Efficient Architecture Design, Deep Learning Techniques, Embedded Systems and Internet of Things.