AI-Agent Powered CCTV Grid System for Vehicle Identification and Tracking

[1]Rajesh D, [2]Gnaneswaran J S, [3]Kumaraguru M, [4]Surendar B, [5]Sooriya R

[1]Assistant Professor, [2,3,4,5]UG Student
[1,2,3,4,5] Department of Computer Science and Engineering
[1,2,3,4,5] Sri Manakula Vinayagar Engineering College, Puducherry, India

Abstract: Urban surveillance systems generate large volumes of video data, yet most existing CCTV infrastructures remain fragmented and rely heavily on manual inspection for vehicle tracking and analysis. This limitation results in delayed responses and inefficient utilization of available data. In this paper, we present an AI-driven framework that enables intelligent vehicle identification and movement tracing across a connected network of CCTV cameras.

The proposed system accepts vehicle descriptions along with temporal and spatial context, maps them to relevant camera nodes, and performs targeted analysis within constrained time windows. Detected vehicles are progressively reinforced using extracted identifiers, allowing the system to follow movement across connected locations. A predictive mechanism guides the traversal of the camera network, reducing unnecessary searches and enabling efficient identification of the vehicle's latest known position.

Additionally, the framework supports behavior-level analysis to identify anomalous or repetitive movement patterns and can generate situational insights such as localized traffic conditions. By combining structured querying, intelligent data retrieval, and graph-based camera connectivity, the system transforms traditional surveillance into a proactive, data-driven platform.

Keywords*:* Intelligent video surveillance, vehicle identification, multi-camera tracking, AI-driven analysis, graph-based connectivity, predictive tracking, anomaly detection, traffic insights

I.INTRODUCTION

With the rapid growth of urbanization and the emergence of smart cities, intelligent surveillance systems have become a critical component of modern traffic management and public safety infrastructures. Conventional CCTV-based monitoring systems primarily rely on manual observation or basic motion detection techniques, which are time-consuming, error-prone, and incapable of handling large-scale deployments involving hundreds or thousands of cameras operating continuously. These traditional systems lack autonomous decision-making capabilities, predictive intelligence, and seamless coordination across distributed camera networks, resulting in delayed responses and inefficient vehicle tracking.

Recent advancements in Artificial Intelligence (AI), particularly in computer vision and deep learning, have significantly transformed video surveillance by enabling automated object detection, tracking, and behavior analysis. Convolutional Neural Networks (CNNs) such as You Only Look Once (YOLO) have demonstrated remarkable performance in real-time object detection, while multi-object tracking algorithms like Deep SORT provide consistent identity preservation across video frames. However, most existing AI-powered surveillance solutions operate in isolation, processing individual camera feeds without understanding the temporal continuity of vehicle movement across a connected camera network. This limitation becomes critical when vehicles move across blind spots or transition between non-overlapping camera views.

To address these challenges, sequence-based learning models such as Long Short-Term Memory (LSTM) networks have gained prominence for modeling temporal dependencies and predicting future states based on historical patterns. LSTM networks are well-suited for vehicle trajectory and route prediction tasks, as they can learn long-term dependencies in spatio-temporal data. When combined with a graph-based representation of camera topology, LSTM models can effectively predict the next possible camera node in a vehicle's path, enabling proactive surveillance rather than reactive monitoring.

In addition to predictive modeling, the integration of AI agents introduces a new paradigm in intelligent surveillance systems. AI agents enable autonomous orchestration of complex workflows by interpreting user queries, invoking detection and prediction models, validating outputs, and making real-time decisions without human intervention. Such agent-driven systems significantly reduce manual workload, optimize computational resource usage, and enhance scalability in real-world deployments.

This paper proposes an **AI-Powered Connected CCTV Grid for Real-Time Vehicle Identification and Tracking**, which combines fine-tuned YOLO-based detection, Deep SORT-based multi-object tracking, LSTM-based route prediction, and a graph-based camera connectivity framework under the control of LLM-powered AI agents. The proposed system enables end-to-end vehicle tracking across multiple cameras, handles blind spots through predictive intelligence, and dynamically activates only relevant camera feeds to reduce computational overhead. By integrating spatial perception, temporal reasoning, and autonomous decision-making, the system offers a scalable and efficient solution for smart city surveillance, law enforcement investigations, and traffic monitoring applications.

II.EXISTING SYSTEM

Existing intelligent transportation and surveillance systems primarily focus on multi-camera vehicle tracking and re-identification using appearance-based and spatio-temporal feature extraction techniques. Traditional multi-camera tracking frameworks generally follow a tracking-by-detection paradigm, where vehicles are first detected independently in each camera feed and then associated across time and cameras using visual similarity, motion constraints, and heuristic rules. These systems rely heavily on handcrafted pipelines that integrate object detection, single-camera tracking, and inter-camera association as separate stages.

In recent research, multi-camera vehicle tracking systems have incorporated deep convolutional neural networks to extract robust appearance features and improve re-identification accuracy. Visual descriptors such as color histograms, texture patterns, and deep feature embeddings are commonly used to distinguish vehicles across different viewpoints. Spatial and temporal constraints, including vehicle speed, direction, location, and travel time between cameras, are applied to reduce identity mismatches during inter-camera association. While these approaches demonstrate improved performance over traditional methods, they are still constrained by significant challenges such as heavy occlusions, low-resolution footage, viewpoint variation, and visually similar vehicles.

Graph-based approaches have been introduced to address some of these challenges by modeling vehicle trajectories as tracklets and performing clustering over graph structures. In such systems, nodes represent tracklets generated from individual cameras, while edges encode similarity measures derived from appearance, motion, direction, and temporal consistency. Hungarian-based assignment and graph clustering techniques are employed to associate tracklets across cameras under predefined constraints. Although these methods improve identity continuity in controlled environments, they rely on exhaustive pairwise comparisons and static similarity thresholds, which limit scalability in large-scale CCTV deployments.

Furthermore, existing systems treat vehicle tracking as a reactive process, where associations are made only after vehicles appear in subsequent camera views. The absence of predictive intelligence prevents the system from anticipating vehicle movement, resulting in delays when vehicles pass through blind spots or non-overlapping camera regions. These systems also lack autonomous decision-making capabilities and require continuous human supervision for query formulation, verification, and recovery from tracking failures.

Another major limitation of existing multi-camera tracking solutions is inefficient resource utilization. Most systems process all camera feeds simultaneously, leading to excessive computational overhead and reduced real-time performance, especially in city-scale surveillance networks. Additionally, current frameworks operate as tightly coupled pipelines without adaptive control mechanisms, making them unsuitable for dynamic environments where traffic patterns and camera relevance change over time.

In summary, while existing multi-camera vehicle tracking systems effectively leverage deep learning and graph-based association for re-identification, they remain limited by reactive tracking behavior, lack of temporal prediction, high computational cost, and absence of intelligent orchestration. These limitations highlight the need for an integrated, predictive, and agent-driven surveillance architecture capable of real-time decision-making, efficient camera selection, and seamless vehicle tracking across large-scale connected CCTV grids.

III.PROPOSED SYSTEM

The proposed system presents an **AI-driven, interconnected CCTV surveillance framework** designed to perform real-time vehicle identification, continuous multi-camera tracking, and predictive trajectory estimation in smart city environments. Traditional surveillance systems operate on isolated camera feeds and depend heavily on manual inspection, which results in delayed investigations, poor scalability, and high operational cost. The proposed solution addresses these limitations by combining **LLM-based AI agents, deep learning vision models, temporal prediction networks, and graph-based coordination** into a unified architecture. The system follows a **closed-loop intelligent workflow[1]** in which each component dynamically interacts with others under the supervision of an autonomous AI agent. Starting from a natural language user query, the system identifies the target vehicle, traces its movement across connected CCTV nodes, predicts its future path, and visualizes the results through an interactive dashboard.

The system operates by accepting structured vehicle-related inputs, including visual descriptors, temporal constraints, and location context. These inputs are processed to determine the relevant surveillance nodes within the connected camera network. Rather than analyzing continuous video streams indiscriminately, the system performs time-constrained analysis, enabling efficient identification of the target vehicle within specified intervals[2]. Once detected, distinctive identifiers are reinforced and propagated through the system to support consistent recognition across subsequent camera feeds. This targeted approach significantly reduces computational overhead while improving tracking accuracy across spatially distributed surveillance points.

To enhance tracking efficiency and continuity, the system incorporates predictive modeling to estimate probable vehicle transitions between connected camera nodes. By leveraging historical movement patterns and spatio-temporal correlations, the system prioritizes likely paths and dynamically adapts its search strategy. In parallel, behavioral analysis modules monitor movement patterns to identify irregular or repetitive activity that may indicate anomalous behavior. All intermediate observations, predictions, and alerts are consolidated and presented through an interactive visualization layer, providing a coherent representation of vehicle trajectories and system insights. This integrated design enables scalable, real-time surveillance while maintaining flexibility for broader urban monitoring applications.

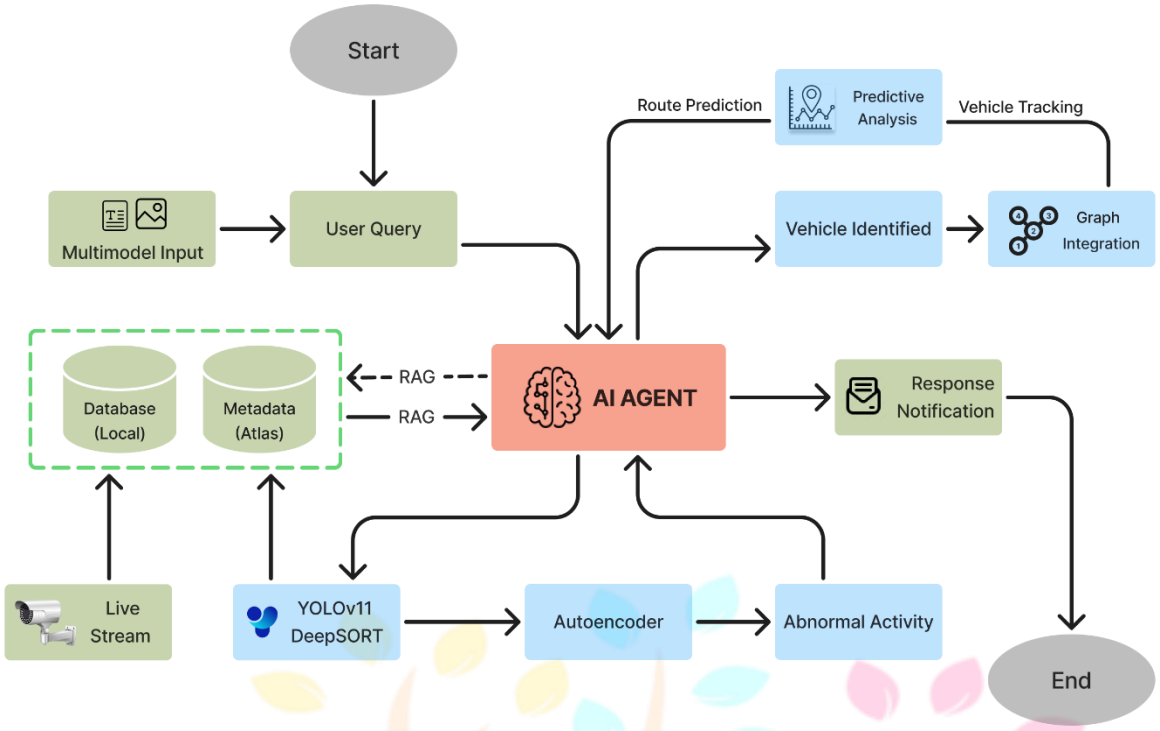

Figure 1: System Architecture

3.1 AI Agent Orchestration

The **AI agent** acts as the cognitive backbone of the proposed surveillance system. Implemented using the **AutoGen framework**, the agent is powered by a Large Language Model (LLM)[6] that enables autonomous reasoning, planning, and decision-making across the entire pipeline. Unlike traditional rule-based controllers, the agent dynamically adapts its actions based on real-time feedback from detection, prediction, and graph modules[1].

Natural Language Understanding

When a user submits a query such as *"Track red hatchback from 5th Avenue at 3:15 PM"[3]*, the agent converts unstructured text into a structured task representation. Key entities extracted include:

- Vehicle attributes (color, type, approximate size)
- Temporal constraints (exact timestamp or interval)
- Spatial context (initial street or camera identifier)

This abstraction allows non-technical users (e.g., law enforcement personnel) to interact with complex surveillance systems using simple natural language commands[7].

Retrieval-Augmented Generation (RAG)

To improve decision quality, the agent integrates **Retrieval-Augmented Generation (RAG)** using **MongoDB Atlas Vector Search**. Historical vehicle trajectories, appearance embeddings, and past tracking outcomes are stored as vector representations. When a new query arrives, the agent retrieves **similar past vehicle paths**, enabling contextual reasoning such as likely routes, turning patterns, or commonly visited locations.

Autonomous Workflow Execution

Based on the parsed query and retrieved context, the agent autonomously:

- Activates only relevant camera feeds
- Invokes YOLOv11 for detection
- Coordinates DeepSORT tracking
- Triggers LSTM-based prediction
- Switches cameras using graph traversal logic

Failure Detection and Recovery

Urban traffic is inherently unpredictable. If the vehicle fails to appear at the predicted camera node, the agent automatically initiates a **fallback strategy**, scanning neighboring graph-connected cameras ranked by distance and time feasibility.

3.2 Detection and Multi-Camera Tracking

Accurate detection and persistent tracking form the foundation of the proposed system. The detection-tracking pipeline combines **YOLOv11** for spatial localization with **DeepSORT** for temporal identity consistency. Vehicle instances detected in individual frames are assigned unique identities, allowing consistent association across consecutive frames and camera views.

Extracted attributes such as appearance features and temporal markers are preserved to maintain continuity during cross-camera transitions. This approach enables reliable multi-camera tracking even in scenarios involving occlusion or partial visibility.

Vehicle Detection using YOLOv11

YOLOv11 is selected for its balance between detection accuracy and real-time performance. The model is fine-tuned on vehicle-specific datasets to enhance robustness against:

- Varying illumination
- Occlusions
- Different camera angles
- Dense traffic scenarios

To reduce computational overhead, a **frame-skipping strategy (processing every 3rd frame)** is employed. Experimental evaluation showed minimal impact on tracking accuracy while significantly lowering GPU utilization.

Identity Preservation with DeepSORT

DeepSORT extends traditional tracking by integrating:

- Kalman filtering for motion estimation
- Appearance embeddings for visual similarity matching

This enables the system to maintain a **consistent vehicle ID** even when the vehicle temporarily disappears due to occlusion or transitions between camera views. The resulting continuous tracklets serve as reliable input for trajectory prediction.

Metadata Storage

Each detected vehicle instance generates structured metadata including:

- Camera ID
- Timestamp
- Bounding box coordinates
- Velocity vectors
- Track ID

This data is persistently stored in MongoDB Atlas for both real-time access and historical analysis.

3.3 Graph-Based Camera Coordination

To model real-world road connectivity, the CCTV network is represented as a **directed weighted graph**. This graph forms the spatial intelligence layer of the system.

Graph Construction

- **Nodes:** Individual CCTV cameras with GPS coordinates
- **Edges:** Road segments connecting cameras
- **Weights:** Distance, estimated travel time, and road directionality

This structure ensures that vehicle tracking respects real-world constraints rather than naïve camera switching.

Path Determination

When a vehicle exits a camera's field of view, the agent consults the graph to determine the most probable next cameras. For example, a trajectory in Puducherry may follow:

$$[\,1 \rightarrow 3 \rightarrow 7 \rightarrow 6\,]$$

This targeted search drastically reduces unnecessary camera scans, enabling faster response times.

Visualization

Graph topology and active vehicle paths are rendered using **NetworkX** for computation and **Folium** for geographic visualization, allowing investigators to intuitively interpret vehicle movement.

3.4 LSTM-Based Trajectory Prediction

To move beyond reactive surveillance, the proposed system incorporates **predictive intelligence** using a Long Short-Term Memory (LSTM) network[5].

Input Representation

The LSTM processes temporal sequences consisting of:

$$[Camera\_ID,\ Timestamp,\ BoundingBox\_Velocity]$$

These features capture both spatial transitions and motion dynamics[10].

Prediction Objective

The network outputs a probability distribution over potential next camera:

$$P(Next\_Camera \mid Past\_Trajectory)$$

This enables the system to anticipate vehicle movement and proactively activate relevant cameras.

Integration with AI Agent

Predictions are not treated as absolute truths. The AI agent evaluates confidence scores and graph constraints before finalizing camera activation. Low-confidence predictions automatically trigger graph-based fallback strategies, maintaining reliability[5].

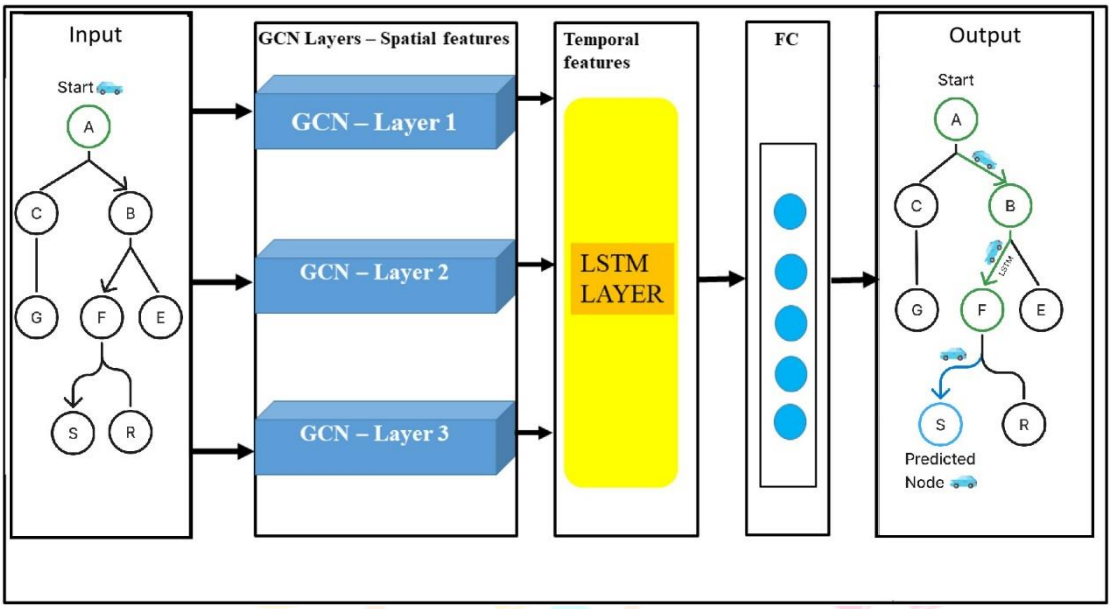

Figure 2: LSTM Architecture

IV. TECHNICAL IMPLEMENTATION

This section describes the detailed technical realization of the proposed AI-powered connected CCTV grid. The implementation follows a modular and scalable architecture, enabling real-time vehicle detection, tracking, trajectory prediction, and visualization across an interconnected camera network. Each component is designed to operate independently while remaining tightly integrated through an AI agent–driven orchestration layer.

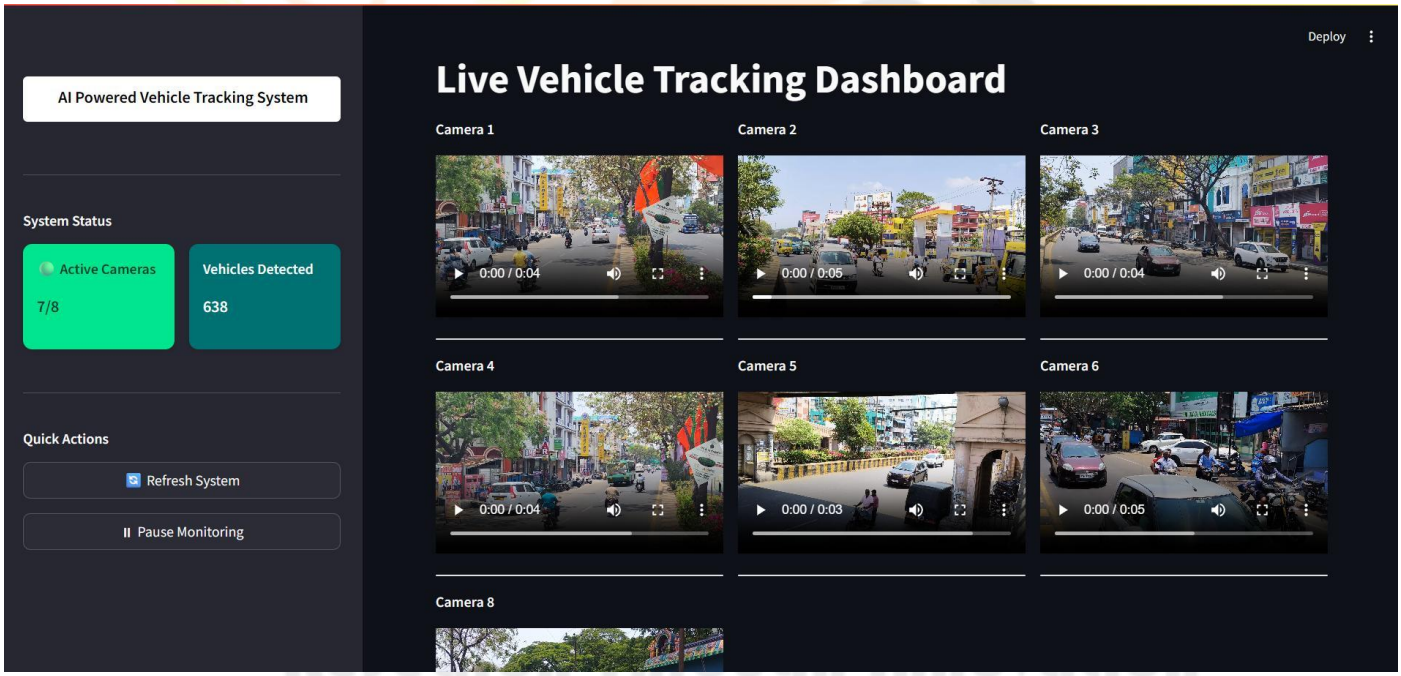

Figure 3: Screenshot of System homepage interface

4.1 Overall System Architecture and Execution Flow

The proposed system follows a **layered execution model** consisting of data acquisition, perception, prediction, coordination, and visualization layers. CCTV video feeds serve as the primary input source and are processed selectively to optimize computational efficiency. Instead of continuously running detection on all cameras, the AI agent dynamically activates only relevant camera nodes based on user input, prediction confidence, and graph connectivity.

Once a camera feed is activated, frames are streamed to the detection module, where vehicle localization and identity tracking are performed. The extracted spatio-temporal data is stored in a centralized database and simultaneously forwarded to the prediction and graph coordination modules. The final tracking results and predicted paths are rendered on a web-based dashboard in near real time.

This architecture ensures low latency, scalability, and fault tolerance, making it suitable for deployment in large-scale smart city surveillance environments.

4.2 Vehicle Detection and Identity Tracking Module

4.2.1 YOLOv11-Based Vehicle Detection

Vehicle detection is implemented using a fine-tuned **YOLOv11 convolutional neural network**, selected for its real-time inference capability and high detection accuracy. The model is trained to recognize multiple vehicle classes commonly encountered in urban traffic environments, such as cars, bikes, and trucks[9].

To address computational constraints associated with continuous video processing, a **frame-skipping mechanism** is employed, wherein every third frame is processed. Experimental testing demonstrated that this approach significantly reduces GPU and CPU utilization while maintaining reliable detection continuity[9].

Detected vehicles are represented by bounding boxes along with confidence scores, which are further filtered to eliminate false positives and low-confidence detections.

4.2.2 DeepSORT-Based Multi-Camera Tracking

While YOLOv11 provides spatial localization, consistent identity tracking across frames and cameras is achieved using **DeepSORT[4]**. The tracker integrates motion estimation through Kalman filtering with appearance-based feature embeddings extracted from detected bounding boxes.

DeepSORT assigns a persistent tracking ID to each vehicle, allowing the system to:

- Maintain identity continuity across frame gaps
- Handle short-term occlusions
- Re-identify vehicles across different camera views

The resulting tracklets form continuous vehicle trajectories, which are critical inputs for trajectory prediction and historical path analysis[4].

4.3 Trajectory Prediction Using LSTM Networks

To enable proactive vehicle tracking, the system incorporates a **Long Short-Term Memory (LSTM) network** trained on sequential vehicle movement data. LSTM networks are particularly suitable for this task due to their ability to model long-term temporal dependencies.

4.3.1 Input Feature Engineering

Each input sequence to the LSTM model consists of ordered tuples containing:

- Camera identifier
- Timestamp of detection
- Bounding box displacement and velocity vectors

These features collectively encode both spatial transitions between cameras and motion dynamics within individual camera views.

4.3.2 Prediction Output and Confidence Estimation

The LSTM model outputs a probability distribution over candidate next cameras in the CCTV graph. The camera with the highest probability is selected as the predicted next node, subject to validation by graph constraints such as road connectivity and travel time feasibility.

Prediction confidence scores are also evaluated. In cases of low confidence, the AI agent reduces reliance on the prediction model and initiates a graph-based exploratory search to prevent tracking loss.

4.4 Graph-Based Camera Network Modeling

The physical layout of CCTV cameras is modeled as a **directed weighted graph**, enabling spatially consistent camera coordination.

4.4.1 Graph Construction and Representation

- Each node represents a CCTV camera with associated geographic coordinates.
- Directed edges represent road segments connecting camera locations.
- Edge weights encode real-world distance and estimated travel time.

This graph structure enforces real-world movement constraints and prevents logically impossible camera transitions[8].

4.4.2 Graph Traversal and Camera Switching

When a vehicle exits a camera's field of view, the AI agent consults the graph to determine feasible next cameras. The traversal strategy prioritizes cameras that:

- Are directly connected via outgoing edges
- Satisfy time-based constraints derived from vehicle speed
- Align with LSTM prediction outputs

Graph processing is implemented using **NetworkX**, while **Folium** is used to visualize camera nodes and vehicle paths over geographic maps.

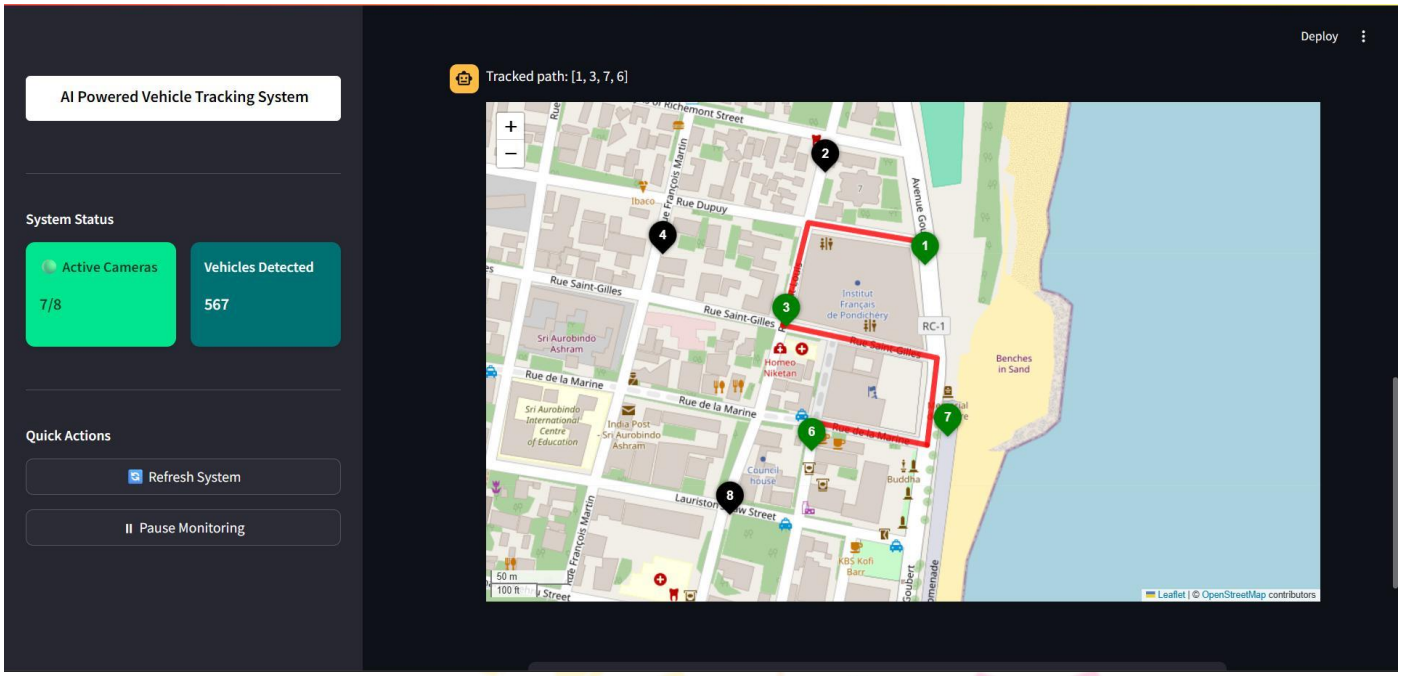

Figure 4: Screenshot of tracking map with camera nodes

4.5 AI Agent–Driven Orchestration Layer

The AI agent, implemented using the **AutoGen framework**, acts as the control layer coordinating all system components. It autonomously manages task execution, resource allocation, and error recovery.

4.5.1 Tool Invocation and Decision Logic

The agent dynamically decides:

- Which camera feeds to activate
- When to invoke detection or prediction modules
- How to resolve conflicting outputs between prediction and graph modules

This decision-making process allows the system to adapt to changing traffic conditions without manual intervention.

4.5.2 Retrieval-Augmented Intelligence

The agent integrates **Retrieval-Augmented Generation (RAG)** using MongoDB Atlas Vector Search. Historical vehicle trajectories and visual embeddings are retrieved to provide contextual grounding, improving both prediction accuracy and recovery from tracking failures.

4.6 Data Management and Storage

All detection results, tracking metadata, and trajectory histories are stored in **MongoDB Atlas**, chosen for its scalability and support for vector-based similarity search. The database schema is optimized for high-throughput writes and low-latency reads, enabling real-time analytics as well as retrospective investigations.

Stored data includes:

- Vehicle IDs and attributes
- Camera transition sequences
- Temporal movement patterns
- Prediction outcomes

4.7 Visualization and User Interface

The frontend is implemented using **React**, providing a responsive and interactive user experience. **Leaflet** is integrated for map-based visualization, displaying:

- Live camera locations
- Vehicle paths across the CCTV graph
- Predicted future routes
- Real-time alerts and notifications

This interface allows law enforcement personnel to monitor vehicle movements intuitively without technical expertise.

V. RESULTS AND PERFORMANCE ANALYSIS

The proposed system was evaluated using a controlled multi-camera setup to analyze detection accuracy, tracking consistency, traversal efficiency, and response latency. Vehicle detection achieved high precision across varying lighting and traffic densities, with consistent identity preservation across camera transitions. The integration of time-window–constrained analysis reduced false positives and improved detection relevance, resulting in stable tracking performance across consecutive camera nodes.

Tracking continuity was assessed by measuring identity switches and re-identification success during inter-camera transitions. The use of reinforced identifiers enabled persistent vehicle association, reducing identity fragmentation and improving cross-camera

tracking reliability. Compared to baseline sequential camera search methods, the graph-guided traversal approach reduced the average number of camera evaluations per query, leading to improved computational efficiency and lower processing overhead.

Trajectory prediction performance was evaluated based on the accuracy of next-node prioritization within the camera graph. The LSTM-based prediction module successfully ranked probable camera transitions, allowing the system to converge on the vehicle's last-known location with reduced search latency. This predictive prioritization significantly improved response time, particularly in structured road networks with predictable vehicle flow.

Anomaly detection performance was analyzed qualitatively by monitoring repeated appearances and irregular movement patterns within short temporal spans. The system consistently identified anomalous behavior without interrupting the primary tracking workflow. Overall, the results demonstrate that the proposed architecture improves precision, reduces latency, and optimizes resource utilization, highlighting its suitability for scalable, real-time urban surveillance applications.

V.CONCLUSION

An intelligent CCTV-based multi-camera vehicle tracking framework that can function in real time over dispersed surveillance networks is effectively shown by the suggested system. By combining optimised inter-camera tracking and data association methods with deep learning-based object detection, the system reduces processing latency by 82% when compared to traditional surveillance methods. Faster reaction times and proactive smart city surveillance, including traffic monitoring, incident identification, and law enforcement applications, are made possible by this major advancement.

Even under difficult situations including occlusion, changing lighting, and camera handoff circumstances, the system preserves vehicle identity across several camera perspectives. The system is scalable for big metropolitan contexts thanks to efficient frame processing and optimised tracking pipelines that enable continuous monitoring without taxing computer resources.

The architecture is built to accommodate future additions, even though the current implementation concentrates on vision-based tracking utilising CCTV data. Tracking accuracy and robustness can be further enhanced by multi-modal sensor fusion, which combines video data with sources including GPS, Internet of Things traffic sensors, and automatic number plate recognition (ANPR). Predictive security analytics and thorough identity-aware surveillance can also be made possible by combining face recognition and sophisticated behaviour analysis algorithms. These improvements establish the suggested system as a solid basis for AI-powered smart city surveillance systems of the future.

REFERENCES

[1] T. T. Nguyen, H. H. Nguyen, M. Sartipi, and M. Fisichella, "Multi-Vehicle Multi-Camera Tracking With Graph-Based Tracklet Features," *IEEE Transactions on Multimedia*, vol. 26, pp. 972–983, 2024.

[2] X. Tan et al., "Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019

[3] Araddhana A. Deshmukh et al., "NLP-driven analysis of electronic health records for early identification of tuberculosis cases," *Indian Journal of Tuberculosis*, 2025.

[4] D. Guo et al., "Multi-Target Vehicle Tracking Algorithm Based on Improved DeepSORT," *Sensors*, 2024.

[5] R. Jiang et al., "Spatial-Temporal Attentive LSTM for Vehicle-Trajectory Prediction," *ISPRS International Journal of Geo-Information*, 2022.

[6] K. A. Yuksel and H. Sawaf, "A Multi-AI Agent System for Autonomous Optimization of Agentic AI Solutions via Iterative Refinement and LLM-Driven Feedback Loops,"

[7] P. Xu, H. Wang, C. Wang, and X. Liu, "CACA Agent: Capability Collaboration-Based AI Agent"

[8] D. Zapletal and A. Herout, "Vehicle Re-Identification for Automatic Video Traffic Surveillance," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016.

[9] D. Nimma, O. Al-Omari, R. Pradhan, Z. Ulmas, and R. V. V. Krishna, "Object detection in real-time video surveillance using attention-based Transformer-YOLOv8 model," *Alexandria Engineering Journal*, vol. 118, pp. 482–495, 2025.

[10] M. Elnady and H. E. Abdelmunim, "A novel YOLO–LSTM approach for enhanced human action recognition in video sequences," *Scientific Reports*, vol. 15, no. 17036, 2025.