

---

# AI Cloud Platform for Smart City Traffic Monitoring and Analysis Using Multi-Dimensional Big Data and Deep Learning Models

---

[Ling Zhao](#) , [Jerry Gao](#) <sup>\*</sup> , [Yukta Mehta](#) , Aishwarya Ashok , Deepthi Desharaju , Padhmavathy Cebolu Srinivasan , Sowmya Manchikanti

Posted Date: 2 June 2026

doi: 10.20944/preprints202606.0144.v1

Keywords: intelligent transportation systems; traffic flow analysis; accident detection; spatio-temporal modeling; multimodal learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# AI Cloud Platform for Smart City Traffic Monitoring and Analysis Using Multi-Dimensional Big Data and Deep Learning Models

Ling Zhao, Jerry Gao \*, Yukta Mehta, Aishwarya Ashok, Deepthi Desharaju, Padhmavathy Cebolu Srinivasan and Sowmya Manchikanti

San Jose State University, San Jose, United States

\* Correspondence: jerry.gao@sjsu.edu

## Highlights

### What are the main findings?

- A unified multimodal framework effectively integrates vision-based detection, tracking, and sensor-based forecasting, enabling joint traffic perception and prediction within a single system.
- The proposed system achieves strong performance across tasks, including high detection accuracy (up to 93.60% precision, 93.91% mAP) and low-error traffic forecasting, demonstrating robust multi-source data fusion capability.

### What are the implications of the main findings?

- The integration of heterogeneous data sources (CCTV, drone, and IoT) provides a scalable approach for real-time traffic intelligence in smart city environments.
- The unified perception–prediction architecture enhances traffic monitoring and decision-making, supporting improved efficiency, safety, and proactive traffic management.

## Abstracts

Rapid urbanization has increased the demand for intelligent traffic management systems capable of real-time monitoring and predictive analysis. Modern smart city applications require the integration of heterogeneous data sources to improve traffic efficiency and safety. However, existing approaches frequently treat vision-based perception and sensor-driven forecasting as distinct processes, limiting their ability to capture dynamic traffic interactions and large-scale spatio-temporal dependencies. To address this limitation, we propose a unified multimodal traffic intelligence framework that integrates YOLOv8-based object detection, DeepSORT-based tracking, and DCRNN-based traffic forecasting, enabling joint perception and prediction within a single system. The framework consists of three tightly coupled modules. First, a vision module performs real-time detection of vehicles and accidents from CCTV and drone data. Second, a tracking module establishes temporal consistency by associating objects across frames. Third, a DCRNN-based forecasting module models traffic flow using IoT sensor data, capturing both spatial and temporal dependencies. In addition, a multi-cluster strategy with overlapping nodes is introduced to enhance scalability while preserving spatial dependencies within large-scale traffic networks. Experimental results demonstrate strong performance across tasks. The detection model achieves up to 93.60% precision and 93.91% mAP, while the integrated accident detection framework maintains robust generalization. The DCRNN model achieves low prediction errors, particularly for short-term forecasting.

**Keywords:** intelligent transportation systems; traffic flow analysis; accident detection; spatio-temporal modeling; multimodal learning

## 1. Introduction

The rapid development of smart city infrastructures has intensified the demand for scalable and intelligent traffic management solutions. Grand View Research estimates that the global intelligent traffic management system market will reach approximately USD 13.77 billion in 2025 and increase to USD 48.67 billion by 2033. Despite this growth, existing urban traffic systems remain limited in their ability to integrate real-time perception with predictive intelligence, particularly under increasingly complex traffic conditions driven by urban expansion.

Recent advances in artificial intelligence have introduced effective solutions for individual tasks, including computer vision-based detection, multi-object tracking, and spatio-temporal traffic forecasting. However, these approaches are typically developed in isolation, limiting their ability to capture dynamic interactions and large-scale spatio-temporal dependencies across heterogeneous data sources.

To overcome these limitations, this study develops a unified multimodal traffic intelligence framework in which perception, tracking, and prediction are jointly integrated within a cohesive architecture. Rather than treating these functions as independent modules, the framework enables coordinated interaction across tasks, thereby supporting consistent traffic representation over both spatial and temporal dimensions.

- At the modeling level, the framework incorporates YOLOv8, DeepSORT, and DCRNN to capture complementary aspects of traffic dynamics, allowing for integrated analysis that reflects both spatial structure and temporal evolution.
- From a data perspective, a multi-source fusion strategy is employed to integrate heterogeneous inputs from vision-based and sensor-driven modalities, including CCTV imagery, UAV data, and IoT sensor streams. This design enhances robustness under varying environmental conditions while providing a more comprehensive representation of traffic states.
- At the system level, the framework is supported by a cloud-native architecture deployed on AWS, which enables scalable, real-time data processing through dynamic resource allocation. The modular design further facilitates efficient handling of high-throughput data streams, improving both system flexibility and operational efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work and situates the present study within the existing literature. Section 3 describes the processes of data collection, preprocessing, and statistical analysis. Section 4 presents the modeling approaches employed for traffic flow analysis, accident detection, traffic tracking, and forecasting. Section 5 evaluates model performance for both CCTV and drone-based data using metrics such as Precision, Recall, and Mean Average Precision (mAP), providing insights into detection accuracy and system reliability. Section 6 outlines the design and implementation of the cloud platform. Finally, Section 7 concludes the paper and discusses directions for future research.

### 2. Related Work

Research on intelligent traffic analysis in smart city environments has evolved along several major directions, including vision-based traffic perception, multi-object tracking for dynamic scene understanding, and traffic forecasting based on sensor networks. In parallel, recent work increasingly emphasizes system-level integration through cloud computing and Internet of Things (IoT) infrastructures.

#### 2.1. Vision-Based Traffic Monitoring

Vision-based traffic monitoring constitutes a fundamental component of intelligent transportation systems, particularly for tasks such as vehicle detection, congestion identification, and incident recognition. In this domain, deep learning models have significantly improved the ability to detect and classify traffic objects from surveillance footage. Early region-based detectors such as Faster R-CNN provided accurate object detection frameworks for vehicle recognition and scene understanding, while real-time detection architectures such as the YOLO family enabled efficient large-scale traffic monitoring in urban environments.[1-3]. By leveraging convolutional neural

networks (CNNs), these approaches offer superior feature representation compared to hand-crafted methods and achieve lower computational cost than traditional sliding-window techniques [11]. As a result, they have been widely applied in CCTV-based systems for vehicle detection, congestion monitoring, and accident identification. Despite these advances, existing systems predominantly focus on object detection and visualization, with limited integration of higher-level analytical capabilities such as causal inference and decision support. Several studies further extend these capabilities. For example, Algiriyage et al. combined YOLO with SORT to estimate traffic flow from surveillance video for real-time analytics [15]. Sindhu employed YOLOv4 for vehicle detection under varying environmental conditions, supporting applications such as collision detection and emergency response [16]. Radojic et al. reviewed advancements in deep learning-based traffic surveillance, emphasizing the role of CNNs in detecting traffic violations [17]. Osman et al. proposed a computer vision-based adaptive traffic signal system that dynamically adjusts signal timing based on vehicle density [18].

Despite these advances, the project-level survey suggests that while many existing systems provide traffic flow visualization, fewer systems explicitly address congestion-cause detection or integrate advanced AI analytics into operational workflows.

## 2.2. Multi-Object Tracking in Traffic Monitoring

While object detection provides frame-level recognition of traffic participants, understanding traffic dynamics requires modeling temporal continuity. Multi-object tracking (MOT) methods address this need by associating detections across consecutive frames, enabling trajectory reconstruction and behavior analysis. Early tracking approaches such as SORT introduced efficient mechanisms for associating detections in real time, while subsequent methods such as DeepSORT incorporated deep feature embeddings to improve robustness in complex traffic environments [4,5]. These methods allow traffic monitoring systems to estimate vehicle trajectories, analyze flow patterns, and capture dynamic behaviors in traffic scenes. Tracking-based approaches are often combined with detection models to provide end-to-end monitoring capabilities. For example, CNN-based detection integrated with filtering techniques such as Kalman filtering has been used to monitor large-scale traffic flows, particularly in aerial imagery scenarios [19]. Such approaches enable continuous monitoring of moving vehicles with reduced human intervention.

However, according to the literature survey, many existing systems emphasize detection and visualization rather than comprehensive trajectory-based analysis. The extent to which tracking is used for higher-level behavioral modeling or decision support is not consistently specified. This suggests that, although MOT provides essential temporal information, its integration into fully operational traffic analysis systems remains limited.

## 2.3. Traffic Forecasting Using Sensor Data

Traffic forecasting is another important research direction aimed at predicting future traffic conditions and enabling proactive traffic management. Early approaches relied on statistical time-series models such as autoregressive integrated moving average (ARIMA) for analyzing traffic flow patterns [6]. With the increasing availability of large-scale traffic data from IoT sensors and monitoring systems, machine learning and deep learning models have been widely applied. Hybrid machine learning approaches have also been proposed to leverage the complementary strengths of different models. By combining linear, nonlinear, and temporal modeling capabilities, such hybrid approaches achieve more stable and reliable traffic flow prediction compared to single-model methods. Recurrent neural networks, particularly long short-term memory (LSTM) models, have demonstrated strong performance in modeling temporal dependencies in traffic data and are commonly used for short-term traffic prediction [7]. Nevertheless, most forecasting models operate independently of real-time perception systems, limiting their ability to incorporate dynamic environmental information derived from vision-based inputs.

More recent studies incorporate spatial dependencies through graph-based deep learning models, such as diffusion convolutional recurrent neural networks (DCRNN) and spatio-temporal graph convolutional networks (STGCN), which model traffic networks as graphs and capture both spatial and temporal correlations [8,9]. These models have shown improved prediction performance in large-scale urban environments. Traffic forecasting systems typically rely on heterogeneous data sources, including IoT sensors, GPS trajectories, and surveillance systems, and are evaluated using metrics such as mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). However, the literature suggests that forecasting models are often developed independently from traffic perception systems. The integration of real-time detection, tracking, and prediction within a unified framework is not consistently specified in existing studies. Table 1 shows a comprehensive overview of different research papers including the above showcased overview, models developed, measurements and evaluation metrics used.

**Table 1.** Comparison of Detection, Tracking, Prediction and System-Level Traffic Studies.

#	Objective	Models Used	Data Source	Metrics
1	High-accuracy object detection	RPN + CNN (two-stage)	PASCAL VOC, COCO	mAP, IoU, FPS
2	Real-time object detection	YOLO (one-stage CNN)	PASCAL VOC, COCO	mAP, FPS
3	Speed-accuracy optimization	Improved YOLO	VOC, COCO	mAP, FPS
4	Real-time tracking	Kalman + Hungarian	Video	MOTA, MOTP
5	Robust tracking	SORT + CNN embedding	Video	MOTA, IDF1
6	Time-series modeling	ARIMA	Loop detectors	MSE, RMSE
7	Short-term prediction	LSTM (RNN)	Sensor data	RMSE, MAE
8	Spatiotemporal prediction	GNN + RNN	PeMS	RMSE, MAE
9	Efficient forecasting	GCN + CNN	Sensor network	MAE, RMSE
10	Traffic safety analysis	CNN, RNN, GNN	Multi-source	Accuracy, F1
11	UAV traffic monitoring	CNN + filter tracking	UAV images	Accuracy
12	Big data traffic prediction	DL + GPU	Big data	RMSE, MAE
13	UAV-based prediction and control	ML + UAV vision	UAV + traffic	Accuracy
14	Vehicle counting	Classical CV methods	Video	Accuracy
15	Traffic flow estimation	YOLO + SORT	Video	Accuracy, MOTA
16	Vehicle identification	YOLOv4	Video	mAP
17	Computer vision survey	Deep learning methods	Multi-source	Accuracy
18	Vision-based traffic control	CV models	Video	Delay, throughput
19	Intelligent traffic system	ML + CV	Video + traffic	Accuracy
20	Adaptive traffic signal control	YOLO + SORT + rule-based	CCTV video	Delay reduction, travel time
21	Smart surveillance systems	DL + IoT	Multi-source	Accuracy
22	IoT traffic monitoring	IoT + fog/cloud	Sensor data	Latency, throughput
23	Edge-cloud traffic framework	Edge + cloud	IoT data	Latency
24	Early IoT traffic system	Cloud + IoT	Sensors	Efficiency

#### 2.4. Review of Current Technology and Solution for Traffic Management System

Despite recent advances, existing system-level traffic management solutions are typically designed to address isolated functionalities—such as data acquisition, monitoring, or prioritization—rather than providing a unified framework that integrates perception, tracking, and prediction. This fragmented design limits their ability to support holistic and intelligent traffic management in complex urban environments.

### 2.4.1. Traffic Management Features

To further substantiate this observation, Table 2 presents a comparative analysis of representative traffic monitoring systems with respect to data modalities, monitoring scope, and analytical capabilities. As illustrated, most existing systems provide only limited functionality, primarily focusing on data collection or basic traffic analysis. Advanced capabilities—such as accident detection, real-time alert generation, and end-to-end traffic analytics—are not consistently supported. Furthermore, the integration of heterogeneous data sources and the development of unified analytical pipelines remain largely underexplored in practical deployments.

**Table 2.** Comparison of Parameters and Dataset Examined for Traffic Management.

Ref.	Purpose	Data Type			Drone	Location			Parameters					
		Video	Sensor	Image		City	Intersection	Traffic Report	Accident	Animals	Count Vehicles	Speed	Alert	Analysis
[10]	To analyze commercial traffic monitoring service implementation with safety focus	Y	N	N	Y	Y	N	Y	Y	N	Y	Y	Y	Y
[11]	Smart traffic monitoring using object detection and drone imagery	Y	N	N	Y	Y	N	N	N	N	N	N	N	N
[12]	Big data + deep learning for large-scale real-time traffic prediction	N	Y	N	N	-	-	Y	N	N	Y	Y	N	N
[13]	Smart traffic lights with predictive modeling and drone images	N	N	Y	Y	Y	N	Y	N	N	N	Y	N	N
[14]	Vehicle counting and movement analysis using image-based systems	N	N	Y	N	N	Y	Y	Y	N	Y	Y	N	N
[21]	Review of video surveillance in smart cities with cloud/edge integration	Y	N	N	Y	Y	N	Y	Y	N	Y	N	Y	N
[22]	Fog + cloud based smart traffic monitoring system	N	Y	N	N	Y	N	N	Y	N	Y	N	N	Y
[23]	Traffic prediction and load balancing at intersections	N	Y	N	N	N	Y	Y	N	N	Y	Y	N	Y
[24]	Intelligent traffic management using VANETs and smart cities	N	Y	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y
<b>Our</b>	<b>Our Results</b>	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y

### 2.4.2. Commercial Smart Traffic Management Systems: A Comparative Analysis

To assess current industry capabilities, a range of commercial smart traffic management systems—including solutions developed by GoodVision, Isarsoft, Econolite, and KaaIoT Technologies—were systematically analyzed. These platforms generally integrate computer vision, IoT sensors, and cloud-based analytics to support real-time traffic monitoring, incident detection, and signal control. However, their capabilities vary significantly, and support for advanced functionalities—such as spatio-temporal traffic forecasting and automated emergency response—remains inconsistent, as summarized in Table 3.

Despite their practical effectiveness, several critical limitations persist. First, most systems emphasize real-time monitoring and descriptive analytics, with limited support for predictive modeling based on spatio-temporal dynamics. Consequently, their ability to anticipate congestion patterns or enable proactive traffic optimization is constrained. Second, existing solutions often rely on single-source data, lacking comprehensive multi-modal data fusion across heterogeneous inputs such as UAV imagery, connected vehicle data, and IoT sensor networks. This restricts their capacity

to achieve comprehensive situational awareness in complex traffic environments. In addition, the application of advanced AI techniques remains limited, typically confined to object detection or basic statistical analysis. More sophisticated capabilities—such as trajectory prediction, anomaly detection, and adaptive decision-making—are not fully realized in current systems. These limitations highlight a clear gap between existing industry practices and the potential of next-generation intelligent traffic management systems.

To bridge this gap, this paper proposes an AI-driven smart traffic management framework that integrates advanced object detection, spatio-temporal forecasting, and multi-source data fusion, enabling more proactive, predictive, and intelligent traffic control.

**Table 3.** Comparison of Traffic Management Systems Across Key Functional Dimensions.

Company	Data Source	Emergency Detection	Traffic Forecasting	Cloud	Count of Vehicles	Type of Vehicles	Traffic Control	Alerts	Traffic Analysis	Accident Reports	Real-Time Processing
Houston	Camera/Sensors	N	N	N	N	N	Y	N	Y	N	N
Isarsoft	Camera	N	N	N	Y	Y	Y	Y	Y	Y	Y
GoodVision	Camera/Drone	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
NEC	Camera/Sensors	N	N	N	Y	Y	Y	Y	Y	Y	Y
Econolite	Camera/Loop/Sensors/Probe	N	Y	Y	Y	N	Y	Y	Y	Y	Y
Lanner Electronics	Camera	N	N	N	Y	Y	N	N	Y	N	Y
ACE	Camera/Sensors	N	N	N	Y	N	Y	Y	Y	Y	Y
EDAG	IoT/Connected Vehicles	N	N	N	N	N	Y	Y	Y	N	Y
Data From Sky	Camera/Drone	Y	N	N	Y	Y	Y	Y	Y	Y	Y
Dhyan	Camera	N	N	N	Y	Y	Y	Y	Y	Y	Y
Sense Traffic Plus	Camera	N	N	N	Y	Y	N	Y	Y	Y	Y
km4city	IoT/Sensors	N	N	Y	Y	N	N	Y	Y	Y	Y
Esri	IoT/GPS/Probe/APIs	N	N	Y	Y	N	N	Y	Y	Y	Y
Ours	CCTV/Drone/IoT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

### 3. Smart City Traffic Data Engineering

#### 3.1. Data Collection

Effective smart city traffic analysis for computer vision applications relies on the integration of diverse and high-quality datasets that support both model training and evaluation across multiple tasks, including traffic flow analysis, accident detection, situational assessment, and traffic forecasting. In this study, a combination of real-world traffic data, high-resolution imagery, and video streams from CCTV systems and unmanned aerial platforms is employed to address challenges such as traffic optimization, incident response, and pattern prediction. A structured data collection strategy was adopted to ensure that the datasets are aligned with specific analytical tasks, thereby facilitating robust model development and more reliable urban traffic management outcomes.

Traffic flow analysis is primarily based on data obtained from Caltrans, which includes information on traffic patterns, roadway conditions, and incident reports collected through roadway sensors, traffic cameras, and vehicle counting systems. In addition, publicly available datasets are incorporated to support visual recognition tasks. These include the COCO dataset for object detection

and segmentation, the COWC dataset for annotated vehicle imagery, and the PeMSBay dataset, which provides real-time IoT sensor data such as vehicle counts, speeds, and incident records collected at five-minute intervals from 325 sensors across the Bay Area. To capture a wide range of traffic scenarios, high-definition images and videos were sourced from platforms such as Google, YouTube, TikTok, and the Pexels API. The VisDrone dataset further contributes video sequences suitable for trajectory analysis, encompassing diverse environments and dynamic objects, including vehicles, cyclists, and pedestrians. Data related to emergency incidents were compiled from multiple sources, including drone footage, UAV videos, and publicly available online media. These data support the analysis of events such as collisions, rollovers, and vehicle fires.

Additional datasets focusing on emergency vehicles—such as police cars, ambulances, and fire trucks—were collected from online platforms, news reports, and social media to enhance detection capabilities in critical scenarios. The vehicle detection component of the dataset, drawing on drone imagery and high-resolution images from VisDrone, emphasizes the identification of cars, trucks, and bicycles. Notable challenges encountered during data preparation include variations in image resolution and the presence of motion blur, both of which affect detection performance and necessitate careful preprocessing.



**Figure 1.** (a–c) Emergency Vehicle data. (d–f) Emergency Incident Data.

### 3.2. Data Preprocessing

Data preprocessing plays a critical role in establishing a reliable foundation for training robust computer vision models. The process aims to standardize heterogeneous data sources while improving data quality and representation, thereby enhancing model accuracy, generalization, and

performance under real-world conditions. In this study, preprocessing encompasses data cleaning, frame extraction, augmentation, annotation, and transformation of both visual and sensor-based data.

### 3.2.2. Data Cleaning

The initial stage involved data cleaning, where the raw dataset was examined for inconsistencies and low-quality samples. A substantial portion of the collected imagery exhibited low resolution, often resembling high-altitude drone footage, and contained irrelevant or sparsely populated scenes. To mitigate these issues, frames with excessive noise, limited visual information, or large empty regions were manually excluded, resulting in a more representative and task-relevant dataset.

### 3.2.3. Sequence Frame Extraction and Preprocessing

To support temporal analysis and object tracking, sequence frame extraction was applied to video data. Using Roboflow, frames were sampled at a rate of 153 frames per minute. These frames subsequently underwent preprocessing operations, including resizing, grayscale conversion, and contrast enhancement, to improve feature visibility. Additional augmentation strategies, such as noise injection, rotational transformations, and brightness adjustments, were employed to increase variability and improve model robustness.



**Figure 2.** Sequence frame extraction samples.

### 3.2.4. Data Augmentation

Further data augmentation techniques were introduced to enhance generalization and mitigate both underfitting and overfitting. These included cropping, rotation, and cutout operations, with all images standardized to a resolution of 640×640 pixels to ensure computational efficiency during

training. Roboflow was used to implement these augmentations systematically, enabling consistent transformations such as 90-degree rotations and brightness variation across the dataset.

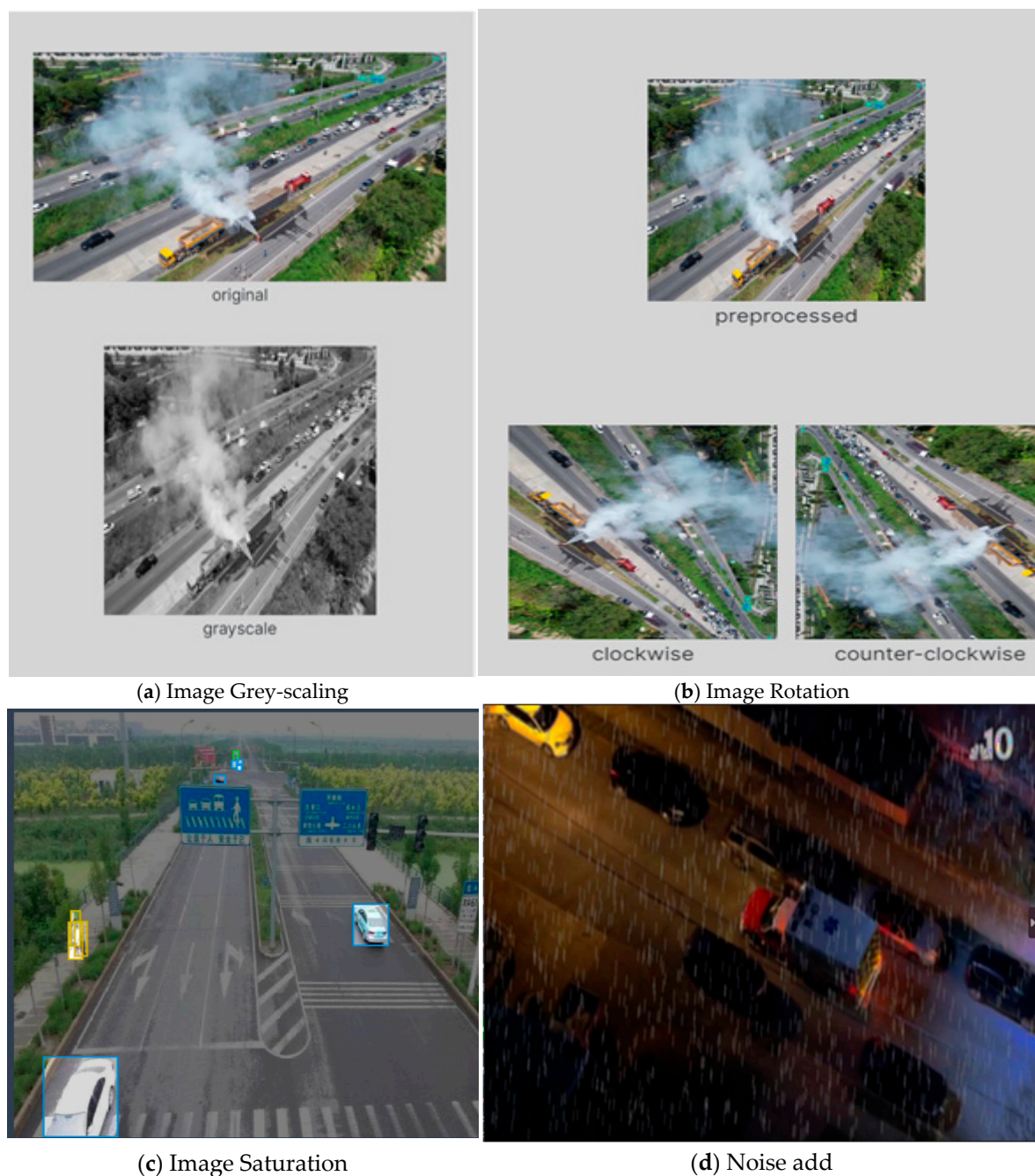


Figure 3. Data Augmentation.

### 3.2.5. Image Annotation

Image annotation was conducted manually using Roboflow's bounding box tools to label objects of interest, including cars, trucks, pedestrians, and animals. In addition to object categories, traffic-related incidents were explicitly annotated and classified into single-vehicle accidents, multi-vehicle collisions, vehicle fire events, and animal-related road incidents. These annotations provide structured supervision for model training and facilitate more precise incident detection and response mechanisms.



**Figure 4.** Image Annotation.

### 3.2.6. IoT Sensor Data Processing

For traffic sensor data, preprocessing was carried out using Python-based tools, including Pandas and Tables. The PeMS dataset, originally stored in HDFS format, was converted into a structured DataFrame to enable efficient data manipulation and analysis. This process involved configuring the necessary data storage environment, creating HDFStore objects, and organizing the data into queryable formats suitable for downstream modeling tasks.

### 3.2.7. Drone Image Augmentation

Drone imagery was further augmented to improve robustness under diverse environmental conditions. Techniques such as Gaussian blurring, noise addition, and weather simulation—including rain, fog, and snow—were applied to better reflect real-world variability and enhance model performance in adverse scenarios.





(c)

**Figure 5.** Image Augmentation: (a) Added noise, (b) Added rain condition samples, (c) Added light condition sample.

### 3.2.8. IoT Sensor Data Transformation

In parallel, IoT sensor data underwent normalization using MinMaxScaler to facilitate stable neural network training. The dataset was partitioned into 80% training and 20% validation subsets to support model evaluation. In addition to bounding box annotations, polygon-based annotations were introduced for selected object classes, particularly emergency vehicles. This approach enables the capture of finer-grained visual details, such as lighting patterns and vehicle markings, thereby improving detection precision in complex urban traffic environments.

### 3.3. Data Preparation

Following data preprocessing and transformation, the dataset was partitioned according to a standard 80/20 split, ensuring that both training and validation subsets maintain representative diversity. To improve annotation quality and support precise object detection, polygon-based labeling was adopted within the deep learning framework. Unlike conventional bounding boxes, which often include irrelevant background regions, polygon annotations allow for a more accurate delineation of objects with irregular geometries. This level of precision is particularly important for emergency vehicles—such as fire trucks, ambulances, and police cars—whose structural features and visual signatures require detailed representation.

As demonstrated in Figures.a, and b, polygon annotations capture critical attributes, including emergency lighting patterns, vehicle contours, and distinctive markings. Such fine-grained labeling enhances the model's ability to differentiate between vehicle categories with greater accuracy. By learning from a training set that faithfully represents these characteristics, and validating performance on a held-out subset, the model achieves improved generalization in complex traffic scenarios.

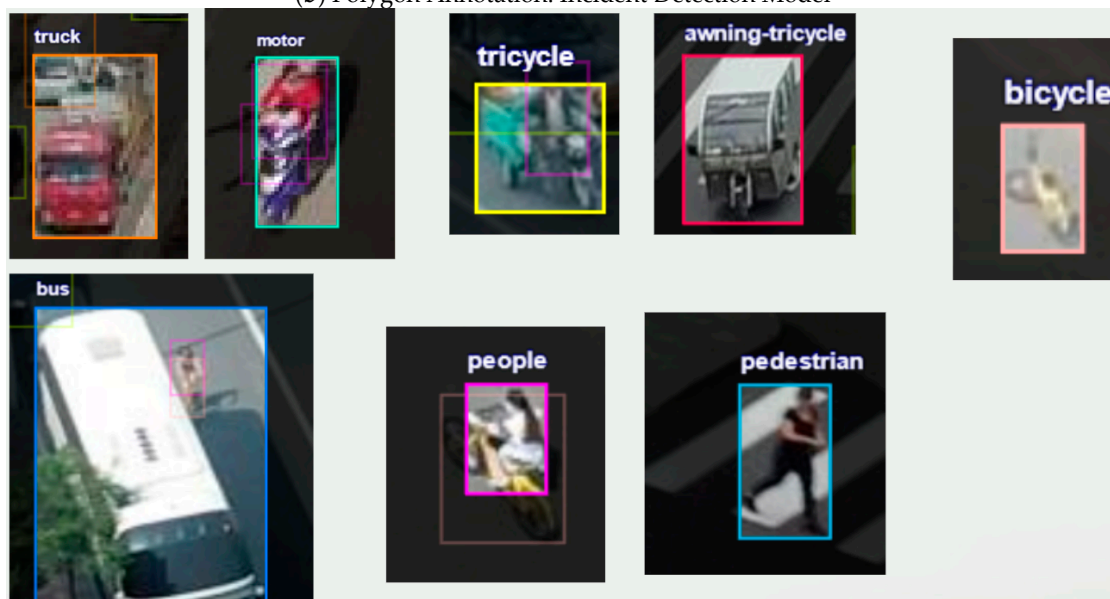
Overall, the use of polygon annotations contributes to more effective feature learning and strengthens the model's capacity to accurately recognize and classify emergency vehicles in real-world urban environments.



(a) Polygon Annotation: Emergency Car.



(b) Polygon Annotation: Incident Detection Model



(c) Polygon Annotation: Vehicle Detection Model

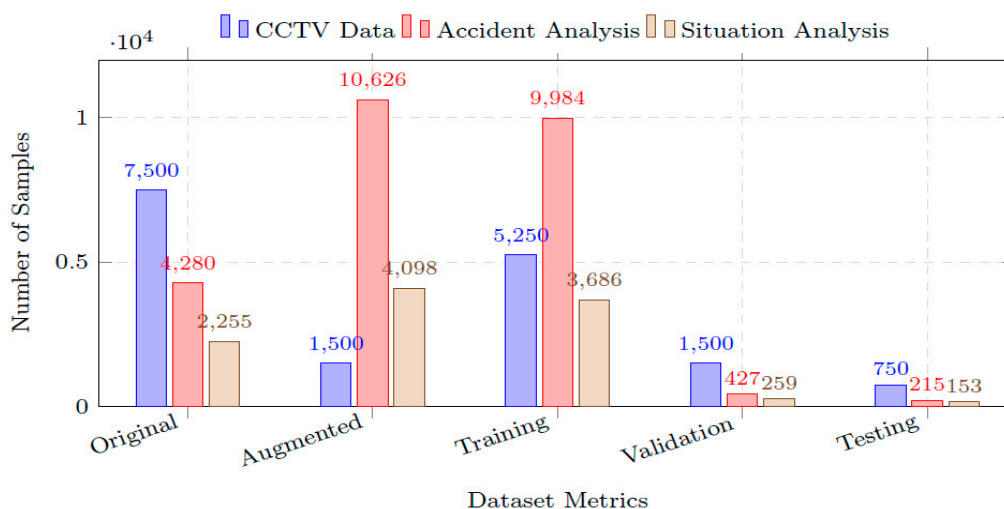
**Figure 6.** Polygon Annotation.

### 3.4. Data Statistics

- IoT Data: The IoT sensor dataset, PeMS-Bay, was collected from the California Performance Measurement System (PeMS) in real-time using over 325 sensor stations across California's highway system. The data is aggregated into 5-minute intervals, covering weekdays from January to May 2024. Detailed statistics of the IoT dataset are presented in Figure 7. This structured partitioning ensures a balanced dataset for model training and evaluation.
- CCTV data: The study utilizes a subset of the Microsoft Common Objects in Context (MS COCO) dataset, focusing on vehicle-related classes such as cars, trucks, motorcycles, and buses. The

subset consists of 7,500 original images, which were further augmented with 1,500 additional images, resulting in a total of 9,000 images. These images were partitioned into 5,250 for training, 1,500 for validation, and 750 for testing, as shown in table I. This structured partitioning ensures a balanced dataset for model training and evaluation.

- Drone data: The drone data sourced from YouTube and TikTok was categorized into three groups: vehicle recognition data, safety incident data, and road hazard data. These datasets were annotated and subsequently partitioned into training, testing, and validation set. Table I provides a breakdown of the drone, CCTV, and IOT datasets.



**Figure 7.** Dataset Statistics.

The two categories of drone-derived data were further annotated to improve predictive performance. For vehicle detection, the dataset was labeled with object classes including cars, trucks, pedestrians, motorcycles, and other large vehicles. In the case of traffic safety incidents, annotations were assigned to specific emergency vehicle types, such as police cars, fire trucks, and ambulances. Additionally, road hazard data were categorized into events including collisions, vehicle fires, and truck rollovers, each accompanied by corresponding labels. A detailed comparison of the original and annotated datasets for each category is provided in Table 4.

**Table 4.** Class-wise Comparison of Situation and Accident Analysis.

Class	Situation	Accident
Car	17726	-
Person	12868	-
Bike	7697	-
Bus	1028	-
Truck	15	-
Police Car	-	1868
Car Collision	-	1692
Car on Fire	-	1365
Fire Truck	-	1360
Truck Rollover	-	668
Ambulance	-	544

## 4. Model Development

There are three important components of this research drone, CCTV and IoT, the models developed for analysis of real-world scenarios. Each module is associated with a task and provides the results that can be used by officials at different levels. Figure 2 shows the combined model architecture of all the modules.

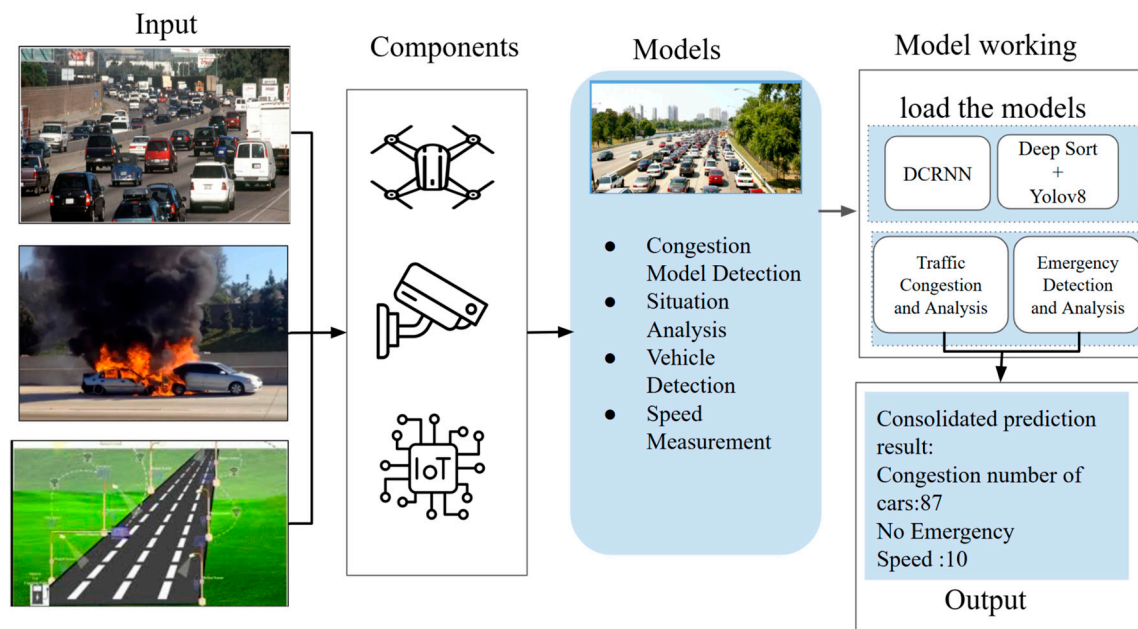


Figure 8. Combined model development architecture.

### 4.1. CCTV-Based Traffic Flow Analysis and Accident Detection Using YOLOv8

The YoloV8 model was used for traffic flow analysis and accident detection on CCTV data. Figure 9 shows the CCTV model architecture presenting the flow to the model. For the CCTV accident detection model YOLOv8 was trained on a dataset curated specifically for accident detection, undergoing 300 epochs of rigorous optimization. The model is ensured to adapt to various scenarios, including changes in weather, lighting, and camera positions. The trained model is capable of identifying and localizing accidents in real-time, offering a powerful tool for traffic monitoring systems. When an accident is detected, the model focuses around the area of interest, visually highlighting the exact location of the accident in the footage. This mechanism is based on YOLOv8's ability to perform object detection by predicting the spatial coordinates of the accident region along with the confidence score. Further analysis is made to identify the severity, type of accident, or the number of objects involved.

For traffic flow analysis Model YoloV8 was trained and refined its parameters to improve its understanding of traffic scenes captured by CCTV cameras. The primary objective of the model was to perform traffic flow analysis based on the speed of vehicles captured in the CCTV footage. By accurately detecting and tracking vehicles over time, the model provided valuable insights into traffic patterns, congestion levels, and overall flow dynamic

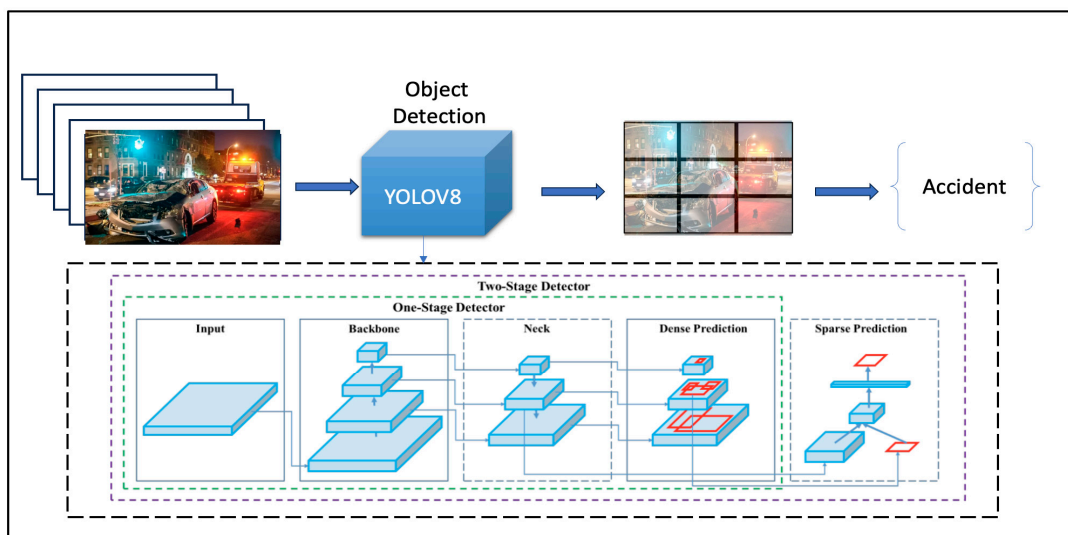


Figure 9. CCTV Model Architecture.

#### 4.2. Drone-Based Vehicle Detection and Tracking Using YOLOv8 and DeepSORT

The drone model is similar to the CCTV model with additional analysis. The drone model is designed to support an innovative traffic management system using an AI-powered drone system that can be used for traffic monitoring, analyzing ground situations, and presenting a detailed analysis. Along with customized YOLOv8 similar to CCTV additional DeepSort algorithms are used to develop the model. Figure 10 presents the model architecture.

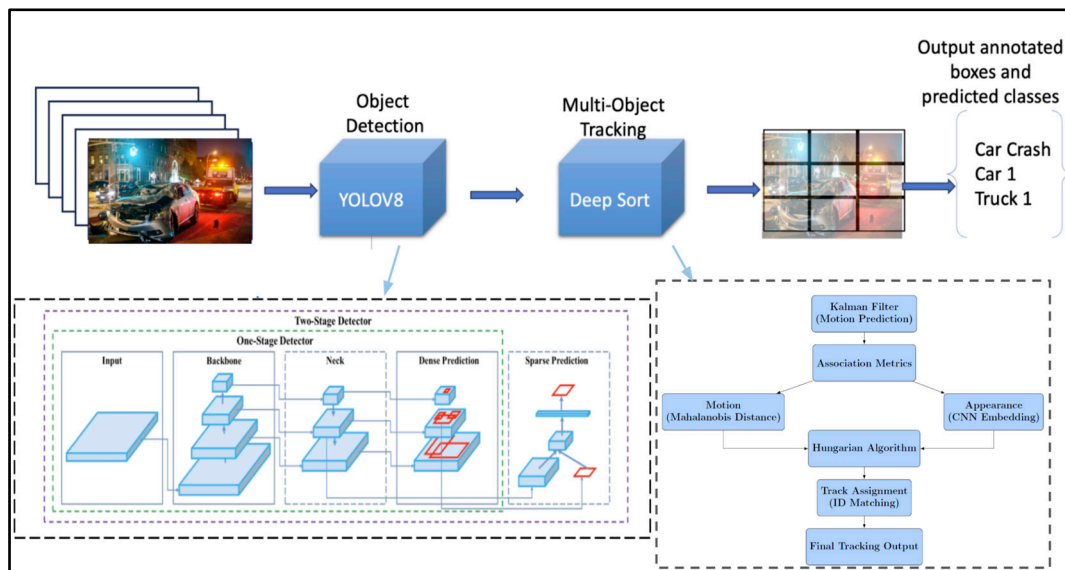


Figure 10. Bird-view of Drone Model Architecture.

The initial steps of the model began with data collection and data preparation from various platforms mentioned in the data collection section above. The 5K annotated images were augmented, and multi-label classification for various scenarios, including vehicle collisions, rollovers, and fires, was considered. To ensure the robustness of the model sequence frame extraction, 153 frames per minute were extracted to get precise details. To improve the prediction efficiency, techniques such as Gaussian blur, seasonal (fog, rain, low light), and different daytime simulations were implemented. To achieve the accurate categorization of accident type, emergency vehicles, pedestrians, and other

irregularly shaped objects were ensured via polygon-based annotations. The dataset was divided into a ratio of 70:20:10 to train, test, and validate to maintain balance for model learning.

After a thorough study and test, we selected YoloV8 for object detection and fine-tuned it for a prepared dataset with a confidence threshold of 0.5, and non-max suppression (NMS) was optimized to reduce false positives to handle crowded locations. Further, to improve the existing results and add unique IDs to objects in different frames, the DeepSort algorithm was integrated for multiobject tracking, using Kalman filters and cosine distance. The deployment pipeline was developed using OpenCV, resizing frames to  $640 \times 640$  pixels and forwarding the generated output to the model. In the output phase of the model, the count of the objects and class is provided, and the focused area is highlighted even under adverse conditions like low light and complex intersections. The developed models are deployed on the AWS cloud and integrated into the system. The system code uses parallel processing, leading to reducing operational costs. The drone model gives a high precision of 87% in high visibility areas and around 70% precision in less visibility areas. Overall this model proves to be an efficient and effective way for situation analysis and accident analysis.

#### 4.3. Traffic Flow Forecasting IOT Sensor -Multi-Cluster Diffusion Convolutional Recurrent Neural Network (DCRNN)

Figure 11 illustrates the architecture of the Multi-Cluster DCRNN, showcasing its capability to handle large-scale traffic networks through parallel execution of multiple DCRNN models. In this setup, the traffic network is divided into 8 partitions (e.g., Partition 1, Partition 2, ..., Partition 8), with each partition representing a subset of sensors and their corresponding spatial dependencies. Each partition is assigned to a distinct DCRNN model that runs independently on a dedicated GPU. This parallel execution across GPUs allows for simultaneous processing of all partitions, dramatically reducing training and inference times. The right side of the diagram presents the encoder-decoder framework of the DCRNN model, which processes the input graph sequence by capturing both spatial and temporal dependencies through diffusion convolution layers. The encoder learns from the historical traffic data, while the decoder generates future traffic predictions for the respective partition. Copy states are transferred between time steps to maintain temporal coherence, ensuring accurate forecasting across future time intervals. This parallel, partition-based architecture enables the Multi-Cluster DCRNN to scale effectively for large traffic networks, providing faster and more efficient predictions without sacrificing accuracy.

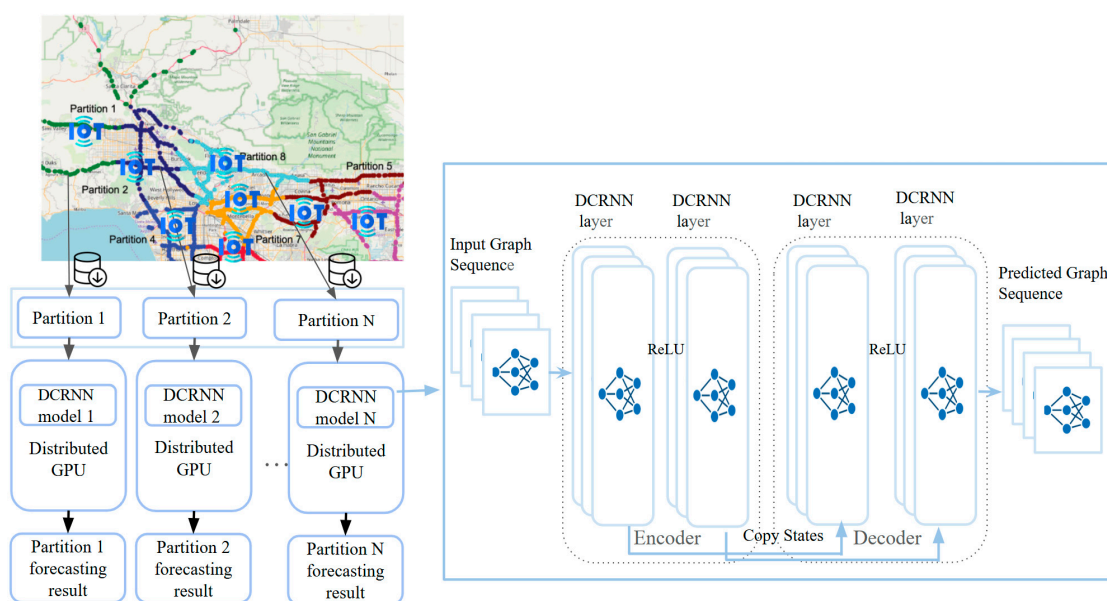


Figure 11. Architecture of Simultaneous execution of DCRNN.

## 5. Case Study & Results

In the context of Smart City Traffic Management, effective analysis of model execution and evaluation results is crucial to ensure accurate and reliable decision-making. For CCTV data, the model's ability to detect and classify vehicles or objects and detect accidents is assessed using key metrics such as Precision, Recall, and Mean Average Precision (mAP). These metrics are essential for evaluating object detection accuracy and identifying false positives or missed detections. Drone based incident and situation analysis further leverage these metrics to monitor events and manage traffic flows effectively.

### 5.1. CCTV Model Traffic Flow Analysis - YoloV8

The model exhibits consistent performance across both training and validation datasets. On the training set, it achieves a precision of 80.10%, indicating a reliable ability to correctly identify traffic-related objects. The recall reaches 85.70%, suggesting that a substantial proportion of relevant objects present in the footage are successfully detected. In addition, the mean Average Precision (mAP) attained 85.9%, reflecting strong overall capability in both object detection and localization. These results indicate that the model has effectively learned representative feature patterns from the training data, allowing it to maintain a balance between detection accuracy and coverage. The comparatively higher recall relative to precision suggests a tendency to capture a broader set of relevant objects, which may be advantageous in traffic monitoring scenarios where missing critical instances—such as vehicles involved in congestion or incidents—could reduce situational awareness. On the validation set, the model achieves a precision of 79.09% and a recall of 83.10%, with an mAP of 83.09%. These results remain close to those observed during training, indicating stable performance when applied to unseen data. The limited discrepancy between training and validation metrics suggests that the model generalizes effectively, with no clear indication of overfitting. Moreover, the consistency of performance across datasets implies that the learned representations remain robust under varying input conditions, including differences in lighting, camera perspectives, and traffic density commonly encountered in CCTV-based monitoring environments. The sustained mAP values above 80% further demonstrate the model's ability to maintain a balanced trade-off between detection accuracy and localization precision across multiple object categories.

Taken together, these findings support the effectiveness of the YOLOv8-based approach for traffic flow analysis using CCTV data. The observed performance characteristics indicate that the model is well-suited for real-time traffic monitoring applications, where both accuracy and robustness are essential for downstream tasks such as congestion assessment and incident detection.

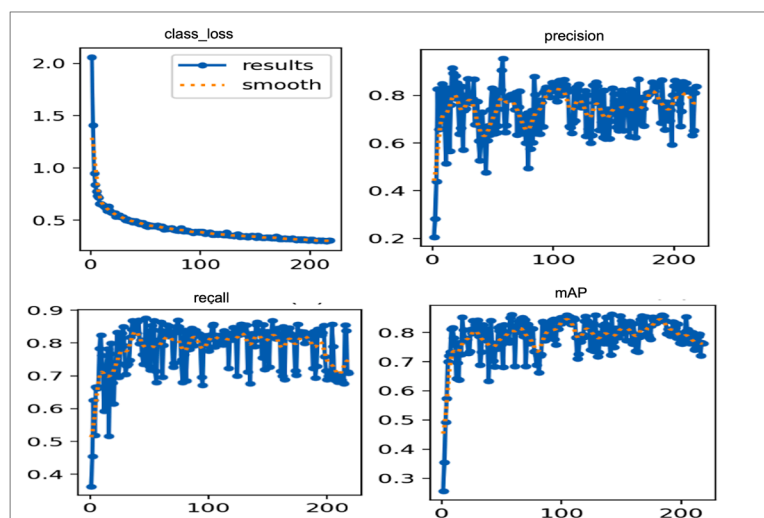


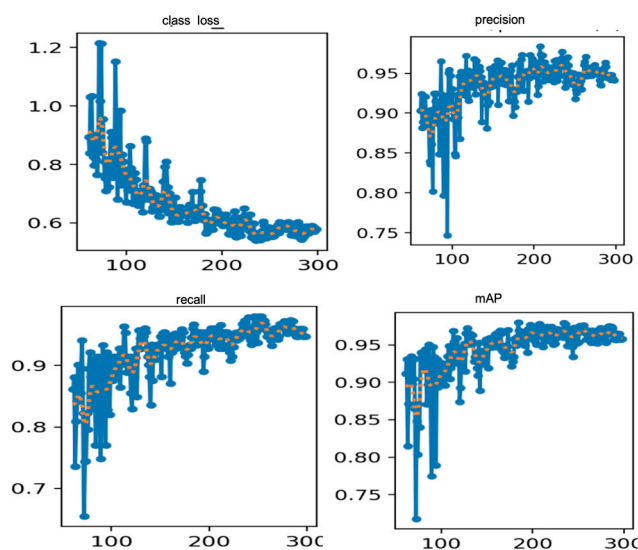
Figure 13. CCTV Traffic Flow Evaluation and Results.

## 5.2. CCTV Accident Detection Model – YOLOv8

During training, the YOLOv8-based accident detection model achieved a precision of 93.60%, a recall of 92.70%, and a mean Average Precision (mAP) of 93.91%. These results indicate strong detection capability, with the model accurately identifying accident-related events while maintaining a relatively low rate of false positives. The observed performance suggests that the model is able to learn discriminative visual features associated with accident scenarios, including both spatial configurations and contextual cues present in CCTV imagery. The high precision reflects reliable identification of accident regions, whereas the elevated recall indicates that most relevant events are successfully captured. Evaluation on the validation dataset yields a precision of 89.09%, a recall of 88.10%, and an mAP of 88.09%. Although these values are modestly lower than those obtained during training, the discrepancy remains limited. This close alignment between training and validation results indicates that the model generalizes effectively to unseen data, with no clear evidence of overfitting. Furthermore, the stability of these metrics suggests that the learned feature representations remain robust under varying conditions, including differences in illumination, camera viewpoints, and environmental factors commonly encountered in real-world surveillance settings.

From an application perspective, the high precision implies that the model can detect accident events with a low incidence of false alarms, which is essential for maintaining the reliability of traffic monitoring systems. At the same time, the strong recall ensures that a large proportion of actual accident events are identified, thereby supporting comprehensive situational awareness. Achieving a balance between these two metrics is particularly important in safety-critical contexts such as traffic incident monitoring, where both detection reliability and coverage are necessary. The mAP metric, which integrates precision and recall across multiple thresholds, further substantiates the overall effectiveness of the model. The consistently high mAP values across both datasets indicate that the model not only detects accident instances accurately but also localizes them with sufficient precision. Such localization capability is important for downstream tasks, including assessing incident severity and facilitating timely response. Taken together, these results demonstrate that the YOLOv8-based approach provides reliable performance for CCTV-based accident detection and is suitable for real-time deployment in traffic monitoring systems. In addition, the model's ability to generate bounding boxes around detected accident regions provides explicit spatial information, which can support subsequent analysis, such as event characterization and automated response mechanisms.

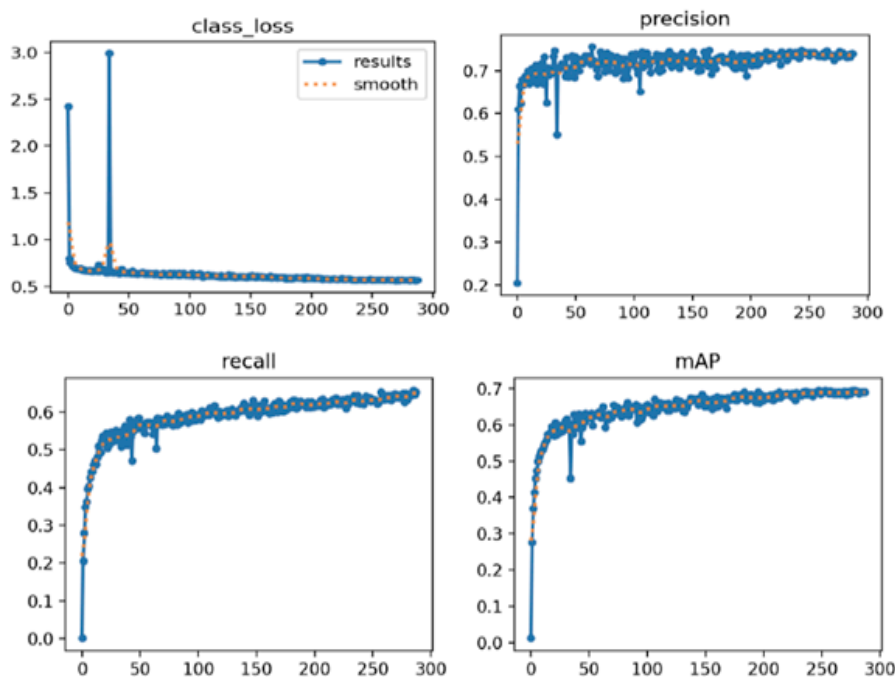
Figure 14 presents the evolution of class loss, precision, recall, and mAP during training, illustrating stable convergence and consistent improvement in performance over time.



**Figure 14.** CCTV Accident Detection Evaluation and Results.

### 5.3. Drone Situation Analysis - YoloV8+DCRNN

Using YOLOv8 architecture, the researcher set out on a thorough trip to improve the multi-label classification model's accuracy and robustness. After training the model for 100–150 In training, it was discovered that the model was underfitting, requiring additional improvement. In order to improve its functionality, more pictures were added. The methodical approach of image augmentation using Roboflow resulted in a noticeable increase in accuracy and a decrease in loss measures. The model was able to train through 300 epochs. With the help of augmented data and transformational methodologies, YOLOv8's inherent strengths in object identification were combined to create a strong classification framework that could distinguish between six different classes: people, automobiles, trucks, bicycles, vans, and buses. The experiment demonstrated the importance of ongoing improvement and the critical role of cutting-edge architectures like YOLOv8 in pushing the boundaries of multilabel classification problems as it mapped its evolution from underfitting to increased accuracy this can be seen in the accuracy Figure 15 achieves a high accuracy rate of roughly 74% of the time, while a low accuracy rate of roughly 64% occurs when it misses anything. It performs admirably overall, earning 69.5%. When we validate it with fresh data, it remains rather stable, correctly identifying patterns over 70% of the time and missing only a small percentage, or 63% of the time. It receives an overall score of roughly 67%, indicating that even with new information, it can recognize objects well. These findings aid in our comprehension of the model's effectiveness and areas for improvement.



**Figure 15.** Drone Situation Analysis evaluation and results.

### 5.4. Drone Accident Detection Model - YoloV8+DCRNN

Accurate accident analysis remains challenging due to the complexity of traffic scenes, which often involve occlusions, heterogeneous visual patterns, and dynamic interactions among multiple objects. Initial experiments relying solely on the YOLOv8 architecture yielded suboptimal performance, particularly in meeting the precision and recall requirements for reliable accident detection. To address these limitations, the DeepSORT tracking algorithm was integrated into the detection pipeline. By associating objects across consecutive frames, this approach introduces temporal consistency and improves the interpretation of motion patterns and event progression. In parallel, data augmentation strategies—specifically rotational transformations—were applied over

150 training epochs to enhance robustness and generalization under diverse visual conditions. The enhanced framework demonstrates clear performance improvements. On the training set, the model achieved a precision of 85.70%, a recall of 80.50%, and a mean Average Precision (mAP) of 88.20%. On the validation set, it maintained a precision of 82.20% and an mAP of 80.20%, while recall decreased to 71.10%. Despite this reduction, the relatively small gap between training and validation results suggests stable generalization performance. Overall, the combination of YOLOv8 for object detection and Deep SORT for tracking provides a complementary and effective framework for accident analysis, enabling reliable detection and localization in complex traffic environments.

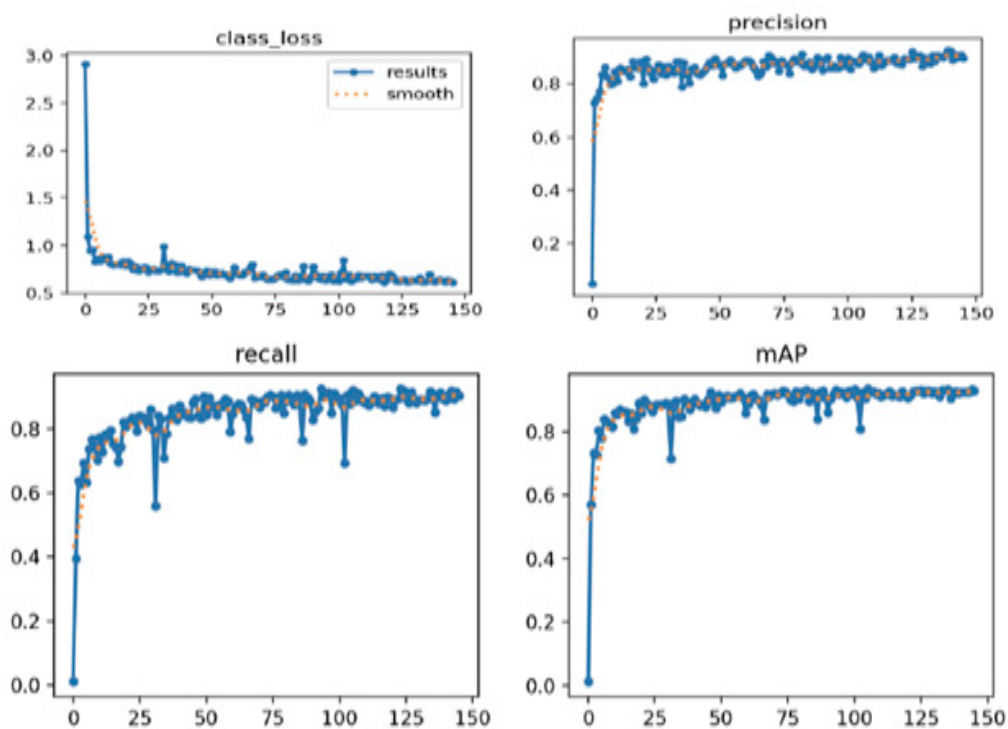


Figure 16. Accident analysis evaluation and results.

Combined evaluation results of CCTV traffic flow analysis, CCTV accident analysis, drone situation analysis and drone accident analysis are shown in Table 5.

Table 5. Performance Comparison of the Proposed Models on Different Tasks.

Model	Phase	Precision	Recall	mAP
CCTV Traffic Flow YOLOv8	Training	80.10%	85.70%	85.90%
	Validation	79.00%	83.10%	88.09%
CCTV Accident Detection YOLOv8	Training	93.60%	92.70%	93.91%
	Validation	89.09%	88.10%	88.09%
Drone Situation Model YOLOv8 Plain	Training	60.80%	57.01%	59.11%
	Validation	58.23%	55.40%	54.01%
Drone Accident Model YOLOv8 +DeepSORT	Training	74.20%	63.80%	69.50%
	Validation	70.20%	63.10%	67.20%
Drone Accident Model YOLOv8 Plain	Training	70.20%	71.01%	70.11%
	Validation	69.23%	68.40%	69.01%
Drone Accident Model YOLOv8 +DeepSORT	Training	85.70%	80.50%	88.20%
	Validation	82.20%	71.10%	80.20%

### 5.5. IoT Traffic Forecasting - DCRNN Model

Performance of the DCRNN model evaluated on the PEMS-BAY dataset across various forecasting horizons (5, 15, 30, and 60 minutes) using three different metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). For the 5-minute horizon, the MAE is 0.85, MAPE is 1.63 and RMSE is 1.54. As the forecasting horizon increases to 15 minutes, the MAE, MAPE, and RMSE values also increase to 1.31, 2.74%, and 2.76 respectively. This trend continues for the 30-minute horizon with MAE, MAPE, and RMSE at 1.66, 3.76%, and 3.78 respectively. Finally, for the 60-minute horizon, the values further rise to 1.98, 4.74%, and 4.62 for MAE, MAPE, and RMSE, respectively, indicating a degradation in prediction accuracy as the forecasting horizon extends. Figure 17 represents the IoT sensor observed and predicted speed.

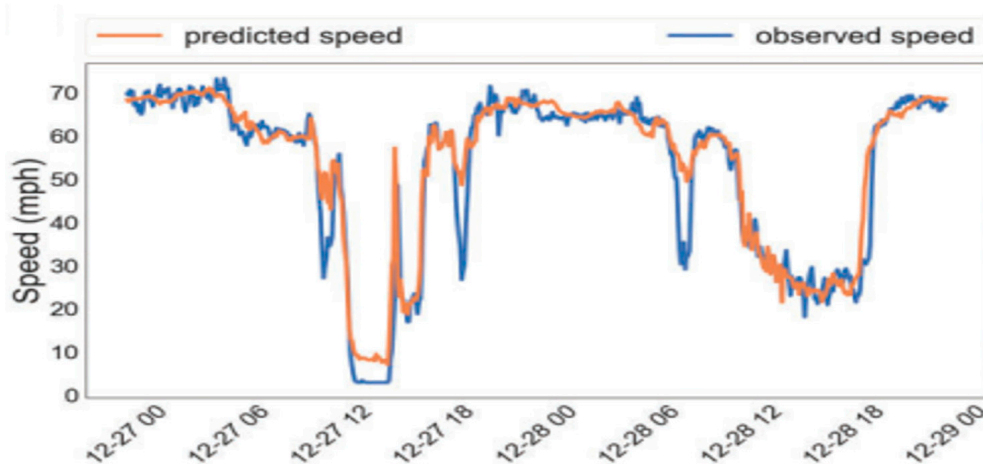


Figure 17. IoT Sensor Model Result.

### 5.6. IOT Sensor – Traffic Forecasting Using Multi Cluster DCRNN

The validation was designed to measure the model's performance based on the number of partitions, the introduction of overlapping nodes, hyperparameter tuning, and multioutput forecasting. MAE Distribution Across Partitions: The model was evaluated on different numbers of partitions, from 2 up to 128 partitions, with each partition trained independently. The distribution of MAE values were calculated for each configuration. The results showed that an increase in the number of partitions (up to 64) reduced the overall MAE, highlighting the model's improved accuracy as partition sizes were optimized. However, further increasing the partitions to 128 resulted in higher errors, indicating a tipping point where partitioning became too granular, causing correlated nodes to split into different partitions, which impacted accuracy. Impact of Overlapping Nodes: To address the issue of spatially correlated nodes ending up in separate partitions, the overlapping nodes method was introduced. By adding neighboring nodes from other partitions, the accuracy of the model improved, particularly for higher partition numbers. The overlapping nodes technique reduced the MAE for the smaller partitions (512 and 1024 partitions), showcasing the significance of spatial correlation in traffic prediction. As a result, including overlapping nodes reduces MAE, especially when the partition size becomes smaller. Multi output Forecasting: The model was also tested for multi output forecasting, predicting both speed and flow simultaneously. The results showed that the multioutput model achieved lower MAE compared to the models predicting only speed or only flow. Error Analysis with Sensitivity Analysis: A sensitivity analysis, using a decision tree model was conducted to understand the factors leading to large errors. forecasting accuracy was improved by reducing the forecasting horizon from 60 minutes to shorter intervals. Hyper parameter Tuning: A hyper parameter search was conducted to further enhance

model performance. This included tuning parameters such as batch size, diffusion steps, and RNN layers.

The tuned model (DCRNN 64 naive hps) outperformed the baseline configuration, reducing MAE across most partitions. Similarly, tuning was applied to the overlapping nodes version (DCRNN 64 overlap hps), which also saw significant improvements in forecasting accuracy. The baseline DCRNN model demonstrates strong short-term forecasting capability but exhibits performance degradation as the prediction horizon increases. To address these limitations, a multi-cluster DCRNN framework with overlapping nodes and multi-output forecasting is introduced, leading to improved prediction accuracy and better preservation of spatial dependencies.

Table 5 and Table 6 compares the MAE values for different model configurations with and without hyperparameter tuning. The tuned models show better performance in terms of the number of nodes with lower MAE values.

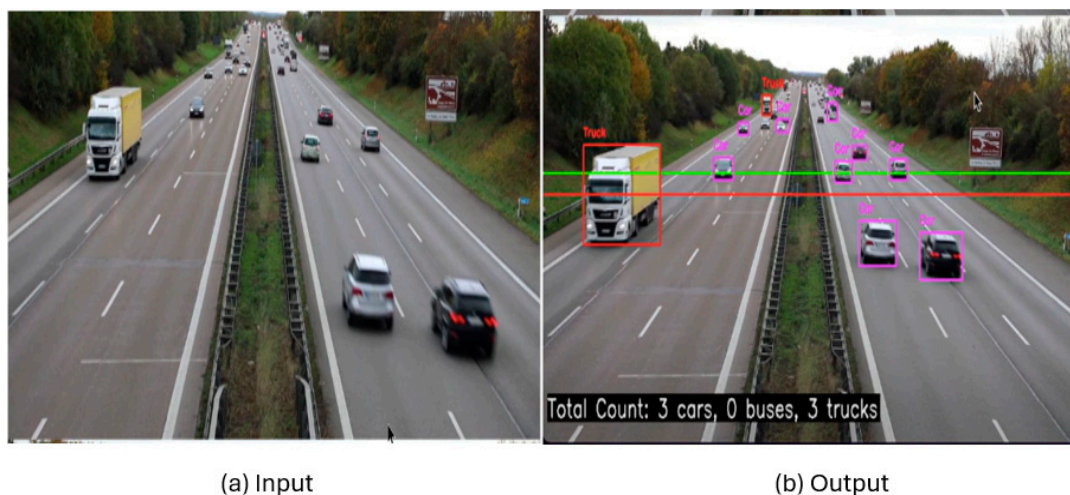
**Table 6.** Graph-Partitioning-Based DCRNN Result(Part 1: Error Metrics).

DCRNN_64 Model	MAE < 1	1 ≤ MAE < 3	3 ≤ MAE < 5	MAE ≥ 5
Naive	1,716	6,729	2,266	449
Overlap	1,837	6,687	2,204	432
Naive_hps	1,920	6,897	1,980	363
Overlap_hps	1,897	6,940	1,972	351

**Table 7.** Graph-Partitioning-Based DCRNN Result(Part 2: Model Details).

DCRNN_64 Model	Trainable Parameters	Training Time (min)	Forecast Time (min)
Naive	14,608	178.67	4.38
Overlap	14,608	221.04	4.88
Naive_hps	19,808	287.05	4.92
Overlap_hps	38,048	461.57	5.83

Output Samples Figure 18 represents CCTV traffic flow model results where a CCTV footage is added as input and in the output the model identifies and counts vehicles within each video frame, providing real-time analytics on traffic density and vehicle types.



**Figure 18.** CCTV Traffic Flow Model Results.

Figure 19 presents a sample result of drone situation analysis, showcasing the identification of cars and trucks in the drone-captured image.

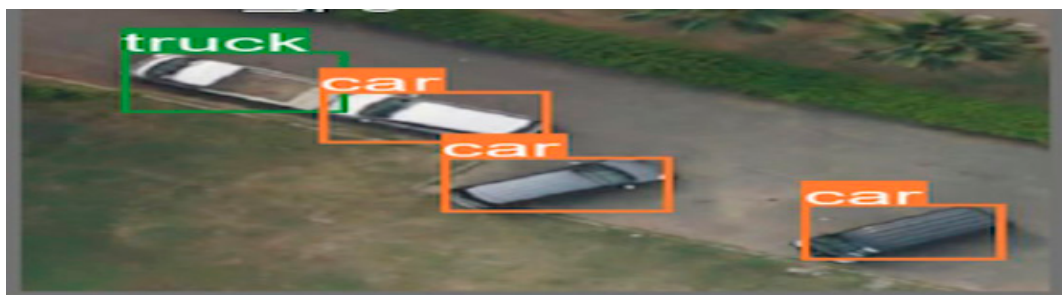


Figure 19. Drone Situation Analysis Results.

This is a sample of drone accident detection results. Figure 20 presents a sample 'Truck Roll-Over Scenario,' where the model generates an alert indicating low congestion severity. It also identifies the type of accident, detects the presence of an emergency vehicle (a fire engine) on-site, and provides data on the number of people, total vehicle count, and the types and quantities of vehicles present.

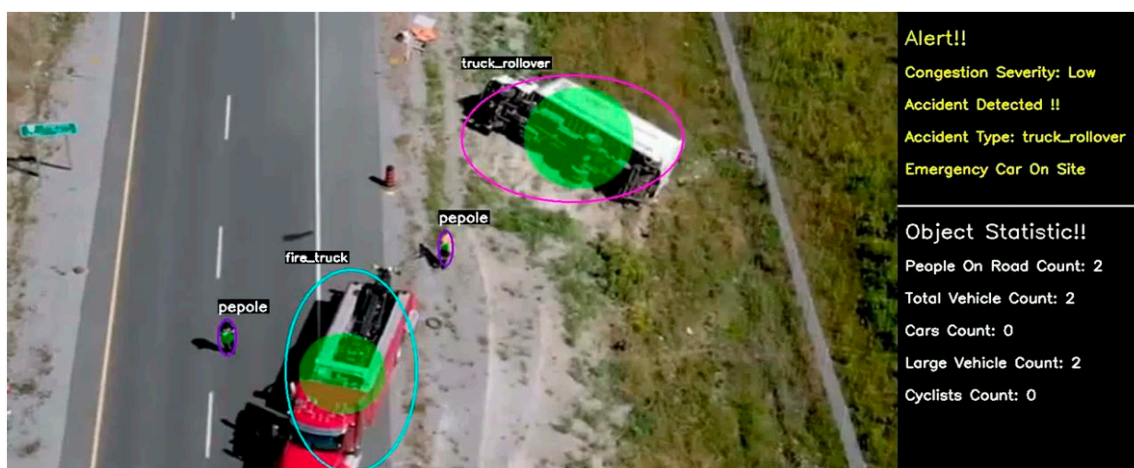


Figure 20. Drone Accident Detection Results.

The figure 21 illustrates traffic forecasting using an IoT sensor model, with the green line on the road map representing a prediction of minimal traffic congestion

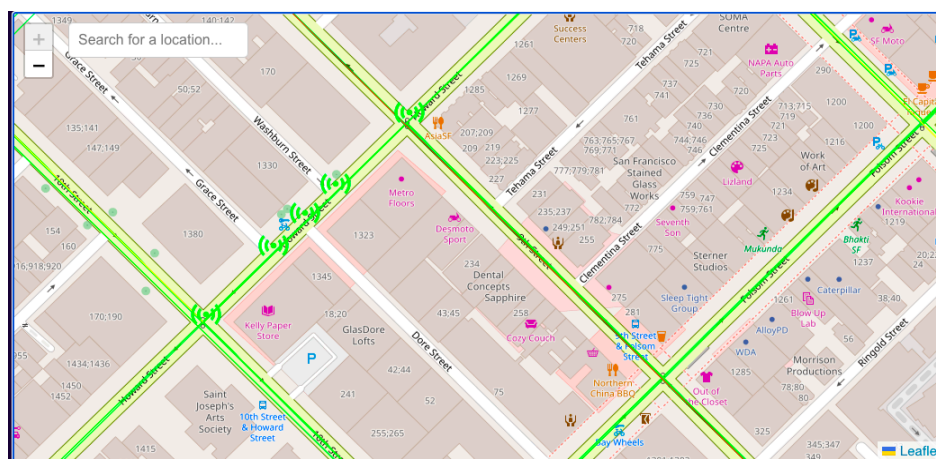


Figure 21. IoT Sensor Traffic Flow Forecast.

## 6. Cloud Platform Design and Implementation

### 6.1. Cloud Platform Design

The proposed web-based system integrates machine learning models with an interactive interface for traffic data processing and visualization. It adopts a Python-based backend and a React frontend, enabling efficient computation and responsive user interaction. The frontend supports dynamic rendering and incorporates secure authentication mechanisms, including login and registration. Additional functionalities, such as device control and location management, allow users to configure system settings.

The dashboard provides real-time traffic analysis and detailed visual reports. Backend services are implemented using the Flask framework, which manages API requests and retrieves real-time traffic video streams based on user-defined locations. A MongoDB database is employed to handle user and location data. Core analytical capabilities are supported by multiple machine learning models. YOLOv8 performs real-time object detection, while Deep SORT enables object tracking across video frames. Temporal dependencies and traffic forecasting are modeled using Diffusion Convolutional Recurrent Neural Networks (DCRNN). These models jointly support congestion analysis and emergency detection. Data preprocessing is conducted in Google Colab, where video data are prepared for downstream analysis. The processed data are then input into the machine learning pipeline for detection and prediction tasks. Frontend interactions trigger backend operations, including real-time data acquisition and model inference. Results are returned to the frontend and visualized through an interactive dashboard.

Overall, the system combines a robust backend with a responsive frontend to provide an effective platform for traffic monitoring and analysis.

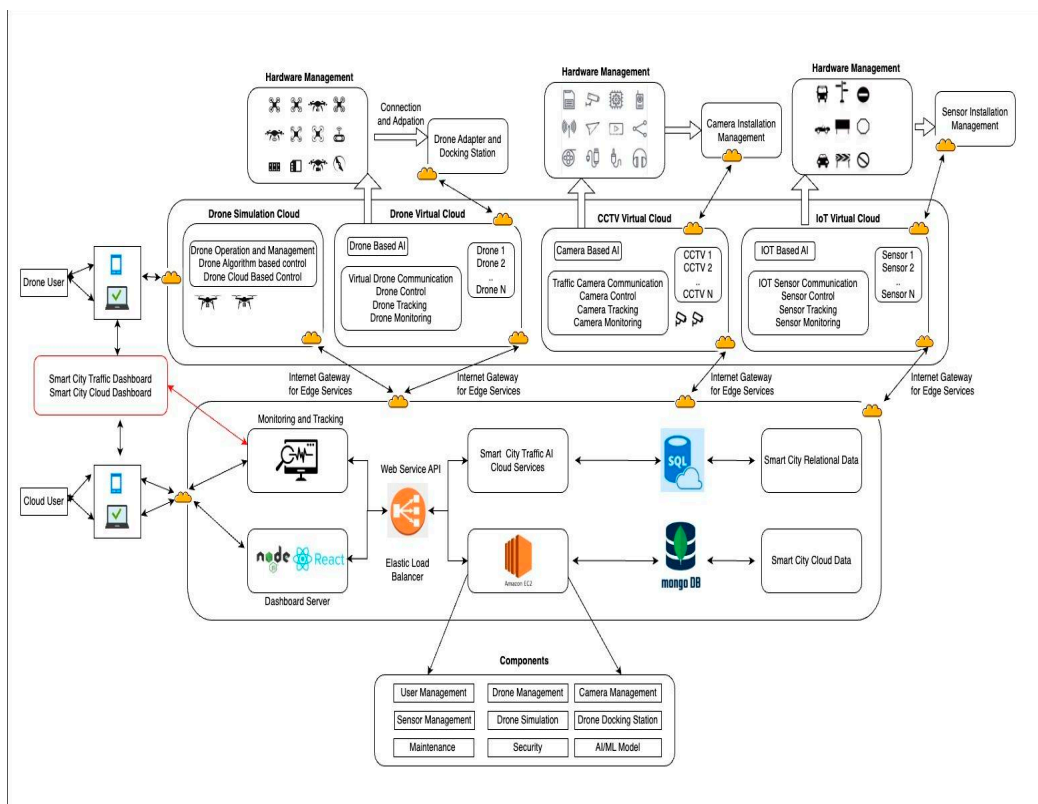


Figure 22. AI-Cloud System architecture.

## 6.2. System Implementation

### 6.2.1. Smart City Traffic Management Dashboard

The system interface is centered on a traffic management dashboard that functions as the primary entry point, providing a comprehensive overview of traffic surveillance operations. A navigation panel on the left offers access to multiple subsystems, including drone, CCTV, and IoT monitoring modules. The dashboard further presents analytical insights through visual components, including a pie chart illustrating drone status (active, inactive, and charging), a line graph depicting weekly drone missions, and a horizontal bar chart summarizing incident types and frequencies. Figure 23 illustrates the Smart City Traffic Management Dashboard.



Figure 23. Smart City Traffic Management Dashboard.

### 6.2.2. Drone Mission Planner

The interface shown in Figure 24 presents the Drone Mission Planner, which supports real-time monitoring and coordination of drone-based surveillance. This module enables traffic authorities to track, plan, and control drone operations across the urban network. A live map interface allows simultaneous tracking of multiple drones, facilitating the identification of congestion, incidents, and emergency conditions. The top navigation bar provides access to key functionalities, including Drone Monitoring, Drone Management, and Alerts and Notifications. Within the monitoring view, users can query specific drones, upon which waypoint information—such as latitude, longitude, speed, and altitude—is displayed. Interactive map icons further provide real-time status updates, indicating whether drones are active or inactive.

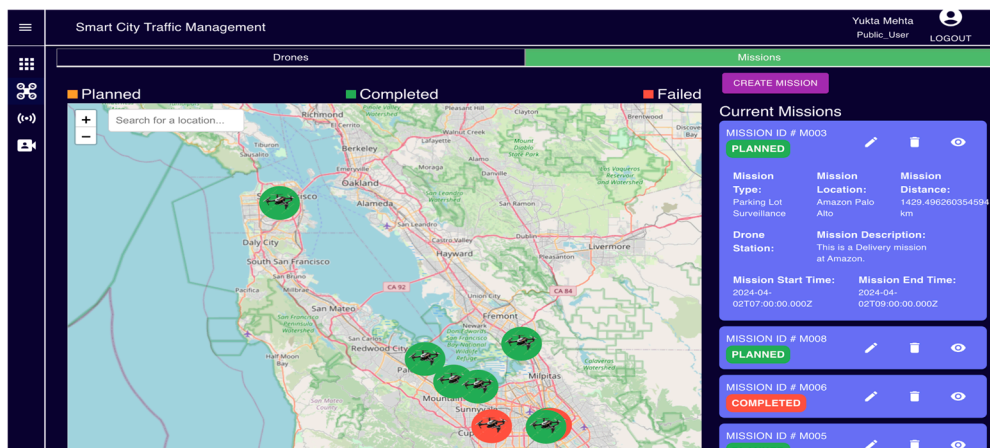
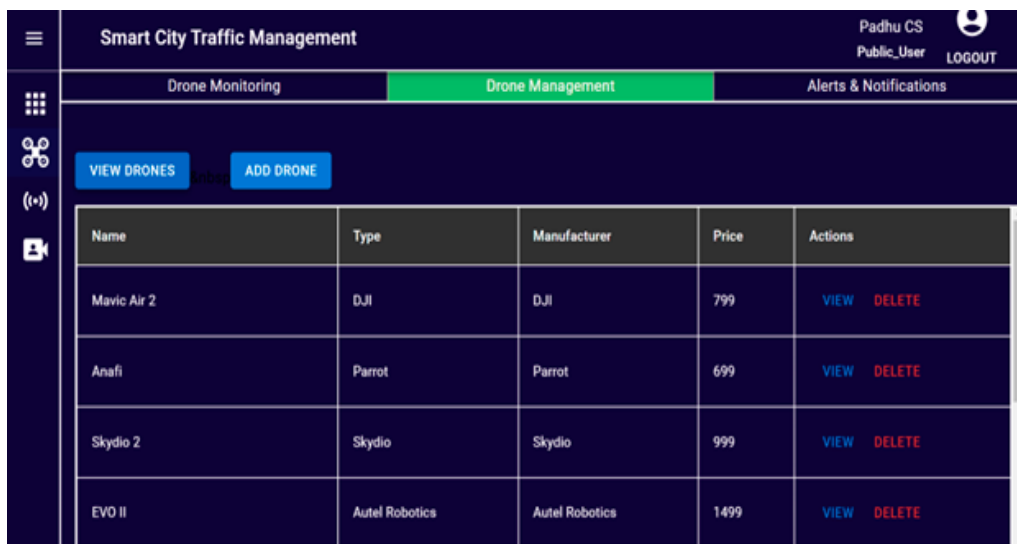


Figure 24. Drone Mission Planner.

### 6.2.3. Drone Management Subsystem

The Drone Management interface, shown in Figure 25 displays detailed information for selected drones, including identifiers, locations, and coordinates. In addition to information retrieval, this module enables authorized users to add, modify, or remove drone missions according to their assigned roles.

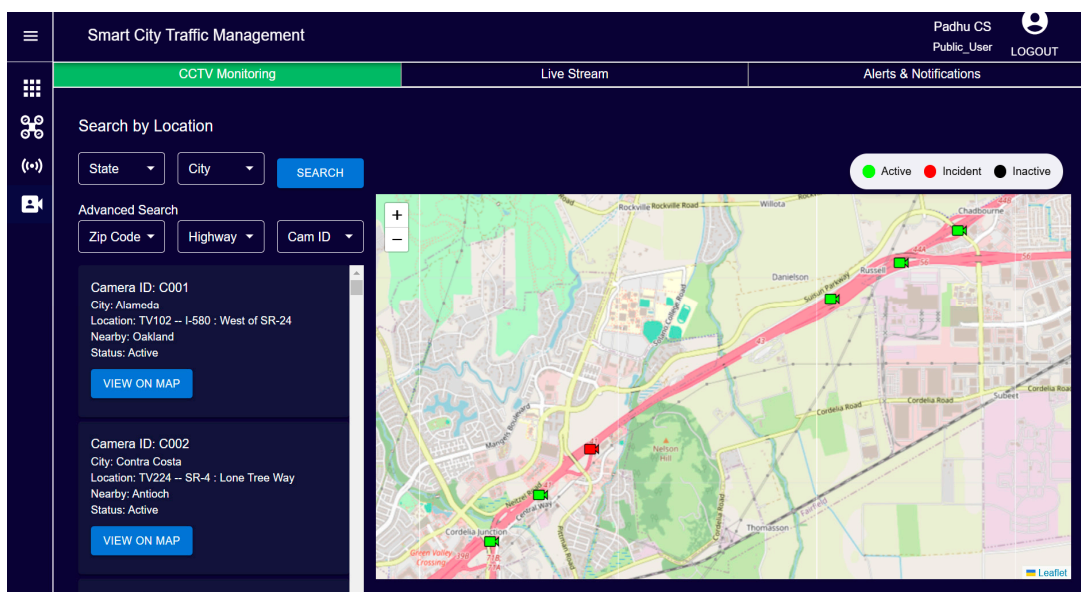


Name	Type	Manufacturer	Price	Actions
Mavic Air 2	DJI	DJI	799	<a href="#">VIEW</a> <a href="#">DELETE</a>
Anafi	Parrot	Parrot	699	<a href="#">VIEW</a> <a href="#">DELETE</a>
Skydio 2	Skydio	Skydio	999	<a href="#">VIEW</a> <a href="#">DELETE</a>
EVO II	Autel Robotics	Autel Robotics	1499	<a href="#">VIEW</a> <a href="#">DELETE</a>

Figure 25. Drone Management System.

### 6.2.4. Smart City Traffic CCTV Management System

The CCTV monitoring dashboard, shown in Figure 26, enables real-time observation of traffic through distributed camera systems. An interactive map visualizes camera locations, supporting efficient coverage assessment. Selecting a camera reveals associated metadata, including camera ID, location, highway, and exit information. Cameras are color-coded to reflect operational status: green indicates active monitoring, red denotes incidents, and black represents inactive units.



Search by Location

State  City  [SEARCH](#)

Advanced Search

Zip Code  Highway  Cam ID

Camera ID: C001  
City: Alameda  
Location: TV102 -- I-580 : West of SR-24  
Nearby: Oakland  
Status: Active  
[VIEW ON MAP](#)

Camera ID: C002  
City: Contra Costa  
Location: TV224 -- SR-4 : Lone Tree Way  
Nearby: Antioch  
Status: Active  
[VIEW ON MAP](#)

Camera ID: C003

Legend: ● Active ● Incident ● Inactive

Figure 26. CCTV monitoring dashboard.

### 6.2.5. Smart City Traffic IoT Management System

Similarly, the IoT dashboard depicted in Figure 27 provides real-time traffic monitoring through distributed sensor devices. Available devices are listed based on selected locations, and detailed attributes—such as device ID, location, highway, and exit—are displayed for each selection. The map interface uses color-coded indicators to represent traffic conditions, with green denoting free flow, yellow indicating moderate traffic, and red corresponding to congestion.

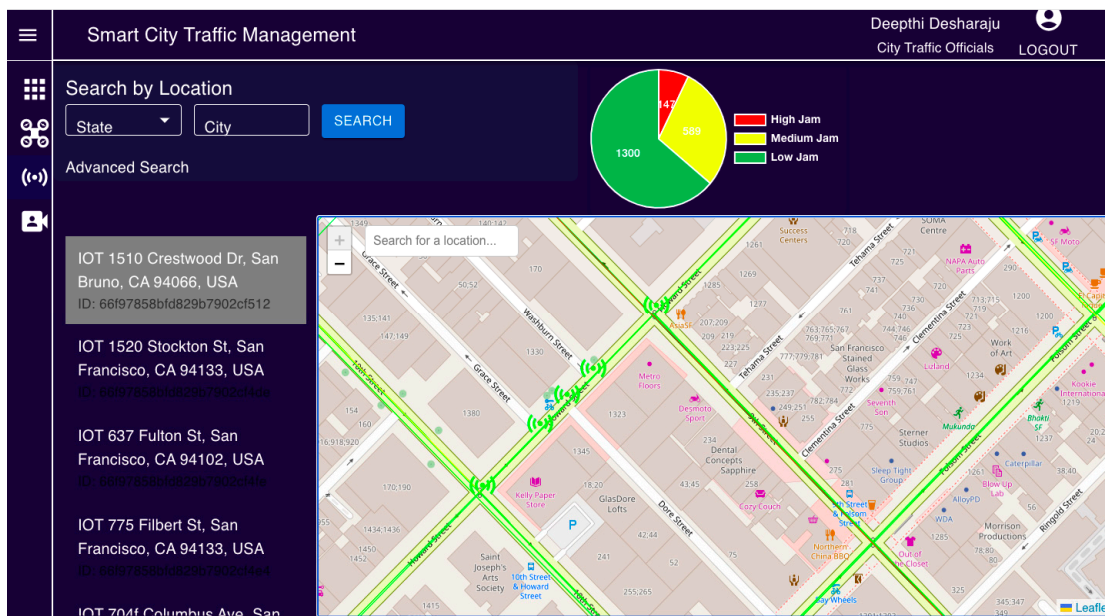


Figure 27. Smart City Traffic IoT Management System.

### 6.2.6. Real-Time Incident Notification

The Alerts and Notifications interface provides access to incident-related data filtered by parameters such as location, drone ID, zip code, highway, and exit number. Tabular displays summarize incident attributes, including type, location, ordinates, and timestamp. Each record includes a detail function that opens a real-time map view, where the selected incident is highlighted to indicate its location and potential impact on traffic conditions.

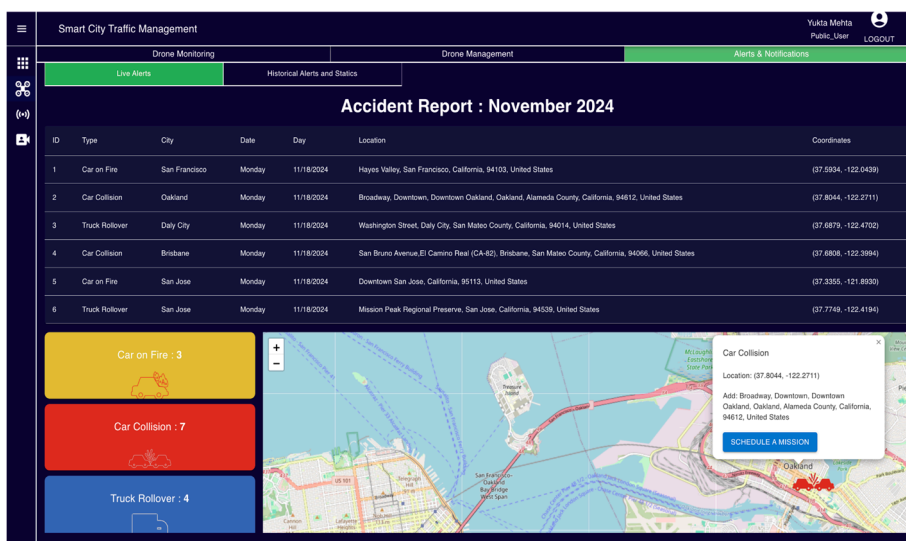


Figure 28. Alerts and Notifications interface.

Collectively, these dashboards constitute integral components of the Smart City Traffic Management System, enabling real-time incident detection, traffic flow analysis, and informed decision-making to support timely and effective traffic management.

## 7. Discussion : UAV-Assisted Data Collection in Smart Traffic Systems

While the primary focus of this study lies in deep learning–driven traffic perception and prediction, UAV-assisted data collection constitutes an important enabling component within the broader architecture of intelligent traffic management systems. Although this work does not explicitly optimize UAV deployment strategies, existing literature reveals a clear and evolving research trajectory that provides valuable context for future system integration. In Internet of Things (IoT) environments, similar data collection mechanisms are extended to more complex systems. UAVs collect data either by direct communication with devices or via cluster heads, while also supporting multi-hop transmission and relay-based communication [25]. In addition, UAVs may operate in hovering or fly-by modes, with trajectory and communication jointly optimized to improve efficiency and coverage. More recent work in urban traffic monitoring shifts toward adaptive data collection strategies, where UAVs follow dynamically updated routes based on traffic conditions and revisit requirements, enabling improved spatiotemporal coverage over multiple monitoring cycles [26].

In contrast, application-driven studies on traffic analysis often adopt a simpler approach, where UAVs perform static or quasi-static data collection, typically by hovering over a target area to capture aerial video for subsequent traffic analysis [27].

## 8. Conclusion and Future Work

This project offers a highly flexible, scalable, and future-ready solution by integrating drones, CCTV systems into a seamless end-to-end framework. The solution, which is intended for a wide range of users, including cloud-based operators and hardware administrators, combines text, video, and picture data to solve problems with infrastructure management, traffic monitoring, and accident detection. The processed data is safely kept in Google Cloud Storage, guaranteeing component accessibility and effectiveness. Every module, CCTV, and drones—has a unique function and uses cutting-edge machine learning algorithms and customized features to produce outstanding outcomes. This project effectively addresses complicated traffic and infrastructure management concerns by integrating state-of-the-art technologies. It provides useful, implementable solutions by utilizing sophisticated deep learning models, scalable cloud solutions, and strong hardware integration. Its efficacy and potential for widespread adoption are demonstrated by its high accuracy rates, user-friendly dashboards, and adaptable features across CCTV, and drone components.

While the proposed framework establishes the practical viability of combining advanced deep learning techniques with cloud-based infrastructure for intelligent traffic management, there remain several avenues through which the system can be further strengthened and extended. One important direction involves the integration of satellite-derived data alongside existing ground-level sensing modalities. Satellite imagery offers broad, continuous spatial coverage of urban environments, making it possible to observe large-scale traffic behavior, evolving land-use configurations, and infrastructural developments. Incorporating such data into the existing IoT-based sensing framework would enable a more comprehensive representation of traffic conditions, bridging localized observations with system-wide dynamics. In addition, the current system already incorporates real-world aerial data obtained from unmanned aerial vehicles (UAVs), providing realistic traffic scenarios for analysis. Nevertheless, these data are primarily sourced from publicly available online platforms and may exhibit variability in quality, coverage, and annotation consistency. Future work will therefore focus on the systematic collection of UAV data under controlled protocols, with standardized acquisition settings and high-quality annotations. Such improvements are expected to enhance data reliability and reproducibility.

Furthermore, expanding the scale and diversity of UAV datasets—across different traffic conditions, geographic regions, and environmental contexts—will contribute to improving model generalization. The use of high-resolution, real-time UAV imagery, combined with more consistent labeling practices, is anticipated to further strengthen object detection accuracy, refine trajectory analysis, and improve the robustness of incident identification in complex urban traffic environments.

Another area for development concerns the implementation of real-time traffic report generation. By integrating streaming data processing with automated analytical pipelines, the system could continuously transform incoming data into structured, interpretable reports. Such outputs would provide timely and actionable insights for urban authorities, supporting more effective responses to congestion patterns, traffic incidents, and emerging risks.

Looking further ahead, this research envisions the evolution of the current framework into a generalized, AI-driven platform for smart city applications. Anchored in scalable, cloud-native architectures, the system could progressively transition toward an AI-centric infrastructure capable of supporting a wider spectrum of urban services beyond traffic management. Designed as a modular and adaptable solution, it holds the potential for deployment across diverse urban contexts, thereby enabling broader scalability and contributing to the development of data-informed, intelligent city systems.

**Author Contributions:** Conceptualization, L.Z. and J.G.; methodology, A.A., Y.M. and D.D.; software, S.M.; validation, S.M.; formal analysis, A.A., Y.M. and D.D.; investigation, A.A., Y.M. and D.D.; data curation, P.C.S.; writing—original draft preparation, L.Z.; writing—review and editing, L.Z.; visualization, L.Z. and S.M.; supervision, J.G.; project administration, J.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work received no external funding.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell.* 2017;39(6):1137-1149.
2. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016:779-788.
3. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. *arXiv.* 2020.
4. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: *IEEE International Conference on Image Processing.* 2016.
5. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *IEEE International Conference on Image Processing.* 2017.
6. Ahmed MS, Cook AR. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transportation Research Record.* 1979.
7. Zhao Z, Chen W, Wu X, Chen PCY, Liu J. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems.* 2017.
8. Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *International Conference on Learning Representations.* 2018.
9. Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: *Proceedings of the International Joint Conference on Artificial Intelligence.* 2018.
10. Razi, X. Chen, H. Li, H. Wang, B. Russo, Y. Chen, and H. Yu, "Deep learning serves traffic safety analysis: A forward-looking review," *IET Intell. Transp. Syst.*, vol. 17, no. 1, pp. 22–71, 2023.

11. A. Rafique, A. Al-Rasheed, A. Ksibi, M. Ayadi, A. Jalal, K. Alnowaiser, H. Meshref, M. Shorfuzzaman, M. Gochoo, and J. Park, "Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images," *IEEE Access*, vol. 11, pp.2993–3007, 2023.
12. M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S.M. Altowaijri, "Smarter Traffic Prediction Using Big Data, in-Memory Computing, Deep Learning and GPUs," *Sensors*, vol. 19, no. 9, pp. 2206,2019.
13. A.A. Salunke, "Enhancing urban traffic management through predictive modelling and drone-captured image analysis for smart traffic lights," *Int. Res. J. Mod. Eng. Technol. Sci.* (2023).
14. Z. Dai, H. Song, X. Wang, Y. Fang, X. Yun, Z. Zhang, and H. Li, "Video-Based Vehicle Counting Framework," *IEEE Access*, vol. 7, pp.64460–64470, 2019.
15. N. Algiriyage et al., "Towards real-time traffic flow estimation using YOLO and SORT from surveillance video footage," *Conference Paper*, Jul. 2021.
16. V. S. Sindhu, "Vehicle identification from traffic video surveillance using YOLOv4," in *Proc. Int. Conf. on Intelligent Computing and Control Systems*, May 2021.
17. V. Radojic et al., "Advancements in computer vision applications for traffic surveillance systems," *Zbornik Radova Sinergija*, Dec. 2023.
18. T. Osman et al., "Dynamic traffic control using computer vision," in *Proc. IEEE CCWC*, Jan. 2017.
19. Sakhuja, "Intelligent Traffic Management System using Computer Vision and Machine Learning," *Innovative Research Thoughts*, vol. 9, no. 5, pp. 1–10, Oct.–Dec. 2023, doi: 10.36676/irt.2023-v9i5-001.
20. M. Sharma et al., "Intelligent traffic light control system based on a traffic environment using deep learning," *Conference Paper*, Dec.2020.[Online].Available:<https://doi.org/10.48047/some-link>
21. Y. Myagmar-Ochir and W. Kim, "A Survey of Video Surveillance Systems in Smart City," *Electronics*, vol. 12, no. 17, pp. 3567, 2023.
22. S. Dhingra, R. B. Madda, R. Patan, P. Jiao, K. Barri, and A. H. Alavi, "Internet of Things-Based Fog and Cloud Computing Technology for Smart Traffic Monitoring," *Internet of Things*, vol. 14, pp. 100175, 2021.
23. Sahil and S. K. Sood, "Smart Vehicular Traffic Management: An Edge Cloud Centric IoT Based Framework," *Internet of Things*, vol. 14, pp.100140, 2021.
24. X. Yu, F. Sun, and X. Cheng, "Intelligent Urban Traffic Management System Based on Cloud Computing and Internet of Things," in *2012 International Conference on Computer Science and Service System*, IEEE, 2012, pp. 2169–2172.
25. Z. Wei et al., "UAV-Assisted Data Collection for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15460–15483, 2022.
26. Y. Bai and Y. Feng, "A Dynamic Unmanned Aerial Vehicle Routing Framework for Urban Traffic Monitoring," *IEEE Transactions on Intelligent Transportation Systems*, 2025.
27. M. A. Khan et al., "Unmanned Aerial Vehicle-based Traffic Analysis: A Case Study to Analyze Traffic Streams at Urban Roundabouts," *Procedia Computer Science*, vol. 130, pp. 636–643, 2018.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.