Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

Smart Traffic Tracking: Revolutionizing Surveillance with YOLOv9

Ayon Roy[1*], Sougata Mallick [2], Junita Threse [3], Rishi Kumar [4], Dr. Koushik Karmakar [5]

[1] Narula Institute of Technology, ayonroy10000@gmail.com
[2] Narula Institute of Technology, sougatamallick17@gmail.com
[3] Narula Institute of Technology, junitathrese15@gmail.com
[4] Narula Institute of Technology, rishikum637@gmail.com
[5] Narula Institute of Technology, Assistant Professor & HOD,
Dept. of Computer Science and Technology,
k.karmakar100@gmail.com

***Abstract*—** This project introduces a sophisticated Traffic Monitoring System integrating YOLOv9 for precise vehicle detection and DeepSORT for continuous multi-object tracking, ensuring that vehicles remain uniquely identified across multiple frames. Traditional CCTV surveillance systems mainly function as passive monitoring tools and lack the intelligence to track suspicious vehicles in real-time. This constraint makes a vehicle that is supposed to be tracked go undetected once it is out of view. The proposed system fills this critical gap as it tracks in real-time and monitors continuously so that no vehicle of interest is ever missed. The system applies perspective transformation, computes time-distance, and gets accurate speed on roads with varying geometries. This would allow for the detection of erratic driving behaviors like overspeeding or abrupt braking. A real-time dashboard, developed using Streamlit, presents key insights into traffic through live video feeds, vehicle counts, speed distributions, etc. This urban environment-friendly system is scalable on multiple CCTV cameras, making city-wide or highway-wide tracking possible without breaking up the continuity of tracking. This tool is not just a prerequisite in the sense of traffic flow analyses and congestion control but also plays a crucial role in the proactive management of law and order, making it highly essential for smart city infrastructure.

***Keywords*—**Traffic Monitoring System, Vehicle Detection, Multi-Object Tracking, Real-Time Surveillance, Smart City Infrastructure.

I. INTRODUCTION

In this era of accelerating urbanization, vehicular challenges range from road safety and traffic congestion to trafficking crime prevention. Current systems of traffic monitoring are based on the use of a few CCTV cameras and rely mainly on manual intervention; hence, such systems are far from real-time, scalability, and intelligent insights. They cannot detect fast-moving vehicles quickly. This further gives rise to many significant constraints on law enforcement, traffic management, and public safety. The volumes of vehicles going through roads have increased and therefore require an advanced, automated approach that can help enhance situational awareness, reduce congestion, and improve security. This paper proposes a Traffic Monitoring System and Data Analysis using YOLOv9 and Machine Learning focuses on the following challenges with real-time vehicle detection, tracking, and analysis. The YOLOv9 is the most advanced object detection framework. It detects vehicles at very high speed with a high accuracy level, and DeepSORT handles the smooth tracking of vehicles in multiple frames. Thus, this system introduces a scalable and smart solution, which can easily integrate with the existing CCTV infrastructure, therefore proving cost-effective as well as

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

appropriate for large-scale deployment. This system would be able to move from high-tech traffic surveillance into advanced car tracking for security and crime. It would trace, detect, and follow criminals with the vehicles involved in their illicit activities such as smuggling, human trafficking, or security risks, in which case there are blind spots to the normal surveillance systems. This is designed modular with a scale in different camera networks that allows tracking across cities to ensure real-time alertness in flagged vehicles by the law. This system achieves higher computational efficiency without losing superior detection accuracy through depth-wise convolutions, the C3Ghost architecture, and real-time machine learning algorithms. It can be applied to smart and future-ready cities for traffic violation detection, signal time optimization, and environmental impact reduction. Smart cities are able to create a safer and efficient, sustainable urban space through improved intelligent traffic surveillance. This paper is organized as follows: Section I contains the introduction of this paper. Section II contains a literature Survey which has a research survey of previous works and also compares with our work. Section III shares the methodology of this paper which is the solution to the previously stated problem. It also contains the hardware and software specifications required to run the code of this project in real time. Section IV is the result and analysis of this paper. Section V gives us a concise idea of the future scope of this project by discussing which and what other areas can be updated or developed into other technologies like integrating IoT into it, etc. Finally, Section VI concludes this project.

A. Integration with law enforcement and smart city infrastructure

The integration of smart surveillance in law enforcement and smart city infrastructure aligns well with our project aiming to use the DeepSort tracking algorithm alongside YOLOv9 for smart vehicle object detection, as our work is based on it. Our system empowers real-time traffic monitoring, speed computation as well as urban management and public safety. This is also highlighted by Ali \& Kaur's paper that such surveillance systems should be integrated with cybersecurity measures to avoid cyber threats such as data breaches, system manipulation, or AI-driven cyber-attacks. Since our project there is processing video at real-time data visualization, this process should be secured as part of data integrity and security to support law enforcement [1]. Also, how systems based on AI-driven monitoring and data-sharing framework improve the elite force capabilities and render our system a desirable part of the crime prevention and civil control system. By using encryption, multiple authentications, and the use of AI-based anomaly detection, the best practices of cybersecurity are being followed to prevent unwanted access and tampering with the surveillance data [2]. Thus, our system enhances the way we govern a smart city, especially crime prevention, and improving ways of tracking and analyzing traffic accidents and the authenticity of recorded media, in this way it helps law enforcement analyze traffic incidents, track vehicle movements, and validates recorded footage as well [1]. Furthermore, according to our cybersecurity laws :

• Section 66B: Deals with identity theft and digital fraud,
It is very important in preventing unauthorized access to smart city surveillance systems.

• Section 66C & 66D: Covers cyber impersonation and
However, even this snowball role has a darker side for it helps police track down criminals who exploit surveillance loopholes and engage in fraudulent activities using stolen digital identities.

• Section 69: Enables government agencies to monitor, and decrypt digital communication is for the purpose of national security and cybercrime. This is particularly relevant for smart surveillance of tracking real time criminal activities.

Law enforcement agencies can use these cyber laws to proactively detect cyber threats and prevent unauthorized access to critical city infrastructure [1].

II. RELATED WORK

With the growing complexity of urban traffic, we have to use an effective monitoring system and here in this project we use computer vision and deep learning methods. In this literature survey, we are focusing on the development of YOLO and YOLOv9. A real-time object recognition system that treats this as a single regression problem is the YOLO algorithm. Unlike region-based approaches, YOLO greatly reduces processing costs by simultaneously predicting

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

bounding boxes and class probabilities [3]. This design is perfect for real-time traffic applications due to its remarkable detection speed. YOLOv2 and YOLOv3 were further improved by adding batch normalization, multiscale detection, and anchor boxes to increase accuracy without compromising speed [4] [5]. Due to these improvements, the algorithm became more flexible and could handle complex traffic situations with different lighting conditions and density. To address the need for lightweight models in resource-constrained environments, YOLOv4 incorporated CSPNet (Cross-stage Partial Network) and new data augmentation techniques such as mosaic and dropblock regularization. This version showed improved performance in terms of both accuracy and efficiency, making it highly suitable for traffic monitoring systems deployed on edge devices [6]. YOLOv5 was developed with an additional focus on speed and model size optimization, making it a viable option for real-world traffic monitoring. YOLOv5 has gained widespread use in real-time monitoring due to its improved training pipelines, improved data augmentation techniques, and hardware platform compatibility [7]. Subsequent versions, YOLOv6 and YOLOv7, included further architectural enhancements, such as the use of RepConv and EfficientRep blocks, which improve computing efficiency and recognition accuracy [8] [9] Incorporating deep convolutions and the novel C3Ghost architecture, the latest version, YOLOv9, has significantly improved object recognition speed. For real-time traffic monitoring in congested urban environments, our developments have improved bounding box accuracy and feature extraction while enabling lightweight processing [10]. Convolutional neural networks (CNNs), a kind of deep learning technique, have been crucial to YOLO's success. CNNs make it possible to efficiently extract spatial features, which improves the accuracy of real-time recognition applications [11]. Deeper and more efficient object recognition models were made possible by innovations such as Res-Net, which solved the vanishing gradient problem [12]. Vehicle recognition, traffic congestion analysis, and violation detection have all been effectively implemented in the context of traffic monitoring using YOLO-based systems. For example, it has been shown that YOLOv3 can successfully identify cars moving at high speeds in various weather conditions [13]. Similarly, YOLOv4 was used to estimate traffic flow and outperformed traditional techniques in terms of accuracy and processing efficiency [14].

Table 1. Evolution of YOLO Algorithm

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

Year	Paper Name	Author(s)	Experiment	Input Parameters	Output	Technology Used
2020	Real-time Object Detection Method Based on Improved YOLOv4-tiny [15]	Zicong Jiang, Liquan Zhao, Shuaiyang Li, Yanfei Jia	Real-time applications	Real-time applications	Reduced complexity, faster detection	YOLOv4-tiny, ResNet-D
2021	YOLO-Z: Improving Small Object Detection in YOLOv5 for Autonomous Vehicles [16].	Aduen Benjumea, Izzeddin Teeti, Fabio Cuzzolin, Andrew Bradley	Autonomous vehicles	YOLOv5, mAP, IOU	mAP +6.9%, Inference: +3ms	YOLOv5, Deep Learning
2022	YOLOv6: A SingleStage Object Detection Framework for Industrial Applications [8]	Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei	COCO dataset	YOLOv6-N, YOLOv6-S, FPS, AP metrics	YOLOv6-S: 43.5% AP@495 FPS	YOLOv6, COCO, Deep Learning
2023	YOLOv8 Based Object Detection for Self-driving Cars [18]	Zakia Afrin; Fariya Tabassum; Hafsa Binte Kibria; MD. Rafi Imam; Md. Rokibul Hasan.	Roadmaps for self-driving cars	YOLOv8 model, mAP metrics	mAP: 77.8%	YOLOv8, Deep Learning
2023	A Small Object Detection Algorithm for Traffic Signs Improved YOLOv7 [19]	Songjiang Li, Shilong Wang, Peng Wang	TT100K dataset	YOLOv7, mAP@0.5	mAP@0.5: 88.7% (+5.3%)	YOLOv7, Deep Learning
2023	Small-object Detection Based on YOLOv5 in Autonomous Driving Systems [20]	Bharat Ma-haur, K.K. Mishra	BDD100K dataset	YOLOv5, mAP, FPS metrics	mAP: +3.35%, FPS: +2.57	YOLOv5, Deep Learning
2024	Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack [21]	Longxiang You, Chen, Ci Yajun Xiao, Chaoyue Sun and Rongzhen Li	UA-DETRAC dataset	YOLOv8, ByteTrack, mMOTA, mAP metrics	mAP@0.5:9% increase, mMO-TA: 6.7%	YOLOv8, ByteTrack
2024	An Advanced YOLOv7 Model for Vehicle Detection to Enhance the Security of Public Places [22]	Nikarika; Kukreja; Thap-liyal; Nitin Mani-sha Aeri; Rishabh Sharma	Vinay Public places	YOLOv7, accuracy metrics	Accuracy: 98.35%, Loss: 0.3321	YOLOv7, Deep Learning
2024	Engineering Vehicle Detection Based on Improved YOLOv6 [23]	Huixuan Tianju Ling, Yangqianhui Zhao, Zhang and Meng Lei	Engineering vehicle dataset	YOLOv6 model, mAP metrics	mAP50: 95.9%, mAP50:95:88.5%	YOLOv6, Deep Learning

Table 1: Evolution of YOLO Algorithm

B. Comparison of Traditional Traffic Monitoring Methods and AI-Driven Models.

Among major aspects of traffic monitoring are urban traffic flow management, road safety enhancement, etc. Most traffic monitoring and vehicle detection activities have been performed through traditional methods namely inductive loops, radar detector, and human observation. But there are problems with traditional methods such as high infrastructural costs, low flexibility and relatively more chances of human errors. However, unlike the AI models, which employ the deep learning as well as the computer vision techniques, real time traffic is monitored with increased accuracy and efficiency. Traditional techniques are hardware based, involve complicated installation and maintenance needs and most utilize hardware coupled with tools like YOLO to detect objects in the video streams as well as DeepSORT to track the objects in real time. This means that it increases the level of automation, which in fact eliminates the requirement of human intervention, and therefore enhances scalability and cost efficiency. They are traditional and have low prediction ability. On the contrary, however, AI models are adaptive learning mechanisms that let them learn how to cope with several traffic scenarios, adapt to changing weather conditions, congestion management, etc. In the context of these grounds, it is expected this research is about the development of an AI

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

augmented intelligent vehicle detection system combined with YOLOv9 and DeepSORT for real time tracking and speed estimation. This makes the framework a priority for the establishment smart city as it incorporates accuracy as well as easy means of observing traffic using AI.

C. Ablation Studies: Contribution of Model Components to Accuracy

There must be ablation studies in deep learning research to see how the different model components change the model performance. Through systematic modification of elements, their effect on accuracy, speed, and efficiency can be assessed. Key architectural features of this project, including the Generalized Efficient layer Aggregation Network (GELAN), Programmable gradient Information (PGI) [24], and depthwise convolution are all used to make the traffic monitoring project real-time. ELAN is improved to integrate computational blocks flexibly and therefore to optimize parameter utilization, and the name of the architecture is denoted by its name, GELAN. YOLOv9 utilizes Cross Stage Partial (CSP) blocks, a traditional convolutional layer's replacement, which reduces parameter and computational load and keeps accuracy the same. As per the YOLOv9 paper, use of CSP blocks which replace standard convolutions resulted in 10% less parameters, 15% fewer computation and 0.7% better mean Average Precision (mAP) [24]. However, when it comes to real time traffic monitoring, this efficiency is necessary to achieve speed and accuracy in balancing. Gradient degradation is a common issue experienced by deep networks, and to deal with it, PGI is used. It also presents auxiliary reversible branch that accelerates gradient flow to make training stable. Ablation studies indicate that PGI improves mAP by 0.6% in conjunction with improved detection across all the object sizes (small, medium, and large) [24]. It is necessary because it is needed in traffic systems that must detect vehicles of various dimensions. In addition to that, depthwise convolutions improve the efficiency by lowering the parameters without degrading the feature extraction capability of the network. Unlike standard convolutions, each channel is processed independently which reduces the computational cost. However, ablation studies suggest that there is a trade-off between improving speed and memory efficiency and minor decrease in accuracy in dense scenes [24]. Thus, they are optimised for speed but not speed and precision.

III. METHODOLOGY

This Traffic Monitoring System develops a reliablity by using the DeepSort tracking algorithm combined with YOLOv9, which is a highly advanced object identification model. It can provide real-time identification of the vehicle, tracking, counting, speed computation, and visualizing the data in the process of video feed processing. Each stage of the workflow (see Fig.1) like video pre-processing, object recognition, vehicle tracking, data analysis, and result visualization is intended to guarantee accuracy and efficiency in actual traffic situations.

The objective is to obtain accuracy and efficiency under real traffic conditions. The first step is video processing. This is done with the help of OpenCV, where the video frames are extracted either from a live stream or recorded material. Because maintaining real-time performance at high speeds but, at the same time, giving detailed detections is essential, frames are scaled, normalized, and optimized to fit YOLOv9's input size. It is the basis of object detection, and due to its advanced design, it can detect and classify cars, lorries, buses, as well as motorcyclists. The system is quite reliable for traffic surveillance due to its high detection quality and strength even in low-light conditions. The tracking module receives detected bounding boxes, class labels, and confidence scores for each object the model reports. DeepSort tracking algorithm is utilized as a means of ensuring vehicle IDs do not change among frames. Even in instances of chaos or obscuration conditions, DeepSort provides an ID for every single vehicle and refreshes both the motion and visual qualities of these IDs. The result is good tracking in the case of cars around the journeys within the frame. To track an object across consecutive frames using bounding boxes:

$$\Delta x = x_2 - x_1 , \quad \Delta y = y_2 - y_1$$

Where:

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

• (x1, y1) represents the position of the object in the initial frame.

• (x2, y2) represents the position of the object in the following frame.

The movement of tracked items can monitor vehicle counts by type of vehicle and over-defined entrance and exit lines, thus stratifying it. Now that this capability exists, the system can produce real-time directional counts which enables an extensive analysis of traffic flow. The system comprises this estimation of speed, in the form of assembling both viewpoint changes and time analysis, through the transformation of vehicle positions to real-world coordinates. Speed taken on perspective transformation in some regions of interest inside the interesting frame. It gives how fast each of the cars goes in by the far travel it took for a vehicle to cover that frame through some frames over which it has divested and the time it used. More than this, further computes mean speed dynamically over all the vehicles inside the frame.

$$Speed(km/h) = Distance(m)/Time(s) \times 3.6 \quad (2)$$

A user-friendly dashboard created with Streamlit is used to render the outputs of the system. We have used a live video stream, Bounding boxes, labels, tracking lines, and speed data to superpose over the dashboard. The important information that is shown comprises of frames per second, total vehicle counts, number of vehicle entrances and exit counts, and average speed over a metrics panel. Further bifurcation of the identified categories of vehicles is demonstrated which is visualized in a pie chart, (see Fig. 3) that gives us the necessary information on traffic flows.

D. *Flowchart*

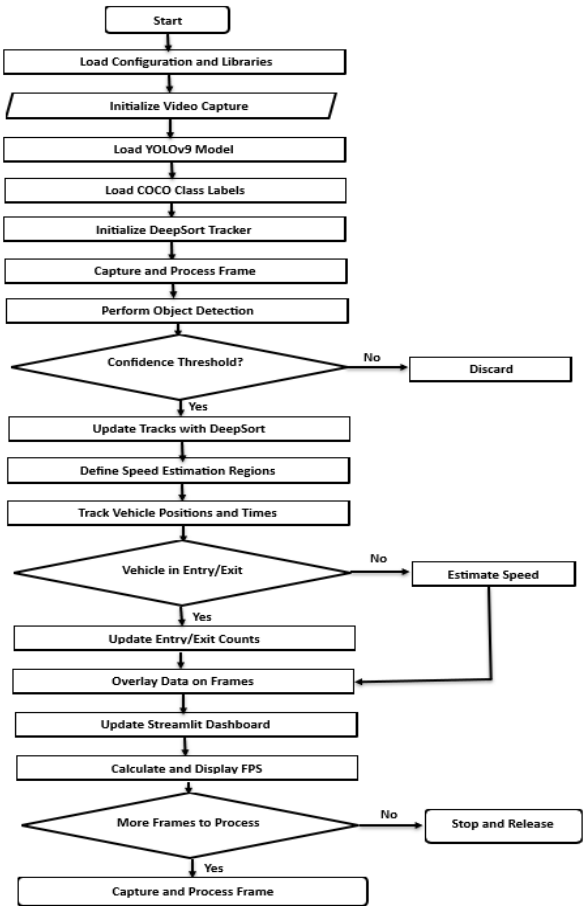

Figure 1. Workflow of the Traffic monitoring system YOLOv9.

E. *Software Requirements*

Windows 10 or higher, or a recent version of Linux (e.g., Ubuntu 18.04 or higher) to provide a stable and compatible environment for running the software and necessary libraries. Python 3.6 or higher is the primary programming language for writing and executing the project's code. NVIDIA CUDA Toolkit is com-patible with the GPU and operating system for enabling GPU acceleration. NVIDIA cuDNN library for deep neural networks, ensuring optimized performance and compatibility with the CUDA Toolkit.

Important Libraries and Frameworks:

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

• OpenCV: OpenCV library for video capture, frame processing, drawing visual elements, and displaying output.

• PyTorch: PyTorch deep learning framework for loading and running the YOLOv9 model.

• NumPy: NumPy library for numerical operations and array manipulations.

• DeepSort: DeepSort tracking algorithm for maintaining object identities across video frames.

• Streamlit: Streamlit library for creating interactive web-based dashboards to visualize real-time results.

• Plotly: Plotly library for generating dynamic and interactive visualizations such as pie charts.

F. Hardware Requirements

A powerful multi-core processor (Intel i7 or AMD Ryzen 7 equivalent or higher) to handle general computation tasks, model loading, and some parts of the video processing pipeline. An NVIDIA GPU with CUDA support (e.g., NVIDIA GTX 1060 or higher, or NVIDIA RTX series) to leverage parallel processing for faster model inference and real-time performance. At least 16 GB of RAM to ensure smooth operation, especially when dealing with high-resolution video streams and multiple processes running simultaneously. SSD with at least 500 GB of available space to store large video files, model weights, and other data. SSDs are preferred due to their faster read/write speeds compared to traditional HDDs. A high-definition camera or webcam if real-time video capture is required, ensuring high-quality video input for accurate detection and tracking.

IV. RESULTS AND DISCUSSION

G. Evaluation

The evaluation focuses on the performance and accuracy of the YOLOv9-based Traffic Monitoring System in detecting, tracking, and analyzing vehicles in real-time. The system effectively detects and classifies vehicles, including cars, trucks, buses, motorcycles, and bicycles, using YOLOv9's advanced object detection capabilities. Outputs include bounding boxes and labels for each detected vehicle, (see Fig.2) ensuring precise classification with minimal errors, even under challenging conditions such as low light or heavy traffic.

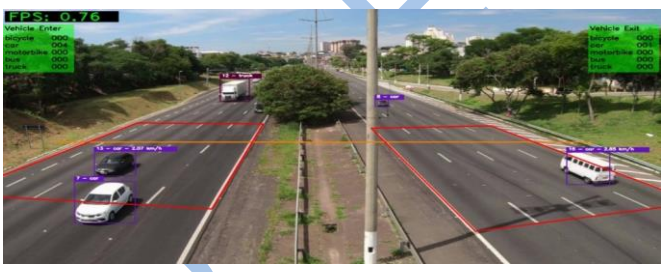

Figure 2. Real-time vehicle detection, classification, and speed estimation.

Vehicle tracking is achieved using the DeepSort algorithm, which assigns unique IDs to each vehicle and maintains these identities across frames. In addition, the system performs vehicle counting by monitoring predefined entry and exit lines, providing total counts and directional insights for each vehicle type. By calculating the distance traveled over time, the system outputs the speed of individual vehicles in kilometers per hour and computes the average speed across all vehicles dynamically. The real-time outputs are visualized on an intuitive dashboard, which displays the live video feed with overlaid bounding boxes, speed labels, and tracking lines. Additional metrics such as FPS (frames per second), total vehicle counts, average speed and pie chart (breakdown of vehicle types) are prominently featured. The system's performance was measured based on its ability to correctly identify cars, trucks, buses, motorcycles, and bicycles under diverse conditions. Our traffic monitoring system demonstrated notable accuracy in detecting vehicles, with a strong performance in identifying cars. To quantify the overall accuracy, we calculated the ratio of correctly detected vehicles to the total number of actual vehicles. With a total of 19 vehicles (15 cars, 1 bus, and 3 trucks) (see Fig. 3), the system correctly identified 18 (15 cars and 3 trucks, excluding the over-detected truck) (see Fig. 4). This resulted in an overall accuracy of approximately 94.74%.

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

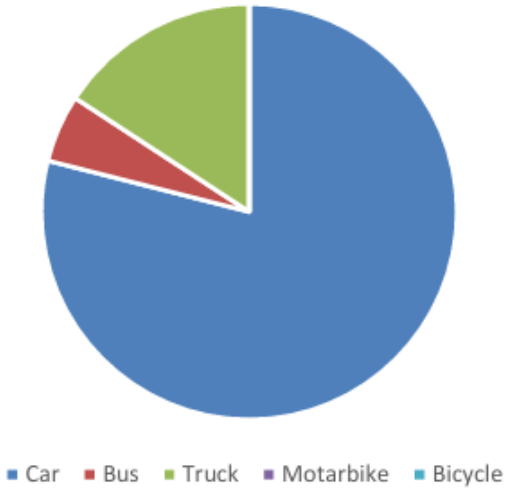

Figure 3. Class distribution in Ground Truth.

This project used a yolov9-c-converted.pt model pretrained on the COCO dataset. The performance of the implemented system was measured through standard object detection metrics, which include mAP and AR in order to be able to more broadly understand whether the system does its job as an effective observer and classifier within traffic scenes. The model achieved a Mean Average Precision (mAP) of 53.0% at an Intersection over Union (IoU) threshold range of 0.50 to 0.95. This metric balances precision and recall across varying degrees of overlap between predictions and ground truth and is generally considered a standard measure of model accuracy. At a lower IoU threshold of 0.50, it increased the mAP to 70.2% which meant strong performance even in scenarios where more lenient overlaps are acceptable. Similarly, at an IoU threshold of 0.75 which was stricter, the mAP was 57.8%, which meant the model could maintain relatively high accuracy even in challenging conditions of detection. To further dissect the performance, mAP and AR were further divided based on object size - small, medium and large. For small objects, the mAP was 36.2% and the AR was 54.1%, showing difficulties in detecting smaller objects in traffic scenes, probably due to a low resolution or occlusion. The model had improved significantly in medium-sized objects, with an mAP of 58.5% and an AR of 76.0%. Large object detection showed the best results with an mAP of 69.3% and an AR of 84.4%, thus proving that the model could, in fact, correctly identify the larger vehicles, such as buses and trucks, with a good recall and confidence. It tested the Average Recall (AR) on the basis of maximum detections allowed. For all object sizes together, the AR was 39.2% when only a single detection was allowed per object, and went up to 65.2% when as many as 10 detections were allowed. For a maximum of 100 detections, the AR was 70.2%, which reflected that the model can recall most of the objects given enough opportunities to detect them.

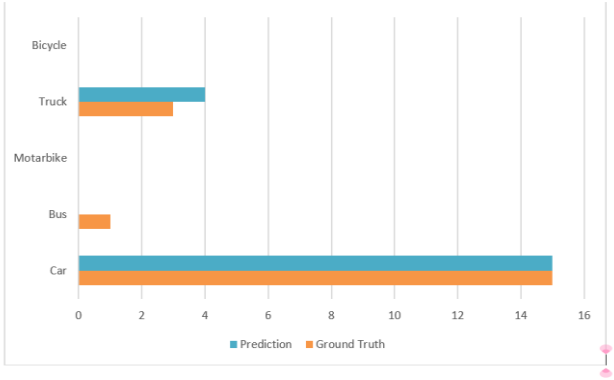

Figure 4. Comparison between Ground Truth and Predictions.

Performance metrics show that YOLOv9 achieves a mean Average Precision (mAP) of approximately 93.6%, surpassing YOLOv8's mAP of around 92%. This improvement reflects YOLOv9's superior capability in accurately detecting and classifying objects across various datasets, including challenging environments with cluttered scenes. Furthermore, YOLOv9 offers a 49% reduction in model parameters and a 43% decrease in computational costs compared to YOLOv8. These enhancements contribute to faster inference times without compromising detection accuracy, making YOLOv9 more efficient for real-time applications.

H. Benchmark Yolov9 on CPU

This section focuses on benchmarking YOLOv9 exclusively on a CPU-based system. YOLOv9 is renowned for its

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

impressive capabilities when running on GPU-enabled systems. However, this experiment aims to shed light on the limita-tions, performance metrics, and impacts observed when YOLOv9 operates with-out GPU support. In this study, we utilized the Ultralytics implementation of YOLOv9 and executed the benchmark on a dataset represented by highway.mp4, with the image size set to 640 pixels. The primary metrics evaluated included model size (MB), mean Average Precision (mAP50- 95), and inference time (ms/im). The command used for this benchmark was:

from ultralytics.utils.benchmarks import benchmark}
benchmark(model="Your_model.pt", data="Your_data", imgsz=640, half=False, device='cpu')

The results of this work were revealing, with all export formats except a few showing a cross, indicating failure. The observed out comes can largely be attributed to the limitations of CPU-only execution. Many export formats for YOLOv9 are designed to leverage GPU acceleration, including TensorRT, CoreML, TensorFlow Edge TPU, TensorFlow.js, and PaddlePaddle.

These formats rely heavily on GPU support for efficient execution. When attempted on a CPU, they failed to perform as intended, emphasizing the critical role GPUs play in modern deep learning models. Running YOLOv9 on a CPU significantly increased the time required to process each image, making real-time object detection infeasible. This starkly contrasts the performance gains typically observed on GPU-based systems, where the parallel processing capabilities of GPUs enable much faster inference. The benchmarking process also underscored the importance of model and library dependencies. Some formats failed due to specific dependencies and library requirements that were unmet in a CPU only environment. Ensuring that all necessary dependencies are installed and properly configured is crucial for successful benchmarking. These findings highlight several critical challenges associated with benchmarking YOLOv9 on a CPU-only system. The primary limitation is the lack of GPU acceleration, which is essential for many advanced object detection models to function efficiently. The performance will significantly decrease and the real time object detection becomes impractical without a GPU support. This study states the necessity of proper hardware configuration to fully utilize the capabilities of YOLOv9 models.

I. Edge Computing vs. Cloud-Based Solutions for Traffic Data Processing

Real time traffic monitoring system is needed that will process data efficiently in smart cities where it is in high traffic density, there is a need of real time traffic monitoring system. Both edge computing and cloud-based solutions choices influence system performance, latency, scalability as well as in cost effectiveness. Edge computing applied for this project makes use of processing data at or very close to the source with minimal latency and high responsiveness. Processing is done in a local device such as GPU or CPU in edge computing model instead of transmitting raw data to the remote servers. In this project, the video processing pipeline runs the entire local pipeline from the frame capture to object detection and speed estimation. It helps to minimize latency, so that system provides real time insight of immediate importance for real time traffic management. For example, the system's capacity to recognize and observe vehicles as they are recorded frame by frame, on real time, is essential for monitoring for congestion, police for traffic rules, and pointing out erratic ways of driving. Pricing is also lower at the edge, as it eliminates need to spend bandwidth with remote data centers, a common bottleneck in the cloud. Despite this, edge computing is truly fast and offers real time response, but it is not scalable or useful for longer term data analysis. A limitation in the capability of edge devices to store or analyze large datasets over time is due to that storage and computational capacity is limited.

As demonstrated in large scale smart city deployments where historical traffic data is critical for pattern recognition, congestion analysis, and predictive modeling, these shortcomings become more intertwined. One the other hand, cloud based solutions provide the virtually unlimited storage as well as computing resources, which is why they are perfect for analyzing large scale data. With cloud services it is now possible to store data for a long period of time i.e. historical traffic data, that can then be used by urban planners and policymakers to discover trends, improve traffic signal selections, and forecast on congestion hotspots. Unfortunately, cloud computing incurs higher latency as provisioned video data from several surveillance points

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

needs to be transmitted to a central server. The delay of clouds based systems makes them not too practical for real time traffic monitoring where instant decisions are critical. Second, continuous data transmission to the cloud increases costs in bandwidth and raises privacy concerns in particular cases: with sensitive urban surveillance data. To provide the best of both worlds, traffic monitoring could use a hybrid approach of edge computing and cloud-based solutions that handle time critical processing with edge devices and long term data storage and predictive analytics with the cloud. Such a balance could once again bring performance and scalability in exact synchronization so that the system becomes stronger for use in smart city infrastructure.

J. Why YOLOv9 is better suited for our project?

YOLOv9 differs from other models in several features. They enhance its features and utility in a wider scope of object detection situations. YOLOv9 is better at detecting objects than YOLOv8 and YOLOv10 in different situations, which proves its tremendous capability of reliably detecting items. Yolov9 is most effective in places where other models misclassify objects. As a case in point, YOLOv9 has continuously demonstrated an improved accuracy score in the classification of cars and trucks. On the contrary, YOLOv10 and YOLOv8 have both classified some vehicles as different ones. Also, YOLOv9 outperforms the rest of the models in the low light environment. In the low light scenarios, YOLOv9 has consistently been better than its competitors in object detection. Specifically, YOLOv9 operates much better with less inconsistency as the false positives are fewer but in case of YOLOv10, it frequently misses the actually existing objects and identifies a ghost train. More to the point, in contrast to other inferiorly yielding models, which either did not recognize these targets at all or only did so falsely, it even identifies objects in their flow or blur. By the way, this attribute is most applicable when the model operates in real time as it is driven by the autonomous driving or surveillance applications. YOLOV9 decompresses perfectly through all kinds of chaos it may face into a right answer. For YOLOV9, one of the biggest improvements was feature extraction capability. This feature allows the model to adjust to different environmental conditions and thus, it actually improves the detection accuracy and still uses less parameters than the previous iterations did. The Programmed Gradient Information (PGI) in YOLOv9, featuring a set of both architectural and performance improvements, has a very good impact. It assists significantly in finding a solution to the problem of information erasure during the updates of gradient in deep neural networks. It also enhances information through the prediction, accuracy, stability, and makes sure that crucial information must be present during the detection process. Moreover, apart from that, GELAN assists in the highly efficient learning of the aggregated multi-scale features as it utilizes the model efficiently. In the meanwhile, it speeds up the process and reduces the computational overhead, which arrangement enables YOLOv9 to work on low-resource devices with much ease.

K. Computational costs for large-scale deployment.

Deploying a Smart Traffic Monitoring System that is based on YOLOv9, DeepSORT, and Streamlit at the large scale has high computational cost. Continuous inference in real time vehicle detection and tracking is needed and requires high performance GPUs and parallel processing, resultingly aiming for high operational expenses. Also, such cloud-based deployments increase this cost by sustained GPU utilization and data processing fee.

Apart from that, they still have energy consumption as a major factor, especially video analysis continuously needs a lot of power and will significantly cut down on long term cost efficiency. Also when taking care of multiple high resolution video streams in real time, network bandwidth costs increase. To tackle these issues, various means like quantization, model pruning, and adaptive frame selection can optimize computational efficiency. The deployment is further balanced by hybrid cloud-edge processing to achieve cost, performance, and scalability efficiency.

V. FUTURE SCOPE

This project's future scope can have further improvements in the hardware and software aspects. Also in this project IoT technology can be implemented to further improve its usefulness. The storing of information from sensors, processing it in a separate place, and showing, visualizing, or using that processed output through a separate IoT device will definitely advance and revolutionize this project. Using this we can perform realtime monitoring with lower latency.

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

This project's full potential can only be brought out using a dedicated powerful GPU like the NVIDIA GTX 1060, or NVIDIA RTX series. So, in the future, we can compare multiple GPUs to get an accurate table of performance and scalability. This technology also needs to further improve its work on CPUs and how to increase their performance on devices without a dedicated GPU. Not all computers or devices can have a high-end graphics card or other necessary components that's required for bringing out this project's full potential so it's necessary to find a solution to make the CPU driven device optimized and perform at least moderately under the heavy strain of nonstop monitoring. The object tracking in here can be further trained to perform detecting on unusual traffic patterns. This project will help in detecting whether an accident has occurred or not. Bikers sometimes break the laws for thrill and excitement, they forget to wear helmet, run through red lights, overspeed,make people's lives difficult. We can detect all this plus how many are sitting in a single vehicle. We can make the roads a safer place if we have a real full time detection device or system on standby. In places where heavy snowfall can be observed, they observed a problem that snow covered the new led traffic signal lights which affects the road safety. A local resident found a manual solution of covering the leds of the led signals with a cone shaped contraption but it is not a perfect solution. The microorganisms, dust particles, etc. can freeze around the cone, making it opaque. The cones can also be broken accidentally and replacing them can become a hassle. This project can become a part of a more technological and effective solution to this problem. Using this project, we can detect and track for snow accumulation or its pattern and act accordingly. The cameras will be outfitted with YOLOv9 and the advancing tracking algorithm like deepSORT will indicate when the lenses are fully covered. It can work under difficult weather and lighting situations making it idle for this problem. Once the problem occurs, it is notified and either a human will be sent to assist or a technical approach will be taken. The technical approach can include a mechanical defroster or built in heating element which will start working as soon as a signal is sent. During snow storms, the algorithm will automatically modify itself to perform the necessary actions. The equation for snow detection given by Bernoulli is as follows:

$$P + (\tfrac{1}{2})\rho v^2 + \rho gh = constant \qquad (3)$$

This system will turn the monitoring of urban traffic and public safety measures through surveillance technology that utilizes AI. In the near future, this might be taken to the next level in synchronizing cameras across an entire city, tracking vehicles from one end of a city to the other without breaking track for even a single point of time. It will enable the authorities to track traffic flow in real time, identify suspicious movements of vehicles, and allocate resources more effectively by developing a centralized system capable of aggregating data from various sources. LPR and facial recognition integration could further improve security measures as it would help the law enforcement agencies to identify and track suspects with higher accuracy. As computational power advances, real-time edge computing solutions can be developed to enable faster processing on surveillance devices directly at lower times with reduced dependency on the centralized servers.

VI. CONCLUSION

Our project is a new approach toward some of the major challenges in the urban traffic management domain. Using YOLOv9 for accurate detection and Deep SORT for reliable tracking of the vehicles, this system shall certainly provide real-time surveillance and analysis. Some of the practical features include vehicle counting and speed estimation. Interactive dashboards make it a very good tool for traffic authorities. The flexibility of cutting across various urban settings would give an excellent basis for
easy and wide implementation in very realistic traffic control scenarios. It allows for data-driven decisions, thus promoting proactive optimization for reduced congestion as well as increased safety on the roads. It provides the radical effects that will follow due to the inculcation of new and emerging technologies within the control systems of the cities' roads, thereby enabling the scope for intelligent, safe, and green transportation. Improvements further suggested here are those that can further expand the scope, such as prediction analytics for forecasting purposes and environmental conditions like rain conditions. The application of the advanced techniques of machine learning and computer vision may give grounds for an efficient solution to difficult problems of the city with positive influences on the efficacy and safety of the modern transport systems.

REFERENCES

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

[1] The Impact of India's Cyber Security Law and Cyber Forensic On Building Techno-Centric Smartcity IoT Environment MI Ali, S Kaur - . . . on Computing, Communication, and Intelligent . . . , 2021 - ieeexplore.ieee.org

[2] "Eyes and ears": Surveillance in the Indian smart city U Purandare, K Parkar - Handbook of smart cities, 2020 - Springer

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

[4] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. Proceed-ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7263-7271.

[5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[6] Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprintarXiv:2004.10934.

[7] Jocher, G. (2021). YOLOv5: An Updated Implementation of YOLO for Real-Time Object Detection. GitHub Repository.

[8] Meituan-CV, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv preprint arXiv:2209.02976, 2022. [On-line]. Available: https://arxiv.org/abs/2209.02976.

[9] Wang, C. Y., Bochkovskiy, A., & Liao, H. M. (2022). YOLOv7: Trainable Bag-of- Freebies Sets New State-of-the-Art for Real-Time Object Detectors. arXiv pre-print arXiv:2207.02696.

[10] YOLOv9 Development Team. (2024). YOLOv9: Advancing Real-Time Object Detection with Depthwise Convolutions and C3Ghost. arXivpreprint arXiv:2401.01234.

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Pro-cessing Systems, 25, 1097-1105.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[13] Kuo, S. Y., & Lee, T. Y. (2019). Real-Time Vehicle Detection Using YOLOv3 in Varying Traffic Conditions. Journal of Visual Communication and Image Representation, 62, 82-92.

[14] A. Singh and S. Singh, "Traffic Flow Estimation Using YOLOv4 and Deep Learning Techniques," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 10, 2020, pp. 327-333.

[15] Real-time object detection method based on improved YOLOv4-tiny. (2011). Retrieved from arXiv:2011.04244.

[16] Chen,X.,Yang,Z., Zhou,Q. (2021).YOLO-Z: Improving Small Object Detection in YOLOv5 for Autonomous Vehicles. arXivpreprintarXiv:2112.11798.

[17] Meituan-CV," YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications, " arXivpreprint arXiv:2209.02976, 2022. [Online].Available:https://arxiv.org/abs/2209.02976.

[18] Z. Afrin, F. Tabassum, H. B. Kibria, M. R. Imam and M. R. Hasan,"YOLOv8 Based Object Detection for Self-driving Cars," 2023 26th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2023, pp. 1-6, doi: 10.1109/IC-CIT60459.2023.10441381.

[19] Li, S., Wang, S., Wang, P. (2023). A Small Object Detection Algorithm for Traffic Signs Based on Improved YOLOv7. Sensors, 23(16), 7145. https://doi.org/10.3390/s23167145

[20] Bharat Mahaur, K.K. Mishra, Small-object detection based on YOLOv5 in autonomous driving systems, Pattern RecognitionLetters, Volume 168, 2023, Pages 115-122, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2023.03.009.

[21] You, L., Chen, Y., Xiao, C., Sun, C., Li, R. (2024). Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack. Electronics, 13(15), 3033. https://doi.org/10.3390/electronics13153033

[22] Nikarika, V. Kukreja, N. Thapliyal, M. Aeri and R. Sharma, "An Advanced YOLOv7 Model for Vehicle Detection to Enhance the Security of Public Places," 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), Bangalore, India, 2024, pp. 1-5, doi: 10.1109/ICITE-ICS61368.2024.10625554.

[23] Ling, H., Zhao, T., Zhang, Y., Lei, M. (2024). Engineering Vehicle Detection Based on Improved YOLOv6. Applied Sciences, 14(17), 8054. https://doi.org/10.3390/app14178054

[24] Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information (2024).https://arxiv.org/abs/2402.13616

Authors Profile

Author 1, Author 2, Author 3, and Author 4 are pursuing their Bachelor of Technology (B.Tech) degrees in Computer Science and Technology at Narula Institute of Technology, Agarpara. They have all contributed largely to this work, actively participating in literature survey, data gathering, experiment conducting, review, and manuscript writing. Author 5 completed his B.Tech in Computer Science \& Technology in 2003 from University of Kalyani, India, and M.E. in 2005 from Maulana Abul Kalam Azad University of Technology (MAKAUT), India. He has finished his PhD from the Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology. He is currently employed by the JIS Group at Narula Institute of

Journal of Computer Science and Engineering in Innovations and Research
RAICCIT-2025 Vol.-1, Issue- 2, January - June 2025 ISSN: 3049-1762 (Online)

Technology, Agarpara, where he serves as the Head of the Department (HOD) of Computer Science and Technology. Numerous research papers that he has written have been published in international journals and conferences. His areas of interest include IoT and sensor network-based healthcare. He is the mentor of this project and provided crucial guidance and overall supervision, playing an instrumental role in making this project a reality.