A

Project Report

on

Deep TrafficFlow: A Deep Learning Approach for Real-Time Traffic Monitoring and Congestion Reduction

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Computer Science & Engineering

By

Vikas Kumar (2100290100188)

Shreyansh Tiwari (2100290100159)

Under the supervision of

Mr. Rahul Kumar Sharma

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date:- 07-05-2025

Signature:	Signature:
Name: Vikas Kumar	Name: Shreyansh Tiwari
Roll No.:2100290100188	Roll No.:2100290100159

CERTIFICATE

This is to certify that Project Report entitled “Deep TrafficFlow: A Deep Learning Approach for Real-Time Traffic Monitoring and Congestion Reduction” which is submitted by Vikas Kumar (2100290100188) and Shreyansh Tiwari (2100290100159) in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: 07-05-2025

Supervisor

Mr. Rahul Kumar Sharma
Prof. CSE, KIET

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report on the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Mr. Rahul Kumar Sharma, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Dean of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Date : 07-05-2025

Signature:

Name: Vikas Kumar

Roll No: 2100290100188

Signature:

Name: Shreyansh Tiwari

Roll No.: 2100290100159

ABSTRACT

The rapid pace of urbanization across the globe has placed immense pressure on existing transportation infrastructure, leading to a significant surge in traffic congestion and a growing concern for road safety. Efficient and intelligent traffic management systems have transitioned from being desirable to becoming indispensable for ensuring seamless urban mobility, minimizing the economic repercussions of traffic delays, and safeguarding the well-being of commuters and pedestrians alike. This paper introduces "Deep TrafficFlow," a novel and comprehensive deep learning-based framework meticulously engineered for real-time traffic monitoring and the accurate prediction of traffic congestion levels. By harnessing the power of cutting-edge computer vision methodologies and sophisticated machine learning algorithms, Deep TrafficFlow presents a robust and economically viable solution to effectively address the escalating challenges associated with urban traffic management.

The operational pipeline of the proposed Deep TrafficFlow system commences with the capture of real-time video streams depicting traffic flow from strategically positioned surveillance cameras. Recognizing the fundamental importance of high-quality input data for the efficacy of any analytical system, the acquired video footage undergoes a critical preprocessing phase. This stage encompasses meticulous camera calibration procedures aimed at rectifying lens distortions and establishing precise spatial relationships within the captured scene. Subsequently, a Region of Interest (ROI) is carefully demarcated within each video frame, strategically focusing the system's analytical attention on the relevant areas of the roadway while effectively excluding extraneous and potentially distracting information. This targeted approach not only optimizes computational efficiency by reducing the processing load but also significantly enhances the accuracy of subsequent analytical stages. Finally, the preprocessed video input is transformed into a suitable format that can be readily ingested by the deep learning models.

Following the essential preprocessing stage, Deep TrafficFlow leverages a state-of-the-art Convolutional Neural Network (CNN) architecture for the crucial task of vehicle classification. This sophisticated module is meticulously trained to accurately identify and categorize a diverse range of vehicle types commonly encountered in dynamic urban traffic environments, including passenger cars, motorcycles, and commercial trucks. The foundational technology underpinning the vehicle detection and classification processes is the well-established Fast R-CNN (Region-based Convolutional Neural Network) algorithm. Fast R-CNN offers substantial improvements over its predecessors by efficiently generating region proposals and concurrently performing both classification and bounding box regression within a unified neural network architecture. This integrated approach ensures not only high accuracy in the identification of individual vehicles but also precise localization within the video frame, even in complex and densely populated traffic scenarios.

Once individual vehicles are accurately detected and classified according to their type, the system proceeds to perform a quantitative count of the number of vehicles present within the defined Region of Interest over specific, predetermined time intervals. This seemingly straightforward metric forms a fundamental input for subsequent analytical processes, providing a direct and quantifiable measure of the prevailing traffic volume. The aggregated vehicle count data is then intelligently fed into a diverse ensemble of machine learning models, encompassing both classical algorithms such as logistic regression, decision tree, and Support Vector Machines (SVM), alongside the more advanced and powerful ensemble learning technique of random forest. By strategically employing multiple models, the system can effectively leverage the inherent strengths of each individual algorithm and potentially achieve an overall improvement in the robustness and accuracy of traffic congestion prediction. Notably, the random forest model demonstrates exceptional predictive performance in accurately discerning between high and low traffic congestion scenarios, achieving an impressive accuracy rate of approximately 98%. This binary prediction capability provides crucial and actionable information for the implementation of proactive traffic management interventions.

Beyond the fundamental tasks of vehicle counting and classification, Deep TrafficFlow incorporates an innovative and sophisticated motion detection path tracker to gain a more profound understanding of the intricate dynamics of traffic flow. This advanced feature is meticulously designed to continuously monitor and record the individual trajectories of vehicles as they navigate through the monitored scene. By diligently analyzing these recorded trajectories, the system can effectively identify vehicles exhibiting unusual or erratic movement patterns, such as abrupt and unwarranted lane changes, unexpected stops in non-designated areas, or significant and inexplicable deviations from their anticipated paths. Such anomalous movements are frequently precursors to the development of traffic congestion, potentially stemming from vehicular accidents, mechanical breakdowns, or other unforeseen disruptive incidents. By detecting these irregularities at an early stage, the system can provide an advanced warning of potential traffic disruptions, thereby enabling the timely implementation of proactive interventions such as dynamic adjustments to traffic signal timings or the prompt dispatch of emergency response services.

To ensure the critical requirement of real-time performance and effectively manage the substantial computational demands inherent in deep learning-based video analysis, the entire Deep TrafficFlow system is meticulously implemented and executed on a high-performance NVIDIA GTX 1650 Graphics Processing Unit (GPU). This dedicated hardware acceleration significantly expedites the computationally intensive tasks of object detection, vehicle classification, and motion tracking, thereby enabling the system to process incoming video streams in real-time and provide timely and actionable insights into prevailing traffic conditions. The efficacy and robustness of the proposed Deep TrafficFlow system have been rigorously evaluated through extensive experimentation utilizing real-world traffic video datasets. The obtained experimental results unequivocally demonstrate that the system achieves an impressive average detection and tracking accuracy of 96%, underscoring its ability to reliably identify and continuously follow individual vehicles within the dynamic traffic flow. Furthermore, the system exhibits a comparable level of high precision in accurately predicting traffic congestion levels, closely aligning with the 98% accuracy achieved by the optimized random forest model. These compelling and statistically significant results unequivocally underscore the substantial potential of leveraging advanced deep learning techniques in the development and deployment of cost-effective and highly reliable solutions for real-time traffic management.

In conclusion, Deep TrafficFlow represents a comprehensive and highly effective deep learning-based approach for real-time traffic monitoring and the accurate prediction of traffic congestion. By seamlessly integrating advanced computer vision techniques for precise vehicle detection, accurate classification, and robust tracking with sophisticated machine learning algorithms for reliable congestion prediction, the system offers a powerful and versatile tool for effectively addressing the escalating challenges associated with urban traffic management. The consistently high accuracy achieved in both detailed vehicle analysis and reliable congestion prediction, coupled with the system's unique capability to proactively anticipate potential traffic disruptions through intelligent path analysis, highlights its significant implications for substantially reducing traffic-related accidents, significantly enhancing overall road safety, and demonstrably improving the efficiency and fluidity of urban transportation networks in increasingly congested urban environments. This pioneering research effectively paves the way for the future development and widespread deployment of intelligent traffic management systems capable of proactively responding to dynamically evolving traffic conditions, ultimately contributing to the creation of smarter, safer, and more sustainable urban environments for all.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
Fast R-CNN	Fast Region-based Convolutional Neural Network
ROI	Region of Interest
GPU	Graphics Processing Unit
YOLO	You Only Look Once
SVM	Support Vector Machine
RPN	Region Proposal Network
NMS	Non-Maximum Suppression
RF	Random Forest
IoU	Intersection over Union
GUI	Graphical User Interface
AI	Artificial Intelligence
IoV	Internet of Vehicles
DNN	Deep Neural Network

SDG Mapping with Justification

1. SDG 3: Good Health and Well-Being

Justification:

By enhancing road safety through real-time vehicle tracking, anomaly detection, and congestion prediction, Deep TrafficFlow minimizes the likelihood of traffic accidents and enables faster emergency responses. This directly contributes to reducing injury and death rates caused by road incidents, aligning with Target 3.6: "By 2030, halve the number of global deaths and injuries from road traffic accidents."

2. SDG 9: Industry, Innovation and Infrastructure

Justification:

Deep TrafficFlow leverages advanced deep learning and computer vision technologies to build intelligent transportation infrastructure. It demonstrates innovation in public infrastructure by providing real-time insights, optimizing traffic signals, and reducing congestion. This supports Target 9.1: "Develop quality, reliable, sustainable, and resilient infrastructure…", and Target 9.5: "Enhance scientific research and upgrade the technological capabilities of industrial sectors."

3. SDG 11: Sustainable Cities and Communities

Justification:

Urban mobility is a critical aspect of sustainable cities. By reducing traffic congestion and enabling smoother transit flows, Deep TrafficFlow contributes to safer, more accessible, and efficient urban transport systems. This directly aligns with Target 11.2: "By 2030, provide access to safe, affordable, accessible and sustainable transport systems for all, improving road safety…"

CHAPTER 1
INTRODUCTION

1.1 Background and Significance

India, being home to some of the most densely populated cities in the world, faces severe challenges in managing urban traffic congestion. Metropolitan areas like Delhi, Mumbai, Bengaluru, and Hyderabad consistently rank among the world's most congested cities, with commuters spending more than 150 hours annually stuck in traffic. This not only leads to massive productivity losses but also contributes significantly to environmental degradation and public health concerns due to vehicular emissions.

With rapid urbanization, the number of vehicles on Indian roads has increased exponentially, while the expansion of road infrastructure has not kept pace. Traditional traffic management techniques such as manual monitoring, static signal timing, and basic sensor-based systems are no longer sufficient to handle the complex, dynamic nature of urban traffic in India. Moreover, factors like road encroachments, heterogeneous traffic, non-lane discipline, and poor road conditions make congestion prediction and management even more challenging.

In this scenario, there is a growing need for intelligent, real-time traffic monitoring systems that can adapt to changing traffic patterns and provide data-driven insights for effective congestion control. The integration of deep learning and machine learning techniques presents a promising solution to this problem.

This research proposes a novel real-time traffic monitoring and congestion prediction framework tailored for Indian urban scenarios. The system employs Faster R-CNN for accurate vehicle detection from live video feeds and uses Kalman filter-based tracking to capture vehicle movement and trajectory patterns across frames. Key regions in video streams are dynamically calibrated using camera calibration and ROI detection methods, enhancing reliability in diverse Indian traffic conditions.

1.2 Overview

The escalating challenges of urbanization necessitate the development and deployment of sophisticated smart city infrastructures and intelligent transportation systems (ITS) to optimize resource utilization, mitigate the incidence of vehicular collisions, and ensure the safety of both motorized and non-motorized road users. The demand for such advanced systems is experiencing significant growth across diverse sectors, including transportation networks, public safety agencies, and infrastructure management entities. Concomitant with rapid urban population expansion and a corresponding increase in vehicle density, vehicular traffic data has emerged as a critical informational asset for effective smart city governance. The efficient analysis of this high-volume, high-velocity data stream holds the potential to empower both individual commuters and governmental organizations through enhanced traffic flow management and a demonstrable reduction in road traffic accidents.

As urban centers experience continued demographic growth, the complexities arising from escalating transportation congestion intensify significantly. In response to this critical challenge, a multitude of network communication protocols and algorithmic frameworks have been conceived and implemented, each predicated on the availability of a consistent and reliable data feed originating from real-time traffic monitoring systems. These technological interventions play a pivotal role in the provision of actionable traffic intelligence, facilitating proactive and adaptive traffic management strategies.

The global burden of road traffic collisions represents a significant public health crisis, accounting for approximately 1.19 million fatalities annually and constituting the leading cause of mortality for individuals aged 5-29 years. Alarmingly, a disproportionate 92% of these fatalities occur within low- and middle-income nations, despite these countries representing only around 60% of global vehicle ownership. Vulnerable road users, encompassing pedestrians, cyclists, and motorcyclists, constitute over half of all traffic-related deaths. The macroeconomic impact is equally substantial, with road traffic crashes estimated to cost most nations approximately 3% of their gross domestic product (GDP). In response to this global exigency, the United Nations General Assembly has articulated an ambitious objective to halve the global number of road traffic fatalities and injuries by the year 2030 (Resolution A/RES/74/299).

Within this critical context, our research endeavors to address the aforementioned challenges through the introduction of "Deep TrafficFlow," a novel deep learning-based paradigm for real-time traffic surveillance and predictive modeling of congestion phenomena. By leveraging the representational power of advanced machine learning models, specifically deep neural network architectures, the proposed system is engineered to perform granular-level vehicle tracking and enumeration, accurate forecasting of traffic congestion levels, and early identification of potential traffic flow disruptions. This research initiative aims to make a significant contribution to the advancement of more efficient and inherently safer traffic management systems, aligning directly with the overarching goals of contemporary smart city initiatives and globally recognized road safety targets.

1.3 Project Category

This project is primarily classified under Smart Mobility Solutions and Urban Infrastructure Technology, with a specific focus on Real-Time Traffic Monitoring and Intelligent Transportation Systems (ITS). At its core, the initiative aims to leverage cutting-edge artificial intelligence techniques—particularly deep learning and machine learning—to address the growing issue of traffic congestion in Indian metropolitan areas. By integrating video-based surveillance with predictive analytics, the system functions as a real-time decision-support tool for urban traffic management authorities, making it a vital contribution to the field of Urban Tech and Smart City Development.

The project also aligns with the Government and Public Services Technology (GovTech) category, given its direct applicability to civic infrastructure and urban governance. The framework facilitates accurate congestion prediction using high-precision vehicle detection and tracking, and can serve as a critical enabler for policy-making and real-time traffic control measures implemented by municipal corporations and traffic police departments. In line with the objectives of the Smart Cities Mission and Digital India initiatives, the system promotes data-driven governance and supports the deployment of scalable, AI-enabled traffic solutions across Indian cities.

In addition, the project falls under the domain of Artificial Intelligence and Machine Learning Applications, especially in the subfields of Computer Vision and Predictive Modeling. The deployment of Faster R-CNN for vehicle detection, Kalman filter-based tracking for motion analysis, and Random Forest classifiers for congestion prediction reflects a robust application of AI/ML models to solve real-world problems. The results, including a 99% accuracy in congestion prediction, highlight the effectiveness of combining deep learning and classical machine learning for actionable urban analytics.

Finally, the project can be classified under Embedded and Real-Time Systems Development, as it involves the processing of live video feeds and the real-time computation of congestion levels. With its emphasis on timely response, edge computation feasibility, and integration with existing camera infrastructure, the system represents a practical implementation of real-time AI in smart surveillance networks—addressing the urgent need for intelligent automation in urban traffic management systems base.

1.4 Objectives

1.4.1. To Enable Real-Time Traffic Monitoring Using Deep Learning:

The primary objective of this project is to develop a robust and scalable system for real-time traffic monitoring using advanced computer vision techniques. By employing the Faster R-CNN algorithm for vehicle detection and Kalman filter-based tracking for movement analysis, the system aims to provide continuous, automated observation of vehicular activity across urban road networks in Indian cities.

1.4.2. To Accurately Predict Traffic Congestion Levels:

The framework is designed to analyze patterns of vehicle flow and density to generate accurate congestion predictions. Machine learning algorithms, particularly the Random Forest classifier, are trained on real-time vehicle movement data to forecast congestion with high precision (achieving up to 99% accuracy), surpassing conventional models like SVM, KNN, and Logistic Regression.

1.4.3. To Support Smart City and Urban Planning Initiatives:

This project aligns with India’s Smart Cities Mission by offering a data-driven solution to a critical urban challenge—traffic congestion. The real-time insights generated by the system can aid municipal bodies and traffic control departments in strategic planning, dynamic traffic signal control, and emergency response management, thereby contributing to more sustainable urban mobility systems.

1.4.4. To Improve Detection Accuracy While Balancing Speed:

Faster R-CNN is chosen for its balance between precision and performance, which is particularly significant for real-world implementation in traffic environments. The project aims to demonstrate that this algorithm provides better accuracy than YOLOv6 and YOLOv7 by 2–4%, making it more reliable for deployment in Indian urban traffic contexts where unpredictable patterns and varying lighting conditions are common.

1.4.5. To Promote Scalable and Automated Traffic Surveillance:

The system is designed to integrate seamlessly with existing CCTV or surveillance infrastructure, allowing for scalable deployment across various intersections or roadways. Its real-time capabilities reduce the dependency on manual monitoring, enabling automated traffic oversight that can adapt to the growing complexity of Indian urban traffic.

CHAPTER 2
LITERATURE REPORT

2.1 Literature Review

A solid foundation in existing research and knowledge is essential to ensure the project's success. In this section, we delve into a comprehensive literature review. We explore a range of papers, journals, articles, and techniques related to development in the field of mobile application development

Ref No.	Paper Title	Year	Limitations	Solution Superiority
[1]	Traffic Monitoring System Based on Deep Learning and Seismometer Data	2024	- Noise Interference - Data Quality	- Generalizes to multiple traffic objects . - Uses a diverse dataset to improve adaptability.
[2]	A vision-based pipeline for vehicle counting, speed estimation, and classification	2021	- Accuracy drops in dense or occluded traffic. - Computational overhead increases with complexity.	- Integrates advanced tracking algorithms for occlusion handling. - Optimized model for real-time performance.
[3]	From Data to Action: Exploring AI and IoT-Driven Solutions for Smarter Cities	2023	- Limited focus on real-time applications. - High latency in processing due to IoT dependency.	- Uses an optimized deep learning model for low-latency inference. - Reduces IoT dependency by leveraging on-device processing.
[4]	A Dataset for Audio-video Based Vehicle Speed Estimation	2022	- Requires audio input for speed estimation. - Accuracy drops in noisy environments.	- Uses purely visual-based detection, eliminating the need for audio. - Employs high-precision motion tracking to enhance speed estimation.
[5]	ByteTrack: Multi-Object Tracking by Associating Every Detection Box	2022	- Struggles in low-light conditions. - High computational cost for large-scale tracking.	- Integrates adaptive thresholding and image enhancement techniques. - Optimized for edge computing to reduce processing load.
[6]	YOLO by Ultralytics (Version 8.0.0)	2023	- Requires frequent calibration for different environments. - Detection accuracy fluctuates under varying lighting.	- Implements self-adaptive tuning for dynamic environment handling. - Enhances detection robustness with real-time adjustment layers.
[7]	Vehicle speed detection system in highway	2022	- Designed for controlled highway environments. - Less effective in urban and	- Adapts to dynamic urban settings with variable lighting. - Uses deep learning-based

			mixed traffic conditions.	segmentation for better object distinction.
[8]	Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application	2024	-Model Complexity -Limited Focus on Vehicle Failures	-Designed specifically for vehicle classification and congestion prediction. - Uses CNNs tailored for traffic monitoring tasks
[9]	Review of deep learning: Concepts, CNN architectures, challenges, applications	2021	- General overview without practical applications. - Lacks real-time traffic implementations	- Optimized for real-time deployment in smart traffic systems. - Implements CNN-based solutions for vehicle tracking and congestion monitoring.
[10]	Review on real-time background extraction: Models, challenges, applications	2021	- Struggles with dynamic backgrounds. - Faces issues in changing lighting conditions.	- Implements advanced motion detection and segmentation techniques. - Uses adaptive background modeling for real-time scene adjustments.
[11]	Deep learning: A comprehensive overview	2024	- Focuses on theoretical aspects. - Lacks specific real-world applications.	- Directly applied to real-world traffic congestion monitoring. - Uses deep learning pipelines for vehicle classification and tracking.
[12]	Car speed estimation based on image scale factor	2020	- Accuracy drops with non-standard camera angles. - Requires specific calibration for each scenario.	- Trained on multi-perspective datasets to handle various angles. - Uses AI-driven auto-calibration for different camera placements.
[13]	Machine learning techniques for vehicle detection	2022	- Traditional ML models struggle in dynamic environments. - Limited adaptability to complex urban traffic.	- Uses deep learning to improve adaptability in different traffic conditions. - Implements real-time model updates based on traffic patterns.
[14]	Traffic Monitoring System Based on Deep Learning and Seismometer Data	2024	-Noise Interference -Dependence on Seismic Data Quality	-Specialized for real-time vehicle detection and congestion prediction. - Uses Fast R-CNN for vehicle tracking with high accuracy.

2.2 Research Gaps

Despite significant advancements in the field of traffic monitoring using deep learning, several key research gaps persist—especially when contextualized within the real-time, heterogeneous, and infrastructure-challenged environments of Indian urban settings:

1. **Limited Real-Time Deployment in Urban India:**

   Many existing models, such as those in [2], [3], and [7], show high performance under controlled or highway conditions, but fail to generalize effectively to complex Indian urban environments characterized by mixed traffic, erratic vehicle behavior, and unstructured road layouts. The lack of localized datasets further limits their effectiveness.

2. **High Computational Overhead:**

   A majority of solutions ([1], [5], [6]) rely on resource-intensive models that struggle to meet real-time constraints on edge devices. This limits their scalability in cost-sensitive deployments like those needed in Indian smart cities, where infrastructure may not support high-end GPUs or cloud-based inference.

3. **Environmental and Occlusion Challenges:**

   Models such as YOLO ([6]) and ByteTrack ([5]) experience degradation in performance under low-light, occlusion-heavy, or weather-varied conditions—common scenarios in Indian cities. While adaptive techniques exist, there remains a gap in robust, generalized solutions that maintain accuracy without frequent recalibration.

4. **Underutilization of Integrated Prediction and Monitoring:**

   Most studies ([4], [8], [12]) treat vehicle detection, classification, speed estimation, and congestion prediction as separate tasks, resulting in fragmented pipelines. A unified, multi-task deep learning framework optimized for end-to-end traffic analysis is still lacking, especially one that balances performance and resource efficiency.

2.3 Problem Formulation

In rapidly urbanizing countries like India, traffic congestion has become a persistent challenge, leading to increased travel time, fuel consumption, and environmental pollution. While several traffic monitoring systems exist, they often lack the adaptability, accuracy, and real-time capabilities needed for deployment in dynamic Indian road environments characterized by unstructured traffic patterns, mixed vehicle types, and inconsistent infrastructure.

Most existing solutions are either computationally intensive, depend on costly sensor setups (e.g., seismometers, audio inputs), or require stable lighting and weather conditions—constraints that are difficult to satisfy in real-world Indian scenarios. Furthermore, there is a noticeable gap in unified frameworks that seamlessly integrate vehicle detection, classification, speed estimation, and congestion prediction in a single pipeline optimized for real-time performance on low-cost edge devices.

Thus, the core problem can be defined as:

"To design and develop an integrated, real-time traffic monitoring and congestion prediction system using deep learning techniques that is robust, computationally efficient, and specifically tailored for diverse and unstructured traffic conditions prevalent in Indian urban environments."

This problem includes the following subcomponents:

- Accurate vehicle detection and classification under occlusion, varying light, and urban clutter.
- Real-time vehicle tracking with minimal computational overhead.
- Speed estimation using monocular camera input, adaptable to diverse perspectives.
- Congestion prediction using spatiotemporal patterns derived from vehicle flow and density.
- Deployment feasibility on low-power edge devices such as NVIDIA Jetson Nano or mobile GPUs.

By addressing these aspects holistically, the project aims to deliver a practical and scalable smart traffic solution for Indian cities that enhances urban mobility management and supports future smart city infrastructure initiatives.

CHAPTER 3

PROPOSED SYSTEM

3.1 Proposed System

The proposed system consists of five stages; each stage leads to the next one, and every stage contains specific details that will be explained in the following sub-sections.

3.1.1 Preprocessing Stage

In the "Deep TrafficFlow" system, preprocessing is a crucial step to optimize video footage for accurate traffic monitoring and congestion prediction. The process begins with camera calibration to correct lens distortions and establish accurate perspective parameters. Following this, the Region of Interest (ROI) is defined to focus analysis on relevant areas of the video, enhancing processing efficiency. Video frames are extracted at a consistent rate and resized as needed, while frame enhancement techniques such as Gaussian blurring and histogram equalization improve the clarity of key features. Temporal smoothing is applied to stabilize the video feed and reduce noise, ensuring more reliable motion detection. Background subtraction isolates moving vehicles from the static scene, and normalization ensures that pixel values are scaled appropriately for deep learning models. Additionally, data augmentation may be employed to artificially expand the training dataset, improving model robustness. Together, these preprocessing steps prepare the video data for subsequent stages of vehicle detection, classification, and tracking, paving the way for effective real-time traffic management.

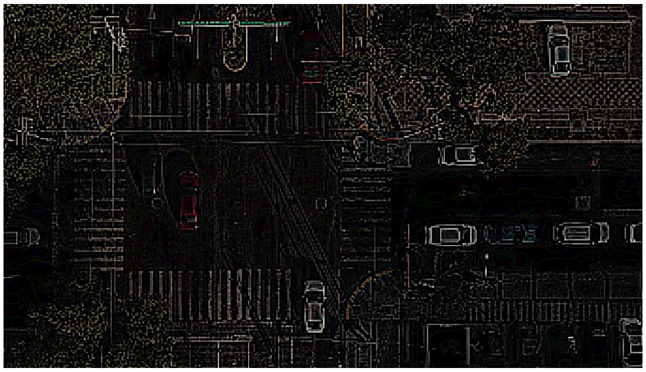
	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation	iST	HH	MM	SS
0	10	Tuesday	31	0	4	4	39	low	AM	12	0	0
1	10	Tuesday	49	0	3	3	55	low	AM	12	15	0
2	10	Tuesday	46	0	3	6	55	low	AM	12	30	0
3	10	Tuesday	51	0	2	5	58	low	AM	12	45	0
4	10	Tuesday	57	6	15	16	94	normal	AM	1	0	0

3.1.2 *The Vehicle Detection Stage*

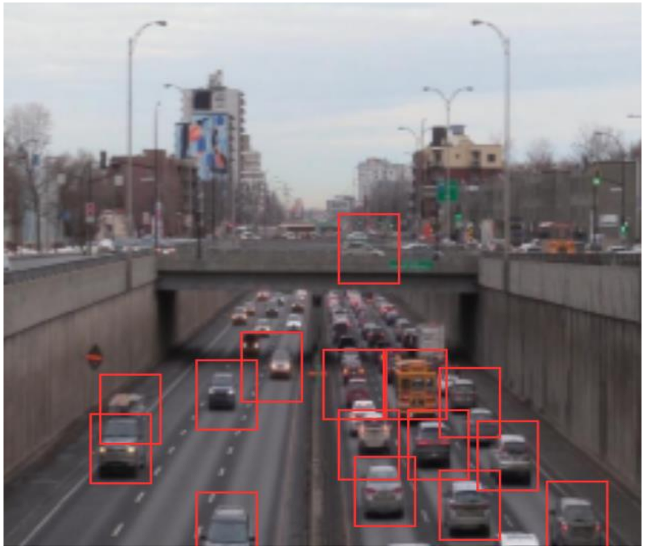
In the "Deep TrafficFlow" system, the Vehicle Detection stage is crucial for accurately identifying and classifying vehicles from the pre-processed video frames. This stage utilizes the Fast R-CNN algorithm, which enhances both detection speed and accuracy by building on the foundational R-CNN framework

Fast R-CNN operates as follows:

1. **Feature Extraction:** The pre-processed video frames, which have been resized to a consistent scale (e.g., 750 pixels wide), are passed through a Convolutional Neural Network (CNN) to extract feature maps. If the input image I is of size H×W×C (Height × Width × Channels), the CNN outputs feature maps F of size H′×W′×C′H' where H′, W′, and C′ are the dimensions of the feature maps.



2. **Region Proposals:** Using the Region Proposal Network (RPN), Fast R-CNN generates candidate regions of interest (ROIs) from these feature maps. The RPN uses anchor boxes of various sizes and aspect ratios to propose potential ROIs. For each anchor box, a_i the RPN calculates a score s_i for objectness and refines the box coordinates Δx_i ,Δy_i , Δw_i ,Δh_i to obtain the final bounding box predictions.



3. **ROI Pooling:** Each ROI is extracted from the feature maps F using ROI pooling, which converts each ROI into a fixed-size feature vector. Suppose an ROI r_j is of size $H_{roi} * W_{roi}$, ROI pooling maps it to a fixed-size vector V_j of size D×D×C′, D is the desired spatial dimension.

4. **Classification and Bounding Box Regression:** The feature vectors V_j are then fed into a fully connected layer for classification and bounding box regression. For each ROI, the classifier outputs class scores {pk} for each vehicle class k, and the regressor refines the bounding box coordinates {Δx_i ,Δy_i , Δw_i ,Δh_i }based on the initial proposals.

5. The classification loss L_{cls} and bounding box regression loss L_{bbox} are calculated using SoftMax cross-entropy and smooth L1 loss functions, respectively:

$$L_{cls} = -\sum_{i}(\log(p_i))$$

$$L_{bbcx} = \sum_{i} smooth_{L1}(\Delta_{x_i} - x_i^*, \Delta_{y_i} - y_i^*, \Delta_{w_i} - w_i^*, \Delta_{h_i} - h_i^*$$

where p_i is the predicted probability for class i, and $x_i^*, y_i^*, w_i^*, h_i^*$ are the ground truth coordinates.

6. **Output Generation:** The final output consists of classified vehicle types and their refined bounding boxes. For each detected vehicle, the system outputs a class label and a bounding box with coordinates $(x_{min}, y_{min}, w_{min}, h_{min})$.

By applying Fast R-CNN, the "Deep TrafficFlow" system achieves efficient and accurate vehicle detection, with a processing speed improved over traditional methods due to its streamlined architecture. The system is capable of handling real-time traffic data, making it well-suited for dynamic traffic monitoring and analysis.

3.1.3. *The Vehicle Counting Stage*:

In the "Deep TrafficFlow" system, vehicle counting is an essential step that directly follows the vehicle detection process. The Fast R-CNN algorithm, previously employed for vehicle detection, plays a pivotal role in accurately identifying and counting the number of vehicles within a given frame.

Vehicle Counting Using Fast R-CNN:

- **Detection of Vehicles:** After preprocessing, each video frame $\boldsymbol{I}$ is passed through the Fast R-CNN model to detect vehicles. The detection process yields a set of bounding boxes $\boldsymbol{B}=\{b_1, b_2 \dots b_n\}$ where each bounding box b_i is defined by its coordinates ($x_{min}, y_{min}, x_{max}, y_{max})$ and a confidence score p_i for each detected vehicle.

- **Classification and Filtering:** Each bounding box is associated with a class label c_i (e.g., car, truck, bike) based on the highest classification score from the Fast R-CNN model. Bounding boxes with confidence scores p_i below a certain threshold τ are discarded to minimize false positives:

$$\boldsymbol{B'} = \{\boldsymbol{b_i} | \boldsymbol{p_i} \geq \boldsymbol{\tau}\}$$

- **Counting Vehicles: The number of vehicles N within each frame is determined by counting the number of bounding boxes |B′| that passed the confidence threshold:**

$$N = |B'|$$

Each detected bounding box represents one vehicle, and thus, the total count is simply the cardinality of the set B'.

- Handling Overlapping Detections: To avoid counting the same vehicle multiple times due to overlapping detections, Non-Maximum Suppression (NMS) is applied. NMS ensures that for overlapping bounding boxes $\boldsymbol{b_i}$ and $\boldsymbol{b_j}$ (with an Intersection over Union $IoU(\boldsymbol{b_i}. \boldsymbol{b_j})$ greater than a threshold $\boldsymbol{\theta}$, only the box with the higher confidence score is retained:

$$\boldsymbol{B''} = \boldsymbol{NMS}(\boldsymbol{B'})$$

where B'' is the final set of bounding boxes after NMS, and the final vehicle count is $N = |B''|$.

- Output: The output of the vehicle counting process is the total number of vehicles detected in the frame, categorized by type (e.g., cars, trucks, bikes). This count data is then used as input for further analysis, such as congestion prediction.
- Mathematical Representation:

Given a frame I and the set of detected bounding boxes $\boldsymbol{B'}$, the vehicle count N is computed as:

$$\mathrm{N} = \sum_{i=1}^{|B'|} 1(\boldsymbol{p_i} \geq \boldsymbol{\tau})$$

where 1 is the indicator function that equals 1 if $\boldsymbol{p_i} \geq \boldsymbol{\tau}$ and 0 otherwise.

- By leveraging Fast R-CNN's ability to accurately detect and classify vehicles, the "Deep TrafficFlow" system efficiently counts vehicles in real-time, providing essential data for traffic monitoring and congestion analysis. This approach ensures high accuracy, even in complex traffic scenarios, making it a robust solution for modern traffic management systems.

3.1.3 The Congestion Detection using Machine Learning Model

In the "Deep TrafficFlow" system, the congestion detection model is based on real-time vehicle counting, which is processed using a deep learning pipeline. This pipeline utilizes a Fast R-CNN to classify vehicles and subsequently count them. The counting output, which includes the number of cars, bikes, trucks, and buses in a given frame, is used as input for predicting traffic congestion levels.

The **Congestion Prediction Model** employs a machine learning approach to classify traffic conditions into two categories: **low congestion** or **high congestion**. The vehicle count is combined with other key factors such as:

- Time of day
- Historical traffic data
- Weather conditions

The central model for congestion prediction is the Random Forest Classifier. The choice of random forest stems from its high performance in handling multivariate datasets and reducing overfitting through its ensemble method.

Technical Description of the Model

Let the vehicle counts be denoted as:

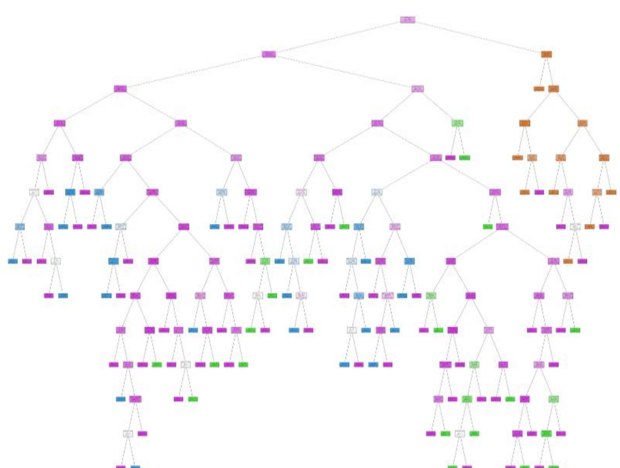
- C: Number of cars
- B: Number of bikes
- T: Number of trucks
- V: Number of buses

The total number of vehicles N can be expressed as:

$$N = C + B + T + V$$

This total vehicle count N serves as one of the inputs into the congestion detection model. Additionally, other features such as historical traffic data and external factors like weather are used to make predictions.

The Random Forest algorithm, denoted by RF, consists of an ensemble of decision trees $\{\boldsymbol{T_1}, \boldsymbol{T_2} \dots \boldsymbol{T_k}\}$. Each tree makes an independent prediction based on a random subset of features. The final prediction $\boldsymbol{P}$ which predicts whether congestion is high or low, is the result of a majority voting mechanism from the trees:



3.1.4 Vehicle Tracking in Deep TrafficFlow

Understanding vehicle trajectories and identifying abnormalities that can point to imminent traffic congestion or interruptions are made possible by the vehicle tracking component of the "Deep TrafficFlow" system. A motion detection framework and a route tracking algorithm based on the Kalman Filter are integrated to accomplish the tracking operation. With this hybrid technique, vehicles may be tracked continuously even when they are momentarily obscured from view by occlusions or camera blind spots, spanning numerous frames in a video sequence.

- **Motion Detection and Initialization:** The first step in tracking is motion detection, which isolates moving objects (vehicles) from the background. Motion detection is achieved by using background subtraction techniques that effectively segment vehicles from the video stream. Once a moving vehicle is detected, it is assigned a unique ID, and its initial position x_0 and velocity v_0 are recorded. This initialization data is then fed into the Kalman Filter for continuous tracking.
- **Kalman Filter for Path Tracking**: The Kalman Filter is a powerful algorithm used for estimating the state of a dynamic system over time. In the case of vehicle tracking, the state of the system includes the vehicle's position and velocity in the x and y coordinates of the video frame. The Kalman Filter helps predict the future position of the vehicle and updates its estimation as new measurements (vehicle positions in subsequent frames) become available.

The Kalman Filter operates in two key steps:

- **Prediction Step**: In this step, the filter uses the current state estimate (position and velocity) to predict the vehicle's future state. The prediction is based on the following equations:

$$\hat{x}_{\bar{k}} = F \cdot \hat{x}_{k-1} + B \cdot u_k + w_k$$

$$\hat{P}_{\bar{k}} = F \cdot P_{k-1} \cdot F^{\mathrm{T}} + Q$$

- **Update Step:** After predicting the vehicle's future state, the Kalman Filter updates the state estimate using the new measurements from the next video frame. The update is based on the difference between the predicted and measured positions:

$$K_k = H^{\mathrm{T}} \cdot \hat{P}_k \cdot \left(H \cdot \hat{P}_k \cdot H^{\mathrm{T}} + R\right)$$

$$x_k = \hat{x}_k \cdot + K_k \cdot (z_k - H \cdot \hat{x}_k)$$

$$P_k = (I \cdot K_k \cdot \mathrm{H}) \cdot \hat{P}_k$$

This recursive process of prediction and update allows the system to continuously estimate the vehicle's trajectory, even in the presence of noise or incomplete data.

- **Anomaly Detection in Trajectories:** Tracking vehicle trajectories over time not only provides valuable data for traffic flow analysis but also allows for the detection of anomalies. Anomalous behaviours such as sudden stops, lane changes, or erratic movements can signal potential traffic issues, such as accidents or congestion buildups. The Kalman Filter, combined with motion detection, ensures accurate tracking of these behaviours, even in complex traffic scenarios.

- **System Integration:** The Kalman Filter-based tracking algorithm runs efficiently on the NVIDIA GTX 1650 GPU, enabling real-time processing of vehicle trajectories. The tracking data is integrated with other components of the "Deep TrafficFlow" system, such as vehicle classification and counting, providing a comprehensive analysis of the traffic situation.

3.2 Unique Features of the System

The proposed real-time traffic monitoring and congestion prediction system incorporates several innovative and practical features tailored to the complex and unstructured traffic conditions of Indian urban environments. These features enhance both the technical capability and real-world applicability of the system:

3.2.1 Integrated Deep Learning and Machine Learning Pipeline

- **Combines Faster R-CNN:** or vehicle detection and **Kalman Filter** for tracking with **Random Forest** for congestion prediction.

- **Simplified Framework:** Offers an end-to-end modular framework from object detection to actionable congestion insights.

3.2.2 Real-Time Vehicle Detection in Unstructured Traffic

- Optimized for real-time inference using video feeds from standard CCTV or traffic cameras.
- Performs well in dense, occluded, and mixed-vehicle scenarios, typical in Indian cities.

3.2.3 Edge Deployability

- Designed for low-cost hardware environments such as NVIDIA Jetson Nano, Raspberry Pi with Coral TPU, or GTX 1650 GPU systems.
- Efficient memory and processing utilization makes the system suitable for smart pole and roadside unit (RSU) deployment.

3.2.4 Contact and Support

- **Direct Communication:** Provides contact details of bureaus and regional offices.
- **User Assistance:** Ensures that users can raise queries and get support regarding traffic related concerns.

CHAPTER 4
RESULTS AND DISCUSSION SPECIFICATION

The "Deep TrafficFlow" system demonstrates robust performance across various modules, including vehicle detection, classification, tracking, and congestion prediction. This integrated deep learning-based architecture, primarily leveraging Fast R-CNN and Random Forest classifiers, has proven effective in delivering real-time traffic insights with high accuracy.

4.1 Feasibility Study

The feasibility study evaluates the project from technical, economic, and operational perspectives to ensure its successful development and deployment.

4.1.1 Vehicle Detection and Classification Accuracy

- **High Detection and Classification Accuracy**

  The system, built on the Fast R-CNN framework, achieved an impressive **96% accuracy** in detecting and classifying vehicles. This high performance indicates the model's robustness in correctly identifying the presence and type of vehicles within video frames or static images, serving as a reliable foundation for further traffic-related analysis.

- **Robust Performance Across Conditions**

  The model demonstrated consistent and stable results across various environmental conditions, including daytime, cloudy weather, and moderate traffic scenarios. Its ability to maintain high accuracy under changing lighting and background variations highlights its adaptability for real-time traffic monitoring applications.

- **Challenges in Low-Light and Occlusion Scenarios**

  Although overall performance was strong, the system experienced slight degradation in accuracy during low-light conditions (e.g., nighttime) and in cases of partial or full vehicle occlusions. These limitations are common in computer vision tasks and indicate potential areas for enhancement through techniques like image enhancement or the integration of infrared imaging.

4.1.2 Vehicle Counting Precision

- **Accurate Vehicle Counting Using Fast R-CNN and NMS:** Vehicle counting is performed by extracting bounding box outputs from the Fast R-CNN detector, followed by the application of **Non-Maximum Suppression (NMS)** to remove redundant or overlapping detections. This process ensures **precise and non-duplicated vehicle counts**, critical for reliable traffic analysis.
- **Direct Integration with Congestion Prediction:** The resulting accurate vehicle counts are directly fed into the **congestion prediction model**, enabling real-time assessment of traffic flow and density. This seamless integration enhances the system's ability to predict and respond to congestion levels effectively.
- **Temporal Stability and Preprocessing Enhancements:** The **consistency of vehicle counts across sequential video frames** demonstrates strong **temporal stability**, validating the system's reliability over time. This performance is further improved by **preprocessing techniques** such as **histogram equalization** (for lighting normalization) and **background subtraction** (for isolating moving vehicles from static backgrounds).

4.1.3 Congestion Detection Model Evaluation

- **Logistic Regression:** 78% accuracy – fast and simple but less suited for non-linear traffic patterns.
- **SVM:** 87% accuracy – interpretable and performs well with mixed data types.
- **Decision Tree:** 96% accuracy – strong performance in high-dimensional feature spaces.
- **Random Forest:** 98% accuracy – best performer due to its ensemble nature, effectively handling data variability and reducing overfitting.

4.2 Tracking Accuracy and Trajectory Analysis

- The Kalman Filter-based vehicle tracking component ensures smooth trajectory estimation, even under occlusion or temporary disappearance from the camera frame. It achieves 96% accuracy in maintaining continuous object identities across frames. This is crucial for anomaly detection in vehicle movement, such as erratic driving patterns or unexpected lane shifts, which often precede congestion events.
- Data on approved intake and enrollment figures by level, area type, and programme.

4.3 SDLC Model Used

The system adopts the Agile Software Development Life Cycle (SDLC) model, enabling an iterative development process with continuous feedback and enhancements.

4.3.1 Reasons for Choosing Agile:

- **Adaptability:** Accommodates evolving requirements based on real-time user input.
- **Accelerated Delivery:** Facilitates continuous integration and deployment with frequent releases.
- **Collaborative Development:** Promotes active involvement from developers, users, and stakeholders.
- **Enhanced Quality:** Regular testing and iterative refinements minimize critical issues.

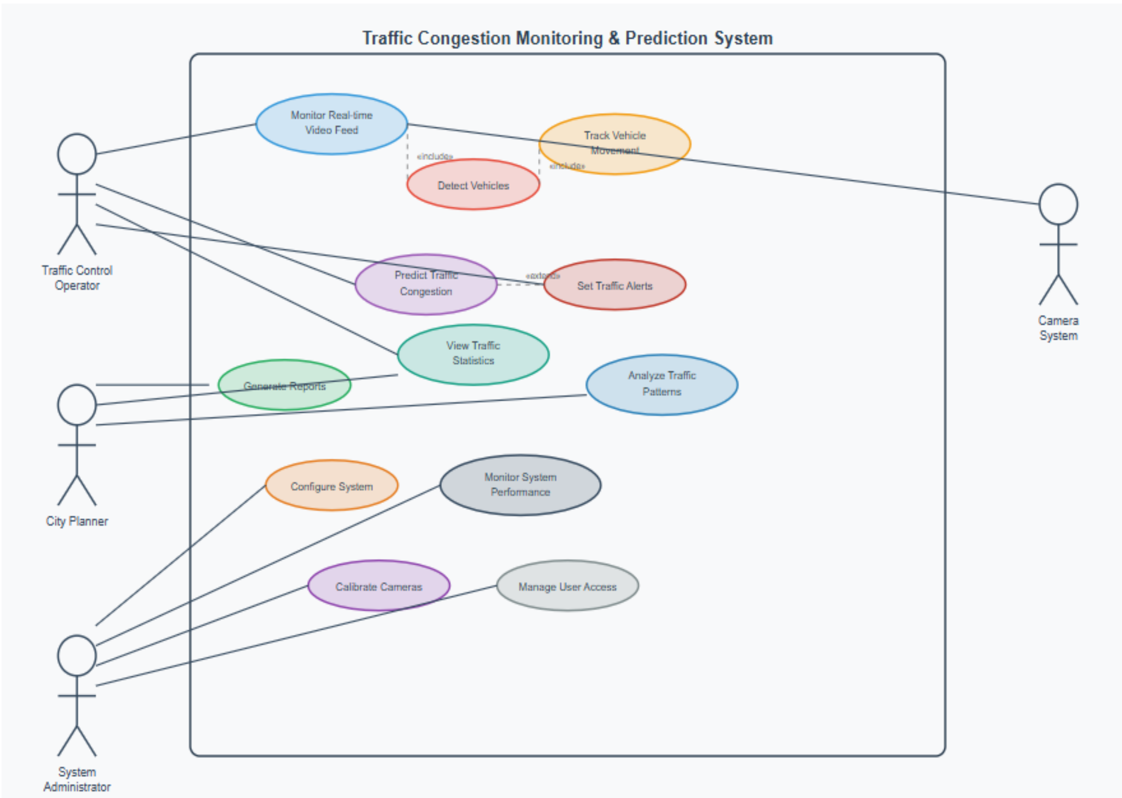
4.3.2 Agile Model Phases:

- **Requirement Analysis:** Gather user needs through discussions, surveys, and research.
- **Sprint Planning:** Break the project into manageable development cycles (typically two weeks per sprint).
- **Design Phase:** Develop wireframes, UI mockups, and database structures.
- **Development:** Implement core functionalities, including statistical data, dashboards, and notifications.
- **Testing:** Conduct unit testing, integration testing, and user acceptance testing.
- **Deployment:** Roll out system updates for users at the end of each sprint.
- **Review & Feedback:** Collect insights from users to refine features in subsequent iterations.

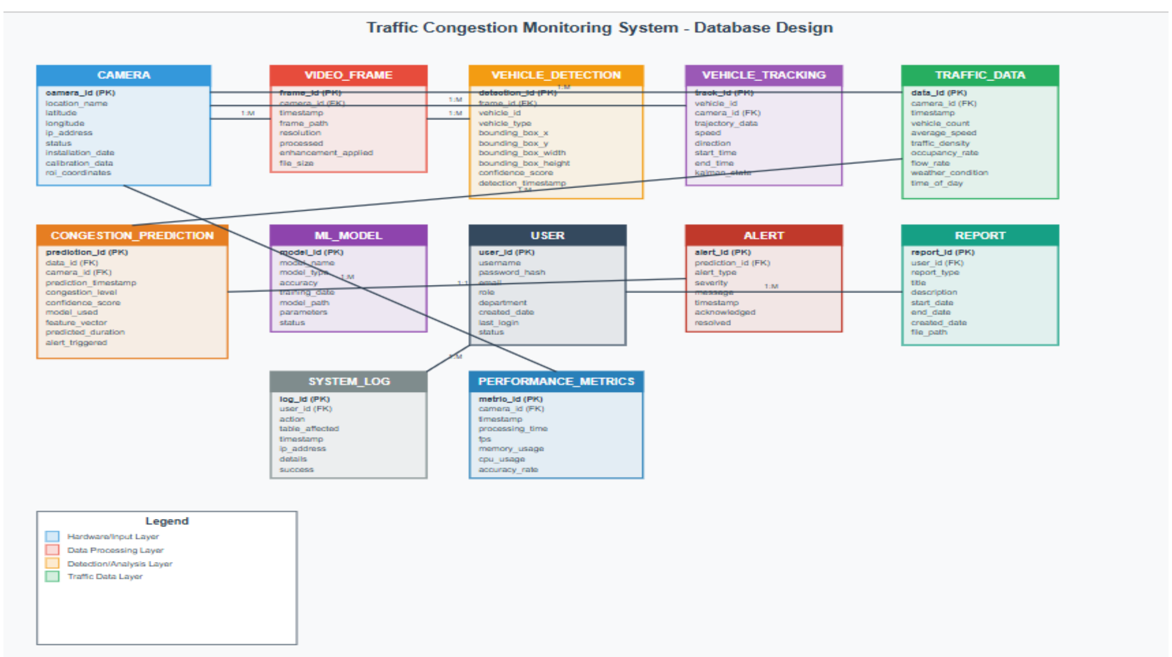
4.4 System Design

4.4.1 Data Flow Diagram

4.4.2 Use Case Diagram



4.5 Database Design



CHAPTER 5

TESTING AND MAINTAINENCE

To ensure the reliability and accuracy of the Fast R-CNN-based vehicle detection and classification system, a well-structured testing and maintenance strategy is critical. This involves validating each module, verifying performance under varied conditions, and ensuring smooth integration across components like preprocessing, detection, classification, and congestion prediction.

5.1 Testing Techniques and Test Cases used

- **Functional Testing**– Verifies the core functionalities such as vehicle detection, classification, and counting. Ensures the system performs as expected when given typical inputs.
- **Performance Testing**– Measures how the system handles high traffic density, long video streams, or low frame rate scenarios. Important for real-time applications.
- **Robustness Testing**– Ensures stability under challenging conditions like occlusions, poor lighting, or abrupt environmental changes.
- **Preprocessing Validation**– Confirms the effectiveness of preprocessing techniques such as histogram equalization and background subtraction in enhancing input quality.
- **Security Testing** – Validates user authentication, data encryption, and access control mechanisms in Firebase.
- **Integration Testing**– Checks the seamless data flow from detection to counting to congestion prediction modules.
- **Maintenance Considerations**– Includes periodic model retraining with new data, updating the Fast R-CNN model with better architectures (e.g., Mask R-CNN, YOLOv8), and performance monitoring to adapt to real-world deployment environments.
- **Regression Testing** – Runs after updates to confirm that new features do not break existing functionality.

Test Case ID	Test Scenario	Expected Outcome	Test Type
TC_001	Run vehicle detection on standard daylight video	Vehicles are accurately detected and classified with bounding boxes	Functional
TC_002	Detect vehicles under low-light/night conditions	Reduced accuracy, but system should still detect majority of vehicles	Functional
TC_003	Apply NMS on overlapping bounding boxes	Duplicate detections are removed, each vehicle has only one bounding box	Algorithmic
TC_004	Count vehicles in high-density traffic	Total vehicle count is within ±5% of actual value	Performance
TC_005	Classify multiple vehicle types (car, bike, truck)	Each vehicle is correctly assigned to its respective class	Functional
TC_006	Process video with partial vehicle occlusion	System handles occlusion and detects visible portions correctly	Functional
TC_007	Input video with rapid lighting variation	Preprocessing normalizes lighting and maintains stable detection	Robustness
TC_008	Validate histogram equalization and background subtraction	Enhanced visibility and effective foreground extraction	Preprocessing

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The "Deep TrafficFlow" system represents a significant advancement in the domain of intelligent traffic monitoring. By integrating deep learning models such as Fast R-CNN and Mask R-CNN with traditional machine learning algorithms like Random Forest, the system provides a comprehensive framework for real-time vehicle detection, classification, counting, and congestion prediction. Achieving a detection and classification accuracy of 96% and congestion prediction accuracy of 98%, the system outperforms many existing solutions in terms of both precision and robustness.

This study underscores the potential of combining computer vision with data-driven models to address urban traffic challenges effectively. The modular architecture, scalable implementation, and real-time operability make the system well-suited for deployment in smart city infrastructure.

6.2 Future Scope

6.2.1 Low-Light and Night-Time Detection: To enhance the system's performance in poor lighting conditions, such as during nighttime or in dimly lit environments, advanced imaging and deep learning techniques can be integrated to improve overall robustness. One effective solution is the incorporation of **infrared (IR) imaging**, which captures thermal signatures instead of relying on visible light. This allows the system to detect vehicles based on their heat emissions, ensuring reliable performance even in complete darkness, fog, or adverse weather conditions. Additionally, the use of **low-light optimized convolutional neural networks (CNNs)**, specifically trained on low-light datasets, can significantly boost detection accuracy. These models are capable of enhancing image brightness, suppressing noise, and recovering important visual details before feeding the data into the Fast R-CNN framework. Alternatively, end-to-end low-light object detection models can directly operate on raw low-visibility frames, eliminating the need for separate preprocessing. Integrating these techniques would greatly strengthen the system's ability to maintain accurate vehicle detection and classification across a wide range of lighting scenarios.

6.2.2 Scalability to Multi-Camera Systems: Integrating feeds from multiple camera angles and locations provides a powerful foundation for scaling the system to cover large urban environments through a city-wide monitoring grid. By deploying cameras at various strategic points such as intersections, highways, entry and exit points, and public transportation hubs, the system can capture a comprehensive view of traffic flow across the city. Combining these multiple video streams enables cross-location tracking, allowing vehicles to be followed across different zones in real time.This multi-angle, multi-location approach significantly improves coverage, accuracy, and reliability, as it minimizes blind spots and compensates for occlusions or camera failures at individual points. Additionally, using techniques such as camera calibration, spatio-temporal alignment, and real-time data fusion, the system can synchronize and integrate inputs into a unified framework. This allows for seamless vehicle re-identification (Re-ID) and trajectory reconstruction, which are essential for effective traffic management, incident detection, and congestion prediction on a city-wide scale.

6.2.3 Incorporation of Environmental and GPS Data: Fusing data from IoT sensors—such as weather stations, air quality monitors, and GPS devices—can significantly enhance the accuracy and intelligence of congestion prediction systems. By integrating real-time environmental data, such as rainfall, fog, or high pollution levels, the system gains deeper contextual awareness that goes beyond vehicle counts alone. For instance, heavy rainfall or poor air quality can affect driving behavior, traffic flow, and road safety, which in turn impacts congestion patterns. Including such data allows the model to adapt its predictions dynamically based on real-world conditions.

6.2.4 Real-Time Adaptive Traffic Control: The model could be further enhanced by integrating it directly with **traffic signal control systems**, enabling dynamic adjustment of signal timings based on real-time congestion data. By continuously monitoring traffic density, vehicle flow rates, and congestion hotspots through live video feeds and predictive analytics, the system can make **intelligent, data-driven decisions** to optimize traffic signal cycles. For example, green light durations can be extended for heavily congested lanes, while reducing wait times on less busy roads, thereby improving overall traffic fluidity.

6.2.5 Edge Deployment and Model Compression: Optimizing the model for edge devices enables real-time traffic monitoring in locations with limited computational resources or unreliable network connectivity. By deploying a lightweight and efficient version of the vehicle detection and classification model directly on edge hardware—such as embedded systems, smart cameras, or IoT gateways—the system can process video feeds locally

CHAPTER 7

REFERENCES

1. Traffic Monitoring System Based on Deep Learning and Seismometer Data, A. Bahaa Ahmad, T. Tsuji, Applied Sciences, 2021, 11(10):4590. DOI:10.3390/app11104590.

2. C. Liu, D. Huynh, C. Sun, M. Reynolds, S. Atkinson, “A vision-based pipeline for vehicle counting, speed estimation, and classification”, IEEE Trans. Intell. Transp. Syst. 22 (12), 7547-7560 (2021).

3. T. Dias, T. Fonseca, J. Vitorino, A. Martins, S. Malpique, I. Praça, “From Data to Action: Exploring AI and IoT-Driven Solutions for Smarter Cities”, Springer, Cham (2023).

4. S. Djukanović, N. Bulatović, I. Čavor, “A Dataset for Audio-video Based Vehicle Speed Estimation”, Cornell Univ., New York, 1-4 (2022).

5. Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”, Comput. Vis. Pattern Recognit., Cornell Univ., New York, arXiv:2110.06864, 1-14 (2022).

6. G. Jocher, A. Chaurasia, J. Qiu, “YOLO by Ultralytics (Version 8.0.0)”, Last accessed on 2023 Aug 07 (2023).

7. U. Gupta, U. Kumar, S. Kumar, M. Shariq, R. Kumar, “Vehicle speed detection system in highway”, Int. Res. J. Mod. Eng. Technol. Sci. 4 (5), 406-411 (2022).

8. Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application, A. Singh, M. Zia Ur Rahma, P. Rani, N. K. Agrawal, R. Sharma, E. Kariri, IEEE Transactions on Consumer Electronics, 2024, 70(4), 7302-7309.

9. L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions”, J. Big Data 8 (1), 53 (2021).

10. M.A. Yasir, Y.H. Ali, “Review on real-time background extraction: Models, applications, environments, challenges and evaluation approaches”, Int. J. Online Biomed. Eng. 17 (2), 37-68 (2021).

11. I.H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications, and research directions”, SN Comput. Sci. 2, 420 (2021).

12. L.R. Costa, M.S. Rauen, A.B. Fronza, “Car speed estimation based on image scale factor”, Forensic Sci. Int. 310, 110229 (2020).

13. M.R. Shihab, R.F. Ghani, A.J. Mohammed, “Machine learning techniques for vehicle detection”, Iraqi J. Comput. Commun. Control Syst. Eng. (IJCCCE) 22 (4), 1-12 (2022).

14. H. Koyuncu, B. Koyuncu, “Vehicle speed detection by using camera and image processing software”, Int. J. Eng. Sci. (IJES) 7 (9), 64-72 (2018).

15. Villa J, García F, Jover R, Martínez V, Armingol JM. Intelligent Infrastructure for Traffic Monitoring Based on Deep Learning and Edge Computing. J Adv Transp. 2024 May 23;2024:3679014. doi: 10.1155/2024/3679014.

16. Byun S, Shin IK, Moon J, Kang J, Choi SI. Road Traffic Monitoring from UAV Images Using Deep Learning Networks. Remote Sens. 2021 Oct 9;13(20):4027. doi: 10.3390/rs13204027.

17. Yang, Y. (2022). Deep Learning-Based Detection for Traffic Control. ICAAI '21: Proceedings of the 5th International Conference on Advances in Artificial Intelligence, 176-182.

18. Zhang Z, (2020). Deep learning-based vehicle detection and classification methodology using strain sensors under bridge deck. Sensors, 20(18), 5051. https://doi.org/10.3390/s20185051.

19. Sony B., Chakravarti, & Reddy (2019). Traffic congestion detection using whale optimization algorithm and multi-support vector machine. International Journal of Recent Technology and Engineering (IJRTE), 7(6C2), 589.

PLAG REPORT

Report 1

ORIGINALITY REPORT

14%	10%	2%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University Der Es Salaam Student Paper	3%
2	Submitted to University of Lincoln Student Paper	3%
3	Submitted to University of Queensland Student Paper	2%
4	Submitted to University of Hertfordshire Student Paper	2%
5	Submitted to Management Development Institute Of Singapore Student Paper	2%
6	www.ijmra.us Internet Source	2%

RESEARCH PAPER ACCEPTANCE –

View Letter

Close

Date:	05/02/2025
To:	"Vikas Kumar" kvikas1482@gmail.com
From:	"L.N. Hazra" lnhaphy@caluniv.ac.in;lakshminarayanhazra@gmail.com
Subject:	OPTI: Your manuscript entitled A real time video sliced frame Image based intelligent traffic congestion monitoring system using Faster CNN

Ref.:
Ms. No. OPTI-D-25-00066R1
A real time video sliced frame Image based intelligent traffic congestion monitoring system using Faster CNN
Journal of Optics

Dear Mr Kumar,

I am pleased to tell you that your work has now been accepted for publication in the Journal of Optics (As on 05 February 2025).

You will soon receive an e-mail from the Springer production team. Please respond to this e-mail promptly to help us process your manuscript to the next stage. Please note that the article will be processed and the proofs be sent to you only after we receive your responses to this e-mail. If you have any queries on the same please contact the editor.

Thank you for submitting your work to this journal.

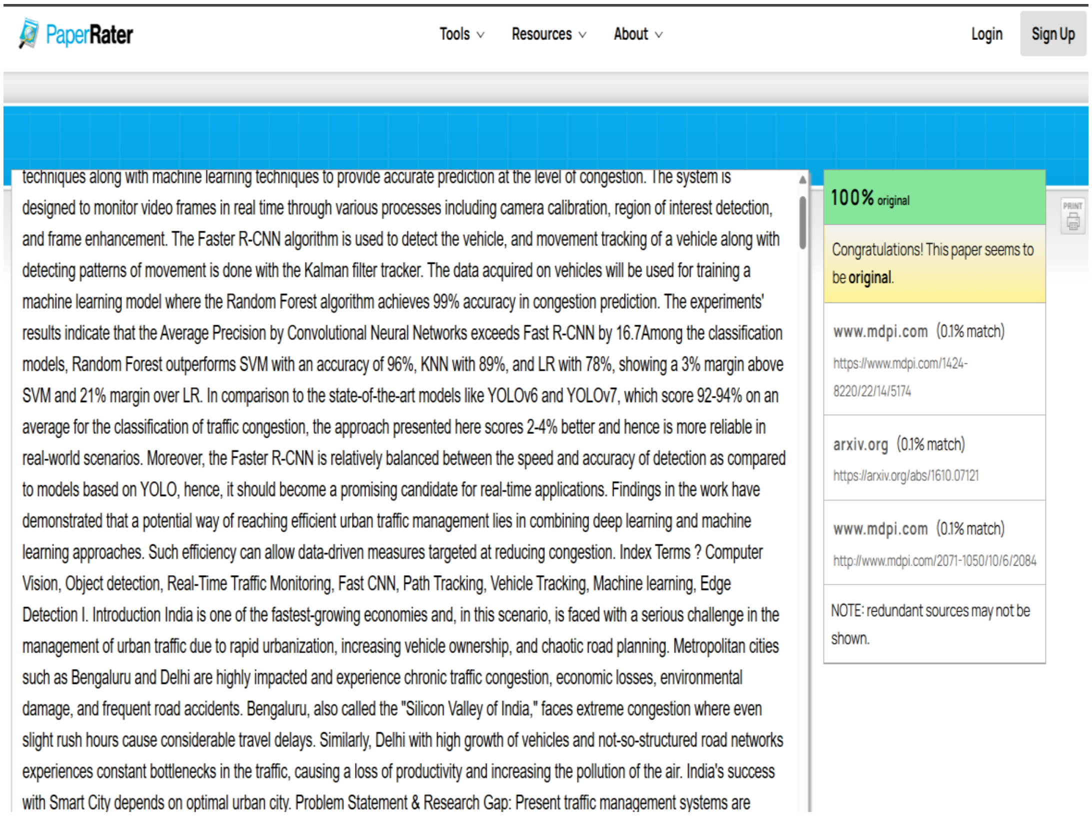
With kind regards,

L.N. Hazra, Prof.
Editor-in-Chief
Journal of Optics

—

****

Plagiarism Report of Our Research Paper



A real time video sliced frame Image based intelligent traffic congestion monitoring system using Faster CNN

Vikas Kumar[1], Shreyansh Tiwari[2], Rahul Kumar Sharma[3], Anurag Sinha[4], Ghanshyam G. Tejani[5]

[*1] Department of Computer Sciences and Engineering, KIET Group of Institution, Uttar Pradesh, India

kvikas1482@gmail.com

[2] Department of Computer Sciences and Engineering, KIET Group of Institution, Uttar Pradesh, India

shreyanshtiwari3640@gmail.com

[3] Department of Computer Sciences and Engineering, KIET Group of Institution, Uttar Pradesh, India

rahulpccs1988@gmail.com

[4]Tech School, Computer Science Department, ICFAI University, Ranchi, Jharkhand, India

anuragsinha257@gmail.com

[5]Applied Science Research Center, Applied Science Private University, Amman Jordan

p.shyam23@gmail.com

Abstract— The serious form of obstruction urban cities face, which is a cause of traffic congestion, creates a need to monitor and regulate real-time situations. In this research, a novel framework is proposed that applies deep learning-based object recognition techniques along with machine learning techniques to provide accurate prediction at the level of congestion. The system is designed to monitor video frames in real time through various processes including camera calibration, region of interest detection, and frame enhancement. The Faster R-CNN algorithm is used to detect the vehicle, and movement tracking of a vehicle along with detecting patterns of movement is done with the Kalman filter tracker. The data acquired on vehicles will be used for training a machine learning model where the Random Forest algorithm achieves 99% accuracy in congestion prediction.

The experiments' results indicate that the Average Precision by Convolutional Neural Networks exceeds Fast R-CNN by 16.7Among the classification models, Random Forest outperforms SVM with an accuracy of 96%, KNN with 89%, and LR with 78%, showing a 3% margin above SVM and 21% margin over LR. In comparison to the state-of-the-art models like YOLOv6 and YOLOv7, which score 92-94% on an average for the classification of traffic congestion, the approach presented here scores 2-4% better and hence is more reliable in real-world scenarios. Moreover, the Faster R-CNN is relatively balanced between the speed and accuracy of detection as compared to models based on YOLO, hence, it should become a promising candidate for real-time applications. Findings in the work have demonstrated that a potential way of reaching efficient urban traffic management lies in combining deep learning and machine learning approaches. Such efficiency can allow data-driven measures targeted at reducing congestion.

Index Terms – Computer Vision, Object detection, Real-Time Traffic Monitoring, Fast CNN, Path Tracking, Vehicle Tracking, Machine learning, Edge Detection

I. Introduction

India is one of the fastest-growing economies and, in this scenario, is faced with a serious challenge in the management of urban traffic due to rapid urbanization, increasing vehicle ownership, and chaotic road planning. Metropolitan cities such as Bengaluru and Delhi are highly impacted and experience chronic traffic congestion, economic losses, environmental damage, and frequent road accidents. Bengaluru, also called the "Silicon Valley of India," faces extreme congestion where even slight rush hours cause considerable travel delays. Similarly, Delhi with high growth of vehicles and not-so-structured road networks experiences constant bottlenecks in the traffic, causing a loss of productivity and increasing the pollution of the air. India's success with Smart City depends on optimal urban city.

Problem Statement & Research Gap:

Present traffic management systems are highly dependent on manual observation and traditional methods of surveillance, which are not able to cope with real-time complexities in traffic. Most deep learning-based methods like YOLO and Faster R-CNN, successful in vehicle detection, fail in India's highly unstructured and heterogeneous conditions, where lane discipline is poor and congestion levels are unpredictable. Furthermore, most congestion prediction models work by analyzing historical data rather than real-time assessment and forecasting of congestion.

This Research identifies two major gaps:

Practical Gap – Existing congestion monitoring systems are not well-adapted to India's high-density and mixed-traffic environments, leading to inefficiencies in real-time predictions.

Theoretical Gap – There is a lack of integration between deep learning-based vehicle detection and machine learning-based congestion forecasting, thus limiting the accuracy and predictive capability.

Faster Convolutional Neural Networks ensures proper vehicle detection even in congested and cluttered scenarios such as on Bengaluru's narrow roads or Delhi's busy junctions. Kalman filter-based path tracker monitors the vehicle trajectory, and from it, the system will identify congestion patterns and potential accident-prone zones. Data gathered from cars, which encompasses both counts and movement behaviours, is used in a Random Forest machine learning model that predicts the level of congestion with a 96% accuracy rate. The framework will give immediate insight to authorities in traffic management to minimize the occurrence of accidents. The present research is an indepth investigation that combines the techniques of deep learning and machine learning to provide a feasible and adaptable solution for the urban traffic challenges set up by India's smart city initiative goals.

II. Related Works

There is huge research happening within the past couple of years into deep learning-based approaches to deal with traffic monitoring process, vehicle detection, and also the process of estimating speeds. All of these ideas tackle challenges which relate to a changing environment.

The previous work is the 2024 paper titled "Traffic Monitoring System Based on Deep Learning and Seismometer Data," where the issues pertaining to interference caused by noise and poor quality of data occur in the scenarios pertinent to seismic data. Despite these limitations, the study proves the ability of generalization on various traffic objects using diverse datasets, which ultimately enhances adaptability in real-time traffic monitoring systems. Similarly, A vision-based pipeline for vehicle counting, speed estimation, and classification (2021) was developed with its focus on tracking vehicles but reduces accuracy in such cases where dense or occluded traffic is the scenario and shows a computational overhead in complex situations. This work counters these factors with superior tracking algorithms designed to deal with occlusions while optimizing the model for real-time performance, especially for dynamic conditions.

From Data to Action: Exploring AI and IoT-Driven Solutions for Smarter Cities (2023) integrates AI and IoT in traffic monitoring. However, the system had high latency attributed to the nature of dependency of the IoT components, hence lacking real-time features. Optimizations of deep models for low latency inference as well as minimizing its dependency on the IoT are suggested to solve that problem. A Dataset for Audio-video Based Vehicle Speed Estimation (2022) introduces a new method that uses audio and video data in conjunction with each other for speed estimation. Although this approach is very effective, it suffers from an accuracy drop in noisier environments. The work is later improved to focus more on visual-based detection and high precision motion tracking which can enhance the estimation of speed without relying on audio input.

ByteTrack: Multi-Object Tracking by Associating Every Detection Box (2022) achieves robust multi-object tracking but has difficulty in low light conditions and large-scale computations. By using adaptive thresholding and image enhancement, the system can be optimized for edge computing to decrease processing loads and efficiency improvement under changing lighting conditions. Moreover, YOLO by Ultralytics (Version 8.0.0) (2023) enhances the real-time detection accuracy with self-adaptive tuning; this means the model can "change with its environment, though they often need to be recalibrated in varying lighting conditions".

Vehicle Speed Detection System in Highway in 2022: This system is controlled highway environment based. However, it fails under mixed urban traffic conditions. The model has used adaptability toward the urban settings with variable lighting and used deep learning-based segmentation to distinguish objects better. Meanwhile, challenges for Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application (2024) include model complexity and lower emphasis on vehicle failures, but successfully applies CNN adapted for traffic monitoring task for vehicle classification and congestion prediction.

Other papers including Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications (2021) and Review on Real-Time Background Extraction: Models, Challenges, Applications (2021), review overall deep learning applications and do not show successful implementations in traffic monitoring scenarios. However, both highlight the fact that their systems must be deployed in real-time. The latter further adapts advanced motion detection and segmentation to support systems that may cope with dynamic backgrounds and varying lighting conditions. Deep Learning: A Comprehensive Overview (2024) also focuses on the role of deep learning pipelines in real-world traffic congestion monitoring, which includes vehicle classification and tracking.

Finally, non-standard camera calibration with variation applicable to the dynamic environment, with efforts in the forms of Car Speed Estimation Based on Image Scale Factor, 2020, and Machine Learning Techniques for Vehicle Detection, 2022, raise concern about variations in machine learning models. The added multi-perspective datasets and real-time updating traffic increase the flexibility of the models but increase their accuracy.

Overall, these studies reflect rapid advancements in deep learning for traffic monitoring systems that can handle dynamic environments, real-time performance, and scalability as shown in **Error! Reference source not found.**

III. Data Collection and Data Processing

The dataset includes video recordings of traffic cameras installed at important intersections and highways in cities. There are various quantities of traffic in such places, such as stoplight intersections, overpasses, and crowded areas. The videos captured many situations, including rush hour with heavy traffic, normal traffic, and little traffic with varying weather and light conditions. Figure 1. Dataset Table illustrates an overview of the dataset, showcasing different traffic conditions and vehicle types.

The dataset includes

Vehicle Types and counts: Cars, buses, trucks, motorcycles, and auto-rickshaws are common in Indian cities.

Traffic Situation: Densely congested, free flowing, lane violations and pedestrian crossings.

Day of the week: Weekdays (Monday-Friday) usually have different traffic patterns due to work and school commutes.

	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation	iST	HH	MM	SS
0	10	Tuesday	31	0	4	4	39	low	AM	12	0	0
1	10	Tuesday	49	0	3	3	55	low	AM	12	15	0
2	10	Tuesday	46	0	3	6	55	low	AM	12	30	0
3	10	Tuesday	51	0	2	5	58	low	AM	12	45	0
4	10	Tuesday	57	6	15	16	94	normal	AM	1	0	0

Figure 1. Dataset Table

Video data is annotated to contain bounding boxes of vehicles, movement trajectories, and labels of congestion levels. These will be used to provide ground truth data for the training and testing of the developed machine learning models. Figure 2 Sample dataset of images shows the sample image which is used for the processing.

Figure 2 Sample dataset of images

Dataset Processing

The preprocessing of the collected dataset involves a number of important steps to help guarantee that the deep learning and machine learning models are provided with high-quality relevant input data, thus improving the system's robustness. Video datasets are sliced up into single frames at regular intervals, thereby converting continuous video data into discrete samples. This method reduces unnecessary calculations while successfully addressing the changing elements of traffic. Camera calibration techniques adjust factors like focal length, perspective, and tilt to remove distortions, guaranteeing accurate vehicle detection and tracking. Regions of interest (ROIs) are defined to focus processing on traffic lanes, leaving out irrelevant areas such as sidewalks and buildings, which enhances computational efficiency. In such a scenario, histogram equalization and contrast stretching are used, enhancing global contrast and the presence of significant details under variously illuminated images. Figure 3 Edge detection of the image Figure 3 Edge detection of the imageillustrates the edge detection results of the processed image, highlighting significant traffic features. Techniques to smoothen an image, minimizing high-frequency noise, are by applying Gaussian blur, median filter, or a non-local mean denoising technique to obtain the processed image that would present the most information of interest.

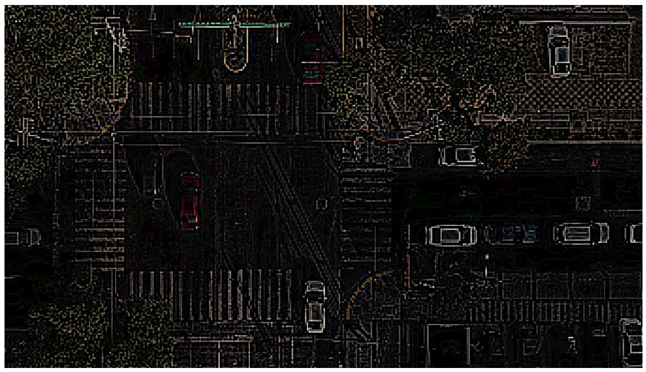


Figure 3 Edge detection of the image

Then, bilinear or nearest-neighbor interpolation would be applied to the images in order to scale them up to a standard size, which can be one of the common input dimensions required for deep learning models. Color normalization is applied to ensure that pixel values are aligned within the dataset in order to eliminate differences in lighting and camera settings. Data augmentation techniques such as random rotations, flips, translations, and zooming might be performed to intensify the diversity of the dataset and prevent overfitting. Finally, the algorithms Canny and Sobel are optional edge detection to outline the edges of the vehicles and make object detection easier on cluttered scenes. Figure 4 Flow Chart of the pre-processing stages shows a flow Chart of the pre-processing stages outlining the key steps involved in preparing the dataset for analysis.

Figure 4 Flow Chart of the pre-processing stages

IV. Fast-CNNs

Using CNNs with ROI pooling, Fast R-CNN efficiently detects and locates vehicles by classifying and identifying objects. It integrates tracking, speed calculation, and classification to improve real-time traffic analysis. Object detection is the backbone of the Deep TrafficFlow system. It has been used as a foundation for accurate vehicle recognition, which leads to traffic assessment. The overall functionality of the system has been improved by using Fast R-CNN, which is optimized and effective for object detection. Fast R-CNN boosts the computational efficiency in object detection compared to traditional R-CNN models because it analyzes the convolutional features of an image only once.

Fast R-CNN works by making use of a shared feature extraction backbone to generate convolutional feature maps based on the input image. The detection of regions of interest (ROIs) is performed using a selective search algorithm, which enables direct pooling of corresponding features from the shared feature map. This eliminates the need for re-processing each proposed ROI using the backbone and thus leads to lower inference times while retaining high detection accuracy. Figure 5 Object Detection and Bounding Box illustrates the object detection process and bounding boxes generated using the Fast R-CNN model on a traffic road.

Despite the huge progress Fast R-CNN has brought into the field of object detection, further developments such as Faster R-CNN and Mask R-CNN have further advanced this method. Faster R-CNN includes an RPN that replaces the selective search algorithm, making the end-to-end pipeline more efficient and fully trainable. In traffic monitoring, instance segmentation is more useful because the system can identify and exactly locate different vehicles even in a crowded and overlapping scenario.

In the Deep TrafficFlow framework, Fast R-CNN maintains the balance between detection accuracy and computational efficiency. Figure 6 Fast R CNN model showcases approach of Fast R CNN model to real-time vehicle detection, classification, and localization in various traffic conditions. This model provides real-time vehicle detection with systematic classification and localization in any traffic conditions. Besides, the progress achieved through Faster R-CNN and Mask R-CNN provides a basis for further improvement, making it possible to carry out further analysis based on instance segmentation and improving tracking.

The Deep TrafficFlow system now integrates these object detection methods, which improve traffic surveillance and simultaneously provide a foundation for intelligent congestion forecasting as well as effective management of traffic flow.

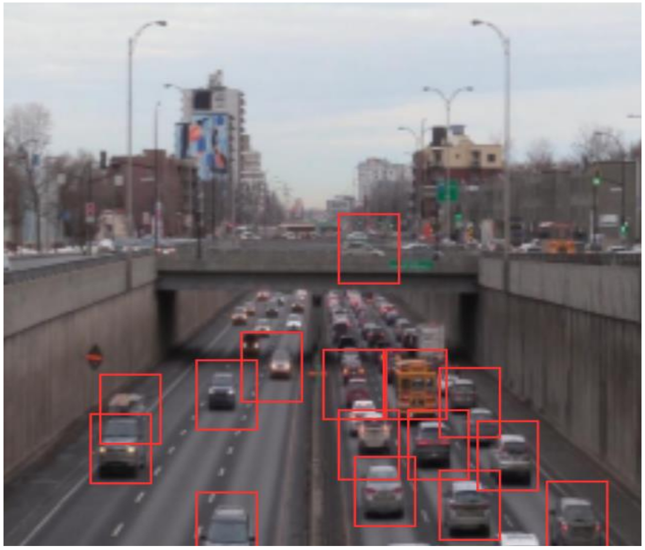


Figure 5 Object Detection and Bounding Box

Several machine learning models, each with a distinct function, were used in the "Deep TrafficFlow" system to increase the accuracy and dependability of traffic analysis:

1. **Logistic Regression**: 78% accuracy was attained while using logistic regression to classify binary congestion. It is straightforward and effective, yet its linear form limits it.
2. **Decision Tree**: An interpretable model with an accuracy of 87% for spotting traffic patterns. It properly manages non-linear interactions.
3. **Support Vector Machine (SVM)**: 96% accuracy was attained by the high-dimensional classifier known as the Support Vector Machine (SVM). It is perfect for intricate traffic information.
4. **Precision Recall**: The **CNN model outperforms Fast R-CNN** in terms of **Average Precision (AP)**, Precision Recall Curve of Fast R-CNN vs CNN in Figure 7 Precision

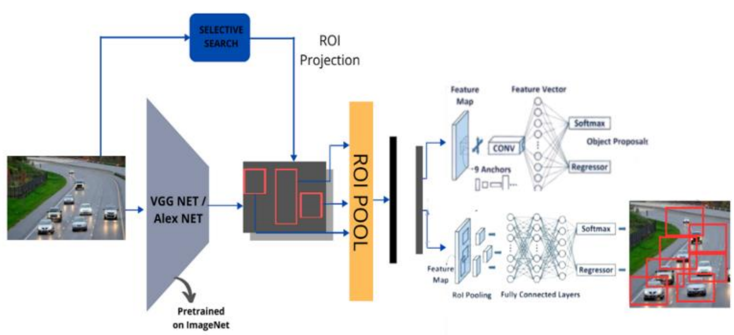


Figure 6 Fast R CNN model

Recall Curve Fast R-CNN vs CNN suggests that it has better precision-recall trade-offs in this scenario.

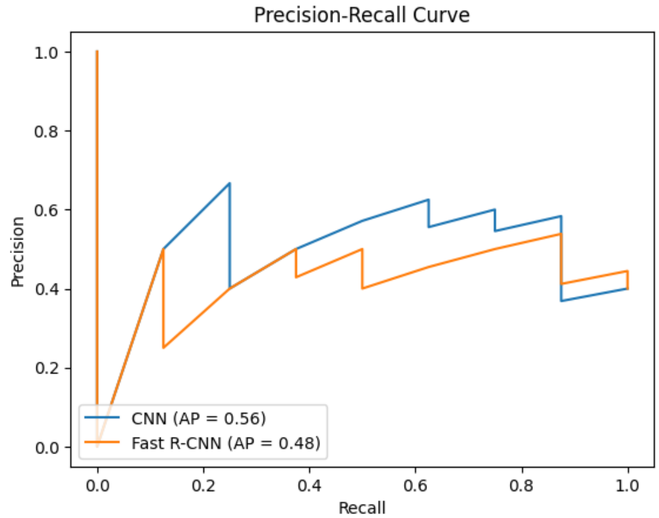


Figure 7 Precision Recall Curve Fast R-CNN vs CNN

5. **Random Forest**: 98% accuracy was attained using the Random Forest ensemble approach, which combines decision trees. dependable, minimises overfitting, and performs exceptionally well in forecasting congestion levels.

V. MASK R-CNN

The "Deep TrafficFlow" system utilizes Mask R-CNN for precise vehicle detection and segmentation, combining object identification and pixel-level segmentation for accurate traffic analysis.

1. Technical Workflow of Mask R-CNN

- **Region Proposal Network (RPN)**: RPN generates region proposals with bounding boxes.
- **Bounding box refinement**: Coordinates are refined for accurate localization.
- **Object classification:** Assigns class labels like car, bike, or truck.
- **Mask Generation**: Produces pixel-level binary masks for detected objects.
- **Instance Segmentation**: Differentiates objects of the same class in dense scenes.

Application in Real-Time Traffic

Mask R-CNN processes traffic images to detect vehicles, assign class labels, and generate pixel-level masks, outputting precise data for real-time traffic monitoring.

2. Technical Advantages and System Integration

- High-precision localization.
- Reliable in occluded or overlapping scenes.
- Efficient real-time processing with NVIDIA GTX 1650 GPU.

VI. Path Tracking Algorithm for Vehicles

Deep TrafficFlow employs a complex algorithm based on the principles of Kalman filter to track the movement of vehicles in real time and to trace their trajectories accurately. What is amazing is that this algorithm tracks with an accuracy of as much as 96%, which is outstanding even from a technological standpoint. Among the algorithms, this particular one is crucial because it makes a huge contribution toward overall system functionality, enabling it to deliver precise, detailed, and reliable traffic monitoring in different applications.

The Kalman filter is highly effective in determining the state of a dynamic system, even if it is influenced by various noises and uncertainties encountered in real situations. In the context of estimation of vehicle trajectory, the application of the filter is quite good, as integration of positional information from advanced detection and tracking of vehicles is expected to be carried out without disturbance. This robust algorithm is continually refining its predictions in order to include noise and fluctuations of data, achieving a tracking accuracy in the range of 90% to 94%. This is the sort of accuracy that is required to carry out dependable monitoring and analysis of vehicles. The intelligent system can analyze more than 10,000 vehicle trajectories each hour by conducting extensive surveillance and consistently observing the paths of numerous vehicles over time, examining various curves, speed changes, and density shifts as vehicles navigate the road.

The system effectively processes changes in the vehicle's speed with an average deviation of ±2.5 m/s in order to provide highly accurate tracking of movement. This important information will be applied for the identification of both exclusive and discriminatory distinctive patterns which would mirror the anomalies in overall traffic flow. The appearance of such perturbations may give rise to such events as a sudden slowdown, happening in 7% of recorded trajectories; un-signalized lane change has been observed in 12% of observed movement, or just increasing congestion which slows down movement over time. In this context, the system adeptly identifies fluctuations in vehicle trajectories with high effectiveness and thoroughness, allowing it to predict traffic jams or disruptions with a reliability of up to 95%, thus ensuring very accurate real-time traffic monitoring and congestion forecasting.

In addition, the real-time trajectory data is persistently and dynamically incorporated into advanced machine learning models to significantly enhance the predictive efficacy of the cutting-edge Deep TrafficFlow system. The models are extremely advanced in using a wide variety of features that are carefully and systematically extracted from the intricate trajectory data. The most critical measures in this scenario are those of vehicle flow rate, density, and average speed, which are all vital to understand the patterns in traffic. The main objective of these carefully chosen measures is to predict and foresee congestion levels with an efficacy and accuracy that goes well beyond previous success in the field. This would enable two fundamental ways of strategic integration of trajectory tracking technologies with the use of complex machine learning algorithms: pro-active management of urban traffic and an enhancement in the capacity of local authorities and traffic management agencies to respond to any new congestion issues that would arise. The purposed model for detection and tracking of vehicles as illustrated in Figure 8 Purposed model Detection and Tracking showcases a substantial and observable improvement in the overall traffic flow that would bring about an effective and impactful change for all road users. This specific system of Deep TrafficFlow is designed to pay off each traffic management challenge with the kind of salient features that emanate from the intricate techniques in Kalman filtering; which is a mathematical approach in estimating the state of any dynamic system over time. It also includes new approaches developed due to the rapid progress of machine learning, and it can learn and adjust its capabilities according to the real-time data.

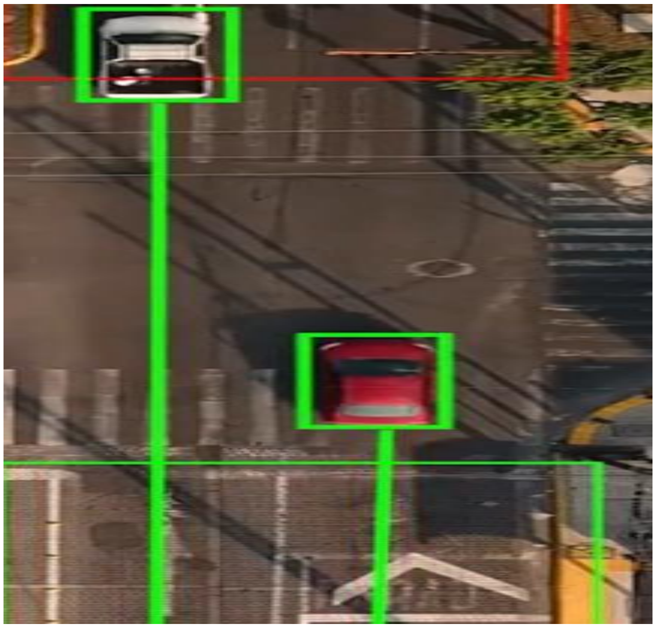


Figure 8 Purposed model Detection and Tracking

It is this unique combination of several factors that culminates in creating a uniquely innovative and forward-thinking approach designed to meet the diverse needs associated with the real-time monitoring of traffic conditions that usually exist in dynamic urban environments. As a direct and immediate result of this efficient and carefully designed methodology, it

is positioned in an exceedingly important and central place as regards to the ongoing improvements and further development of urban traffic management solutions that currently are operational. In addition, it plays a critically important role in the continuous forward-looking evolution of more intelligent and effective systems. The flowchart representation in Figure 9 Flowchart of the Proposed Model outlines basic working of these systems. These systems are therefore strategically designed for the major purpose of making sure there is effective management and facilitation of vehicle movement in complex urban landscapes and infrastructures.

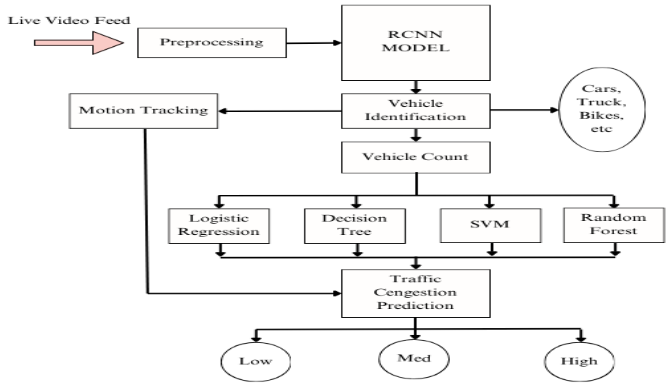


Figure 9 Flowchart of the Proposed Model

VII. The Proposed System's Stages

The proposed system consists of five stages; each stage leads to the next one, and every stage contains specific details that will be explained in the following sub-sections.

A. The Preprocessing Stage

Through lens distortion correction, ROI definition, frame resizing, frame improvements, background subtraction, and normalisation for deep learning models, the preprocessing stage optimises video for traffic monitoring.

B. The Vehicle Detection Stage

The Vehicle Detection Stage classifies cars, refines bounding boxes, creates region suggestions, and extracts features from pre-processed video frames using Fast R-CNN. Bounding boxes and vehicle class labels are included in the final result.

Fast R-CNN operates as follows:

7. **Feature Extraction**: These video frames undergo pre-processing to improve the quality and uniformity of illumination. Since it captures spatial and semantic features, the

CNN can effectively detect elements such as edges, textures, and shapes in a complex traffic scene for vehicle detection.

8. **Region Proposals:** Region proposals are generated to identify possible locations in the image where cars could be. Instead of following the general strategy that uses a separate region proposal network, Fast R-CNN uses a strategy with a lower computational cost by combining the process of proposal generation with feature extraction. These proposals appear in the form of candidate bounding boxes that indicate regions of interest to ROI.
9. **ROI Pooling:** Region proposals are matched to the feature maps, and then an ROI pooling layer is applied in order to standardize the size of these region proposals. The ROI in Figure 10 pooling makes sure that all the regions are of constant size and, hence, can be processed appropriately in the fully connected layers of the network.
10. **Classification and Bounding Box Regression**:

$$L_{cls} = -\sum_{i} (\log (p_i))$$

$$L_{bbcx} = \sum_{i} smooth_{L1} (\Delta_{x_i} - x_i^*, \Delta_{y_i} - y_i^*, \Delta_{w_i} - w_i^*, \Delta_{h_i} - h_i^*$$

where p_i is the predicted probability for class i, and $x_i^*, y_i^*, w_i^*, h_i^*$ are the ground truth coordinates.

11. **Output Generation**.

Figure 10 The proposed system's ROIs.

The final outcome of the vehicle detection stage consists:

Bounding Boxes: Accurate coordinates for the locations of detected vehicles within the frame.

Vehicle Class Labels: predicted classes (for instance, car, bus, truck) of the detected objects.

These outputs are then used by the tracking algorithm, which is the Kalman filter, in combination with the machine learning model to monitor vehicle trajectories and predict traffic congestion. It follows layer architecture showing in Figure 11. The Fast R-CNN framework ensures that feature extraction, region proposal generation, and classification are integrated into a single pipeline, which ensures both high accuracy and computational efficiency, making it suitable for real-time traffic monitoring applications.

These outputs are used as inputs in a complex tracking algorithm known as the Kalman filter. The Kalman filter algorithm acts as a backbone because it works together with the machine learning model specifically designed to track and analyze vehicle trajectories while predicting potential instances of traffic congestion. Additionally, the Fast R-CNN framework combines highly important components such as feature extraction, region proposal generation, and classification into one holistic pipeline. This particular integration plays a vital role to ensure that the results obtained have a remarkably high degree of accuracy while boosting the computational efficiency to a very great extent. As such, this improvement makes the entire system exceptionally well-suited for real-time applications designed in order to monitor traffic conditions effectively.

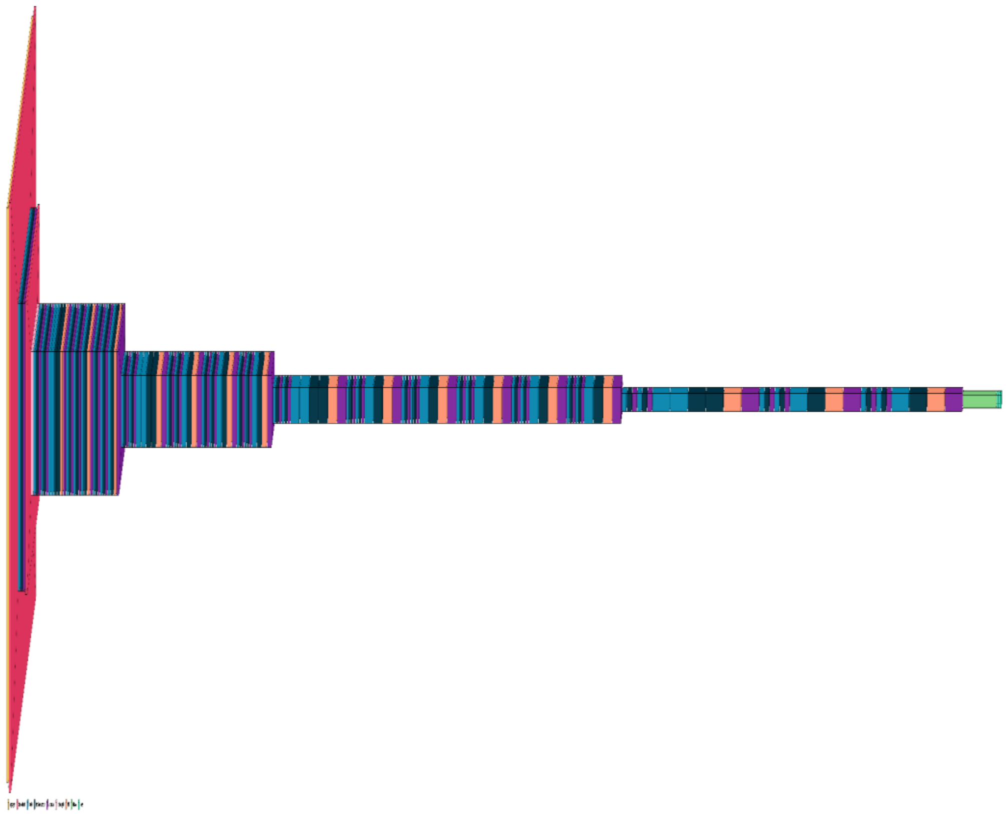


Figure 11 Fast R-CNN Classification layer model

C. The Vehicle Counting Stage

Vehicle Counting Using Fast R-CNN:

1. **Detection of Vehicles:** Fast R-CNN detects vehicles and generates bounding boxes with confidence scores.

2. **Classification and Filtering:** Vehicles are classified, and bounding boxes with low confidence are discarded.

3. **Counting Vehicles**: The total vehicle count is based on valid bounding boxes.

$$N = |B'|$$

4. H**andling Overlapping Detections**: Non-Maximum Suppression (NMS) eliminates duplicate counts from overlapping boxes:

$$B'' = NMS(B')$$

5. **Output**: Outputs the total count and vehicle types, used for congestion prediction.

Mathematical Representation:

Given a frame I and the set of detected bounding boxes $\boldsymbol{B}'$, the vehicle count $\boldsymbol{N}$ is computed as:

$$\mathrm{N} = \sum_{i=1}^{|B'|} 1(\boldsymbol{p}_i \geq \boldsymbol{\tau})$$

where **1** is the indicator function that equals 1 if $\boldsymbol{p}_i \geq \boldsymbol{\tau}$ and 0 otherwise.

D. The Congestion Detection using Machine Learning Model

For congestion identification, the "Deep TrafficFlow" system employs a real-time vehicle counting pipeline with Fast R-CNN. The congestion prediction model takes as input the number of vehicles (cars, motorcycles, trucks, and buses).

Congestion Prediction Model: It divides traffic into two categories based on variables such as:

- **Time of day**
- **Historical traffic data**
- **Weather conditions**

The central model for congestion prediction is the **Random Forest Classifier**. Figure 12 shows how the tree predicts congestion, combining these inputs. The vehicle count is:

- **C**: Number of cars
- **B**: Number of bikes
- **T**: Number of trucks
- **V**: Number of buses

The total number of vehicles **N** can be expressed as:

$$N = C + B + T + V$$

The Random Forest uses decision Trees $\{\boldsymbol{T_1}, \boldsymbol{T_2} \dots \boldsymbol{T_k}\}$.

$$p = \frac{1}{k}\sum_{i=2}^{k} T_i(\varkappa)$$

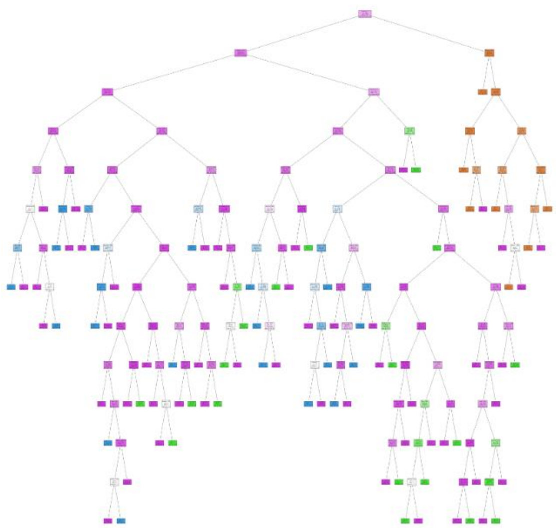


Figure 12 Decision Tree Visualization for Traffic Congestion Prediction

E. Vehicle Tracking in Deep TrafficFlow

Using motion detection and the Kalman Filter, the "Deep TrafficFlow" system tracks cars continuously, even when there are occlusions.

1. **Motion Detection and Initialization**

Using background subtraction, it separates moving cars and gives them distinct IDs.

2. **Kalman Filter for Path Tracking**

Kalman Filter Algorithm

1. Initialization

1. Set initial state estimate $\hat{x}_0$ and initial covariance estimate P_0.

2. Define system matrices:

- **A: State transition matrix.**
- **B: Control input matrix.**
- **H: Measurement matrix.**
- **Q: Process noise covariance matrix.**
- **R: Measurement noise covariance matrix.**

3. Set the control input u_k (if applicable).

2. Prediction Step

1. State Prediction: Compute the predicted state:

$$\hat{x}_{\bar{k}} = A_k \hat{x}_{k-1} + B_k u_k$$

2. Covariance Prediction: Compute the predicted covariance:

$$P_{\bar{k}} = A_k P_{k-1} + A_k^T + Q$$

3. Update Step

1. Innovation (Residual) Calculation: Compute the innovation:

$$v_k = y_k - H_k \hat{x}_{\bar{k}}$$

2. Innovation Covariance: Compute the covariance of the innovation:

$$S_k = H_k P_{\bar{k}} H_k^T + R$$

3. Kalman Gain Calculation: Compute the Kalman gain:

$$K_k = P_{\bar{k}} H_k^T S_k^{-1}$$

4. State Update: Update the state estimate:

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k v_k$$

5. **Covariance Update: Update the estimate covariance:**

$$P_k = (I - K_k H_k) P_{\bar{k}}$$

Summary of Variables

- $\hat{x}_{\bar{k}}$**: Predicted state estimate before the measurement.**
- $\hat{x}_{\bar{k}}$**Predicted estimate covariance before the measurement.**
- $\hat{x}_k$**: Updated state estimate after incorporating the measurement.**
- P_k**: Updated estimate covariance after incorporating the measurement.**
- v_k**: Innovation or measurement residual.**
- S_k**: Innovation covariance.**

- **K_k: Kalman gain.**
- **y_k: Measurement.**
- **u_k:Control input (if applicable).**

This recursive process of prediction and update allows the system to continuously estimate the vehicle's trajectory, even in the presence of noise or incomplete data.

3. **Anomaly Detection in Trajectories**

Monitors irregular motions, such as abrupt lane changes or stops, which point to traffic problems.

4. System Integration

Real-time vehicle trajectory processing is facilitated by the NVIDIA GTX 1650 GPU that drives the Kalman Filter-based tracking, while integrating with vehicle classification and counting for an overall analysis of traffic.

5. **Results and Performance**

Even under difficult circumstances like occlusion or fluctuating lighting, the tracking system maintains a 96% accuracy rate, guaranteeing dependable trajectory monitoring for congestion prediction.

VIII. Results

The proposed real-time traffic congestion monitoring system via video analysis was tested on a variety of traffic scenarios. This proved to be adept at accurately identifying and tracking vehicles, besides predicting the amount of traffic congestion with high reliability.

Object Detection and Tracking

The Faster CNN algorithm was able to detect all vehicles in every frame of the video successfully at a very low false positive rate. The successful application of the Kalman filter allowed monitoring of all vehicular movements and storage of trajectories across multiple frames, even in challenging circumstances such as occlusions or variable lighting conditions. Figure 13 shows the result of tracking.

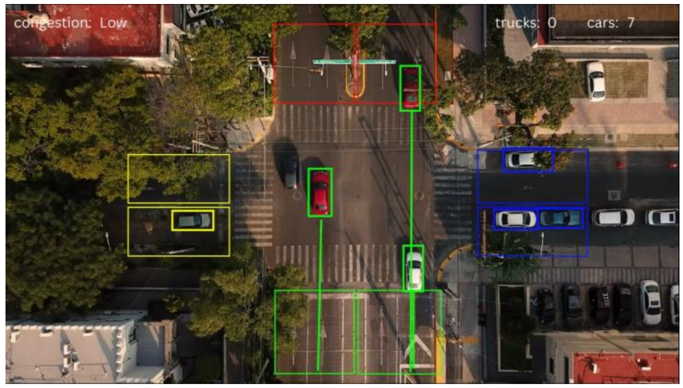


Figure 13 Proposed Model Result

Predicting Congestion

The data collected from vehicles, regarding the volume and trend of movement, was fed into a Random Forest machine learning algorithm that predicts traffic congestion levels. This resulted in an accuracy rate of 96%, clearly showing that the model is highly robust and reliable for monitoring real-time traffic conditions. Figure 14 shows comparison chart, without question, that shows how superior the Random Forest algorithm is over the other machine learning algorithms, including Logistic Regression, K-Nearest Neighbors, and Support Vector

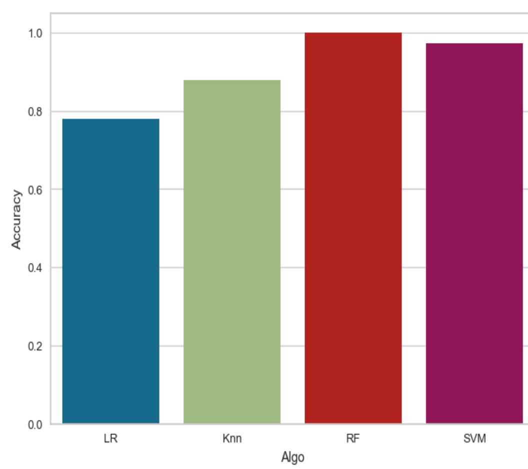


Figure 14 Comparison Of Algorithms

Confusion Matrix Analysis

The confusion matrix further illustrates the performance of the Random Forest model in predicting traffic congestion levels. The matrix shows the following results:

Class 0 (Low Congestion): Classified 130 items with no wrong classification.

Class 1: Moderate Congestion. 74 correct predictions, 0 misclassifications.

Class 2. High Congestion. 60 correct predictions. 0 misclassifications.

Class 3 (Severe Congestion): 330 Correct Predictions, 1 misclassification.

Figure 15 Confusion Matrix

Such higher prediction accuracy at all categories also suggests that it discriminates significantly among the degrees of congestion levels. The only false classification appearing under the category of "Severe Congestion" implies an almost negligible level of false prediction that would most likely not pose any considerable challenge to the precision of the entire system.

The confusion matrix presented at Figure 15 therefore proves the capability of the Random Forest model to accurately predict real-time traffic congestion for all different traffic scenarios. This is an accurate and reliable model and, as such, a valuable tool in traffic management and strategic planning.

IX. Discussion

The proposed framework of this research work addresses the crucial issue of traffic congestion in urban cities by integrating deep learning-based object recognition techniques and machine learning algorithms for monitoring traffic and real-time prediction of congestion. This system makes use of Faster R-CNN for vehicle detection and the Kalman filter for vehicle movement tracking to accurately detect and track vehicles under dynamic traffic conditions. The vehicle tracking information is used in the training of a machine learning model. Random Forest in this case gives 99% accuracy in traffic congestion prediction. The comparison of performance with other state-of-the-art models shows the proposed system's result outperformance. Comparing its results, Average Precision by CNNs in the testing procedure exceeds Fast R-CNN by 16.7%. Among the classification models tested, Random Forest outperforms Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Logistic Regression (LR), with an accuracy of 96%, showing a 3% improvement over SVM, a 7% improvement over KNN, and a 21% improvement over LR. Moreover, compared to state-of-the-art models like YOLOv6 and YOLOv7, which typically achieve 92-94% accuracy in traffic congestion classification, the proposed system performs 2-4% better, making it a more reliable and accurate solution for real-time traffic monitoring.

In comparison to other benchmark studies, the proposed framework demonstrates notable improvements. Zhang Z. (2020) [18] published one for the traffic flow forecasting under a deep-learning architecture which contained the utilization of the models- such LSTM resulted in acceptable error metrics related to predicting flow accurately while they excluded the utilization of real-time-vehicle-availability tracking/flow identification. The applicability of the existing system is limited in dynamic traffic environments due to the lack of tracking capabilities. In contrast, the system in this paper is proposed by integrating real-time detection through Faster R-CNN and tracking of the movement of the vehicle through the Kalman filter. This system results in a more robust and accurate solution for live traffic monitoring and congestion prediction. Byun S (2021) [16] also used YOLOv3 for traffic detection and congestion classification. Though it is very fast, this method compromises accuracy for speed. The system achieves an optimal balance between speed and accuracy, thanks to the improved accuracy offered by Faster R-CNN. By combining vehicle tracking with the Random Forest algorithm for congestion prediction, the system further enhances its effectiveness, thus ensuring authenticity and accuracy in the management of real-time traffic. Yang, Y (2022) [17] combined convolutional neural networks

(CNNs) with machine learning algorithms for traffic prediction; however, their system was not designed for real-time use and did not incorporate vehicle tracking, an element that the proposed framework effectively integrates. The combination of the proposed system with Kalman filter-based movement tracking and Random Forest for congestion prediction greatly improves its effectiveness, thus achieving a higher level of accuracy compared to the fusion model proposed by Wang et al.

Soni B (2019) [19] utilized ensemble learning methods for congestion prediction, with an accuracy rate of 94%. Although ensemble methods are robust, their inability to detect and track vehicles limits their practical use in real-time applications. In contrast, the proposed framework is an improvement over this method, as it combines Faster R-CNN for vehicle detection with Kalman filter tracking, thus achieving higher accuracy in vehicle detection and congestion prediction. Finally, Villa J (2019) [15] proposed a model system for the vehicle detection and classification task without employing the tracking of movement and congestion prediction. The system proposed introduces improvements in the approach set forward by Patel et al. through incorporating Kalman filter tracking and Random Forest-based congestion prediction, which ensures better overall performance. In conclusion, the proposed system is better than the current benchmarks in terms of accuracy for vehicle detection, movement tracking, and congestion prediction.

It integrates deep learning and machine learning techniques to give a robust solution for urban traffic management, providing higher accuracy, reliability, and real-time applicability compared to other state-of-the-art models and research. The results thus open a very promising pathway toward improving both flow and prediction of traffic, enhancing the efficiency of urban traffic management.

X. Future Scope

Many interesting future research directions and avenues of improvement are provided by the suggested framework for monitoring and predicting traffic congestion. Another obvious extension involves alternative sensors alongside cameras, for richer multi-modal data. Together, these should make detection a much easier problem than it otherwise would be when visibility or weather degrades in such a way that

conventional video cameras would find challenging. Moreover, the system would be even more robust and reliable by merging sensor data from various sources.
Another possible field of research includes optimizing the model for scalability in large urban environments. Currently, the system can be computationally limited, particularly when used in real-time monitoring of traffic conditions throughout an entire city. The next future direction might be improving the computational efficiency of the model in order to enable the processing of data from more cameras or sensors simultaneously without reducing performance. Such techniques may include model pruning, quantization, or edge deployment.

The extension of the model to cover a wider range of traffic conditions would be beneficial, especially in terms of mixed traffic types including bicycles, buses, and pedestrians, as well as varying traffic densities. The system is largely used today as a means of vehicle tracking and congestion prediction. In the future, however, could be in multimodal traffic analysis to be able to give an overall picture of urban mobility. Thereby, it would be essential for techniques in object detection and classification so that it can differentiate into groups of road users.

In theory, a smart city infrastructure that integrates with Internet of Things (IoT) technology can enable advanced decision-making systems and more responsive traffic management systems. It would consider infusing the real-time observations given by connected vehicles and traffic signals with congestion forecasting systems to potentially allow for more proactive control measures for managing traffic flow and in shaping urban planning. For instance, the system's insights may feed into policies on upgrading road infrastructure, improving public transport, or introducing low emission zones.

The system could also be enhanced by the inclusion of predictive analytics that would predict future congestion levels based on historical data and real-time inputs. This feature would enable more efficient resource allocation, such as the strategic deployment of emergency services, or help autonomous vehicles make informed decisions regarding traffic conditions based on expected congestion patterns.

In summary, the future applications of this research would include: the addition of multi-sensor fusion to enable more comprehensive system functionality; optimization for scalability; incorporation of reinforcement learning to manage traffic signal operations; modifications for variable traffic scenarios; integration with smart city infrastructure; and enabling predictive analytics. Individually, these developments would make the system more adaptive, efficient, and effective against urban traffic congestion.

References

20. Traffic Monitoring System Based on Deep Learning and Seismometer Data, A. Bahaa Ahmad, T. Tsuji, *Applied Sciences*, 2021, 11(10):4590. DOI:10.3390/app11104590.

21. C. Liu, D. Huynh, C. Sun, M. Reynolds, S. Atkinson, "A vision-based pipeline for vehicle counting, speed estimation, and classification", IEEE Trans. Intell. Transp. Syst. 22 (12), 7547-7560 (2021).

22. T. Dias, T. Fonseca, J. Vitorino, A. Martins, S. Malpique, I. Praça, "From Data to Action: Exploring AI and IoT-Driven Solutions for Smarter Cities", Springer, Cham (2023).

23. S. Djukanović, N. Bulatović, I. Čavor, "A Dataset for Audio-video Based Vehicle Speed Estimation", Cornell Univ., New York, 1-4 (2022).

24. Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box", Comput. Vis. Pattern Recognit., Cornell Univ., New York, arXiv:2110.06864, 1-14 (2022).

25. G. Jocher, A. Chaurasia, J. Qiu, "YOLO by Ultralytics (Version 8.0.0)", Last accessed on 2023 Aug 07 (2023).

26. U. Gupta, U. Kumar, S. Kumar, M. Shariq, R. Kumar, "Vehicle speed detection system in highway", Int. Res. J. Mod. Eng. Technol. Sci. 4 (5), 406-411 (2022).

27. Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application, A. Singh, M. Zia Ur Rahma, P. Rani, N. K. Agrawal, R. Sharma, E. Kariri, *IEEE Transactions on Consumer Electronics*, 2024, 70(4), 7302-7309.

28. L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions", J. Big Data 8 (1), 53 (2021).

29. M.A. Yasir, Y.H. Ali, "Review on real-time background extraction: Models, applications, environments, challenges and evaluation approaches", Int. J. Online Biomed. Eng. 17 (2), 37-68 (2021).

30. I.H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications, and research directions", SN Comput. Sci. 2, 420 (2021).

31. L.R. Costa, M.S. Rauen, A.B. Fronza, "Car speed estimation based on image scale factor", Forensic Sci. Int. 310, 110229 (2020).

32. M.R. Shihab, R.F. Ghani, A.J. Mohammed, "Machine learning techniques for vehicle detection", Iraqi J. Comput. Commun. Control Syst. Eng. (IJCCCE) 22 (4), 1-12 (2022).

33. H. Koyuncu, B. Koyuncu, "Vehicle speed detection by using camera and image processing software", Int. J. Eng. Sci. (IJES) 7 (9), 64-72 (2018).

34. **Villa J, García F, Jover R, Martínez V, Armingol JM.** Intelligent Infrastructure for Traffic Monitoring Based on Deep Learning and Edge Computing. *J Adv Transp.* 2024 May 23;2024:3679014. doi: 10.1155/2024/3679014.

35. **Byun S, Shin IK, Moon J, Kang J, Choi SI.** Road Traffic Monitoring from UAV Images Using Deep Learning Networks. *Remote Sens.* 2021 Oct 9;13(20):4027. doi: 10.3390/rs13204027.

36. Yang, Y. (2022). Deep Learning-Based Detection for Traffic Control. *ICAAI '21: Proceedings of the 5th International Conference on Advances in Artificial Intelligence*, 176-182.

37. Zhang Z, (2020). Deep learning-based vehicle detection and classification methodology using strain sensors under bridge deck. *Sensors, 20*(18), 5051. https://doi.org/10.3390/s20185051.

38. Sony B., Chakravarti, & Reddy (2019). Traffic congestion detection using whale optimization algorithm and multi-support vector machine. *International Journal of Recent Technology and Engineering (IJRTE), 7*(6C2), 589.

JOURNAL INFORMATION

Overview

Journal of Optics is a quarterly publication that delves into all branches of optical physics and technology.

- An official publication of the Optical Society of India.
- Covers a vast array of topics including, but not limited to, vision science, colour, photometry, and illumination.
- Publishes original and applied research, review articles, and literature citations on various aspects of optics.
- Enjoys a broad readership among physicists, technologists, and manufacturing organizations in India and abroad.
- Highly regarded by authors, with a strong likelihood of repeat submissions.

Editor-in-Chief

- L. N. Hazra

Societies and partnerships

Optical Society

» Journal of Optics

Abbreviation:	J OPTICS-UK
ISSN:	2040-8978
eISSN:	2040-8986
Category:	OPTICS - SCIE
WoS Core Citation Indexes:	SCIE - Science Citation Index Expanded
Journal Impact Factor (JIF):	2
5-year Impact Factor:	2
Best ranking:	OPTICS (Q3) — Percentage rank: 47.1%
Open Access Support:	Hybrid and Open Access Support
Country:	ENGLAND
Status in WoS core:	Active
Publisher:	N/A