J Opt
https://doi.org/10.1007/s12596-025-02606-w

RESEARCH ARTICLE

A real time video sliced frame image based intelligent traffic congestion monitoring system using faster CNN

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

Received: 12 January 2025 / Accepted: 5 February 2025

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.

Keywords Computer vision · Object detection · Real-time traffic monitoring · Fast CNN · Path tracking · Vehicle tracking · Machine learning · Edge detection

✉ Vikas Kumar
kvikas1482@gmail.com

✉ Shreyansh Tiwari
shreyanshtiwari3640@gmail.com

Rahul Kumar Sharma
rahulpccs1988@gmail.com

Anurag Sinha
anuragsinha257@gmail.com

Ghanshyam G. Tejani
p.shyam23@gmail.com

1 Department of Computer Sciences and Engineering, KIET Group of Institution, Ghaziabad, Uttar Pradesh, India

2 Tech School, Computer Science Department, ICFAI University, Ranchi, Jharkhand, India

3 Applied Science Research Center, Applied Science Private University, Amman, Jordan

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

	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

Fig. 1 Dataset Table

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 in-depth 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.

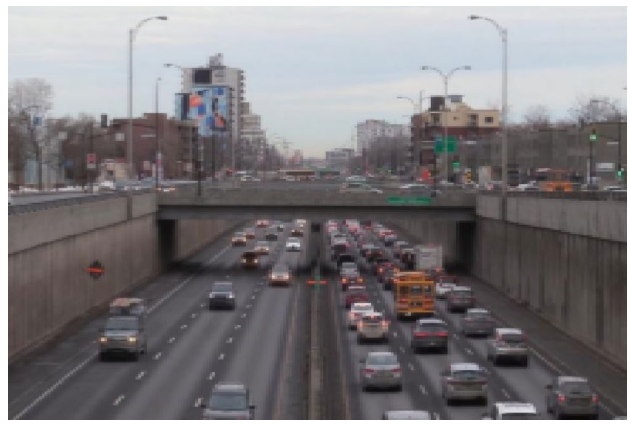

Fig. 2 Sample location of location

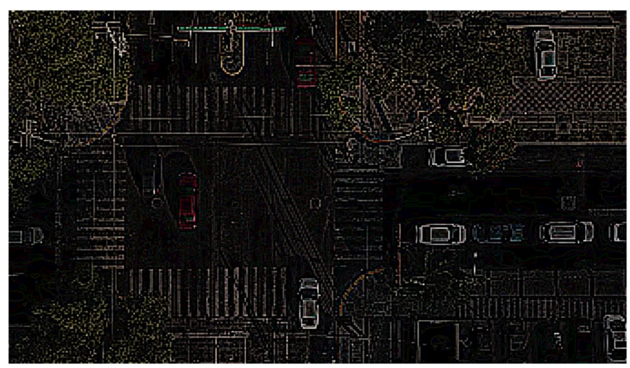

Fig. 3 Edge detection of the image

Fig. 4 Flow Chart of Pre-Processing

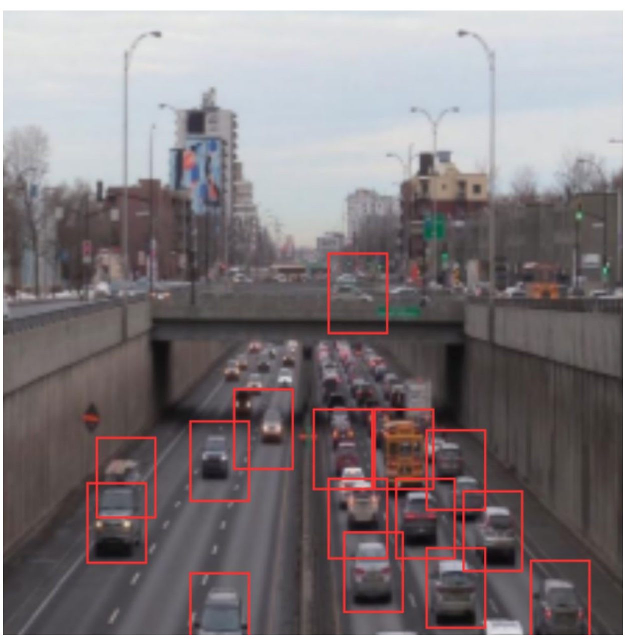

Fig. 5 Object Detection and Bounding Box

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

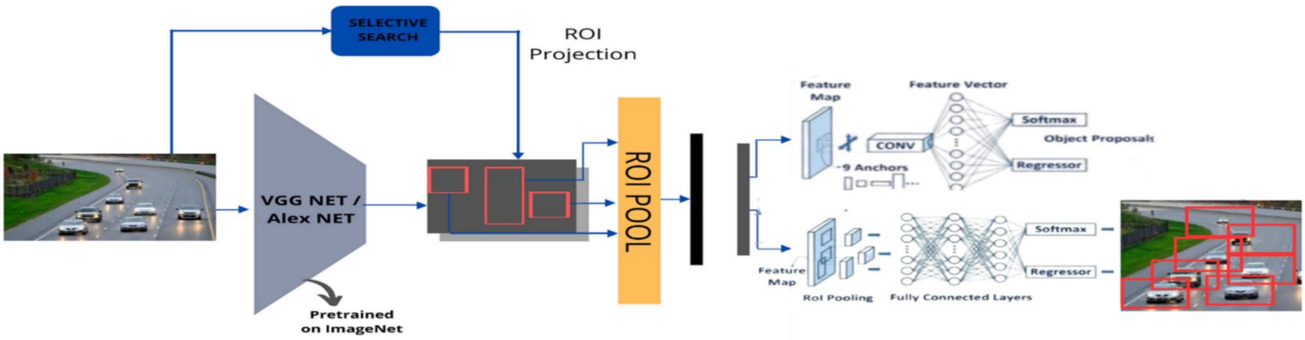

Fig. 6 Fast R-CNN model

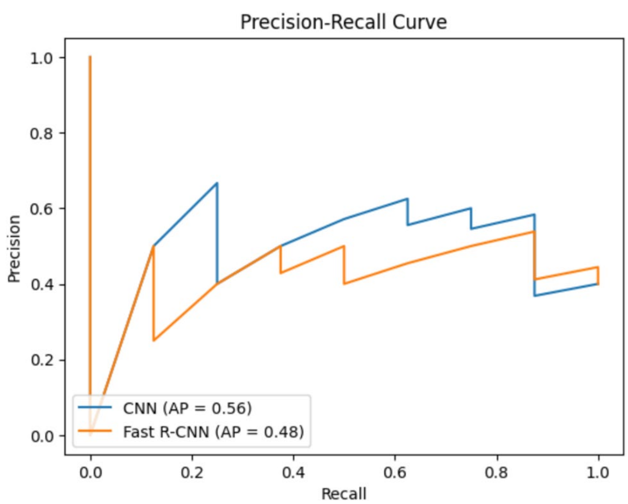

Fig. 7 Precision-Recall Curve for CNN vs. Fast R-CNN

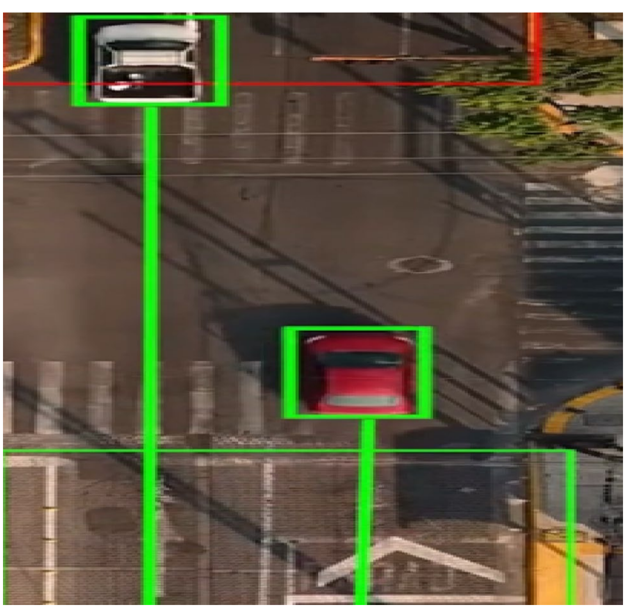

Fig. 8 Purposed model Detection and Tracking

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

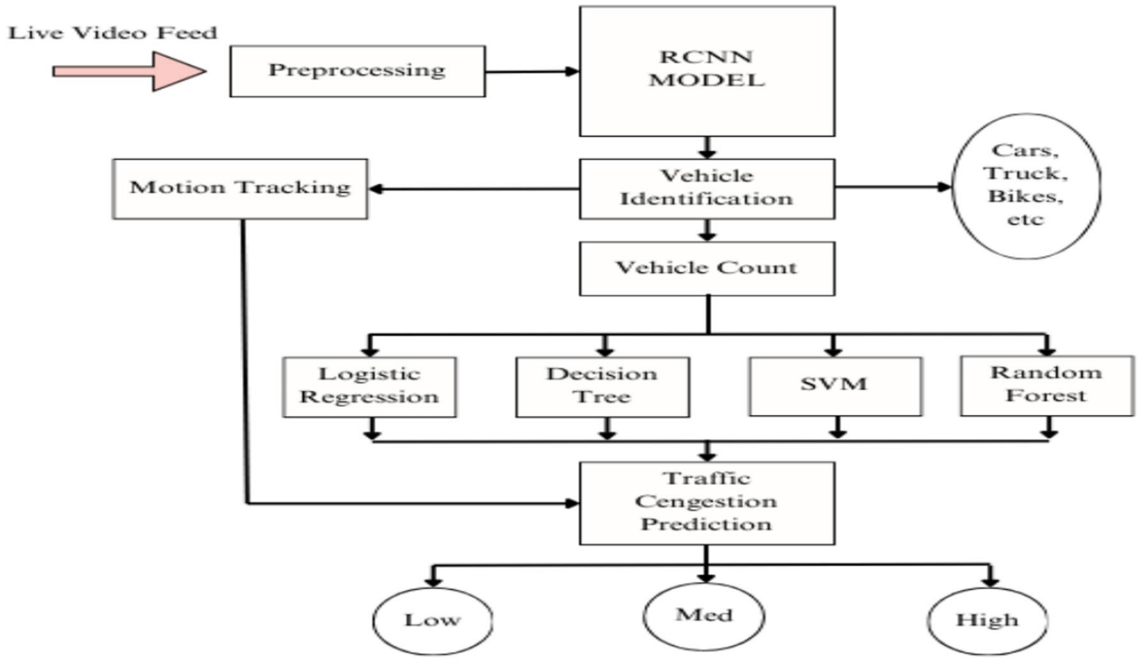

Fig. 9 Flowchart of the Proposed Model

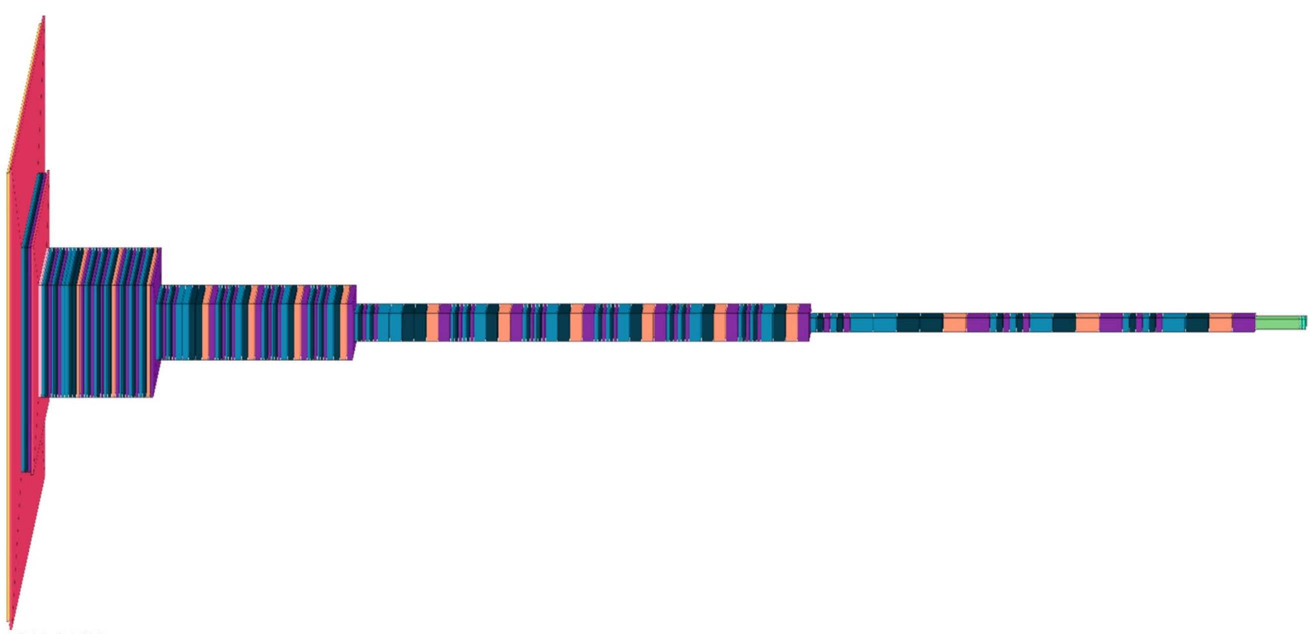

Fig. 10 Fast R-CNN for Car Classification model layers

(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.

Fig. 11 The proposed system's ROIs

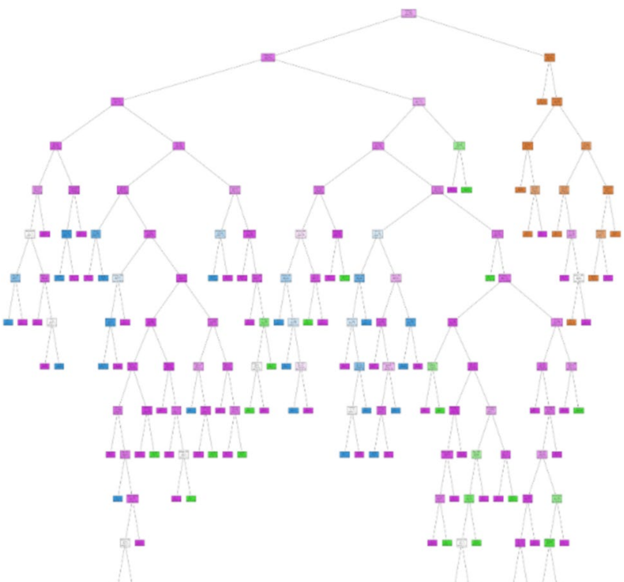

Fig. 12 Decision Tree Visualization for Traffic Congestion Prediction

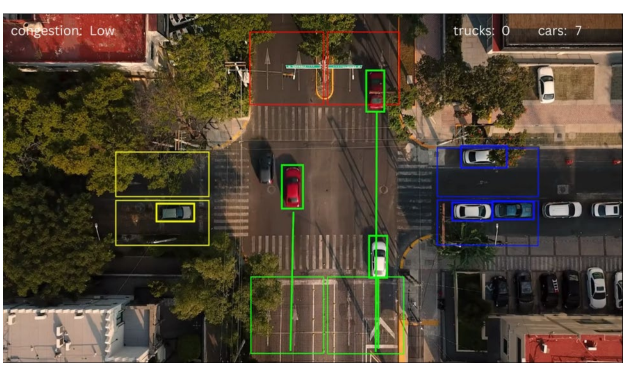

Fig. 13 Proposed Model

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.

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

Ref No	Paper Title	Year	Limitations	Solution Superiority
[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 mixed traffic conditions	- Adapts to dynamic urban settings with variable lighting - Uses deep learning-based 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

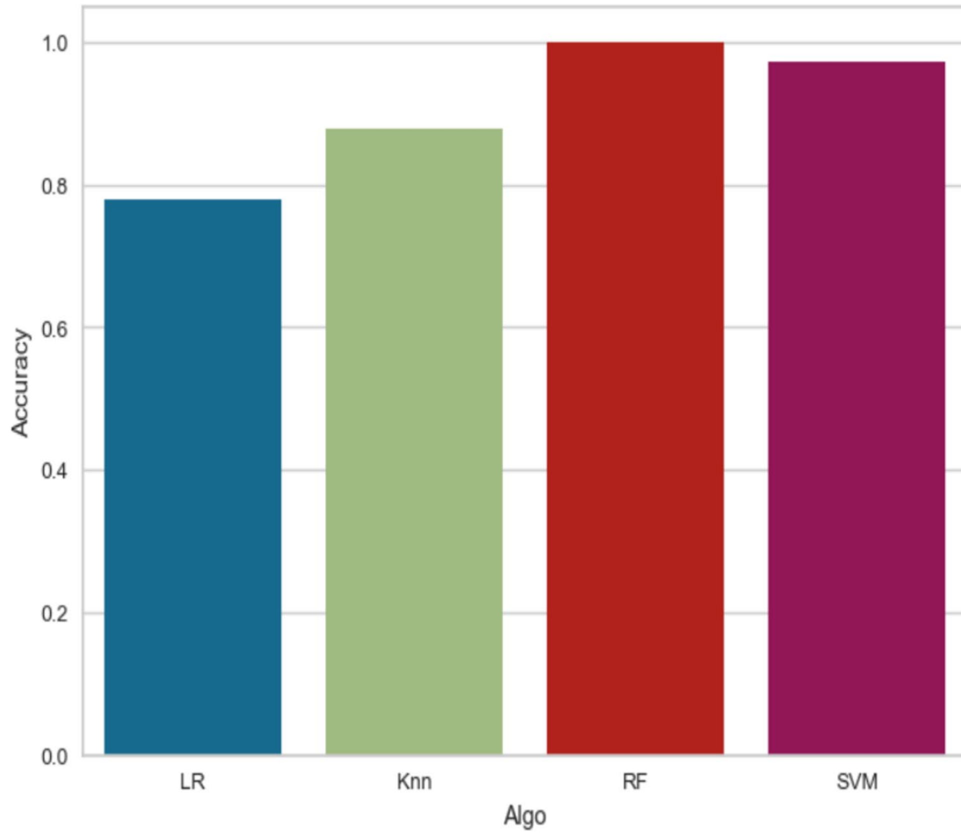

Fig. 14 Comparison of Algorithms

Fig. 15 Confusion Matrix

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 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.

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 shows the sample image which is used for the processing.

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 illustrates 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.

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 shows a flow Chart of the pre-processing stages outlining the key steps involved in preparing the dataset for analysis.

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 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 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.

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 Fig. 7 suggests that it has better precision-recall trade-offs in this scenario.
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.

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.

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 Fig. 8 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.

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 Fig. 9 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.

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:

1. **Feature Extraction**: These video frames undergo preprocessing to improve the quality and uniformity of illumination. The processed frames are further input into a CNN to yield feature maps. 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.
2. **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.
3. **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 Fig. 10 ROI 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.
4. **Classification and Bounding Box Regression:**

$$L_{cls} = -\sum_i \left(\log(\boldsymbol{p}_i)\right)$$

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

where $\boldsymbol{p}_\mathbf{i}$ is the predicted probability for class $\boldsymbol{i}$, and $\boldsymbol{x}_i^*, \boldsymbol{y}_i^*, \boldsymbol{w}_i^*, \boldsymbol{h}_i^*$ are the ground truth coordinates.

5. **Output Generation:**

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 Fig. 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.

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.

$$N = |B\prime|$$

Handling Overlapping Detections: Non-Maximum Suppression (NMS) eliminates duplicate counts from overlapping boxes:

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

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 N is computed as:

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

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 show 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 $\{\boldsymbol{T}_1, \boldsymbol{T}_2 \ldots \boldsymbol{T}_k\}$.

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

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:**
   - o **A: State transition matrix.**
   - o **B: Control input matrix.**
   - o **H: Measurement matrix.**
   - o **Q: Process noise covariance matrix.**
   - o **R: Measurement noise covariance matrix.**
3. **Set the control input u_k (if applicable).**

2 Prediction Step

1 State Prediction: Compute the predicted state:

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

2 Covariance Prediction: Compute the predicted covariance:

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

3 Update Step

1 Innovation (Residual) Calculation: Compute the innovation:

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

2 Innovation Covariance: Compute the covariance of the innovation:

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

3 Kalman Gain Calculation: Compute the Kalman gain:

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

4 State Update: Update the state estimate:

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

5 Covariance Update: Update the estimate covariance:

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

Summary of Variables

- **$\hat{x}_{\overline{k}}$: Predicted state estimate before the measurement.**
- **$\hat{x}_{\overline{k}}$ 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.

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 show the result of tracking.

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 show is a 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 Machines.

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.

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 Fig. 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.

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 et al. [15] 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 et al. [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 [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.

Sony et al. [18] 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 et al. [19] 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.

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

1. A.B. Ahmad, T. Tsuji, Traffic monitoring system based on deep learning and seismometer data. Appl. Sci. **11**(10), 4590 (2021). https://doi.org/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, 2022), pp.1–4
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. A. Singh, M.Z.U. Rahma, P. Rani, N.K. Agrawal, R. Sharma, E. Kariri, D.G. Aray, Smart traffic monitoring through real-time moving vehicle detection using deep learning via aerial images for consumer application. IEEE Trans. Consum. Electron. **70**(4), 7302–7309 (2024). https://doi.org/10.1109/TCE.2024.3445728
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. Z. Zhang, Deep learning-based vehicle detection and classification methodology using strain sensors under bridge deck. Sensors **20**(18), 5051 (2020). https://doi.org/10.3390/s20185051
16. S. Byun, I.K. Shin, J. Moon, J. Kang, S.I. Choi, Road traffic monitoring from UAV images using deep learning networks. Remote Sens. **13**(20), 4027 (2021). https://doi.org/10.3390/rs13204027
17. Y. Yang, Deep Learning-Based Detection for Traffic Control. *ICAAI '21: Proceedings of the 5th International Conference on Advances in Artificial Intelligence*, 176–182 (2022)
18. B. Sony, A. Chakravarti, M.M. Reddy, Traffic congestion detection using whale optimization algorithm and multi-support vector machine. Int. J. Recent Technol. Eng. (IJRTE), **7**(6C2), 589 (2019)
19. J. Villa, F. García, R. Jover, V. Martínez, J.M. Armingol, Intelligent infrastructure for traffic monitoring based on deep learning and edge computing. J. Adv. Transp. **23**(2024), 3679014 (2024). https://doi.org/10.1155/2024/3679014