Cureus Journal of Computer Science

Open Access Original Article

Traffic Congestion Prediction Using Machine Learning Algorithm

Trishita Das [1] , Ipshita Chatterjee [1] , Sanjoy Mondal [2]

1. Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, IND
2. Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, IND

Corresponding author: Sanjoy Mondal, sanjoymondal@soa.ac.in

Received 11/08/2024
Review began 11/09/2024
Review ended 01/24/2025
Published 01/27/2025

DOI:
https://doi.org/10.7759/s44389-024-01981-y

Abstract

Traffic congestion poses a significant challenge in urban environments, leading to delays, increased fuel consumption, and heightened environmental pollution. Addressing this issue, our study presents an innovative dynamic traffic signal control system designed to adapt to real-time traffic conditions. Traditional traffic management systems typically rely on fixed schedules, which are inefficient when dealing with fluctuating traffic volumes. In contrast, our proposed system leverages advanced technologies to optimize traffic flow dynamically. At the core of the system are convolutional neural networks, which process video footage from CCTV cameras to accurately detect and count vehicles. This real-time data enables the system to dynamically adjust traffic signal timings, prioritizing lanes with heavier congestion and minimizing overall delays. By effectively responding to changing traffic patterns, the system reduces wait times and alleviates congestion, as validated through extensive simulations and experimental studies. The proposed approach is both cost-effective and scalable, making it particularly suitable for densely populated regions, such as urban centers in India. Additionally, the system capitalizes on existing infrastructure, providing a practical and efficient solution to urban traffic challenges. Future developments aim to integrate features like emergency vehicle detection and weather condition monitoring, further enhancing the system's responsiveness and optimizing urban traffic management.

Categories: AI applications, Expert Systems, Predictive Analytics
Keywords: traffic analysis, traffic congestion, machine learning, convolution neural network (cnn), deep learning

Introduction

In today's rapidly evolving world, where efficient mobility is crucial for both economic growth and social development, traffic congestion remains a significant hurdle. Traffic jams not only lead to wasted time and fuel but also contribute to environmental harm and place substantial pressure on urban infrastructure. In this context, integrating machine learning algorithms into transportation systems emerges as a promising strategy to mitigate congestion and improve mobility. The idea of using machine learning to predict traffic jams embodies the convergence of data science and transportation engineering, offering a proactive solution to traffic issues. By leveraging historical and real-time data along with advanced algorithms, transportation authorities and urban planners can gain valuable insights into traffic patterns, predict congestion hotspots, and implement targeted measures to reduce their negative impact [1-3]. Central to this approach is machine learning techniques to analyze large and diverse datasets. These datasets include various factors such as traffic flow, weather conditions, road infrastructure, and socio-economic elements. Utilizing supervised, unsupervised, and reinforcement learning algorithms, predictive models can be created to identify patterns within this extensive data, enabling accurate traffic congestion forecasts [4,5].

Supervised learning techniques, including regression and classification algorithms, are fundamental to traffic prediction models. By training these algorithms on historical data, such as traffic volumes, vehicle speeds, and time of day, predictive models can learn to associate certain features with congestion events. These models can then extrapolate past data to forecast future traffic conditions, allowing timely interventions to improve traffic flow. Unsupervised learning algorithms, such as clustering and anomaly detection methods, provide additional tools for traffic prediction. These algorithms can identify clusters of similar traffic behaviors and detect unusual patterns that signal potential congestion. By utilizing unsupervised learning, transportation authorities can gain deeper insights into traffic dynamics and proactively address emerging congestion hotspots before they become severe [6,7].

Moreover, reinforcement learning algorithms show promise in optimizing real-time traffic management strategies. By simulating traffic scenarios within a dynamic environment, reinforcement learning agents can learn optimal control policies to reduce congestion and enhance traffic efficiency. These algorithms can adapt to changing traffic conditions through continuous interaction with the transportation system, refining their decision-making processes to offer adaptive solutions for complex urban environments. However, implementing machine learning-based traffic prediction systems involves several challenges. These include integrating disparate data sources, ensuring the accuracy and reliability of

How to cite this article
Das T, Chatterjee I, Mondal S (January 27, 2025) Traffic Congestion Prediction Using Machine Learning Algorithm. Cureus J Comput Sci 2 : es44389-024-01981-y. DOI https://doi.org/10.7759/s44389-024-01981-y

predictive models, and addressing concerns related to data privacy and cybersecurity. Additionally, effective collaboration among stakeholders - including government agencies, transportation authorities, and technology providers - is essential to fully realize the potential of machine learning in alleviating traffic congestion.

In summary, predicting traffic jams using machine learning algorithms represents a significant shift in transportation management, providing a data-driven approach to addressing the challenges of urban mobility. By harnessing artificial intelligence and big data analytics, transportation systems can move from reactive to proactive strategies, ultimately improving the efficiency, sustainability, and resilience of urban transportation networks. A typical scenario of the traffic piling up is depicted in Figure *1*.

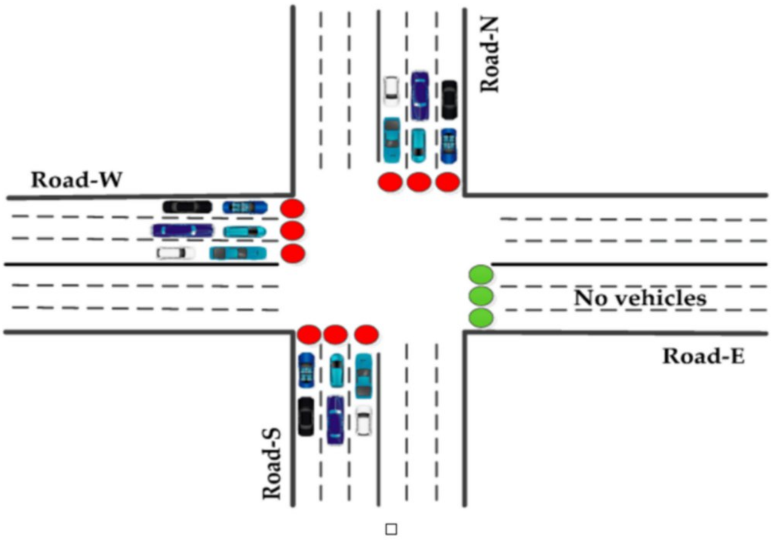

FIGURE 1: Conventional scenario of traffic piling up

Literature survey

Throughout the process of conducting the project, we have surveyed and gone through multiple documentations and research papers to obtain information in various fields that are relevant to our project. A study [8] presents an innovative approach to traffic light control aimed at improving traffic flow efficiency. The system utilizes real-time traffic flow data to dynamically adjust traffic light timings, thereby optimizing traffic management at junctions. By leveraging this real-time data, the system can adapt to changing traffic conditions, prioritizing green signals for areas with heavier traffic and minimizing delays for areas with lighter traffic. The proposed system offers a promising solution to address the challenges of traffic congestion, particularly in urban environments where traffic dynamics are complex and variable. Through simulations and experiments, the effectiveness of the intelligent traffic light control system is demonstrated, showcasing its potential to significantly improve traffic flow efficiency and reduce congestion. Overall, the research contributes to the advancement of intelligent transportation systems by introducing a dynamic and adaptive approach to traffic light control, with implications for enhancing overall road safety, reducing travel time, and improving the commuting experience for motorists.

Unlike conventional traffic control systems that rely on physical sensors, the proposed system captures light signals emitted by vehicles and analyzes them to gauge traffic density in real-time. This non-intrusive method offers several advantages, including cost-effectiveness and minimal infrastructure requirements. By harnessing visible light communication (VLC) technology, the system can accurately monitor traffic flow without the need for additional hardware installations, making it suitable for deployment in diverse urban environments. The intelligent traffic control system dynamically adjusts signal timings based on the detected traffic density, prioritizing green signals for areas with heavier traffic and optimizing traffic flow to minimize congestion. Through simulations and experiments, the efficacy of the system is evaluated, demonstrating its potential to significantly improve traffic management and reduce travel time for commuters [9,10].

The paper further discusses incorporating advanced capabilities into the system, including the ability to identify emergency vehicles and provide alternative routes via a mobile application in times of heavy congestion. Such features add effectiveness and safety to the system by allowing rapid responses for emergencies while offering drivers real-time traffic updates and alternate routes to avoid delay. Overall, this research presents a new intelligent traffic management approach using VLC technology that is affordable and adaptive in easing traffic and improving flow in urban areas. This system is highly promising as it utilizes visible light signals from vehicles to streamline traffic operations and enhance the driving experience. Ghazal et al. [3] have proposed an intelligent traffic light system that will increase flow and reduce congestion. The design involves the integration of advanced tools like infrared (IR) sensors and microcontrollers for accurate detection of vehicles and real-time volume of traffic. It differs from the usual fixed-timing system, as the signal duration is optimized through live traffic data.

Through the use of IR sensors, the system continuously monitors traffic flow, detecting the presence of vehicles and analyzing traffic density in specific directions [11,12]. This real-time data is processed by microcontrollers embedded within the traffic lights, enabling the system to adaptively regulate signal changes to prioritize green lights for areas with heavier traffic and minimize delays for areas with lighter traffic improved overall commuting experience for motorists. Moreover, Chaoura et al. highlight the system's scalability and adaptability, which can be easily integrated into existing traffic infrastructure with minimal additional hardware requirements [13]. This scalability makes the system suitable for deployment in diverse urban environments, offering a cost-effective solution for addressing traffic congestion and enhancing overall road safety [14]. In summary, the research paper presents a comprehensive approach to smart traffic light control, leveraging advanced technologies to optimize traffic flow and reduce congestion at junctions. By dynamically adjusting signal timings based on real-time traffic data, the system offers a promising solution for improving traffic management efficiency and enhancing the commuting experience for motorists in urban environments. This particular research paper, apart from addressing the obvious problem with traffic management, also addresses other important factors like a dedicated pass-through route for emergency vehicles like ambulances and fire engines that need a congestion-free path at any cost. It also includes a mobile application in order to provide users with alternate routes when there is an expected congestion.

Problem definition

Control the traffic flow at intersections or on routes by changing the timings of traffic lights according to vehicle density rather than fixed time periods. This project will produce an easy traffic management system where algorithms are used to regulate signals dynamically based on the volume of traffic in real time. The idea is to ensure that daily commuters save a lot of time by eliminating delays on the road. Traditional timed traffic signals always lead to bottlenecks, but our density-based method prevents this. By using real-time analysis, this approach helps deal with urban congestion, particularly in densely populated regions such as India. Since the algorithm reduces time lost in traffic, it provides a pragmatic, cost-effective solution that easily integrates with existing systems. This initiative offers a scalable approach to improving mobility and mitigating gridlock in cities.

Optimize traffic flow at a junction/along a route by enabling traffic light control based solely on vehicle count rather than time duration. This project aims to build a simple, easy-to-implement traffic control system using algorithms that vary signals based on the density of traffic present at an instant at that signal. By doing so, we want commuters to travel for hours on a daily basis to save as much time as possible while on the road. Traffic signals regulate traffic flow, but conventional timing methods often lead to congestion. Inspired by real-time traffic density analysis, our algorithm offers a dynamic solution. By prioritizing traffic flow over fixed intervals, we optimize junctions for smoother travel. Backed by research, our approach addresses the pressing issue of urban congestion, which is particularly relevant in high-density areas like India. With the potential to save countless hours lost to traffic, our algorithm offers a cost-effective, scalable solution. Together with existing infrastructure, it presents a practical step towards easing congestion and improving urban mobility.

Traffic signals are an integral part of the road network system of any city/state/country due to the apparent fact that they are used to control and maintain the smooth flow of traffic. However, the conventional methods used to operate these traffic signals are not as effective anymore due to the increased number of vehicles. This causes unavoidable traffic jams. Therefore, we have devised an algorithm that relies on traffic density at a particular traffic light rather than relying on time to regulate traffic flow at a multi-lane junction. Rather than allotting a specific time interval for each side of a junction at a traffic light, we analyze the density of traffic in a particular direction and, using our simplified algorithm and, based on the count, regulate the traffic lights between red signal and green signal. All of this happens in real time as and when more traffic keeps piling up.

Materials And Methods

Proposed methodology

The proposed traffic management system starts with a CCTV camera, which is usually present at most traffic signals in most traffic junctions in India. With rapid technological advancements and the reduced cost of cameras, we hope that even remote locations and villages will soon implement them, making our project future-proof. The footage captured by the camera, aimed at the vehicles present at the junction, is sent to the system for further processing. The system creates a virtual counter line on the road and feeds the footage to the algorithm for analysis. The proposed convolutional neural network (CNN) model for the system consists of a convolutional layer, max and min pooling layers, and a fully connected layer. Each layer has distinct parameters that can be optimized and performs a different task on the input data. After the footage is passed through this algorithm, the system can detect vehicles individually. Once a vehicle is detected and passes through the virtual counter line, the vehicle count increases by 1. We can now factor in various conditions, such as the presence of paramedics ahead, and dynamically change the vehicle count to vary itself based on the vehicle density. As soon as the input count is reached and the conditions are satisfied, we can turn the traffic signal green. This approach ensures that signals at busier intersections turn green more quickly, preventing traffic jams. The system architecture of the proposed model is shown in Figure 2.

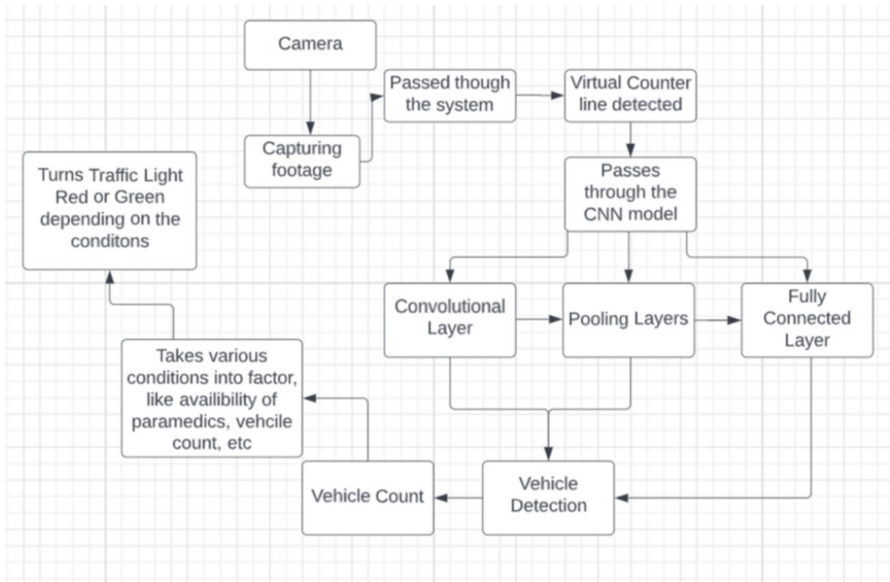

FIGURE 2: System architecture of the proposed model

Traffic signals regulate traffic flow, but conventional timing methods often lead to congestion. Inspired by real-time traffic density analysis, our algorithm offers a dynamic solution. By prioritizing traffic flow over fixed intervals, we optimize junctions for smoother travel. Backed by research, our approach addresses the pressing issue of urban congestion, which is particularly relevant in high-density areas like India. With the potential to save countless hours lost to traffic, our algorithm offers a cost-effective, scalable solution. Together with existing infrastructure, it presents a practical step toward easing congestion and improving urban mobility. Our approach involves the utilization of CNN to streamline the process of vehicle counting at specific intersections. CNNs, a deep learning neural network, specialize in image detection and processing, particularly in interpreting pixel-level input. CNNs are designed to extract features from input data and make accurate predictions by combining convolutional layers, pooling layers, and fully connected layers. Here, CNNs are chosen over other techniques like regression models or reinforcement learning (RL) for this specific application because they excel at processing image and video data, particularly for object detection, recognition, and classification.

Overview of CNN

The following points highlight why CNNs are a good fit for processing video footage in this context:

Suitability for Image-Based Tasks

Feature detection and extraction: CNNs are designed to automatically examine spatial hierarchies of features (like edges, textures, and objects) from images or video frame captures. That is why CNN is highly efficient in detecting and counting vehicles from traffic footage.

The potency for variations: CNNs handle variations in lighting, perspective, and blockages much better than traditional methods like regression models.

Real-Time Vehicle Detection

Unlike regression models, preferably for numerical predictions, CNNs specialize in visual data analysis.

This is crucial for accurately identifying the number of vehicles, which is the key input for dynamically adjusting traffic signals.

Comparison With RL

RL drawbacks: RL could be utilized to control traffic signals based on historical data and simulations, but it typically requires a predefined environment model and extensive training. RL systems might be usable for optimizing signal timings once the vehicle counts are obtained but are not inherently designed for image processing.

Complementary role: CNNs can be combined with RL to create an end-to-end system where CNNs handle perception (vehicle counting) and RL optimizes decision-making (traffic control).

Convolutional layers: Convolutional layers form the backbone of deep CNNs, where specialized filters or feature maps are applied to input images. These layers are responsible for detecting patterns and features within the image data. Parameters such as filter size, stride, and padding play a crucial role in determining the network's ability to capture relevant features effectively. Convolutional layers extract hierarchical representations of features by convolving filters across the input image, gradually learning to identify complex patterns. A block diagram of the CNN model is depicted in Figure 3.

Pooling layers: Pooling layers, typically inserted after convolutional layers, serve to downsample feature maps, reducing computational complexity while retaining crucial spatial information. Max pooling and average pooling are two standard pooling techniques employed in CNNs. Max pooling selects the maximum value within a defined region, highlighting the most prominent features. On the other hand, average pooling computes the average value within an area, smoothing out the feature representations. By incorporating pooling layers, CNNs achieve spatial invariance and robustness to variations in input data.

Fully connected layers: Fully connected layers, positioned at the end of the CNN architecture, serve as the transition from feature extraction to classification. These layers connect every neuron in one layer to every neuron in the subsequent layer, allowing for comprehensive information integration and nonlinear mapping. Before reaching the output layer for classification, fully connected layers aggregate the learned features and transform them into a format suitable for decision-making. By incorporating fully connected layers, CNNs can effectively capture complex relationships and nuances in the input data, enabling accurate classification and prediction.

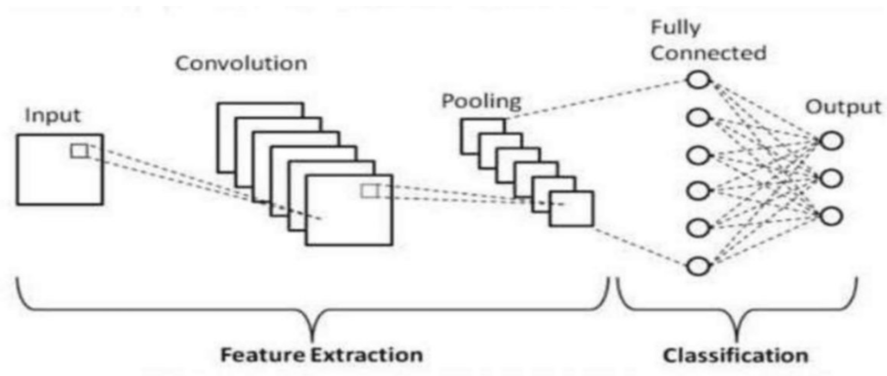

FIGURE 3: Convolutional neural network model

For a new, unfinished piece of software, a use case diagram is the primary means of expressing the system/software requirements that must be met. Use cases specify the expected behavior (what) but not the specific steps to take to achieve that behavior (how). Once use cases are determined, they may serve as a symbol for either written or graphical depiction (i.e., use case diagram). The potential of use case modeling to aid in system design from the user's perspective is a cornerstone concept. It is a helpful tool for explaining system behavior to consumers since it describes all externally visible behaviors inside the system. Use case diagrams are often easy to understand. Without displaying the details of the use cases, not all interactions between use cases, actors, and systems are outlined here.

CNN Architecture

The CNN architecture is designed for efficient processing and analysis of the input images for vehicle count estimation. It starts with a convolutional layer to extract hierarchical features from the images.

These layers use 3 × 3 kernel size, stride of 1, and ReLU as an activation function to bring important patterns and details in the data to the forefront. The layers after convolutional will have pooling layers, a form of max pooling layer that reduces the spatial features to dimensionality. This eliminates vital features while minimizing complex calculations and ensures efficiency while computing. Finally, through the fully connected layers, these extracted features are fed to provide a unified representation, which, through these layers, aid in the classification because of patterns learned in the past layers connected to the activity that needs to be taken care of: vehicle counting. Lastly, the output layer provides the numeric estimate of vehicle count. This output will be utilized to adjust traffic signal timings, thereby optimizing the traffic flow based on the real-time vehicle density.

Algorithm Overview

Input processing: The CNN receives input images captured by CCTV cameras at traffic intersections.

Feature extraction: Convolutional layers extract hierarchical features from the input images, detecting patterns relevant to vehicle identification.

Spatial pooling: Pooling layers downsample the feature maps, retaining essential spatial information while reducing computational complexity.

Nonlinear mapping: Fully connected layers aggregate the extracted features and map them to higher-level representations, preparing them for classification.

Classification: The final fully connected layer classifies the input images based on the learned features, determining the presence and count of vehicles at the intersection.

Algorithm Pipeline

It begins with preprocessing, in which OpenCV functions are used to prepare the input footage for analysis. First, the footage is converted to grayscale, reducing data to intensity values. Then, it uses Gaussian blur to reduce noise, thereby increasing the clarity of vehicle edges. The final step involves morphological transformations that enhance the contours of vehicles, making them more detectable. This is essential to ensure accurate counts of vehicles passing through the monitored area. A dynamic signal control system integrates the results of the vehicle count data. The scheduling algorithm processes the counts and adjusts the green signal time for each of the lanes, allowing for longer light times to lanes with higher density of traffic and optimizing traffic flow while mitigating congestion.

Results

Simulation environment

The original image once passed through the CNN deep learning algorithm goes through various packages from OpenCV, which initially grays out the background via cvtColor. Gaussian blur blurs the noise. Next, the video is dilated, and morphology is performed. Now, when the video/image moves from the pooling layers to the fully connected layer, it is able to detect the edges and is finally able to focus on the object to be tracked.

We will alter the image in order to produce better results. Here, the image will be converted.

Thereafter, we will use Gradient Blur to remove the rest of the noise from the picture. The Gaussian blur method may be used in image processing. It is also often used in the field of graphic design to smooth out the picture and reduce noise before it undergoes any more pre-processing. The Gaussian blur method may be used to reduce picture noise while simultaneously blurring the image. This pre-processing technique will be implemented by using Gaussian blur function ().

The photo will be blown up here. Dilation is a morphological strategy for filling up empty areas in pictures with elements, often known as kernels (structured fragments). Keep in mind that it is the polar opposite of erosion.

It is time to perform a morphological modification using the kernel. Here, we use a morphology-Ex strategy, which directs the function to carry out certain image processing operations. The second point is about the processes involved; for example, you may need circular or elliptical kernels. The morphology-Ex approach to OpenCV implementation will make use of the obtain structural element function.

There is now a need for a "car cascade" to detect vehicles. To get started, we need to upload the files to Colab (if you are using Colab, add the cascade files to the same folder) and set the path to the car cascade source. The photographs will be used to train a deep neural network in OpenCV using the built-in Cascade

Classifier function. To make the best use of this method, we must find several objects, such as cars.

Using the previously supplied contours, we will draw a rectangle around any automobiles we can reliably identify. You can see how it marks each automobile it locates with a red rectangle.

For the first time, a video will be used to identify and tally vehicles. To create a movie from still photos, we need to use the cv2.VideoWriter() function. The first required argument is a file path with the appropriate extension; the second is the output format's codec; and the last three are the number of frames per second, the width, and the height.

Discussion

Result analysis

The system was tested with real-world traffic footage from dense urban intersections in India. The testing environment was challenging and practical for evaluating the system's performance. To enhance its performance, pretraining datasets such as Cityscapes and KITTI were used. These datasets provided a strong base for diverse and labeled traffic scenarios. Some preprocessing steps are essential for improving vehicle recognition. These steps included applying Gaussian blur to reduce noise, using dilation to emphasize vehicle contours, and employing contour detection to effectively isolate and recognize vehicles in the video. The simulations were conducted on hardware featuring an Intel Core i5 CPU and an AMD Radeon R-430 GPU, which provided the necessary computing power to process traffic data and test the system in real time.

Figure 4 depicts the typical preprocessing steps used in a CNN model for object or lane detection in autonomous driving. The "Original" image shows a highway scene, presumably the input image. The "Canny edges" step detects edges in the scene, focusing attention on key structures such as lane lines. A "Blur" is applied, using a smoothing filter to eliminate noise and enhance features. Finally, the "Threshold with ROI mask" step applies an ROI mask, focusing on the relevant area - the road - and isolates it from the rest of the image for further processing. These steps help the model detect lanes and objects more accurately.

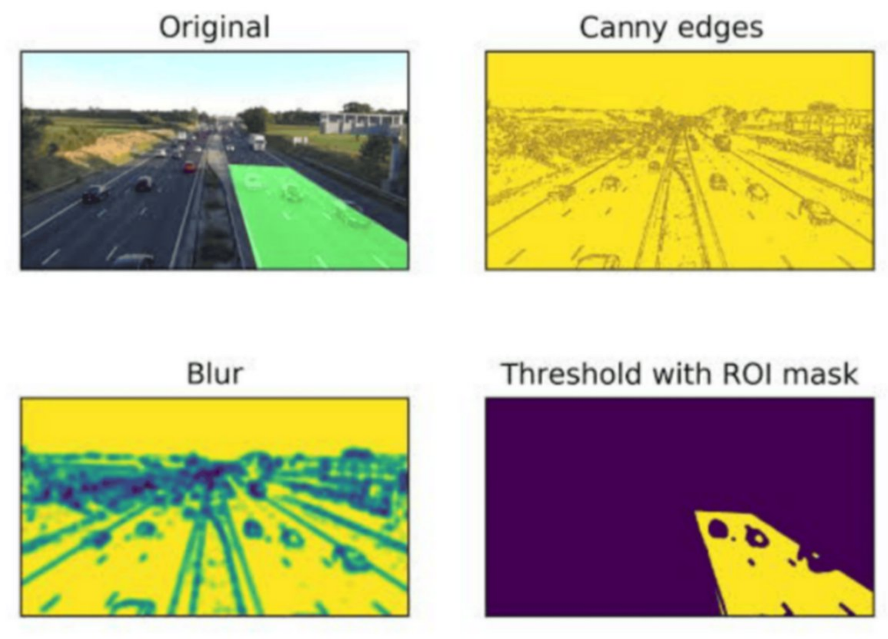

FIGURE 4: The steps involved in a convolutional neural network model

In Figure 5, a video shot from a flyover on the Cuttack-Bhubaneswar Highway in Odisha is shown. We successfully separated the background from the foreground. Using the CNN algorithm, we were able to detect the vehicle as an object, as shown in the third image. The video also captures a motorcyclist on the road within a regular video frame. It is important to note that the video has not been processed with any algorithm, meaning no detection, tracking, or analysis features have been applied to this frame. Unlike the previous images, this frame does not contain any bounding boxes, vehicle counters, or traffic signal indicators. This frame serves as a baseline, showing the raw, unprocessed video footage before any computer vision or algorithmic enhancements were applied for vehicle monitoring or counting.

FIGURE 5: Regular video, not passed through the algorithm

Once the object has been detected, it is passed through a virtual count line, and when the object crosses the count line, the counter increases by 1. Based on this counter, we can determine the number of vehicles at a particular traffic stop. This way, the traffic signal with the highest number of cars at the junction can be given a green signal, while that with least cars can be held longer, optimizing traffic flow since it is no longer time-dependent. In another video shot under low lighting conditions near Berhampur in Odisha, the traffic light changes from red to green based on various user-entered conditions. For example, if there is a paramedic center within a few kilometers of the area, the light will change from red to green faster than if there is none. The counter line is positioned at 350 px, with detection boxes of 60 × 60 px. The input video has a resolution of 480 × 720 px. We are collecting inputs from users for vehicle counts to optimize traffic flow, as it varies depending on location, and these inputs also consider other factors, such as the availability of paramedics.

The output of a processed video frame that has passed through layers of a CNN is shown in Figure 6. The processing draws attention to interesting features, transforming the background and the motorcycle rider into high contrast forms like this: with high contrast, close to the form of a binary, where edges are very evident and prominent as shapes. This creates the rider and motorcycle as a white silhouette against a dark background, thereby simplifying details in features that could be easily detected by the algorithm or recognized as an object. Such transformations, achieved through CNN processing, enhance relevant features while filtering out the noise of irrelevant backgrounds.

FIGURE 6: Video snippet after it passes through various layers of CNN

CNN: Convolutional Neural Network

A continuous video of 810 frames was used from the flyover for vehicle detection. For the given video, we separated the background, passed it through the CNN algorithm, and detected the vehicles passing. We then compared it with the conventional 1,920 × 1,080 video that is not a part of the traffic system algorithmic network. With a measured count of 85 in the original video versus 65 in the processed video at a given point, the percentage deviation was calculated to be 23.52% using the formula D = (Xm - Xt)/Xt × 100. Thus, our system was 76.48% accurate in detecting the vehicles. A conventional traffic stop holds vehicles for approximately 90 s. Using our algorithm, we were able to turn the traffic signal from red to green in less than 30 s. Analyzing two scenarios, our system reduced the average stop at traffic from 90 s to less than 30 s, and from 60 s to 17 s, relying on vehicle count. Our method is fairly close to the actual number of vehicles when compared to the original count. The system was still able to recognize even the tiniest objects in low-light situations.

The final output of an object detection system, where a person riding a motorcycle has been identified within the frame, is presented in Figure 7. The rider and motorcycle are enclosed in a red bounding box, indicating that the algorithm has successfully recognized and localized the object of interest. What might be seen is a processed output from a trained neural network that detects vehicles or people on the road. Such bounding boxes are used to track or monitor objects, which is especially useful in autonomous driving or analyzing traffic patterns. The background and road remain unchanged from the original image.

FIGURE 7: Final output where the object is being detected by the vehicle

Two frames of a vehicle counting system are depicted in Figure 8. In the first frame, no vehicle was present within the region of interest, and the vehicle counter displays "0". In the second frame, a vehicle has been identified entering the detection region and is enclosed by a red box, while the value in the counter has updated to "1" with the vehicle now present. This setup shows how the system automatically counts vehicles as they enter the area on the road defined for detection. The differences between the two frames indicate that the change in the counter is from 0 to 1 upon the detection of a vehicle.

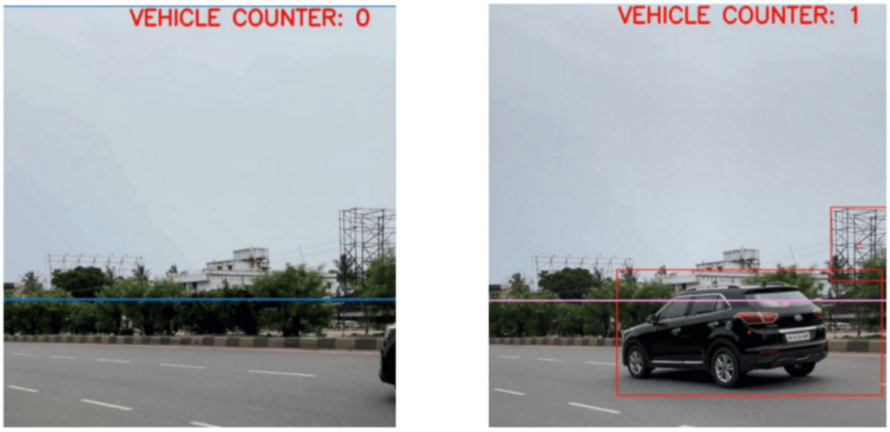

FIGURE 8: Representation of vehicle count changing from 0 to 1

Figure 9 presents a traffic monitoring system that detects the presence of moving vehicles and reacts to changes in the traffic signal. In the first frame, the light is red, and the vehicle counter shows "6," indicating the presence of a paramedic vehicle. The second frame portrays the light as green, with the vehicle counter now reading "49" due to additional vehicles detected as they move through the intersection. The system tracks vehicle movement using red bounding boxes and evolves with changes in the traffic light, thus helping to monitor and provide priority to certain vehicles, such as emergency responders, at the crossing.

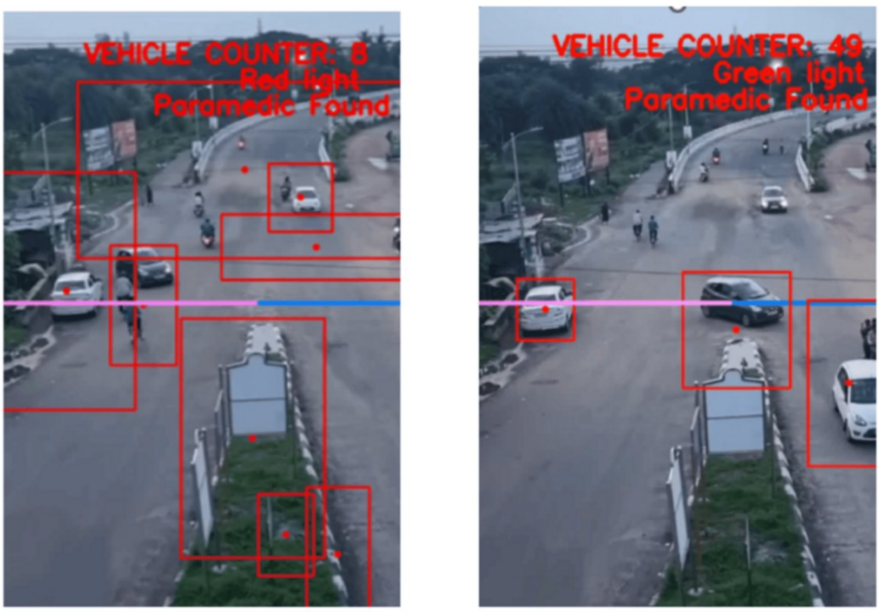

FIGURE 9: Representation of traffic light changing from red to green

Overall outcome

The system demonstrated excellent detection accuracy with an intersection over union score of 92% for vehicle detection. This means the system could identify vehicles accurately and uniformly, with minimal overlap or misclassification errors. In terms of computational efficiency, the system can process traffic footage at 24 frames per second. This real-time performance ensured that vehicle detection and signal adjustment could be done promptly in busy urban environments. The system produced significant improvements in traffic flow, reducing the average waiting time for vehicles at intersections by 40% compared to traditional fixed-schedule traffic signal systems. This indicates the effectiveness of the system in dynamically managing traffic congestion and optimizing the timings of the signals.

The system allows for efficient urban traffic management through its dynamic adaptation of real-time traffic conditions. It prioritizes high-traffic lanes, thus cutting down delays and improving overall road efficiency. Its flexibility to different traffic patterns ensures smooth and effective traffic management. Future enhancements of the system can also include additional features to enhance its capabilities. One of the improvements will be the inclusion of emergency vehicle detection, which would be made using unique visual marks or specific models to identify ambulances, fire trucks, and other priority vehicles. This feature would result in faster response times to emergencies. Another possible upgrade will be weather adaptation, by incorporating weather application programming interfaces. This feature would make the system adapt to signal time variations during adverse weather conditions like heavy rain or fog for even smoother and safer traffic flow.

Environmental impact

Optimized traffic flow significantly reduced vehicle idle times, bringing an approximate annual reduction of 15% in the emission of CO_2 per intersection, thereby contributing to improved air quality and environmental sustainability.

Conclusions

By saving approximately 1 min on every single cycle at every traffic signal, throughout the day, we can save hours of time for the commuters who would otherwise waste so much time waiting while being stuck in a traffic jam. We all know how valuable time is and this extra time saved can be utilized productively, and there will also be lesser pollution since the engines of the vehicles would have to be turned on for a lesser duration. While image detection using CNN is common, our system goes beyond the conventional system by facilitating object detection in real-time using a video input over conventional set of images. Our system was able to reduce this traffic wait times by up to 75%. While conventional traffic system relies just on time, we have not only incorporated vehicle density into consideration but also taken other factors like availability of paramedics nearby to alter the traffic optimization accordingly.

We were successfully able to carry out a simulated traffic scenario wherein the flow of traffic is controlled by a custom-built algorithm devised by us. Using this algorithm, we were able to achieve the free flow of traffic, and when compared with the conventional method, we were able to save an average of 60 s per cycle. If implemented on a large-scale basis in partnership with the traffic department, this algorithm can solve a major problem that all of us have been facing for years. With the increase in the number of CCTV cameras across traffic signals pan India, starting from cities and moving on to smaller villages, the proposed system can be expanded to a good reach within the coming years. In the future, we plan to work closely with the traffic department in order to implement the algorithm in real time at one particular junction on a test basis before the complete roll out. We also plan on adding some incentives like emergency vehicle detection and evacuation, alternate route suggestions, and weather-based route updates with the inclusion of bad road alerts and pothole alerts through a mobile application. The system effectively addresses multiple-lane congestion by implementing an algorithm that resolves conflicts through weighted prioritization. This approach gives higher priorities to lanes with higher cumulative vehicle counts, ensuring that traffic flow is optimized across all lanes while minimizing overall delays. Environmental challenges, such as varying weather conditions and lighting changes, are managed using adaptive image preprocessing techniques. Methods such as histogram equalization are used in enhancing brightness and contrast so that system performance is accurate with good vehicle detection regardless of external conditions.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Sanjoy Mondal, Ipshita Chatterjee, Trishita Das

Acquisition, analysis, or interpretation of data: Sanjoy Mondal, Ipshita Chatterjee, Trishita Das

Drafting of the manuscript: Sanjoy Mondal, Ipshita Chatterjee, Trishita Das

Critical review of the manuscript for important intellectual content: Sanjoy Mondal, Ipshita Chatterjee, Trishita Das

Supervision: Sanjoy Mondal

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Li Z, Li C, Zhang Y, Hu X: Intelligent traffic light control system based on real time traffic flows. 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA. 2017, 624-625. 10.1109/CCNC.2017.7983196
2. Kai Y: Intelligent traffic control system based on visible light communication. 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan. 2020, 52-53. 10.1109/ECICE50847.2020.9301946
3. Ghazal B, ElKhatib K, Chahine K, Kherfan M: Smart traffic light control system. 2016 Third International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA), Beirut, Lebanon. 2016, 140-145. 10.1109/EECEA.2016.7470780
4. Shinde SM: Adaptive traffic light control system. 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India. 2017, 300-306. 10.1109/ICISIM.2017.8122189
5. Chen XF, Shi ZK, Zhao K: Research on an intelligent traffic signal controller. Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, Shanghai, China. 2003, 1:884-887. 10.1109/ITSC.2003.1252076
6. Li Z, Shahidehpour M, Bahramirad S, Khodaei A: Optimizing traffic signal settings in smart cities. IEEE Transactions on Smart Grid. 2017, 8:2382-2393. 10.1109/TSG.2016.2526032
7. Wang H, Wei X, Yao J, Zhang Y: Traffic flow prediction using machine learning methods. 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China. 2021, 30-35. 10.1109/MLBDBI54094.2021.00014
8. Meena G, Sharma D, Mahrishi M: Traffic prediction for intelligent transportation system using machine learning. 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), Jaipur, India. 2020, 145-148. 10.1109/ICETCE48199.2020.9091758
9. Xu Y: Machine learning based traffic prediction and congestion control algorithms in software defined networks. 2024 International Conference on Interactive Intelligent Systems and Techniques (IIST), Bhubaneswar, India. 2024, 285-289. 10.1109/IIST62526.2024.00035
10. Rao SG, RamBabu R, Kumar BSA, Srinivas V, Rao PV: Detection of traffic congestion from surveillance videos using machine learning techniques. 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Dharan, Nepal. 2022, 572-579. 10.1109/I-SMAC55078.2022.9987342
11. Manisha V, Bharti S, C Naveen K: Comparative study of machine learning algorithms for traffic accident prediction and prevention. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India. 2024, 1-5. 10.1109/ICCCNT61001.2024.10723916
12. Hossain MN, Ahmed N, Wazid Ullah SM: Traffic flow forecasting in intelligent transportation systems prediction using machine learning. 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India. 2022, 1-5. 10.1109/INCOFT55651.2022.10094319
13. Chaoura C, Lazar H, Jarir Z: Predictive system of traffic congestion based on machine learning. 2022 9th International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco. 2022, 1-6. 10.1109/WINCOM55661.2022.9966448
14. Kafy MA, Faisal SI, Rahman ML, Moni R, Shanmuganathan H, Raza DM: Traffic congestion prediction using machine learning. 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India. 2024, 1290-1295. 10.23919/INDIACom61295.2024.10498418