A Smart Traffic Management System Powered by Artificial Intelligence Designed to Optimize Traffic Light Timings in Real-Time for Congested Areas.

Mr. Arun Kumar K,
Assistant professor
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
arunkumar.k.it@sathyabama.ac.in

Aswin S
UG Student
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
aswinkpkm@gmail.com

Christo Sham K,
UG Student
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
Christosham05@gmail.com

Ms. L. Mary Gladence, Professor
Department of Information Technology Sathyabama Institute of Science and Technology
Chennai, India
Marygladence.it@sathyabama.ac.in

***Abstract*— The proposed development plan is for an AI-assisted smart traffic control system. This system would employ live camera images from traffic surveillance cameras installed in cities to monitor, assess, and optimize traffic flow at the city's traffic junctions. OpenCV technology would be employed in this project to constantly assess live video taken from traffic cameras installed at cities to detect cars, ascertain the level of congestion, observe patterns of traffic, and gauge traffic jams across every traffic intersection in real time. The system utilizes smart image analysis and machine learning algorithms to evaluate traffic flow from all directions and then optimizes traffic signal timings accordingly in an instant. Unlike traditional fixed-time traffic signals, which follow an installed schedule, this auto-learning system would instantly react accordingly to prevailing conditions, allocate relatively equal green time to congested routes, and decrease idling time at lesser-congested routes. The system would work continuously, learning and adjusting accordingly according to prevailing rush hours, accidents, events, or weather. Its positive impact would include reducing traffic jams, fuel, and emissions, and idling time as well, and make roads much safer. It can be easily scaled up and installed at all city traffic crossings with ease.**

Keywords:

AI-enabled smart traffic management, real-time traffic monitoring, computer vision, OpenCV, live camera feeds, traffic density estimation, vehicle detection, congestion analysis, adaptive traffic signal control, dynamic light timing optimization,

I. INTRODUCTION

However, as the size of cities gets larger, with roads filled with cars, it becomes impossible for commuters to avoid a traffic jam if they have to commute through it. What can be done by FIXED TIME SCALES AND HUMAN EFFORT cannot be varied according to changes that occur every minute, especially when their are more crossroads.

However, the arrival of computer vision and artificial intelligence changed this scenario. Computer vision responds to traffic problems in time by quickly assessing what is happening in the particular traffic environment. We are able to watch traffic conditions around the clock by assessing images. Images provided by OpenCV help us understand exactly where the traffic is and to what extent, allowing it to vary according to road user behavior.

This proposal will look at the intelligent traffic control system whereby the system will utilize artificial intelligence to analyze the traffic patterns using OpenCV and live video captures by the cameras. The idea in this case will be to learn and program the traffic light schedules in real time depending on the observations the system will be able to get from the traffic flow patterns.

II. LITERATURE SURVEY

Basically, the major hallmarks of traditional traffic management system was fixed-time signalization. Those three phases; green, yellow, and red were manually set via looking at past flow volume information and manually counting road users in inter

Recently, research has shown that classic fixed-time signals method lacked situational awareness due to the fact that incidents like rush traffic, accident, or colder weather cannot be adjusted in real time based on the flow of traffic volume to avoid traffic congested lanes and those leading to traffic delays and increasing fuel consumption and emissions.

When digital imaging and computer vision progressed surveillance cameras and other sensors were used as an alternative to monitoring traffic Scholars found it feasible to utilize video data from surveillance cameras for traffic monitoring, as opposed to traditional sensors. There have been various research studies conducted on image processing methodologies such as background subtraction, motion detection, edge tracking and object tracking to detect vehicles and estimate traffic density from raw video feed The availability of OpenCV permitted the application of real-time traffic analysis more easily and quickly due to more efficient algorithms. Camera sensors were much cheaper not only in installation but also in (maintenance), besides sight aspect.

As the study progressed, the researchers came to a realization on how to combine the capabilities of machine learning techniques with deep learning. The CNN became the best technique to analyze vehicles following the advent of the latest trends in deep learning. The objective was analyzing vehicles as they were moving along the roads despite the existence of challenging environmental conditions such as occlusions and illumination. Advanced level detectors used in the existing studies consisted of YOLO, Fast R-CNN, and SSD.
After the successful implementation of reliable car detection, scientists continued to work in developing adaptive traffic signals, ensuring optimum timing of green, yellow, and red signals. Starting from rule-based systems to fuzzy logic, genetic algorithms, and reinforcement learning, various optimization techniques have been used to analyze the reduction in waiting and queue length at the traffic signals during the experiments. In one of the experiment trials, priority systems have been introduced for emergency and public transport vehicles.

A new study investigates how AI-based traffic management might be integrated into the smart city environment. Consider a scenario in which cameras and IoT sensors, working in harmony with each other and cloud computing and traffic management centers, ease congestion on city roads. It is also envisioned that projects like these will have positive results in terms of being good for the environment-reduced pollution and fuel emissions due to reduced traffic congestion. In short, smart and camera-based traffic management seems to have many benefits related to environmental soundness and cost-effectiveness for solving complex urban transport problems.

Some researches also involve analyzing the combination of signals from several cameras and the information exchange between multiple intersections to enhance system efficiency. Particularly, it should be noticed that separate optimization of each intersection may result in congestion of roads adjacent to an intersection. That is why Coordinated Signal Control using synchronized video signals from multiple cameras over multiple intersection points. Based on the upstream and downstream traffic conditions, the green waves ensure the smooth flow of vehicles on the primary roads. The result indicates that the coordinated strategies increase overall speeds and commuter satisfaction while lowering stop-and-go phenomena.

An important and developing field of current research work includes the application of AI technology for traffic management. Some factors that researchers emphasize as important for the application of AI for computer vision traffic analysis include the effects of the brightness of night, shadows, and occlusion by larger vehicles. Edge computing can also be considered as an emerging trend.
Many research works also focus on the effects of intelligent traffic control systems on the social, economic, and environmental aspects of such cities. Results show that these systems are capable of optimizing fuel consumption, reducing carbon dioxide emissions, and minimizing air pollutants due to reduced idling times. Economic studies show that they are also economically viable due to reduced waiting times in traffic.

III. METHODOLOGY

Intelligent traffic starts with data collection from CCTV cameras put in place at key junctions where traffic signal and flow is regulated. As they relay live video feeds from various angles in parallel, of course few cameras. Footage is also analyzed concurrently to ensure that for data processing minimal delay time is experienced. Unlike many other setups, it relies less on presence of physical sensors; making it a possible roll out on metropolitan-wide scale.

The next step is to do post processing CASTSBATCH [lakukan pos proses] on the captured images to enhance vehicle detection accuracy. This includes actions like image resizing, noise cleaning for enhancing detail sharpness, conversion in to gray scale, and its contrast stretching for better clarity. Most of these steps are covered through OpenCV – the very reasons why the complete system can exhibit real-time performance.

In the third phase, we focus on vehicle identification, tracking and density of traffic estimation by employing computer vision and artificial intelligence. This is where sophisticated object detection algorithms recognize cars in each frame and segregate them by size and type as well. It measures traffic density by way of counting the detected vehicles and determining lane occupancy. That density information is being updated in real time to show the current traffic situation at each intersection accurately. Vehicle detection that correct and reliable is crucial for properly analyzing traffic flow .

The 4th stage compress reduce of the traffic signals based on live count of the flow. To this effect, an adaptive algorithm at each directional green shall adjust its time span with respect to volume at every point in consideration to ensure fair and efficient flow. In this case, this system prioritizes heavily congested lanes and eliminates useless delays on routes with less traffic. The adaptive control will respond quickly to changed characteristics and thence optimize junctions in reducing congestion.

It does this to ensure the system provides for continual monitoring, feedback, and scalability. Continuous reassessment of traffic conditions occurs, and real-time changes in the timings of the signals are made to maintain optimum flow in traffic. A system with such architecture can be easily scaled up for multi-intersection networks and centralized traffic management platforms for city-wide traffic management. Performance metrics like wait time, queue length, and throughput are recorded for various analyses and further enhancements. Therefore, it allows for a sound, scalable, and intelligent backbone for these modern traffic management systems.

The other aspect of this approach is learning about traffic congestion and even predicting it. The technology observes a series of time intervals of how the vehicles are moving in order to determine the speeds and directions of the vehicles, as well as where the cars congregate. The technology observes changes over a period of time to be able to detect typical congestions or disruptions in traffic flow. It helps the technology predict when the traffic is about to become congested so that it actions the traffic signals earlier to prevent congestion.

Behind the technology is the integration of components and processing in real time. This involves the detection of traffic and the optimization of signals that are brought together in one process, which makes the flow of data seamless and rapid. This technology applies the best processing of multithreading and buffers that receive video streams with minimal dropped frames. This is developed to run in real-time, which updates the status of traffic signals within the required time frames that are required during road safety operations.

The last step emphasizes the evaluation and verification process of the efficiency of the system. This method will be tested under various traffic, during peak hours, mixed traffic patterns, and even unanticipated traffic congestion. The performance indicators include average delay, queue size, and signal efficiency, and it will be compared to the fixed-time systems. The outcomes prove the benefits of this proposed technique and provide a feasible solution to intelligent traffic control systems.

System Architecture

Presented herein is a breakdown of the entire journey of the smart railway violation detection system-from where the data is grabbed to developing the Django Web application. This happens in an intricate web where machine learning, web development, and analysis of the data in real time intersect.

1) Data Collection and Processing

The images are obtained from “Robo flow Universe”, a high-quality dataset source. The videos showing the trips on the footboard comprise the dataset, as well as other necessary images for training. Collected images are further added into the dataset repository and enter into the process of image augmentation, labeling, resizing, and normalization.

2) Training and building models: YOLOv8 & CNN

Having prepared the data, two models can be created:

YOLOv8, for real-time detection of the violation of the footboard rule in context.

A CNN used in face recognition and gender classification for the purpose of passengers' verification.

3) Python & Django Backend Development

The trained models are stored in .pt files for YOLO and .h5 files for the CNN. Python is used to facilitate the integration of the trained models and the backend of the app. Django acts as the framework for deployment.

"Key features include:

But what does the future hold?

Loading a model and making predictions

- OTP Authentication

Facial Recognition and Gender Classification

- violation detection logic
- Database operations

It is necessary to have an alert system because it helps ensure that the backend environment is strong and secured.

4) Front end development: HTML, CSS, JavaScript

The elements of GUIs are built by using:

- Tkinter

- HTML 5 markup

- CSS for styling, responsive web design •JavaScript for interactivity Front-end pages include: Landing, Registration, and Login* Home/Dashboard - Order booking - Real-time Footboard Display These pages are associated with the Django backend system.

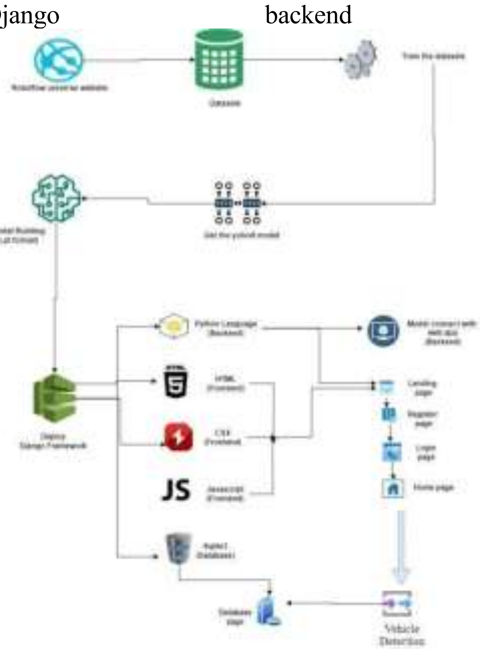

Fig : Architecture Diagram

3. Integrating Model with Web Application

This project integrates a trained YOLOv8 and CNN model in a Django backend. Such a system would enable real-time functionality such as:

Facial Recognition at Login

Classifications concerning gender

Train Footboard Tra vel Detection using live camera feeds.

These predictions, in turn, reinforce user verification to improve safety among passengers.

4. User Workflow: Login → Booking → Detection

It is here that the user journey begins-the landing page.

a. Registration and Login

OTP authentication helps users to log in to secure pages. - When a user logs in, a picture of them is captured, hence verifying their identity by face recognition and gender detection. b. Ticket Booking Successful subscribers, upon verification, are taken further for ticket booking to book their travel. c. Real-Time Monitoring Video streaming by cameras on the train is done continuously during the transit. - The YOLOv8 model detects footboard violations or unsafe behavior. In case a violation is detected, the system creates an alert, logs this incident in a database, and can optionally send email notifications to authorities/staff.

Data Analysis:

We can improve our heartbeat detection by incorporating recurrent neural networks such as Simple RNN, LSTMs, which are recognized for their effectiveness in detecting patterns in audio. Librosa is a library which provides functionality to extract essential audio elements from the heartbeats record, namely the heart rate, tone, and rhythm. Relying on all this information, our models can distinguish normal heartbeats from the unhealthy ones more effectively. It can be implemented using a Django application, which would improve user engagement by allowing users to interpret or even foretell the result of their heartbeats from the audio output obtained from the application.

MODULE DIAGRAM

GIVEN INPUT EXPECTED OUTPUT
Input: data
Output: visualized data

Algorithm implementation:

It is essential to be able to measure how well various machine learning algorithms work. You can create a testing platform using Python and scikit-learn to pit several algorithms against each other. This testing platform can be your boilerplate code for various machine learning applications, helping you pit several algorithms against each other. Every algorithm will work in its own peculiar way, and tools such as cross-validation help you understand their performance on new datasets, helping you shortlist two or three algorithms.

For instance, when you come across a new data set, you pause to evaluate the data. You look at the data in different perspectives to get several pieces of information. The same approach should be considered when making decisions on the various models to use. For example, you need to determine approximately how accurate the various models of ML that you want to try may be. Various data visualizations may aid in the determination of this approximate accuracy.

LeNet7 is considered to be a prominent precursor to convolutional networks. This network was built by Yann LeCun and his team during the early 1990s. This was an improvement over previous machine learning techniques that could not detect handwritten digits, like those on checks and postal addresses.
Its architecture consists of multiple levels:
- Input Layer: This layer accepts an image, which is normally a 32x32 image of a digit in grayscale form.
- Convolutional Layers: The network applies two convolutional layers to the image to extract features from the image. The convolutional kernels traverse the image to determine the features.
- Sub-sampling/Pooling Layers - Following each and every convolution layer, there exists a pooling layer that reduces the size of the feature maps.

- Dense Layers: The resultant feature maps are flattened and arranged as inputs to dense layers, the standard layers of a neural network. - Output Layer: The final fully-connected layer houses the predictions, which include ten neurons for the digits 0-9. Although LeNet is one of the earlier works in deep learning, predating the deep learning revolution in 2015, and utilizes much less complex models than the giant models such as ResNet, VGG, and Inception, its foundational elements such as convolution, pooling, feature extraction, or multi-layer processing brought about the deep learning revolution in the late 1990s and early 2000s when deep learning models were able to perform image classification.

YOLOV8 :

YOLOv8, from the company Ultralytics, targets fast and efficient object detection in videos. At the same time, they strive to make progress in image detection as well. To optimize image and video detection, they introduced updates to drag out as much performance as they possibly could. This too is a single-pass detector.

What's worth noting is that its strength lies in its new skeleton that is built according to a CSP-inspired pattern with an additional convolutional layer that aims to extract distinctive features more efficiently with fewer computations. Moreover, the decoupled head enables faster convergence during training because it tackles both classification and localization simultaneously but through individual trainings. The decoupled head is anchor-free and has a default configuration of scaling and aggregating features.

A novel loss function is implemented in YOLOv8, which merges Distribution Focal Loss with an IOU -based box localization loss to make more precise boxes predictions. For example, this improved training pipeline is further fortified with the help of mosaic augmentation, mix-up, and Strong Label Smoothing in order to make it more robust and highly adaptive towards noise perturbation, changes in lighting, partial occlusions, and varied background scenarios. Such attributes are extremely beneficial when dealing with railway alignment systems, where camera angles, lightening conditions, as well as changes in passengers poses are never the same throughout. YOLOv8 is available in various sizes: n, s, m, l, x, enabling users to pick versions precipices on their speed versus model accuracy trade-offs.

The architecture also allows for multitasking and can perform object detection, image segmentation, and key point detection that shows how YOLOv8 is a more flexible model for a variety of computer vision applications. And is deployable as a model easily due to its support for ONNX, Tensor RT, and Open VINO and also can be run on platforms ranging from large cloud devices to real server devices. YOLOv8, when it comes to real-time surveillance can process multiple frames per second (when less it maintains a high level of accuracy) which is perfect for constant monitoring and instant detection of any misbehaviour. The software is extremely lightweight which results in it being highly performant and can be easily added to backend systems like Django for seamless integration into web-based safety monitoring applications. Kicking off with unsafe foot boarding travel

Deployment:
Django (Web Framework):

- ➢ Django is a micro web framework developed in Python.
- ➢ It's a micro-framework and does not carry any special tools or libraries.
- ➢ It doesn't have a sub-caste of database abstraction, form confirmation, or other factors in which being formerly third-party libraries provides common functions.
- ➢ does, however, support extensions, which offer the addition of operation features as if added to Django directly.
- ➢ Extensions involve object-relational mappers, create confirmation messages, implement the uploading of running, enabling colorful open authentication technologies, among general frame-related gadgets.

IV. RESULT AND DISCUSSION

This smart system for business operation grounded on AI was evolved with live videotape aqueducts and with pre-stored business scripts to define its capability in detecting vehicles and analyses of business inflow. This system successfully detected vehicles under different business conditions ranging from low business to high- viscosity business The vehicle discovery model YOLOv8 showed good discovery for colorful vehicles similar as buses, machine, and motorcycles indeed in grueling lighting conditions or partial occlusion. The results confirm the systems robustness and trustability for real business monitoring. In order to measure the impact of the system, we calculated the standard object discovery criteria similar as perfection, recall mean Average Precision(chart). As the experimental result shows that the trained model achieved remarkable discovery with veritably low false cons and negatives. The system performs real- time conclusion.

The results of business viscosity estimation were that the system estimated lane operation and the number of vehicles present in colorful directions at an crossroad. Grounded on this, the adaptive signal timing system acclimated green light timings to favor the lanes with high traffic. The conducted simulations and real- time tests show a considerable reduction in average delay times and line lengths, when compared with fixed- time business light systems, hence revealing the eventuality of the system in effectively managing the inflow of business.

In this aspect, the objectification of the discovery model in the Django- grounded web operation assured that any commerce by the stoner was smooth and results of discovery were shown in real time. Real- time bounding boxes of vehicle findings along with archived data of business were made available to the druggies on an systematized web interface. All records of discovery and literal data related to business were efficiently stored within the database module, enabling farther analysis of business trends and effectiveness of the system over time. Despite compelling performance during the evaluation process, a limitation arose. Poor visibility in bad rainfall conditions, similar as heavy rain or fog, slightly reduced the delicacy of discovery. either, camera placement and angle came critical to carrying good discovery performance. These aspects can be further bettered by better estimation of cameras, data addition, and model fine- tuning. Overall, the results indicate that the proposed system is doable, scalable, intelligent for real- time vehicle discovery, and smart operation of business inflow, but with room for farther advancements and wider connection in metropolises.

V. Conclusion

This article presents an idea for the development of an intelligent traffic management system using AI. It uses live camera feeds with computer vision and deep learning to detect traffic patterns and classify the flow of traffic. This is a combination of the YOLOv8 object detector, OpenCV, and Django, which are put together for Web development; hence, the system can visualize real-time traffic density calculations with effective and exact vehicle detection. The key conclusion derived here is that camera-based intelligent traffic systems may be more effective in the near future compared to traditional traffic sensors.

The adaptive analysis herein outperforms fixed-time signals by adjusting green-light duration according to current traffic volume, helping keep intersections smoothly flowing without roadway bottlenecks. The positive integration of backend processing, frontend visualization, and database management into an easy-to-work-with system is that the user can view, through a web interface, the current road conditions with associated historical data.

Of course, limitations are also noted in the study. Performance may vary according to environmental conditions and even camera placement. Those constitute the future scope of work that may be done: increasing the dataset, building more sophisticated models, and using multiple cameras for increased robustness. In a nutshell, the system proposed is indeed scalable, intelligent, and affordable for the modern city traffic control system and thus should demonstrate bright perspectives of real-world deployment to improve traffic safety, reduce congestion, and increase the efficiency of transport.

References

1. Hu, P. & Ramanan, D. (2017). Chancing bitsy faces. By Proceed of the IEEE paper on CV . In this paper, the authors present a strong technique for finding very small faces in images. They use high-resolution images and detectors that work well for different sizes of faces.
.
2. Yang, S., Luo, P., Loy, C.C., & Tang, X. (2016). WIDER FACE: A face discovery standard. In Proceedings of CVPR. This dataset is one of the most commonly used and includes a large number of facial images. It is used to train models that can detect faces in different situations.
3. "World's largest selfie." This article is available online at GSM Arena and talks about the event where the biggest selfie in the world was taken. The link is https://www.gsmarena.com/nokia_lumia_730_captures
4. Single-photo density count via multi-column CNN . In Proceedings of CVPR. The paper introduces the MCCNN model, which is designed to count people in crowded images. It especially helps when people are of different sizes.
5. Wu, B., and Nevatia, R. (2005). Discovery of multiple, incompletely occluded humans in a single image using a Bayesian combination of edge let part detectors. In Proceedings of ICCV. This work presents a Bayesian framework for detecting people who are only partly visible in images.
6. Viola, P., Jones, M. J., and Snow, D. (2005). Detect a patterns of moving and present . forgien Journal of CV (IJCV). A Basement approach join movement and present cues for detect .
7. Wang, M., and Wang, X.(2011). Automatic adaption of a general rambler sensor to a specific business scene. By Proceed of the IEEE paper on CV . This work focuses on perfecting rambler discovery delicacy by conforming models to specific surroundings.
8. Idrees,Soomro.H,and M.Shah(2015). Detecting humans in thick crowds using locally- harmonious scale previous and global occlusion logic. IEEE Deals on Pattern Analysis and Machine Intelligence(TPAMI). A fashion for handling traffic and occlusion in extremely thick crowds.
9. Idrees, Saleemi,H Seibert,I and I. Shah,(2013). Multi-source multi-scale counting in extremely thick crowd images. In Proceed of the IEEE paper on CV. This exploration introduces a robust viscosity estimation frame for large- scale crowd counting.
10. Zhang, C., Li, H., Wang, X., and Yang, X.(2015). Cross-scene density count via deep CNN . By of the IEEE Paper on CV . This paper presents a CNN-grounded approach that generalizes crowd counting across different surroundings.