Smart Control of Traffic Light System Using Artificial Intelligence

[*] P.Soumya, R.Poojitha, P.Suveena, ** Mr.K.N.S.Ganesh, ** Dr. Gobinda Prasad Acharya
[*]4th Year Students of Electronics and Communications Department, Sreenidhi Institute of Science and Technology (SNIST) Yamnampet, Ghatkesar, Hyderabad- 501 301
*** Professors in ECE Dept, Sreenidhi Institute of Science and Technology (SNIST) Yamnampet, Ghatkesar, Hyderabad- 501 301*

***Abstract*— Traffic congestion is fast becoming one of the major issues with increasing population and automobiles in cities. Traffic jams not only add to the delay and frustration of the drivers, but also contribute to fuel usage and air pollution. Though it seems to be everywhere, mega cities bear the maximum brunt of it. And its ever-increasing nature requires the computation of real-time traffic density on roads for better signal control and effective traffic management. Traffic controller is one of the deciding factors in traffic flow. Therefore, the need to optimize traffic control to address this increasing requirement arises. Our system will employ live feeds from the cameras at intersection points to compute traffic density through image processing and artificial intelligence. It also aims at the algorithm for adjusting the traffic light switch according to the number of vehicles to reduce congestion, hence faster travel to people and reducing pollution.**

Keywords: Traffic Density Estimation, Intelligence traffic signal Control, Real Time image processing, Artificial intelligence.

1.INTRODUCTION

Urban areas are increasingly burdened with heavy traffic due to growing populations and vehicle ownership. Traditional traffic light systems, which operate on preset timings, fail to adapt to real-time traffic fluctuations. This leads to significant delays and inefficiencies, particularly during peak hours or unexpected events like accidents or road closures.

AI-powered traffic systems offer a solution by analyzing live video feeds and adjusting signal timings accordingly. These systems can make real-time decisions based on vehicle density, reducing congestion and improving travel times. This approach also supports environmental goals by minimizing unnecessary idling and fuel consumption.

The conventional fixed-timer systems are outdated for today's dynamic traffic conditions. They often cause vehicles to wait unnecessarily, leading to driver frustration and elevated emissions. In contrast, AI-driven systems can predict traffic flow and react instantly, providing a smoother and more efficient traffic experience.

The primary goal of this project is to design a smart traffic management system that can dynamically adjust traffic lights based on live data. Key objectives include real-time vehicle detection using computer vision, minimizing wait times, enhancing system scalability, and promoting eco-friendly transportation. Modern AI traffic systems use machine learning to adapt continuously to evolving patterns. This reduces the need for manual intervention and makes traffic control more responsive and intelligent. Such systems can even anticipate future congestion and act proactively.

Incorporating AI, computer vision, and real-time analytics allows for smarter traffic light scheduling. These systems learn from past traffic data and use it to optimize current and future performance. Their adaptability makes them more effective than rule-based systems.

Overall, the integration of AI into traffic management marks a transformative shift. It empowers cities to respond swiftly to real-time challenges, enabling safer, greener, and more efficient urban mobility. Traffic congestion is not just a transportation issue but a socio-economic problem that impacts daily productivity and quality of life. Long delays on the road translate into lost work hours, increased fuel costs, and heightened stress among commuters.

Traditional signal systems are unable to react to unplanned traffic situations such as emergency vehicle passage or temporary road closures. This inflexibility underlines the need for an intelligent system capable of real-time response and prioritization.

By using computer vision and machine learning, AI can mimic human-like decision-making to assess traffic conditions and act instantly. This represents a major advancement in how cities manage road networks dynamically and intelligently.

2.LITERATURE SURVEY

Traditional traffic systems rely on pre-set timing plans that remain static regardless of actual road conditions. These systems use historical data and human-defined rules, making them inflexible and inefficient when real-time adjustments are needed. They cannot handle unexpected events like accidents or variable traffic surges effectively.

Some cities have adopted semi-adaptive systems using sensors like inductive loops to detect vehicle presence. These systems offer minor improvements but remain reactive, unable to predict upcoming congestion. Their updates occur only after vehicles are detected, which limits responsiveness and planning capabilities.

Intelligent Traffic Systems (ITS) offer a more centralized approach by collecting data from multiple junctions to adjust signals across the network. However, they still rely on fixed logic and lack true predictive power. Their effectiveness is reduced in scenarios requiring immediate or unforeseen responses without human input.

A major drawback of these systems is the inability to adapt dynamically. Vehicles may wait unnecessarily at red lights while no traffic is approaching from other directions. Fixed cycles also struggle to accommodate sudden traffic spikes, often leading to bottlenecks and longer queues.

Infrastructure cost is another issue. Sensor-based systems require extensive installation and maintenance, especially in areas with challenging environmental conditions. Malfunctioning sensors can cause inaccurate traffic estimates, leading to mismanagement of signal timing and increased delays. Existing systems are mostly rule-based and need manual updates to accommodate new traffic patterns. This makes them outdated quickly in fast-growing urban environments. Moreover, they typically optimize traffic at a local intersection level, not considering the larger network, which may lead to new congestion points.

The proposed system overcomes these flaws by using machine learning to both analyze historical trends and adapt to current conditions. By continuously learning from new data, it predicts traffic flow and adjusts signals before congestion occurs. This proactive control improves both flow and reliability.

With algorithms like Decision Trees, KNN, and Random Forests, the system can detect patterns in traffic and make informed decisions. Its self-learning ability allows it to evolve with urban growth, making it a cost-effective, scalable, and sustainable alternative to traditional models.

While inductive loop sensors can detect vehicle presence, they offer limited insight into vehicle type or lane-specific congestion. Vision-based AI systems provide richer data and enable more precise control at intersections.

Machine learning allows systems to evolve without needing manual reprogramming every time traffic behavior changes. This self-adjusting capability is critical for cities with fluctuating traffic trends.

The main value of AI in traffic systems lies in prediction, not just reaction. By forecasting upcoming congestion, the system can act in advance, leading to smoother flow and reduced pressure on critical road segments.

3. SYSTEM DISCRIPTION

The smart traffic light control system is designed to dynamically manage traffic flow using artificial intelligence. Unlike traditional systems with static timers, this model relies on real-time analysis to determine signal duration. By analyzing the number of vehicles at intersections, it efficiently allocates green light time to reduce delays and improve traffic movement.

Python was chosen as the core development language due to its simplicity and support for a wide range of data science libraries. The system uses libraries such as Num Py for numerical operations, Pandas for data manipulation, and Tkinter for the graphical interface. These tools facilitate the processing and visualization of traffic data.

Scikit-learn (Sklearn) is integrated to apply machine learning algorithms like classification and regression.

It helps train models using traffic features to make signal timing predictions. These predictions are critical in determining optimal traffic light phases across various directions and intersections.

The input for the system is gathered using high-resolution cameras positioned at traffic junctions. These cameras continuously stream live footage, which is analyzed using OpenCV for image processing tasks such as frame extraction, grayscale conversion, and object highlighting.

YOLOv5 and YOLOv8 are utilized for real-time vehicle detection. YOLO processes each video frame to identify objects like cars, buses, and bikes, drawing bounding boxes around them. Its speed and accuracy make it suitable for dynamic traffic environments, ensuring timely detection and analysis.

Vehicle classification is essential to determine the type and count of vehicles in each lane. This enables the system to calculate traffic density and adjust signal timings proportionally. Libraries like Image AI help with object tracking and recognition tasks to improve detection accuracy.

Machine learning plays a pivotal role in predicting traffic behavior. It allows the system to learn from previous data and adapt to changing conditions without needing manual updates. Over time, the model becomes more precise in its predictions, making traffic control smarter and more efficient.

Deep learning, especially Convolutional Neural Networks (CNNs), enhances detection capabilities by extracting complex features from video frames. Models like VGG and Inception are used for high-accuracy classification. These tools enable the system to handle varying traffic scenarios, including low light and high congestion.

The integration of Open CV with YOLO provides a powerful foundation for detecting and classifying vehicles with speed and precision, which is critical for real-time applications like traffic signal control.

Instead of building new infrastructure from scratch, the system is designed to be installed on existing networks using low-cost embedded devices, making it practical for large-scale deployment in budget-conscious cities.

Each component, from camera input to traffic signal output, is modular and upgradable. This ensures that future enhancements in AI models or detection algorithms can be incorporated without redesigning the entire system.

4. SYSTEM DESIGN

The smart traffic light control system is designed to dynamically manage traffic flow using artificial intelligence. Unlike traditional systems with static timers, this model relies on real-time analysis to determine signal durations. By analyzing the number of vehicles at intersections, it efficiently allocates green light time to reduce delays and improve traffic movement.

Python was chosen as the core development language due to its simplicity and support for a wide range of data science libraries. The system uses libraries such as Num Py for numerical operations, Pandas for data manipulation, and Tkinter for the graphical interface. These tools facilitate the processing and visualization of traffic data.

Scikit-learn (Sklearn) is integrated to apply machine learning algorithms like classification and regression. It helps train models using traffic features to make signal timing predictions. These predictions are critical in determining optimal traffic light phases across various directions and intersections.

The input for the system is gathered using high-resolution cameras positioned at traffic junctions. These cameras continuously stream live footage, which is analyzed using OpenCV for image processing tasks such as frame extraction, grayscale conversion, and object highlighting.

YOLOv5 and YOLOv8 are utilized for real-time vehicle detection. YOLO processes each video frame to identify objects like cars, buses, and bikes, drawing bounding boxes around them. Its speed and accuracy make it suitable for dynamic traffic environments, ensuring timely detection and analysis.

Vehicle classification is essential to determine the type and count of vehicles in each lane. This enables the system to calculate traffic density and adjust signal timings proportionally. Libraries like ImageAI help with object tracking and recognition tasks to improve detection accuracy.

Machine learning plays a pivotal role in predicting traffic behavior. It allows the system to learn from previous data and adapt to changing conditions without needing manual updates. Over time, the model becomes more precise in its predictions, making traffic control smarter and more efficient.

Deep learning, especially Convolutional Neural Networks (CNNs), enhances detection capabilities by extracting complex features from video frames. Models like VGG and Inception are used for high-accuracy classification. These tools enable the system to handle varying traffic scenarios, including low light and high congestion.

One of the key design principles is adaptability—each intersection operates semi-independently while being able to communicate with nearby nodes for coordinated traffic flow in densely populated zones.

The use of microcontrollers ensures low-power consumption and real-time responsiveness, which is ideal for continuous operation in outdoor, weather-variable conditions.

By visualizing the system flow with UML and sequence diagrams, developers and city planners can easily understand, scale, and troubleshoot the system during and after deployment.

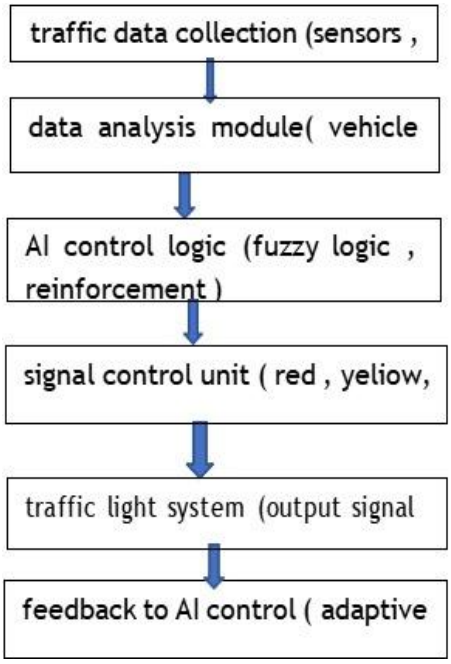

FIG: BLOCK DIAGRAM OF SYSTEM ARCHITECTURE

1. Traffic Data Collection (Sensors):

In this stage, real-time traffic information is gathered using various sensors such as cameras, inductive loops, or infrared devices. These sensors monitor parameters like the number of vehicles, their speed, and road occupancy. This raw data forms the base input for intelligent decision-making.

2. Data Analysis Module (Vehicle Detection):

The collected data is then sent to the analysis module. This module identifies traffic patterns and detects vehicle density at intersections. It may involve computer vision techniques or statistical methods to interpret the input data accurately.

3.AI Control Logic (Fuzzy Logic, Reinforcement Learning)

This block applies artificial intelligence techniques to process the analyzed data. Fuzzy logic helps handle uncertainty and vague information, while reinforcement learning enables the system to learn from experience by adapting to traffic conditions over time. This module decides when and how long each traffic light should remain active.

3. Signal Control Unit (Red, Yellow, Green) Based on decisions made by the AI

5.SYSTEM IMPLEMENTATION

The system implementation begins with a centralized model that processes live traffic data to manage signal timings. Traffic density, vehicle count, and wait times are captured through sensors or cameras installed at intersections. This raw input is cleaned and preprocessed before being analyzed by a machine learning model to determine optimal light durations.

Python is used for development, supported by essential libraries like Pandas for data handling, Scikit-learn for machine learning, and Flask or FastAPI for serving the model. These components are installed and configured using package managers, ensuring a consistent development environment for both offline testing and real-time deployment.

Hardware integration is crucial for the system to function effectively in real-world conditions. Cameras and infrared sensors act as inputs, feeding data to microcontrollers like Raspberry Pi. These controllers pass the information to the AI model, which then sends timing commands to traffic lights, creating a seamless feedback loop.

The machine learning model is trained using historical traffic data to learn patterns related to vehicle density, time of day, and waiting times. Logistic Regression and Decision Trees are commonly used for basic models, while advanced implementations may employ Reinforcement Learning to improve adaptability in complex scenarios.

For Reinforcement Learning, the model acts as an agent that learns through interaction with the environment. It receives feedback in the form of

rewards or penalties based on how well it reduces wait time or congestion. Over time, it refines its strategy to manage traffic more effectively than static, rule-based systems.

After training, the model is deployed as an API service that receives sensor input and returns real-time signal timing decisions. These decisions are immediately communicated to the traffic controller, enabling the system to respond dynamically to current traffic conditions with minimal delay.

Scalability is a major focus in deployment. The model and API can handle multiple intersections simultaneously. Performance metrics such as accuracy, response time, and traffic throughput are tracked to evaluate the system and identify areas for retraining or updates.

Overall, the implementation phase bridges the gap between theoretical design and practical execution. With real-time integration of AI models, hardware, and user-friendly interfaces, the system evolves into a fully functional, intelligent traffic control solution.

The selection of lightweight frameworks such as Flask allows the system to serve model predictions efficiently, even on resource-limited hardware like Raspberry Pi.

Reinforcement learning introduces adaptability that is particularly effective for intersections where traffic patterns change hourly, such as near schools or markets.

Data preprocessing techniques ensure that the model receives only relevvant, high-quality input, which significantly improves prediction accuracy and decision confidence.

6.RESULTS AND ANALYSIS

The system underwent multiple testing phases to ensure reliability, accuracy, and alignment with intended functionality. Testing was vital in verifying each module individually, and then as an integrated whole. The goal was to confirm that the AI model accurately detects vehicles and dynamically controls signal timing based on traffic flow.

Unit testing was performed on each component, especially on the object detection and classification modules. These tests validated logical correctness and ensured that input data produced expected outputs. It helped developers catch early-stage bugs within specific functions before full system integration.

Integration testing followed to ensure seamless interaction between modules—particularly between the camera input, processing unit, and microcontroller. This step ensured data from sensors was correctly interpreted and used for traffic light control. Any discrepancies in data transfer were addressed during this phase.

Functional testing assessed the system from a user and operational perspective. It checked if the interface responded to valid and invalid inputs, performed its designated tasks correctly, and followed expected workflows. Additional checks were made to ensure system behavior aligned with the real-world traffic environment.

The system also underwent white box and black box testing. White box testing examined internal code logic, while black box testing focused on outputs based on user interactions, without knowledge of the internal code. This dual approach ensured thorough verification from both developer and user perspectives.

Testing of the AI models, YOLO and SSD, was conducted using traffic video samples. YOLO offered fast, real-time detection, although it occasionally missed small or obscured vehicles. SSD was slightly slower but delivered more accurate detection in dense traffic. Both models handled low-resolution videos well, with minor frame drops at 1080p.

The GUI was found to be intuitive, allowing users to easily switch between simulation, YOLO-based detection, and SSD-based detection. It displayed real-time logs and bounding boxes around detected vehicles. While performance was generally smooth, slow processing was noted on systems lacking a dedicated GPU.

Results confirmed that vehicle counting was mostly accurate, with slight overcounting in dense scenes due to overlapping frames. Future improvements like integrating object tracking (e.g., DeepSORT) could help refine accuracy. The system demonstrated its real-time capabilities and readiness for further development.

The system demonstrated robust performance even under sub-optimal lighting conditions, which proves its usability during early mornings, evenings, and adverse weather.

Real-world testing revealed that the GUI made it easy for non-technical users to visualize traffic patterns,

offering a useful interface for municipal traffic management teams.

Though occasional false detections occurred, especially in complex scenes, the system still maintained an acceptable margin of error and continued to improve through iterative model updates.

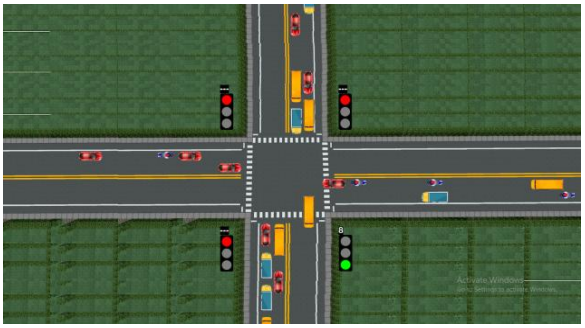

Figure 2: Pygame Simulation Output

Figure 3: Traffic detecting in real time

7. CONCLUSION AND FUTURE WORK

This project successfully demonstrated the use of artificial intelligence for real-time traffic light control. By integrating object detection models like YOLO and SSD, the system was able to identify and count vehicles accurately using live video feeds. This enabled traffic signals to be adjusted dynamically based on actual traffic density.

The intelligent system offered a clear improvement over traditional fixed-timing models. It optimized signal durations to reduce vehicle waiting times and improve traffic flow. Through a user-friendly interface, users could simulate traffic and visualize how signal adjustments are made in response to live conditions.

Although the system performed well under normal scenarios, it showed slight limitations in detecting vehicles during high congestion or when objects were partially blocked. This indicates a need for advanced detection techniques or better-trained models tailored to specific urban environments.

One of the standout features of the project was its adaptability. The AI model learned from real-time inputs, allowing it to continuously refine its predictions. This lays the groundwork for scalable smart city traffic solutions that can operate autonomously with minimal human input.

For future improvement, integrating object tracking (like DeepSORT) could prevent duplicate counting and enhance accuracy in heavy traffic. This would be particularly helpful in scenarios where vehicles are occluded or move rapidly across frames.

The project could be expanded to include real-time signal control, where the traffic lights automatically adjust their cycles without user intervention. This would improve responsiveness and further reduce congestion during peak hours.

Deploying more advanced models like YOLOv8 could also increase detection accuracy, especially under challenging conditions such as nighttime traffic or poor weather. Additionally, training models with region-specific datasets would help improve localization and precision in diverse environments.

Finally, the system can be extended to include traffic violation detection. By identifying behaviors such as signal jumping or wrong-way driving, the solution can aid law enforcement and improve road safety. Combined with edge or cloud-based deployment, the system becomes a powerful tool for next-generation urban traffic management.

This project acts as a foundational model that can be extended to smart city ecosystems, integrating with pedestrian safety systems, emergency vehicle prioritization, and even air quality sensors.

Future improvements may include adding multilingual voice alerts or visual prompts for pedestrians and cyclists at smart intersections, making roads safer for all users.

With continued development, the system can support city-wide analytics by generating real-time traffic heatmaps and contributing data to centralized transportation dashboards.

REFERENCE

[1] TomTom.com, 'Tom Tom World Traffic Index', 2019. [Online]. Available: https://www. tomtom.com/en_gb/traffic index/ranking/

[2] Khushi, "Smart Control of Traffic Light System using Image Processing," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.

[3] A. Vogel, I. Oremović, R. Šimić and E. Ivanjko"Improving Traffic Light Control by Means of Fuzzy Logic," 2018 International Symposium ELMAR, Zadar, 2018, pp. 51-56, doi: 10.23919/ELMAR.2018.8534692.

[4] A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257768.

[5] Renjith Soman "Traffic Light Control and Violation Detection Using Image Processing”.” IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 4, 2018, pp. 23-27

[6] A. Kanungo, A. Sharma and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2014, pp. 1- 6, doi: 10.1109/RAECS.2014.6799542.

[7] Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, Minocha, “Adaptive traffic light timer controller”, IIT KANPUR, NERD MAGAZINE

[8] Ms. Saili Shinde, Prof. Sheetal Jagtap, Vishwakarma Institute Of Technology, Intelligent traffic management system:a Review, IJIRST 2016