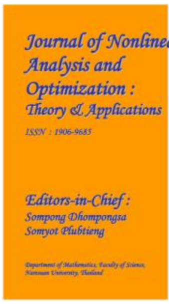
Journal of Nonlinear Analysis and Optimization
Vol. 15, Issue. 2, No.5 : 2024
ISSN : **1906-9685**



A SMART TRAFFIC MANAGEMENT SYSTEM THAT ADJUSTS SIGNALS TO IMPROVE ROAD CONGESTION

S. DHIVYA, Associate Professor, Department of Computer Science &Engineering, Sri Shakthi Institute of Engineering & Technology, Coimbatore, Tamil Nadu, India, Email: dhivyasivanandan@gmail.com
R. GAYATHRI Student, Department of Computer Science &Engineering, Sri Shakthi Institute of Engineering &Technology, Coimbatore, Tamil Nadu, India, Email: gayathri12.ramesh@gmail.com

ABSTRACT

This paper aims to overcome real time road traffic congestion by creating smart traffic system. This system automatically adjusts traffic signals in real time to avoid congestion. Using machine learning and computer vision, the system tracks traffic flow. It analyzes the current road conditions and changes the traffic lights to improve vehicle movement. It adjusts traffic signals in real-time to optimize the flow of vehicles by analyzing data continuously. This reduces waiting times, lowers the fuel consumption, and decreases air pollution. As a result, the system not only makes commutes faster and more efficient but also contributes to a cleaner environment.

1. INTRODUCTION

In modern cities, traffic congestion poses significant challenges, contributing to extended travel times, increased pollution, and commuter stress. Traditional traffic lights operate on fixed schedules, unable to adapt to varying traffic conditions, which limits their effectiveness during peak hours. Recent advancements in technology—particularly in computer vision, machine learning, and real-time data processing—present a transformative approach to managing urban traffic more efficiently.

A smart traffic management system uses live data from cameras and advanced algorithms to dynamically adjust traffic signals, reducing congestion and improving road efficiency. Key tools such as YOLO (You Only Look Once) and OpenCV enable the system to monitor vehicle types, speeds, and traffic patterns, allowing for real-time response based on actual conditions rather than static schedules. Python's robust libraries for video processing and machine learning facilitate the integration and continuous improvement of these systems. This paper explores how such a system not only minimizes traffic delays but also promotes a more sustainable, responsive approach to urban traffic management.

2. LITERATURE REVIEW

2.1 Real-Time Traffic System Using Crowd-Sourced GPS Data

Real-time, accurate traffic data has become vital for urban commuters and drivers. With the rise of GPS-enabled smartphones, a traffic monitoring system leveraging cell phone GPS data is highly practical. As vehicles with a smartphone app drive through various routes, they collect valuable information such as vehicle speed, geolocation, and road conditions. This data is sent to a web service, stored in a database, and used to create a color-coded map reflecting near real-time traffic conditions citywide. Compared to traditional monitoring systems, this GPS-based approach requires minimal maintenance, rapid deployment, and broad area coverage, enabling more informed route choices and improved urban traffic flow.

2.2 Dynamic Road Traffic Management Based On Krushkal's Algorithm

The Dynamic Vehicle Navigation System (DVNS) is designed to manage real-time traffic issues like congestion, accidents, and bottlenecks. By mapping traffic junctions as nodes and assigning traffic rates as link weights, the system uses Kruskal's algorithm to determine optimal routes based on factors such as traffic rate, vehicle speed, and shortest path. Users enter their source and destination on Google Maps, and DVNS provides the best route using a dynamic routing table, which traffic personnel periodically update via mobile devices. While effective, this system relies on manual updates, which may delay or impact route accuracy in sudden traffic events

2.3 Implementation Of Dijkstra's Algorithm To Find An Effective Route Traffic Jam On a Busy Hour

Traffic congestion is a prevalent issue in Indonesia due to factors like high vehicle volume, narrow roads, and road activities. This research applies Dijkstra's Algorithm to find effective routes that help avoid traffic jam points, tackling the Shortest Path Problem. The algorithm determines the shortest route to a destination and removes paths associated with congestion, yielding an efficient route that balances both distance and traffic conditions. While effective, the approach relies on pre-identified congestion points, which may not reflect real-time conditions, as Dijkstra's Algorithm doesn't adjust dynamically to sudden changes, such as accidents, potentially limiting its reliability in unpredictable traffic scenarios.

2.4 A Review of Yolo Algorithm Developments

This paper provides an analysis of the You Only Look Once (YOLO) algorithm and its advanced versions, underscoring its foundational role in AI-driven object detection. It examines YOLO's development, highlighting key similarities and differences among its versions and contrasting it with traditional Convolutional Neural Networks (CNNs). The analysis confirms that YOLO has undergone continuous improvements, reflecting ongoing advancements in algorithmic design and performance. Additionally, the paper explores YOLO's methods for target recognition and feature selection, demonstrating its applicability beyond traditional uses, such as targeted image recognition in finance. By summarizing YOLO's evolution, the study offers valuable insights for object detection applications across diverse fields.

2.5 A Review of Object Detection Models Based On Convolutional Neural Network

This chapter reviews state-of-the-art object detection models based on Convolutional Neural Networks (CNNs), categorizing them into two main approaches: two-stage and one-stage models. It examines the evolution of two-stage models from R-CNN to the latest Mask R-CNN, alongside one-stage detectors from YOLO to RefineDet. The paper discusses architectural innovations and training details for each model, highlighting their contributions to improved accuracy and performance in object detection tasks. It concludes that two-stage models typically offer higher accuracy by separating object localization and classification, while one-stage models prioritize speed by integrating these processes. By comparing the training methodologies and architectural components, the review provides valuable insights into the strengths and trade-offs of each model, illustrating how CNN-based object detection has evolved to address diverse real-world needs.

2.6 Vehicle Counting For Traffic Management System Using Yolo and Correlation Filter

This article introduces a video-based vehicle counting method aimed at estimating road traffic density for intelligent transportation systems. Utilizing the widespread deployment of cameras, including handheld devices, the proposed approach processes highway traffic videos through three stages: object detection with the YOLO algorithm, object tracking using correlation filters, and vehicle counting. YOLO is chosen for its effectiveness in detection, while correlation filters provide high accuracy and speed for tracking multiple vehicles. The study concludes that combining these techniques offers a reliable and efficient solution for vehicle counting. Experimental results demonstrate the method's ability to accurately detect, track, and count vehicles in real-world scenarios, underscoring its potential for integration into real-time traffic management systems to enhance monitoring and traffic management capabilities.

2.7 Vision-Based Vehicle Detection And Counting System Using Deep Learning In Highway Scenes

This article presents a video-based vehicle counting method designed to estimate road traffic density for intelligent transportation systems. The approach processes highway traffic videos in three stages: object detection using the YOLO algorithm, object tracking with correlation filters, and vehicle counting. YOLO is effective for detection, while correlation filters offer high accuracy and speed for tracking. The study concludes that this combined method provides a reliable solution for vehicle counting, with experimental results demonstrating its effectiveness in accurately detecting, tracking, and counting vehicles in real-world scenarios. This technique shows promise for integration into real-time traffic management systems, improving monitoring and traffic management.

2.8 A Yolo-Based Traffic Counting System

This paper presents a vehicle counting system using the YOLO framework, aimed at addressing inaccuracies in traffic flow counting due to short recognition failures. The system architecture includes three blocks: a Detector for generating vehicle bounding boxes, a Buffer for storing vehicle coordinates, and a Counter for counting vehicles. To determine if vehicles in consecutive frames are the same, simple distance calculations are employed, along with checkpoints to minimize false detections. The proposed system was validated using videos from various locations and angles, achieving high counting accuracy under sufficient ambient light. The results indicate that this method offers a practical solution for real-time traffic monitoring, effectively reducing counting errors without relying on complex tracking algorithms.

2.9 Artificial Intelligence Approach For Optimizing Traffic Timings On Urban Road Network

This paper discusses the application of artificial intelligence techniques to optimize signal timings in urban traffic control systems. The proposed method consists of two main processes: training and optimization. During the training phase, two neural network models are employed: a multilayer model to establish the relationship between signal timings and traffic objectives, and a Kohonen feature map model to enhance computational efficiency and estimation accuracy. In the optimization phase, Cauchy machine learning and genetic algorithms are utilized to find optimal signal timings while avoiding local minima, thereby reducing total traffic delays and stop frequencies. The results indicate that the AI-driven approach outperforms conventional methods, highlighting its potential for developing advanced traffic control systems that can adapt to real-time conditions.

2.10 Intelligent Traffic Management System

This paper introduces an intelligent traffic management system that uses RFID technology to collect essential traffic data and reduce travel times. The system consists of passive RFID tags, readers, a microcontroller, a GPRS module, and a central server. It calculates average vehicle speeds at junctions and uses Dijkstra's algorithm to find the fastest routes, providing real-time information to users via an in-vehicle interface.

Additionally, the RFID system can trace stolen vehicles, monitor traffic violations, facilitate toll collection, and manage vehicle taxes. This multifunctional approach highlights the versatility of RFID technology in optimizing traffic flow and offering valuable services to users and traffic authorities.

2.10 Intelligent Traffic Light Control Using Distributed Multi-Agent Q Learning

This paper discusses the application of AI and IoT, collectively known as AIoT, to improve traffic light control systems in smart cities. By utilizing various sensors, such as surveillance cameras, the proposed intelligent traffic light solution employs distributed multi-agent Q-learning to monitor and manage both motorized and non-motorized traffic in real time. This approach considers local traffic conditions as well as data from neighboring intersections to optimize traffic flow. Numerical simulations demonstrate that the solution significantly enhances performance metrics like vehicle and pedestrian queue lengths and waiting times at intersections compared to existing methods.

2.11 Real-Time Adaptive Signal Control In A Connected Vehicle Environment

This paper highlights the benefits of using connected vehicle data in traffic signal control to improve signal phase allocation at intersections. Traditional systems rely on point detectors, which can't accurately measure vehicle speed and location. The proposed adaptive signal phase allocation algorithm optimizes phase sequence and duration through a two-level optimization process, targeting the minimization of vehicle delay and queue length. Simulations in VISSIM show that this algorithm outperforms traditional fully actuated control, reducing total vehicle delay by up to 16.33% in high penetration scenarios. The findings underscore the advantages of integrating connected vehicle technology for smarter traffic management systems.

3. RESEARCH METHODOLOGIES

3.1 Existing System

In most urban cities around the world, traffic light signals are the primary means of managing traffic on the roads. Over the years, different types of traffic control systems have been developed, including vehicle-actuated lights and static traffic lights. Despite these advancements, the timing of these signals is usually fixed, and the patterns are predetermined in the system. These lights don't adapt to real-time traffic conditions, meaning the duration of red, yellow, and green lights remains the same, regardless of whether the road is crowded or empty. As a result, these systems often lead to inefficiencies, especially during rush hours or in the event of unexpected traffic buildup.

3.2 Module Implementation

3.2.1 Data Collection

In the smart traffic management system, data collection is a critical initial step that involves gathering relevant traffic data to train and evaluate the models. This data can be collected from various sources:
Camera Feeds: Live video feeds from traffic cameras installed at intersections are captured using OpenCV. These feeds provide real-time footage of vehicular movement, which is essential for understanding traffic flow and patterns.
Synthetic Data: In addition to real-world data, synthetic datasets can be generated using Pygame to simulate traffic conditions. This involves creating various traffic scenarios with different vehicle densities, speeds, and behaviors to enrich the training data.
Existing Datasets: Publicly available traffic datasets, such as those from traffic management agencies or research institutions, can be leveraged. These datasets often contain annotated images and videos that help in training object detection models.
Sensor Data: If available, data from traffic sensors (like inductive loop sensors) can be integrated. This data typically includes vehicle counts, speed, and lane occupancy, providing valuable insights into traffic dynamics.

3.2.2 Data Preprocessing:

Once the data is collected, preprocessing is necessary to ensure that it is clean and suitable for model training. The preprocessing steps include:
Annotation: For supervised learning with YOLO, collected images or video frames must be annotated with bounding boxes around detected vehicles. This involves labeling the images with object classes (e.g., car, truck, bus) and generating the corresponding annotation files in the format expected by YOLO.
Resizing and Normalization: Images are resized to the dimensions required by the YOLO model (typically 416x416 pixels) to ensure consistency. Normalization of pixel values (scaling from 0-255 to 0-1) is also performed to improve model performance.
Data Augmentation: To increase the robustness of the model, data augmentation techniques can be applied. This includes transformations such as rotation, flipping, cropping, and brightness adjustment to generate diverse training examples from existing data.
Splitting the Dataset: The preprocessed data is divided into training, validation, and testing sets. Typically, a common split is 70% for training, 15% for validation, and 15% for testing. This ensures that the model can be trained effectively while also being evaluated on unseen data.

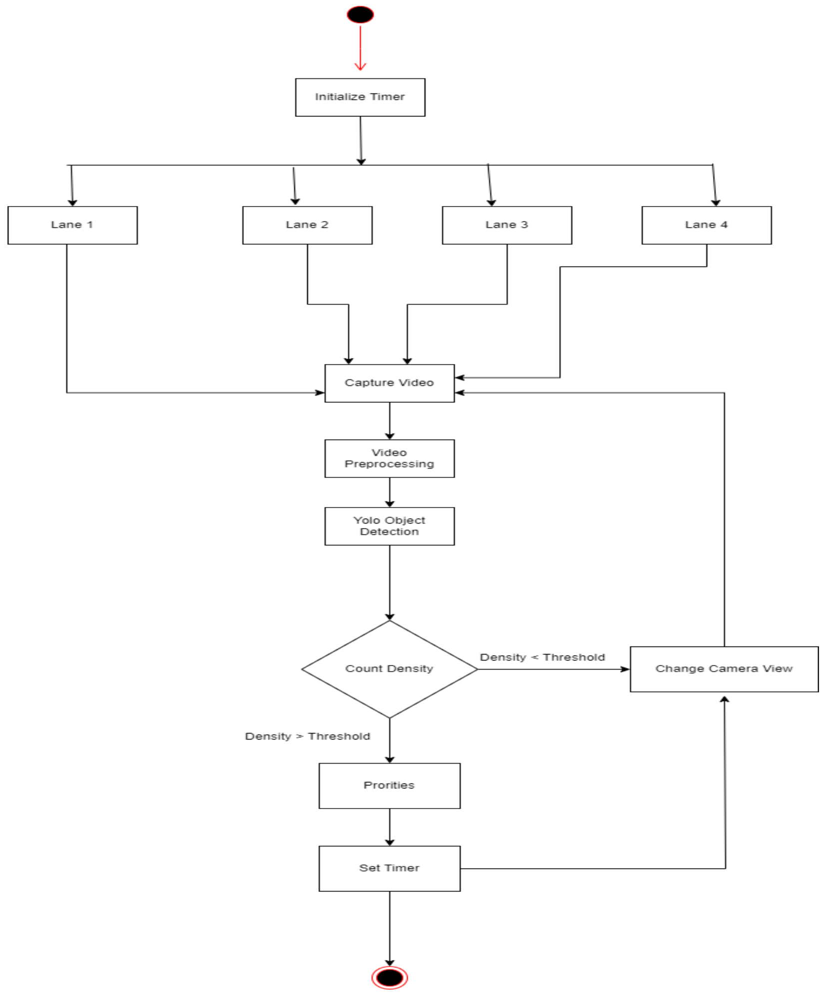
3.2.3 Model Development:

Model development involves designing and configuring the object detection model that will be used in the smart traffic management system. This phase consists of:
Choosing the Model Architecture: YOLO is selected due to its speed and accuracy in real-time object detection. Depending on the project's requirements, different versions of YOLO can be chosen, such as YOLOv3 or YOLOv4. Here we choosen YOLOv3.
Setting Up the Training Environment: A suitable training environment is established using frameworks like TensorFlow or PyTorch, depending on the preferred implementation of YOLO. This includes installing the necessary libraries and configuring the hardware (such as GPU support) for efficient training.
Configuring Hyperparameters: Key hyperparameters are set, including learning rate, batch size, and the number of epochs. These parameters influence how the model learns and must be tuned for optimal performance.

Integrating OpenCV: OpenCV is utilized to handle image processing and preprocessing tasks seamlessly within the model pipeline, ensuring smooth integration with the YOLO object detection process.

3.2.4 Model Training:
Model training is the process of teaching the model to recognize and classify objects (vehicles) in traffic scenarios. The training phase involves several steps:
Loading Data: The training data, which includes images and corresponding annotations, is loaded into the training pipeline. This data is shuffled to ensure that the model learns from diverse examples in each epoch.
Validation: The model's performance is monitored using the validation dataset after each epoch. Metrics such as precision, recall, and mean Average Precision (mAP) are computed to evaluate the model's accuracy.
Fine-tuning: Based on the validation results, hyperparameters may be adjusted, and additional training may be performed to improve the model's accuracy. Techniques like learning rate scheduling or early stopping can be employed to enhance training efficiency.
Testing: Once training is complete, the model is evaluated on the testing dataset to assess its performance on unseen data. This final evaluation provides an indication of how well the model will perform in real-world scenarios.
Model training is a critical phase that equips the YOLO model with the ability to detect and classify vehicles accurately, thereby enabling the smart traffic management system to adapt traffic signals based on real-time data.

3.5 Results
The project results are divided into two main sections: the evaluation of the vehicle detection model and the assessment of the proposed adaptive traffic management system.
Evaluation of Vehicle Detection Model:
The vehicle detection module was tested with various images, achieving a detection accuracy of 75% to 80%. This accuracy, while satisfactory, is hindered by the limited quality and variety of the training dataset. To improve this, using real-life footage from traffic cameras is recommended. By processing videos from live feed cameras with the YOLO and OpenCV frameworks, we detected vehicles, tracked their movements, and classified them, which is essential for traffic analysis. The traffic density data derived from the detection module was integrated to optimize traffic signal timings, prioritizing lanes with higher density to reduce congestion. A simple traffic simulation was developed using Pygame to demonstrate the system's functionality.
Evaluation of the Proposed Adaptive System:
To assess the effectiveness of the adaptive traffic management system, we conducted 15 traffic movement simulations over five minutes each, calculating the total number of vehicles passing through the junction. The results indicated that the adaptive system can significantly enhance traffic management by dynamically responding to real-time data and improving traffic flow efficiency under varying conditions.



4. SYSTEM ANALYSIS

4.1 About The Software

4.1.1 OpenCV in Smart Traffic Management

OpenCV is integral to the smart traffic management system, capturing and processing real-time video from traffic cameras. It improves footage quality through resizing and noise reduction, while its object detection capabilities identify vehicles, pedestrians, and road signs.

By calculating metrics like vehicle counts and speeds, OpenCV helps adjust traffic signal timings dynamically, extending green lights for congested lanes to improve traffic flow**.**

4.1.2 YOLO in Smart Traffic Management

YOLO serves as the primary object detection model, enabling real-time detection and classification of vehicles. It processes images efficiently, allowing quick adjustments to traffic signals based on vehicle counts and types.

YOLO enhances safety by recognizing vulnerable road users and can predict traffic patterns, helping the system optimize signal timings to reduce congestion and improve overall traffic efficiency

4.2 Screenshot

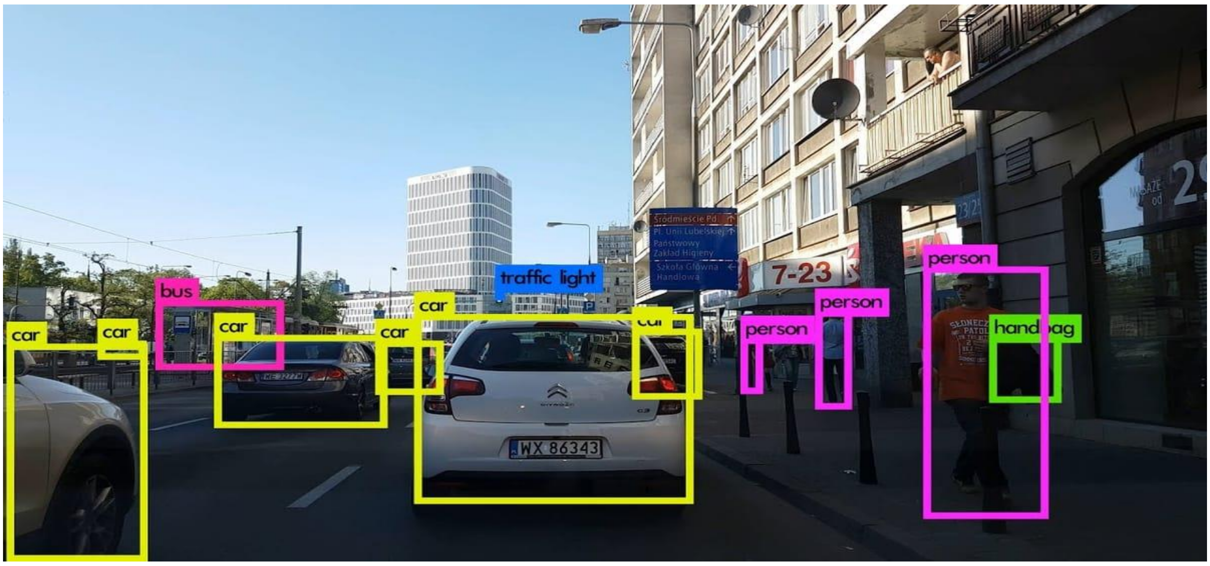


Fig 4.2.1

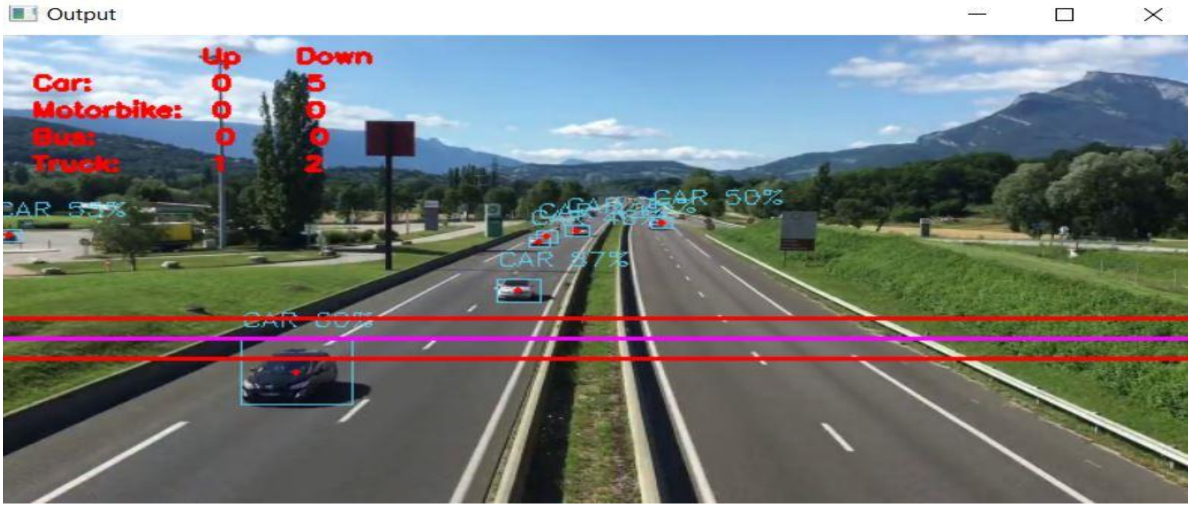


Fig 4.2.1

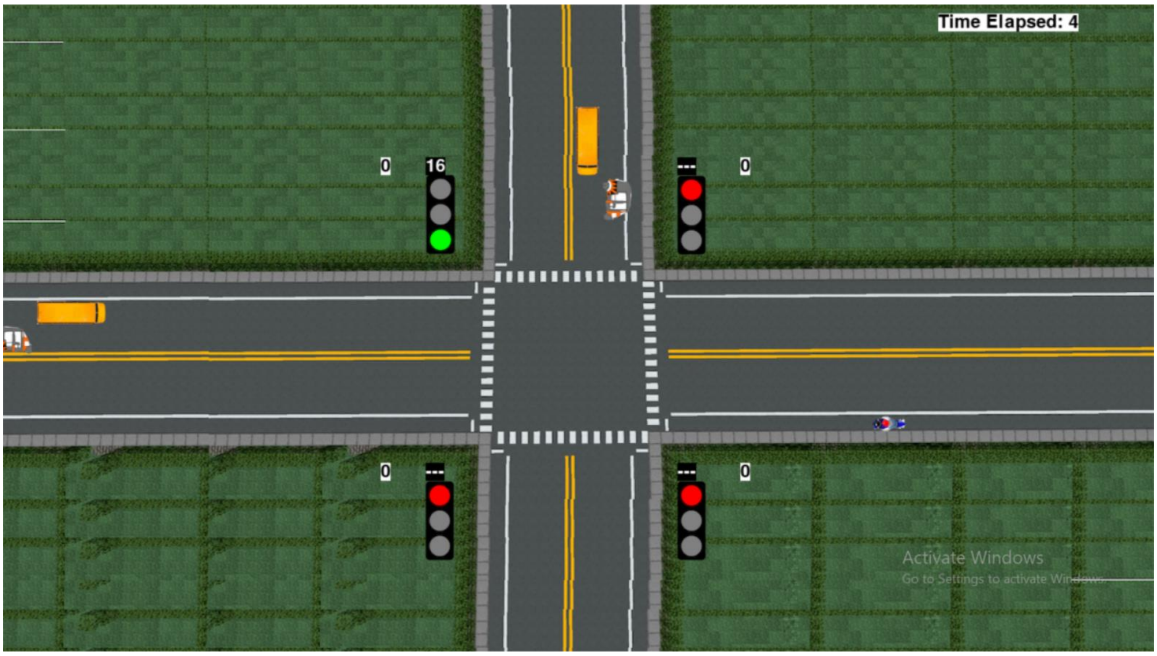


Fig 4.2.3

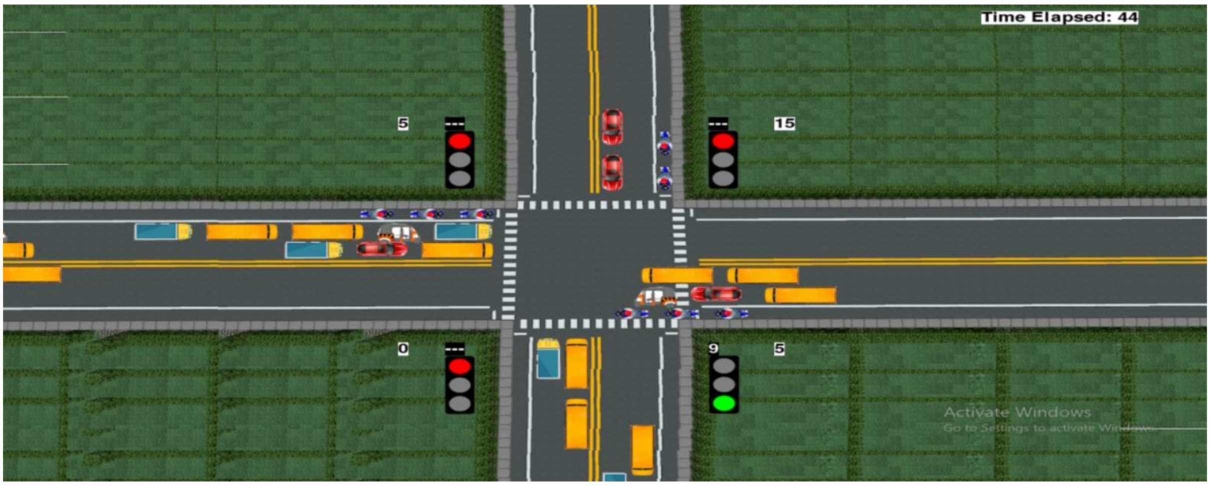


Fig 8.2.4

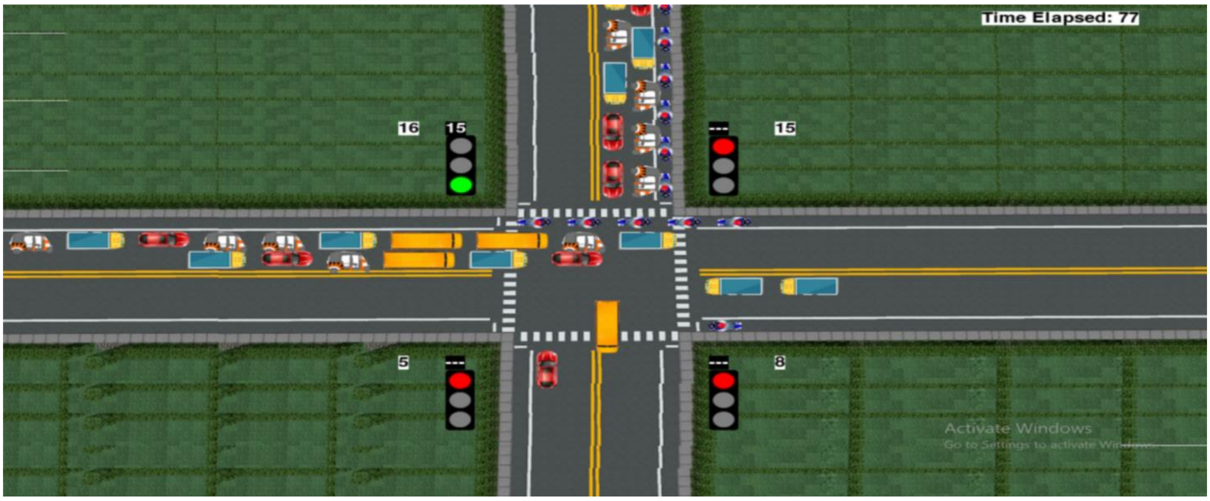


Fig 4.2.5

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

Implementing a smart traffic management system utilizing computer vision and machine learning technologies has shown promising results in enhancing traffic flow and reducing congestion at urban intersections. By leveraging OpenCV and YOLO for real-time vehicle detection, the system can dynamically adjust traffic signal timings based on current traffic conditions, significantly improving the overall efficiency of road usage. The initial evaluation of the vehicle detection model, with an accuracy range of 75% to 80%, highlights the need for better datasets to optimize performance further. However, the successful integration of traffic density data for signal optimization demonstrates the system's potential to provide a more fluid driving experience for commuters

5.1 Future Enhancement

While the current system lays a robust groundwork, several enhancements can be implemented to further improve its effectiveness. Data enrichment involves collecting more diverse and high-quality datasets that capture varied traffic scenarios, including different weather conditions and peak vs. off-peak hours. Additionally, integrating supplementary data sources like GPS information from vehicles and historical traffic patterns will provide deeper insights, leading to better decision-making. Simultaneously, implementing user feedback mechanisms—such as mobile applications that provide real-time traffic updates and suggestions—can enhance community engagement and improve the user experience. By gathering feedback from commuters regarding their travel experiences, the system can be fine-tuned to address real-world challenges, making it more adaptive and responsive to the needs of the users. Together, these enhancements will significantly elevate the system's capability and effectiveness in managing urban traffic.

6. REFERENCE

1. Md. Al Amin and Md. Rofi Uddin, Real time traffic monitoring system using crowd sourced GPS data, 2015.
2. Shashikiran Venkatesha, T. T. Sampath Kumar, N. Sathish Kumar and V. Venkateswaran, Dynamic road traffic management based onkrushkal's algorithm, 2011.
3. P Sembiring, A S Harahap and K S Zalukhu, "Implementationof Dijkstra's algorithm to find an effectiveroute to avoid traffic jamon a busy hour", International Conference on Science and Technology 2018, 2018.
4. Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma, "A Review of Yolo Algorithm Developments", Procedia Computer Science, Volume 199, 2022.
5. Sultana, F., Sufian, A., Dutta, P. (2020). A Review of Object Detection Models Based on Convolutional Neural Network. In: Mandal, J., Banerjee, S. (eds) Intelligent Computing: Image Processing Based Applications. Advances in Intelligent Systems and Computing, vol 1157.
6. C. S. Asha and A. V. Narasimhadhan, "Vehicle Counting for Traffic Management System using YOLO and Correlation Filter," 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2018, pp. 1-6.
7. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. Eur. Transp. Res. Rev. 2019, 11, 51.
8. Lin, J.-P.; Sun, M.-T. A YOLO-Based Traffic Counting System. In Proceedings of the 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taichung, Taiwan, 30 November–2 December 2018; p. 18366282.
9. Syunichi SEKI, Takashi NAKATSUJI, Syuestu SHIBUYA, Terutoshi KAKU, ”Artificial Intelligence Approach for Optimizing Traffic Signal Timings on Urban Road Network”,IEEE 1994.
10. P. Manikonda, A. K. Yerrapragada and S. S. Annasamudram, "Intelligent traffic management system," 2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT), Semenyih, 2011, pp. 119-122.
11. Y. Liu, L. Liu and W. -P. Chen, "Intelligent traffic light control using distributed multi-agent Q learning," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017, pp. 1-8, doi: 10.1109/ITSC.2017.8317730.
12. Yiheng Feng, K. Larry Head, Shayan Khoshmagham, Mehdi Zamanipour, "A real-time adaptive signal control in a connected vehicle environment", Transportation Research Part C: Emerging Technologies,Volume 55, 2015, Pages 460-473.