

Smart Traffic Signal Optimization Using Real-Time Vehicle Detection

Thanuja B

thanuja.mca2024@adhiyamaan.in

Second Year MCA

Adhiyamaan College of Engineering, Hosur, India

Under the Guidance of Mrs. Divya Hariprasad, MCA

ABSTRACT

Traffic congestion has become a major problem in urban areas due to the rapid increase in the number of vehicles. Traditional traffic signal systems operate using fixed timing intervals without considering real-time traffic conditions, which often results in inefficient traffic flow and increased waiting time. This paper proposes a Smart Traffic Signal Optimization System using Real-Time Vehicle Detection to improve traffic management at intersections. The system uses computer vision and deep learning techniques to detect and count vehicles from traffic camera video feeds. A pre-trained YOLO (You Only Look Once) object detection model is used to identify vehicles such as cars, buses, trucks, and motorcycles in real time. Based on the detected vehicle count, the system dynamically calculates the green signal duration using a traffic density-based algorithm. The system is implemented using Python, OpenCV, YOLO, and Flask. Experimental results show that the proposed system improves traffic flow efficiency and reduces waiting time at intersections.

KEYWORDS

Smart Traffic System, Vehicle Detection, Computer Vision, YOLO, Traffic Signal Optimization, Deep Learning, Intelligent Transportation System

1. INTRODUCTION

Traffic congestion has become a critical issue in urban environments due to the continuous growth of the number of vehicles and limited road infrastructure. Traditional traffic signal systems rely on fixed time intervals for controlling traffic flow at intersections. These systems operate without considering the actual number of vehicles waiting at each lane, which often results in inefficient traffic management.

For example, a signal may remain green for a lane even when there are very few vehicles, while another lane with heavy traffic must wait for a long time. This imbalance leads to traffic congestion, increased travel time, and unnecessary fuel consumption.

Recent advancements in Artificial Intelligence, Computer Vision, and Deep Learning have opened new opportunities for developing intelligent traffic management systems. These technologies enable

computers to analyze visual information from cameras and make real-time decisions based on the observed data.

The proposed Smart Traffic Signal Optimization System uses a real-time vehicle detection approach to monitor traffic conditions at intersections. The system processes video frames from traffic cameras and detects vehicles using the YOLO deep learning model. The number of detected vehicles is then used to dynamically determine the green signal duration.

By adjusting signal timing according to traffic density, the system improves traffic flow efficiency and reduces congestion. This approach can play an important role in building intelligent transportation systems for future smart cities.

2. LITERATURE REVIEW

Several research studies have explored intelligent traffic management systems using artificial intelligence

and image processing techniques. Many existing systems use sensors such as inductive loop detectors and infrared sensors to monitor traffic conditions. However, these sensors are expensive and require physical installation on roads.

Computer vision-based traffic monitoring systems provide a more flexible and cost-effective solution by using cameras to detect vehicles and analyze traffic patterns. Deep learning models such as Convolutional Neural Networks (CNNs) have shown significant improvements in object detection accuracy.

One of the most widely used object detection algorithms is YOLO (You Only Look Once). YOLO is designed for real-time object detection and provides high accuracy with fast processing speed. It detects multiple objects in a single image by dividing the image into grid cells and predicting bounding boxes around objects.

Recent studies have successfully applied YOLO for vehicle detection and traffic monitoring applications. These systems demonstrate that deep learning models can effectively identify vehicles and estimate traffic density in real time.

The proposed system builds upon these advancements by integrating YOLO-based vehicle detection with a dynamic signal timing algorithm to optimize traffic signal control.

3. PROPOSED SYSTEM

The proposed Smart Traffic Signal Optimization system is designed to dynamically adjust traffic signal timing based on real-time vehicle detection. Unlike traditional traffic signal systems that operate using fixed time intervals, the proposed system uses computer vision and deep learning techniques to analyze traffic conditions and optimize signal duration.

The system utilizes traffic camera video feeds as input and processes them using the YOLO (You Only Look Once) object detection algorithm. The video frames are analyzed to detect different types of vehicles such as cars, buses, trucks, and motorcycles. After detecting vehicles in each frame, the system counts the number of vehicles present in the traffic lane.

Based on the vehicle count, the system calculates the green signal duration dynamically using a predefined formula. The calculated signal duration ensures that lanes with higher traffic density receive longer green signal time, while lanes with lower traffic density receive shorter signal duration.

The proposed system improves traffic management efficiency by reducing waiting time, minimizing congestion, and improving vehicle flow at intersections. The system is implemented using Python with OpenCV for image processing and YOLO for vehicle detection.

This intelligent approach enables the traffic signal system to adapt to real-time traffic conditions, making it suitable for modern smart city traffic management systems.

4. SYSTEM ARCHITECTURE

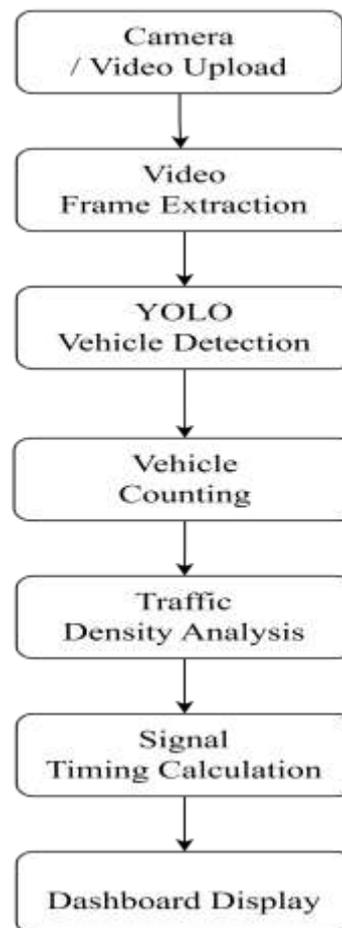


Figure 1: System Architecture of Smart Traffic Signal Optimization

Architecture Description

The system architecture is designed using a layered approach consisting of three major layers: **Presentation Layer, Application Layer, and Data Layer**. Each layer performs specific functions to ensure smooth operation of the system.

Presentation Layer

The presentation layer is responsible for interacting with the user and displaying the system output. This layer provides the interface where traffic monitoring results and vehicle detection outputs are visualized. The results of vehicle detection, bounding boxes around detected vehicles, and signal timing information are displayed through a graphical interface developed using Flask.

This layer ensures that the system output can be easily monitored and understood by the traffic management authority.

Application Layer

The application layer performs the core processing operations of the system. It is responsible for analyzing the video input, detecting vehicles, counting the vehicles, and calculating signal timing.

This layer includes several modules such as frame extraction, vehicle detection using the YOLO algorithm, vehicle counting, traffic density analysis, and signal timing calculation. The application layer processes each video frame in real time and generates the required signal timing output.

Data Layer

The data layer manages the input and processed data within the system. It handles the storage and processing of video frames captured from the traffic camera. The detected vehicle information and traffic density data are temporarily stored and used by the system for signal timing calculation.

The data layer ensures that the required information is efficiently accessed by the application layer for further processing.

5. System Modules

The proposed Smart Traffic Signal Optimization system is divided into several functional modules. Each module performs a specific task in processing the traffic video, detecting vehicles, and calculating optimal signal timing. These modules work together to ensure accurate vehicle detection and efficient traffic signal management.

5.1 Video Input Module

The Video Input Module is responsible for capturing traffic video from the source. In a real-world environment, this module receives live video footage from traffic surveillance cameras installed at road intersections. For system testing and development purposes, pre-recorded traffic videos can also be used as input.

The captured video stream serves as the primary input to the system. The quality and positioning of the camera play an important role in ensuring accurate vehicle detection. The video feed is continuously transmitted to the next stage of processing where frames are extracted and analyzed.

5.2 Frame Extraction Module

Using the OpenCV library, the video stream is processed and divided into frames at a specific frame rate. Each frame is then passed to the vehicle detection module for further analysis.

This module plays an important role in preparing the video data for computer vision processing and ensures that the system can analyze traffic conditions continuously.

5.3 Vehicle Detection Module

The Vehicle Detection Module is the core component of the system. It uses the YOLO deep learning algorithm to detect vehicles in each video frame.

YOLO is a real-time object detection model that can identify multiple objects within an image. It works by dividing the image into a grid and predicting bounding boxes and class probabilities for each object detected in the frame.

The model is trained to recognize various types of vehicles such as:

- Cars
- Buses
- Trucks
- Motorcycles

Once the model detects a vehicle, it draws a bounding box around the object and labels it accordingly. This module ensures fast and accurate vehicle detection even in dynamic traffic conditions.

5.4 Vehicle Counting Module

After detecting vehicles in each frame, the Vehicle Counting Module calculates the total number of vehicles present in the traffic lane. The system counts the detected vehicles based on the bounding boxes generated by the YOLO detection model.

The vehicle count provides an estimate of traffic density at the intersection. This information is essential for determining how long the traffic signal should remain green for a lane.

5.5 Traffic Density Analysis Module

The Traffic Density Analysis Module evaluates the level of congestion based on the vehicle count obtained from the previous module. It analyzes whether the traffic density is low, moderate, or high.

By continuously monitoring vehicle count across frames, the system can determine the overall traffic conditions at the intersection. This analysis is crucial for calculating signal timing dynamically.

The module helps ensure that heavily congested lanes receive longer signal durations, while lanes with fewer vehicles receive shorter signal durations.

5.6 Signal Timing Calculation Module

The Signal Timing Calculation Module determines the optimal duration for the green signal based on the detected vehicle count. This module uses a predefined formula to calculate the signal duration dynamically.

The signal timing is calculated using the formula:

$$\text{Green Signal Time} = \max(10, \text{Vehicle Count} \times 2)$$

This formula ensures that the signal duration increases as the number of vehicles increases, while maintaining a minimum green signal time of 10 seconds.

By dynamically adjusting the signal duration, the system helps reduce waiting time and improves overall traffic flow efficiency.

5.7 Traffic Signal Control Module

The Traffic Signal Control Module is responsible for applying the calculated signal timing to control the traffic signals. Based on the calculated duration, the system determines how long the signal should remain green before switching to the next lane.

In a real-world implementation, this module can be integrated with actual traffic signal hardware to automatically control signal lights at road intersections.

6. DATA FLOW DIAGRAM (DFD)

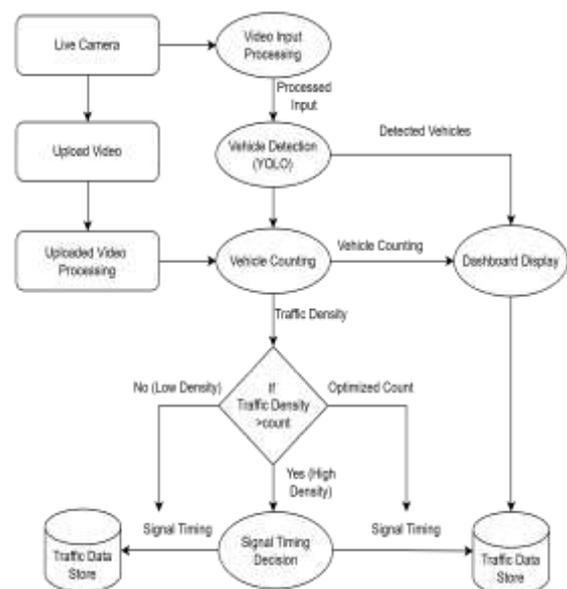


Figure 2: Data Flow Diagram of the Proposed System

The Data Flow Diagram illustrates how information flows through the system.

The process begins with video input from traffic cameras, which is sent to the vehicle detection module.

The detection module processes the video frames and identifies vehicles using the YOLO model.

The detected vehicles are then counted and analyzed to determine traffic density. This information is forwarded to the signal timing module, which calculates the optimal green signal duration.

Finally, the system sends the calculated signal timing to the traffic signal controller, which adjusts the signal duration accordingly.

7. Methodology

The proposed system follows a structured methodology to detect vehicles from traffic video and optimize traffic signal timing based on real-time traffic density. The system begins by capturing video input from traffic cameras installed at road intersections. The captured video stream is processed continuously and divided into individual frames to enable real-time analysis using image processing techniques.

Each frame is analysed using the YOLO deep learning model, which is capable of detecting multiple objects within an image. The model identifies vehicles such as cars, buses, trucks, and motorcycles and generates bounding boxes around the detected objects. After vehicle detection, the system counts the number of vehicles present in each frame to estimate the traffic density at the intersection.

8. Implementation

The system is implemented using a combination of software technologies and libraries.

Programming Language: Python

Libraries Used:

- **OpenCV** – For image processing and video frame extraction
- **YOLO** – For real-time vehicle detection
- **NumPy** – For numerical operations and calculations

Framework

Flask – Used to create a web-based interface for displaying vehicle detection results and signal timing information.

The system processes video frames continuously and performs vehicle detection in real time. The detected vehicles are counted and used to calculate signal timing dynamically.

9. Results Analysis

The experimental results show that the proposed system can effectively detect vehicles from traffic video frames using the YOLO object detection model. The system successfully identifies different types of vehicles and generates bounding boxes around them, enabling accurate vehicle counting. During testing, the system was able to process video frames efficiently and adjust signal timing according to traffic density. Lanes with higher vehicle counts received longer green signal durations, while lanes with fewer vehicles were allocated shorter signal times. This adaptive signal control mechanism helps improve traffic flow and reduce waiting time at intersections.

10. CONCLUSION

The Smart Traffic Signal Optimization System demonstrates the effective use of computer vision and deep learning techniques for improving traffic management. By detecting vehicles in real time and dynamically adjusting signal timing, the system ensures efficient traffic flow and reduces congestion at road intersections.

The use of YOLO for vehicle detection provides high accuracy and real-time performance, making the system suitable for practical applications in urban traffic management. The proposed system can be integrated with smart city infrastructure to develop advanced intelligent transportation systems.

11. FUTURE ENHANCEMENTS

Future improvements to the system may include:

- Multi-intersection traffic signal coordination
- Emergency vehicle detection and priority signalling

- Integration with IoT-based traffic management systems
- Cloud-based traffic monitoring platforms
- Use of more advanced deep learning models for improved detection accuracy

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *Computer Vision and Pattern Recognition*, 2018.
- [3] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.
- [4] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, London, 2011.
- [5] S. S. S. Prasad, V. K. Prasad, and M. S. Kumar, "Intelligent Traffic Control System Using Image Processing," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 6, no. 4, pp. 428–432, 2017.
- [6] M. S. Bhaskar and K. Karthik, "Smart Traffic Signal Control System Using Image Processing," *International Journal of Engineering Research & Technology*, vol. 8, no. 5, pp. 421–425, 2019.
- [7] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
- [8] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature Journal*, vol. 521, pp. 436–444, 2015.
- [10] S. Kato, T. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle Control Algorithms for Cooperative Driving with Automated Vehicles and Intervehicle Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, 2002.
- [11] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Computer Vision and Pattern Recognition*, 2020.
- [12] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, 2017.
- [13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [15] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "MultiNet: Real-Time Joint Semantic Reasoning for Autonomous Driving," *IEEE Intelligent Vehicles Symposium*, pp. 1013–1020, 2018.
- [16] H. K. Patel and M. S. Thakkar, "Real-Time Vehicle Detection and Counting Using Computer Vision Techniques," *International Journal of Computer Applications*, vol. 179, no. 12, pp. 25–29, 2018.
- [17] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, 2001.
- [18] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *International Conference on Learning Representations (ICLR)*, 2015.

[20] M. A. Hossain, M. A. Rahman, and M. S. Islam, “Smart Traffic Management System Using Computer Vision and Internet of Things,” *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 3, pp. 345–354, 2020.