Automated Vehicle Entry and Exit Recording System Using AI

BHUVANESH P, ELAYARAJAN R, HARIHARAN C, KANAGADURGA E

1,2,3 5th Semester B.E. Students Department of Computer Science and Engineering
E.G.S Pillay Engineering College, Nagapattinam,
Tamil Nadu, India
4 Professor, Department of Computer Science and Engineering
E.G.S Pillay Engineering College, Nagapattinam,
Tamil Nadu, India

How to Cite this Article:
P, B., R, E. & C, H. (2026). Automated Vehicle Entry and Exit Recording System Using AI. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05). https://doi.org/10.55041/ijcope.v2i5.796

https://doi.org/ 10.55041/ijcope.v2i5.796

Abstract – The rapid growth of vehicles in urban areas has increased the need for intelligent traffic monitoring and automated vehicle analysis systems. Manual vehicle monitoring methods are time-consuming, less efficient, and prone to human errors. To overcome these limitations, this project proposes an **AI-Based Vehicle Counting and Crossing Time Analysis System** using Computer Vision and Deep Learning technologies. The proposed system uses the **YOLOv8** object detection model to detect and track vehicles from traffic video streams in real time. Different types of vehicles such as cars, buses, trucks, and motorcycles are identified automatically using Artificial Intelligence techniques. A virtual checkpoint line is created within the video frame to monitor vehicle movement and count vehicles whenever they cross the predefined region. The system also integrates **EasyOCR** technology for optional license plate recognition operations. The OCR module extracts alphanumeric characters from vehicle number plates and converts them into machine-readable digital text. The detected vehicle information, crossing activity, and monitoring statistics are displayed through a **Streamlit-based dashboard interface**.The proposed system improves traffic monitoring efficiency, reduces manual effort, minimizes counting errors, and supports intelligent transportation applications. The project can be used in smart traffic monitoring systems, parking management, highway analysis, toll gate monitoring, and smart city surveillance applications. The developed system demonstrates the successful integration of Artificial Intelligence, Computer Vision, OCR, and Real-Time Video Processing technologies for intelligent vehicle monitoring and analysis.

Keywords – Artificial Intelligence, YOLOv8, Vehicle Detection, Vehicle Counting, OCR, OpenCV, Traffic Monitoring, Computer Vision

INTRODUCTION

In recent years, the rapid increase in the number of vehicles has created major challenges in traffic management, road safety, parking control, and transportation monitoring. Traditional manual vehicle monitoring methods are inefficient, time-consuming, and highly dependent on human operators. Manual counting and monitoring may also lead to inaccurate records and delayed traffic analysis. To overcome these limitations, Artificial Intelligence and Computer Vision technologies are increasingly used in intelligent transportation systems. The AI-Based Vehicle Counting and Crossing Time Analysis System is developed to automate vehicle monitoring and traffic analysis operations using Deep Learning techniques. The proposed system uses the YOLOv8 object detection model to detect and track vehicles from real-time video streams or recorded traffic videos. Vehicles such as cars, buses, trucks, and motorcycles are identified automatically with high accuracy. A virtual checkpoint line is created inside the video frame to monitor vehicle movement. Whenever a vehicle crosses the checkpoint line, the system automatically updates the vehicle count and records the crossing activity. This helps in analyzing traffic density, vehicle flow, and movement patterns efficiently. The system also integrates OCR (Optical Character Recognition) technology using EasyOCR for optional license plate recognition. The OCR module extracts alphanumeric characters from vehicle number plates and converts them into machine-readable digital text. This feature improves vehicle identification and monitoring capabilities. The project is implemented using Python, OpenCV, YOLOv8, Streamlit, and EasyOCR technologies. A Streamlit-based dashboard is used to display live vehicle detection results, traffic statistics, OCR outputs, and real-time monitoring information through a user-friendly interface. The proposed system reduces manual effort, improves monitoring accuracy, supports intelligent traffic management, and provides an efficient solution for modern transportation systems. The project can be applied in highways, parking areas, toll gates, smart cities, educational institutions, and industrial surveillance systems**.**

PROBLEM STATEMENT

Traffic monitoring and vehicle management have become major challenges due to the rapid increase in the number of vehicles on roads. Traditional manual vehicle counting and monitoring methods are inefficient, time-consuming, and prone to human errors. Continuous manual observation of traffic using security personnel is difficult, especially in highways, toll gates, parking areas, and crowded urban regions. Existing traffic monitoring systems often require expensive hardware and may not provide accurate real-time vehicle analysis. In many cases, manual vehicle counting cannot efficiently track multiple vehicles simultaneously, resulting in inaccurate traffic statistics and poor traffic management decisions. Another major issue is the difficulty in analyzing vehicle movement and crossing activities from video surveillance systems. Conventional systems lack intelligent automation for detecting, tracking, and counting vehicles automatically. Identifying vehicle information manually from CCTV footage is also a challenging and time-consuming task. To overcome these limitations, there is a need for an intelligent automated system capable of detecting, tracking, and counting vehicles in real time using Artificial Intelligence and Computer Vision technologies. The system should accurately monitor vehicle movement, analyze crossing activities, and optionally recognize vehicle license plates automatically with minimal human intervention. The proposed AI-Based Vehicle Counting and Crossing Time Analysis System aims to provide an efficient, accurate, and automated solution for intelligent traffic monitoring and vehicle analysis applications.

OBJECTIVES

The main objective of this project is to develop an AI-Based Vehicle Counting and Crossing Time Analysis System for automatic vehicle detection, tracking, and counting using Computer Vision technologies.

1. To develop an AI-based vehicle monitoring system for automatic traffic analysis.

The system reduces manual vehicle monitoring using Artificial Intelligence technologies.

2. To detect vehicles automatically using the YOLOv8 model.

The system identifies vehicles such as cars, buses, trucks, and motorcycles from traffic videos.

3. To track moving vehicles with unique IDs.

Vehicle tracking helps avoid duplicate counting and improves monitoring accuracy.

4. To count vehicles automatically when crossing a checkpoint line.

The system updates vehicle count in real time whenever a vehicle crosses the line.

5. To analyze vehicle movement and traffic flow efficiently.

The project helps understand vehicle movement patterns and traffic density.

6. To reduce manual effort and human errors in traffic monitoring.

Automation improves accuracy and saves time in vehicle analysis operations.

LITERATURE REVIEW (SUMMARY)

Many researchers have developed intelligent traffic monitoring systems using Artificial Intelligence and Computer Vision technologies. Traditional vehicle monitoring methods mainly depended on manual observation and sensor-based systems, which often produced inaccurate results and required high maintenance. Recent developments in Deep Learning and Image Processing have improved vehicle detection and traffic analysis efficiency significantly. Several research works used Convolutional Neural Networks (CNN) and YOLO-based object detection models for real-time vehicle detection and tracking. YOLO algorithms became popular because of their high detection speed and accuracy in traffic surveillance applications. Researchers also integrated vehicle tracking techniques to monitor vehicle movement and avoid duplicate counting during video processing. Several existing systems implemented OCR technology for automatic license plate recognition. OCR-based approaches helped convert vehicle number plates into machine-readable text for vehicle identification and monitoring purposes. OpenCV and EasyOCR are commonly used in modern traffic surveillance projects for image preprocessing and text extraction. Recent studies also focused on intelligent transportation systems and smart city applications using AI-based traffic analysis. These systems improved traffic monitoring accuracy, reduced human effort, minimized manual errors, and supported real-time traffic management operations. Based on the literature survey, it is observed that AI-based vehicle detection, tracking, counting, and OCR technologies provide efficient solutions for modern traffic monitoring systems. The proposed project extends these concepts by integrating YOLOv8, vehicle tracking, checkpoint crossing analysis, OCR recognition, and a Streamlit-based real-time dashboard into a single intelligent monitoring system.

PROPOSED SYSTEM

The proposed system is an AI-Based Vehicle Counting and Crossing Time Analysis System. It is developed to automatically monitor traffic videos and count vehicles without manual effort. The system uses Artificial Intelligence, Computer Vision, and real-time video processing techniques. In this system, a traffic video or CCTV video is given as input. The video frames are processed using OpenCV. Each frame is sent to the YOLOv8 object detection model, which detects vehicles such as cars, bikes, buses, and trucks.

After detecting vehicles, the system tracks each vehicle using a unique tracking ID. This helps the system follow the same vehicle across multiple frames and prevents duplicate counting. A virtual checkpoint line is placed inside the video frame. When a vehicle crosses this line, the system automatically increases the vehicle count. This helps to analyze vehicle flow, traffic density, and movement direction.

The system also includes an optional OCR module using EasyOCR. This module reads vehicle number plates from detected vehicle images and converts them into digital text. A Streamlit dashboard is used to display the output in a user-friendly way. The dashboard shows uploaded video, detected vehicles, vehicle count, vehicle type, OCR result, confidence score, and real-time processing status.

The proposed system improves traffic monitoring accuracy, reduces manual work, saves time, and supports smart transportation applications such as highways, parking areas, toll gates, and smart city surveillance systems.

SYSTEM ARCHITECTURE

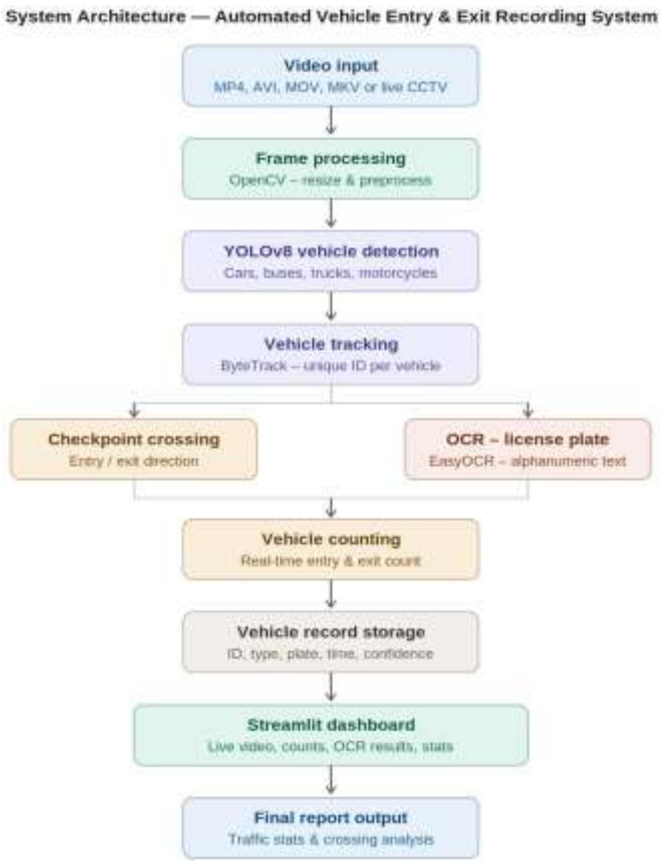

Fig. 1: System Architecture – Automated Vehicle Entry and Exit Recording System

The system architecture of the AI-Based Vehicle Counting and Crossing Time Analysis System explains the flow of data from video input to final output display. Based on the GitHub project, the system is divided into different modules such as video input, YOLOv8 vehicle detection, vehicle tracking, checkpoint crossing detection, OCR processing, vehicle counting, data storage, and Streamlit dashboard.

A. Video Input

The first stage of the system is video input. The user uploads a traffic video or provides a live CCTV camera feed through the Streamlit dashboard. The uploaded video acts as the primary source for vehicle monitoring and traffic analysis. Different video formats such as MP4, AVI, MOV, and MKV are supported by the system.

B. Frame Processing

After receiving the video input, the system processes the video frame by frame using OpenCV. Each frame is extracted continuously from the video stream for further analysis. Frame resizing and preprocessing operations are performed to improve detection speed and reduce computational complexity during real-time processing.

C. YOLOv8 Vehicle Detection

The processed frames are passed to the YOLOv8 object detection model. YOLOv8 is a Deep Learning-based object detection algorithm used to identify vehicles in real time. The system detects different vehicle types such as cars, buses, trucks, and motorcycles by generating bounding boxes around detected vehicles along with confidence scores.

D. Vehicle Tracking

After vehicle detection, the tracking module assigns a unique tracking ID to each detected vehicle. The tracking system continuously follows vehicle movement across multiple frames. This helps prevent duplicate counting and improves monitoring accuracy during continuous traffic flow analysis.

E. Checkpoint Crossing Detection

A virtual checkpoint line is placed inside the video frame for crossing analysis. The system continuously checks vehicle positions and determines whether a vehicle crosses the predefined checkpoint line. The crossing direction such as upward or downward movement is also identified during this process.

F. Vehicle Counting

Whenever a vehicle crosses the checkpoint line, the system automatically increases the vehicle count. Vehicle counting is performed in real time without human intervention. This module helps analyze traffic density, vehicle movement patterns, and road usage statistics efficiently.

G. OCR Licence Plate Recognition

The system also includes an optional OCR module using EasyOCR. The OCR module extracts vehicle number plate regions from detected vehicle images and converts the license plate characters into machine-readable digital text. Image preprocessing techniques are used to improve OCR accuracy.

H. Vehicle Record Storage

The detected vehicle information is stored temporarily during processing. The system stores details such as vehicle ID, vehicle type, license plate text, confidence score, frame number, and detection time. These records help generate vehicle statistics and monitoring reports.

I. Streamlit Dashboard Output

Finally, all processing results are displayed through the Streamlit dashboard interface. The dashboard shows real-time vehicle detection output, vehicle counts, OCR results, confidence values, live processed video frames, and traffic monitoring statistics. The dashboard provides a user-friendly interface for intelligent traffic analysis and monitoring operations.

METHODOLOGY

The methodology of the proposed **AI-Based Vehicle Counting and Crossing Time Analysis System** explains the step-by-step working process used for vehicle detection, tracking, counting, OCR recognition, and traffic monitoring. The system uses Artificial Intelligence, Computer Vision, and Deep Learning techniques for automated traffic analysis.

STEP 1: VIDEO COLLECTION

The first step of the methodology is video collection. The system accepts traffic videos or live CCTV camera feeds through the Streamlit dashboard. The uploaded video acts as the input source for vehicle monitoring and traffic analysis.

STEP 2: FRAME EXTRACTION AND PREPROCESSING

The input video is processed frame by frame using OpenCV. Each frame is extracted continuously from the video stream. Image preprocessing techniques such as resizing and frame optimization are performed to improve processing speed and detection accuracy.

STEP 3: VEHICLE DETECTION USING YOLOv8

After preprocessing, each frame is passed to the YOLOv8 object detection model. YOLOv8 detects vehicles such as cars, buses, trucks, and motorcycles in real time. Bounding boxes are created around detected vehicles along with confidence scores.

STEP 4: VEHICLE TRACKING

The detected vehicles are tracked continuously using tracking algorithms. Each vehicle is assigned a unique tracking ID, which helps the system follow vehicle movement across multiple frames and avoid duplicate counting.

STEP 5: CHECKPOINT CROSSING ANALYSIS

A virtual checkpoint line is placed inside the video frame. The system continuously checks whether tracked vehicles cross the checkpoint line. When a vehicle crosses the line, the crossing event is detected automatically along with the movement direction.

STEP 6: VEHICLE COUNTING

Whenever a crossing event occurs, the vehicle count is updated automatically. The system counts vehicles in real time and maintains traffic statistics efficiently without manual intervention.

STEP 7: OCR-BASED LICENCE PLATE RECOGNITION

The system also performs optional license plate recognition using EasyOCR. The OCR module extracts text from vehicle number plates and converts it into digital machine-readable format for vehicle identification.

STEP 8: VEHICLE DATA STORAGE

The detected vehicle details such as vehicle ID, vehicle type, license plate number, confidence score, frame number, and detection time are stored temporarily during processing for monitoring and analysis purposes.

STEP 9: DASHBOARD VISUALIZATION

Finally, all results are displayed through the Streamlit dashboard interface. The dashboard shows live vehicle detection output, vehicle count, OCR results, processed video frames, and real-time traffic monitoring statistics in a user-friendly manner.

STEP 10: RESULT GENERATION

After completing video processing, the system generates the final output containing vehicle counting results, traffic statistics, OCR outputs, and crossing analysis information. The generated results help improve intelligent traffic monitoring and transportation management applications.

RESULT AND DISCUSSION

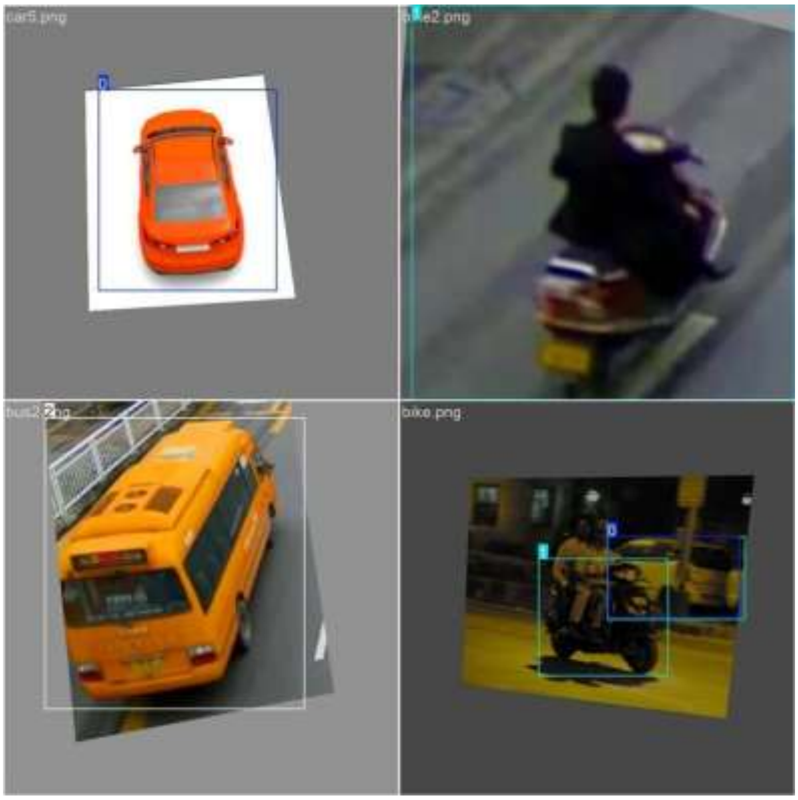

Fig. 2: Vehicle Detection Samples – YOLOv8 bounding boxes on car, bike, bus, and motorcycle

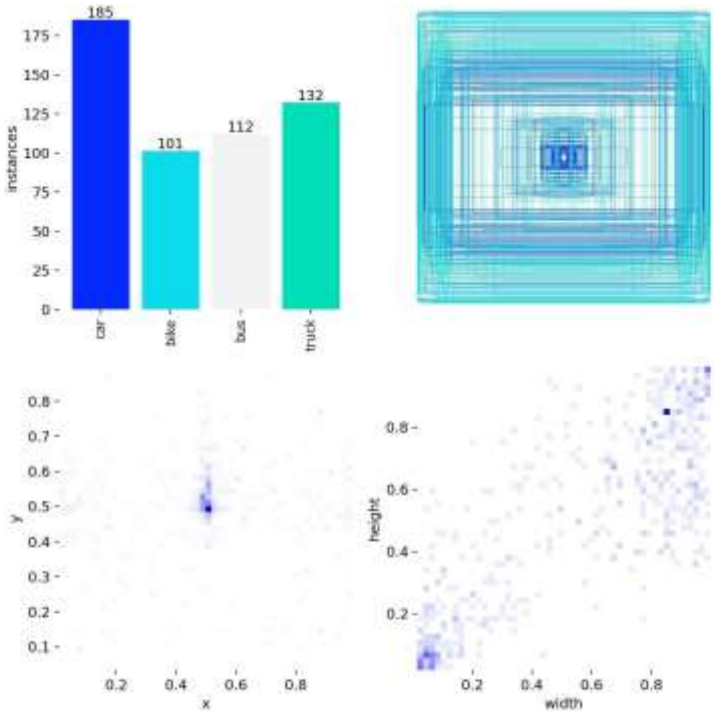

Fig. 3: Dataset Distribution – Instance count per class and bounding box spatial analysis

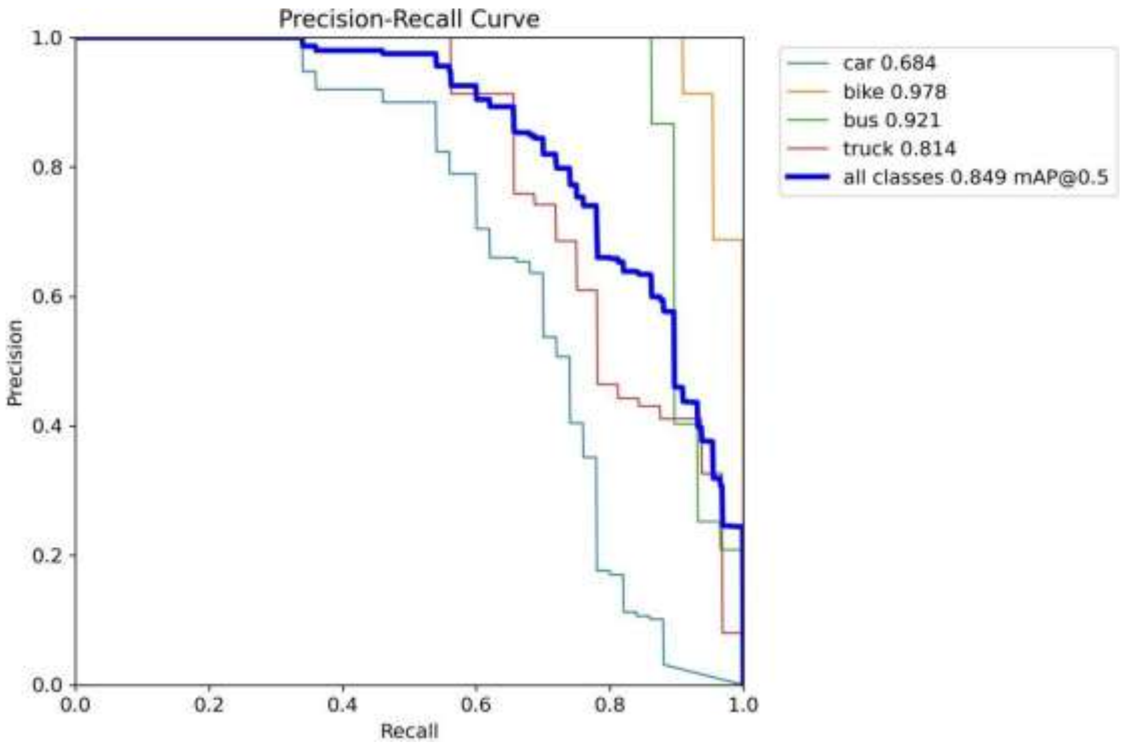

Fig. 4: Precision-Recall Curve – mAP@0.5 = 0.849 across all vehicle classes

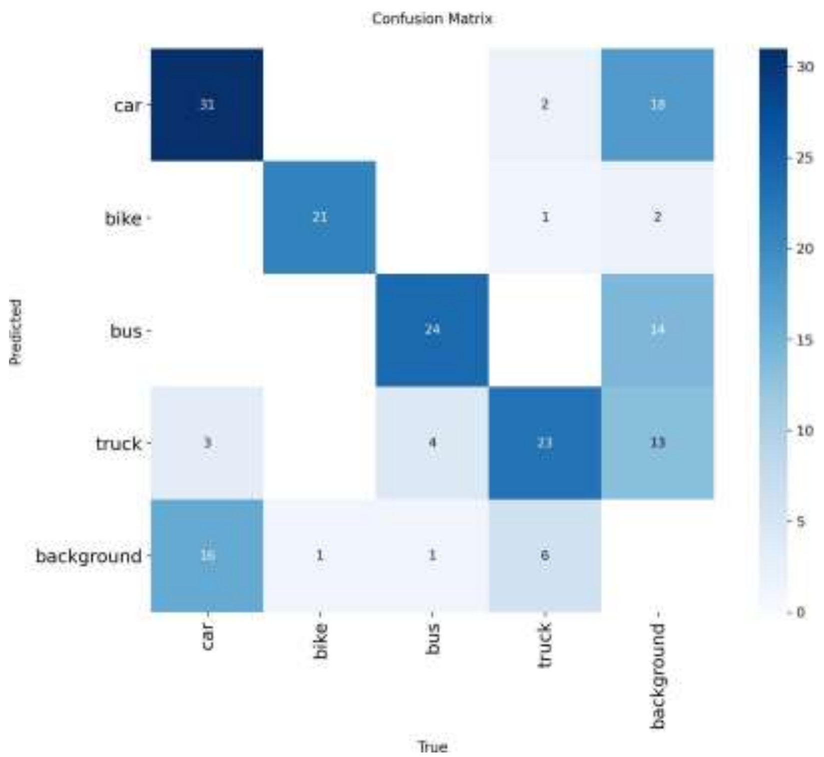

Fig. 5: Confusion Matrix – Predicted vs. True vehicle class classification results

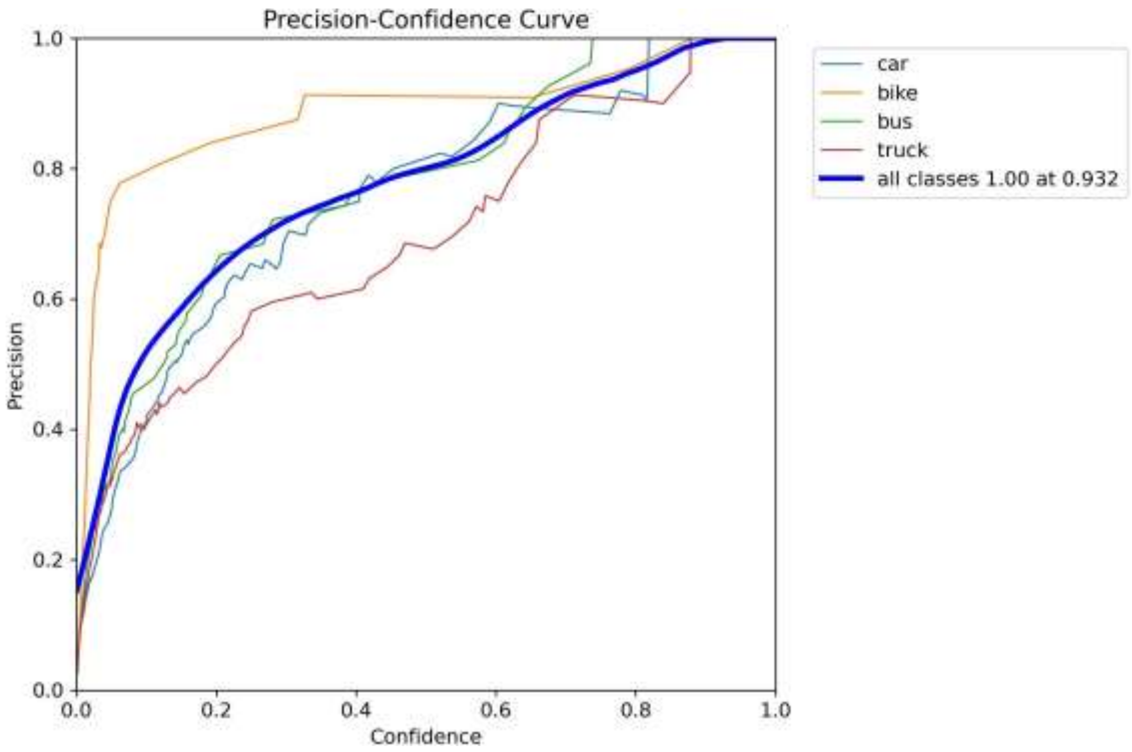

Fig. 6: Precision-Confidence Curve – All classes reach precision 1.00 at confidence 0.932

The AI-Based Vehicle Counting and Crossing Time Analysis System was implemented based on the GitHub project source code. The system uses a Streamlit dashboard for video upload and real-time output display. The uploaded traffic video is processed frame by frame using OpenCV, and the processing pipeline is handled by the RealtimeProcessor module. The result of the system shows that vehicles are detected successfully using the YOLOv8 model. The VehicleDetector module identifies vehicle classes such as car, motorcycle, bus, and truck. For every detected vehicle, the system generates a bounding box, class name, confidence score, and center point. This helps the system clearly identify vehicles from traffic video frames.
After detection, the system tracks vehicles using YOLOv8 tracking with ByteTrack configuration. Each vehicle is assigned a unique tracking ID. This tracking ID helps the system follow the same vehicle across multiple frames and avoids duplicate counting. The checkpoint crossing module successfully detects whether a vehicle crosses the predefined virtual line. The Checkpoint class checks the previous and current position of the vehicle. If the vehicle crosses the line, the system identifies the direction as upward or downward movement.
The VehicleCounter module updates the vehicle count only when a new vehicle crosses the checkpoint line. This improves counting accuracy because the same vehicle is not counted repeatedly. The system also maintains direction-wise counts such as up, down, entry, and exit. The OCR module using EasyOCR reads license plate text from detected vehicle regions. The OCRHelper module crops the lower part of the detected vehicle image, preprocesses it using grayscale conversion and thresholding, and extracts text from the license plate area. OCR output depends on image clarity, lighting, and plate visibility. The VehicleStore module stores detected vehicle details during the current session. The stored data includes vehicle ID, vehicle type,

license plate text, detection time, confidence score, and frame number. Low-confidence detections are ignored to reduce false records. The Streamlit dashboard displays the final output in a user-friendly way. It shows the live processed video, total vehicle count, vehicle type statistics, FPS, detection table, OCR result, and processing status. The dashboard also allows the user to adjust confidence threshold, checkpoint line position, OCR option, and frame processing rate. Overall, the project successfully performs vehicle detection, tracking, counting, OCR-based license plate recognition, and real-time dashboard visualization. The system reduces manual vehicle counting effort, improves traffic monitoring accuracy, and supports intelligent transportation applications such as highways, toll gates, parking areas, and smart city traffic monitoring.

ADVANTAGES

- The system automatically detects vehicles using Artificial Intelligence and YOLOv8 technology. It reduces manual monitoring effort and improves traffic analysis efficiency.
- The system counts vehicles automatically whenever they cross the checkpoint line. This helps monitor traffic flow and vehicle movement accurately in real time.
- Each detected vehicle is assigned a unique tracking ID for continuous monitoring. This prevents duplicate counting and improves overall system accuracy.
- The EasyOCR module extracts vehicle number plate text automatically from video frames. This helps improve vehicle identification and monitoring operations.
- The Streamlit dashboard provides a simple and interactive interface for traffic monitoring. Users can view live detection results, vehicle counts, OCR output, and traffic statistics easily.
- The system supports intelligent transportation and smart city applications. It can be used in highways, parking areas, toll gates, industries, and surveillance systems for efficient traffic management.

LIMITATIONS

The proposed AI-Based Vehicle Counting and Crossing Time Analysis System has certain limitations during real-time traffic monitoring and analysis. The overall system performance mainly depends on the quality of the input video and environmental conditions. Poor lighting conditions, rain, fog, shadows, or unclear CCTV footage may reduce vehicle detection accuracy and affect OCR performance.

The OCR-based license plate recognition module may not always provide accurate results if the number plate is blurred, damaged, tilted, partially hidden, or captured from a long distance. Fast-moving vehicles may also reduce text recognition accuracy during frame processing. The system requires sufficient hardware resources such as sufficient RAM capacity, processor speed, and GPU support for efficient real-time processing. Low-end systems may experience processing delay, frame skipping, or reduced FPS performance during continuous traffic monitoring. Vehicle detection and tracking accuracy depend on proper camera placement and viewing angle. Incorrect camera positioning may cause incomplete vehicle visibility, tracking errors, and inaccurate vehicle counting results. In highly crowded traffic conditions, overlapping vehicles may create difficulties in tracking and checkpoint crossing analysis.

The current system mainly supports common vehicle categories such as cars, buses, trucks, and motorcycles. Detection accuracy for uncommon vehicles or non-standard objects may be lower compared to trained vehicle classes.

The project currently stores detected vehicle data temporarily during processing sessions. Permanent cloud-based storage, advanced database management, and large-scale multi-camera synchronization are not fully implemented in the current version of the system. Although the system performs real-time processing, handling multiple high-resolution video streams simultaneously may reduce overall performance and increase computational complexity. Continuous software operation and proper system configuration are also necessary for stable monitoring performance.

FUTURE SCOPE

The proposed AI-Based Vehicle Counting and Crossing Time Analysis System can be further improved using advanced Artificial Intelligence and smart transportation technologies. In future developments, the system can be integrated with real-time CCTV cameras for continuous live traffic monitoring in highways, smart cities, toll gates, and parking areas. Cloud-based storage and database systems can also be integrated to store vehicle records permanently and support large-scale traffic analysis. Future versions of the system may support multi-camera synchronization for monitoring multiple traffic locations simultaneously. The OCR module can be improved using advanced Deep Learning techniques to achieve better license plate recognition accuracy under low-light conditions, rain, fog, and blurred video frames. GPS and IoT technologies can also be integrated for intelligent vehicle tracking and smart transportation management.

Future enhancements may include automatic traffic violation detection, suspicious vehicle identification, smart parking management, traffic congestion analysis, and AI-based predictive traffic monitoring systems. Mobile application support can also be developed for remote traffic monitoring and alert notification systems.

The system can further be expanded for smart city surveillance, intelligent transportation systems, industrial security monitoring, and automated highway traffic management applications.

CONCLUSION

The proposed AI-Based Vehicle Counting and Crossing Time Analysis System was successfully developed and implemented using Artificial Intelligence, Computer Vision, and Deep Learning technologies. The system effectively performs automatic vehicle detection, tracking, counting, OCR-based license plate recognition, and real-time traffic monitoring using traffic video streams. The YOLOv8 object detection model successfully identified different vehicle types such as cars, buses, trucks, and motorcycles with good accuracy. Vehicle tracking techniques helped assign unique IDs for each detected vehicle, which reduced duplicate counting and improved monitoring efficiency. The checkpoint crossing mechanism also worked effectively by automatically updating the vehicle count whenever vehicles crossed the predefined virtual line. The EasyOCR module successfully extracted vehicle license plate text from clear video frames and improved vehicle identification capabilities. The Streamlit dashboard provided a simple and user-friendly interface for displaying live processed video output, vehicle statistics, OCR results, and traffic monitoring information.

The proposed system reduced manual effort, minimized human errors, improved traffic monitoring accuracy, and supported intelligent transportation applications. The project demonstrates the successful integration of YOLOv8, OpenCV, OCR, vehicle tracking, and real-time dashboard visualization technologies into a single intelligent traffic analysis system. The developed system can be effectively used in highways, parking areas, toll gates, industries, educational institutions, and smart city surveillance applications. Future improvements such as cloud integration, advanced OCR models, IoT support, and multi-camera synchronization can further enhance the performance and scalability of the system. Overall, the project provides an efficient, automated, and intelligent solution for modern traffic monitoring and vehicle analysis applications using Artificial Intelligence technologies.

REFERENCES

1. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
2. Glenn Jocher, Ayush Chaurasia, and Jing Qiu, "YOLOv8 Object Detection Model," *Ultralytics*, 2023.
3. Adrian Rosebrock, *Practical Python and OpenCV*, PyImageSearch Publications, 2019.
4. Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
5. Jaided AI, "EasyOCR Documentation," *GitHub Repository*, 2023.
6. Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.
7. Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B., "Simple Online and Realtime Tracking," *IEEE International Conference on Image Processing (ICIP)*, 2016.
8. Zhang, S., Benenson, R., and Schiele, B., "CityPersons: A Diverse Dataset for Pedestrian Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
9. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.
10. Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer Publications, 2011.