International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 1, Jan-Feb-2025, ISSN (Online): 2395-566X

OpenCV- Based Intelligent Vehicle Surveillance and Time Stamping System

Professor Dr.J.Preetha, Assisstant Professor Mr.R.Viswanathan, A Rasidha Begum, S Pooja, R Jona
Dept. of IT
Kongunadu College of Engineering and Technology, Trichy, India

Abstract- Automated traffic monitoring solutions have become necessary due to the difficulties of manual monitoring systems and the exponential growth in vehicular traffic. The new Advanced Vehicle Detection System described in this work uses sophisticated computer vision algorithms to identify, recognize, and log vehicle data in real time. Utilizing OpenCV, CNN (Convolutional Neural Network), YOLO (You Only Look Once), and OCR (Optical Character Recognition) technologies, the suggested system detects automobiles and records license plate information. In addition, the system gives law enforcement, traffic management, and institutional surveillance a reliable and scalable approach by automating the entry and exit timestamp logging process. Mostly we are developed for the college buses which has been include arrival and Departure time with an owner details and also the vehicle claim the insurance or not, These also updated the count of vehicle that are recognized by the entry and the exit time. Experimental results demonstrate the system's high precision and efficiency, ensuring its practical applicability in real- world scenarios. This practical and efficient system is an excellent example of how technology can address real- world challenges in monitoring and managing vehicles.

Index Terms- Vehicle Detection, Real-Time Monitoring, YOLO, CNN, OCR, OpenCV, Automated Traffic Management.

I. INTRODUCTION

The rising number of vehicles has altered city environments, turning traffic control into an essential concern for policymakers, law enforcement, and urban planners. Traditional approaches to vehicle monitoring, relying on human operators and rudimentary surveillance tools, are insufficient to handle the complexities of contemporary traffic. Additionally, the necessity for precise and effective data gathering for upcoming urban development and traffic oversight has become crucial [1].

Recent developments computer vision have created new opportunities for the automation of vehicle detection and monitoring. In this context, creating a real-time Advanced Vehicle Detection System is particularly significant. This project offers a holistic approach to traffic management issues by combining OpenCV for image pre-processing, YOLO for detecting objects, CNN for classifying vehicles, and OCR for recognizing license plates [2]. The main goal of this system is to identify vehicles in real-time, read their license plates, and record essential information, like timestamps, into a database for further analysis. This information can serve multiple purposes, such as analyzing traffic patterns, supporting law enforcement, and managing access in restricted zones. Furthermore, the system's modular architecture guarantees flexibility for different settings and scalability for extensive implementations [3]. The integration of advanced technologies like YOLO and OCR guarantees high accuracy and reliability, even in adverse conditions [4]. This Advanced Vehicle Detection System is to better traffic control, the suggested system could boost public safety by facilitating rapid reactions to vehicle-related offenses[5]. This system is highly applicable in remote or rural areas, where manual monitoring is often limited. It can enhance road safety by detecting their plates, monitoring and preventing theft on colleges or others [6].

II. RELATED WORKS

[1]. The implementation of vehicle detection algorithms for autonomous vehicles has been widely explored, particularly in the context of self-driving cars navigating complex environments such as toll roads. This work highlighted the challenges of accurately identifying and tracking vehicles in real-time, which is crucial for the safe and efficient operation of autonomous vehicles. The authors implemented machine learning models, particularly convolutional neural networks (CNNs), to enhance the accuracy and speed of vehicle detection, with a focus on toll road environments. In the broader landscape of vehicle detection for autonomous driving, multiple approaches have been explored. For example, earlier studies have utilized traditional computer vision techniques, such as background subtraction, optical

International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 1, Jan-Feb-2025, ISSN (Online): 2395-566X

flow, and edge detection algorithms, to identify moving objects on the road. However, these methods often struggle with complex or dynamic environments, where occlusion, lighting conditions, or road irregularities may affect detection performance.

The transition to deep learning models, has significantly improved vehicle detection, as deep learning algorithms are able to learn more nuanced features of vehicle appearances and behaviors. This shift towards machine learning has become a major trend in the development of self-driving systems, particularly for applications like toll booth detection, where vehicles exhibit consistent patterns.

[2]. A study by Bush and Dimililer (2022) focused on both static and dynamic pedestrian detection algorithms for visual-based driver-assistive systems, which are essential for ensuring that vehicles can safely interact with pedestrians in real-time. Their work combined static detection, which focuses on identifying pedestrians at fixed locations, with dynamic detection, which accounts for pedestrians moving across the road. This hybrid approach allowed for a more comprehensive detection system capable of responding to both stationary and moving threats, thus enhancing the overall safety of autonomous vehicles in complex urban environments.

In comparison to traditional vehicle detection algorithms, pedestrian detection often presents more significant challenges due to the variability in human appearance, movement, and environmental factors. Bush and Dimililer (2022) utilized advanced computer vision techniques, including deep learning and feature extraction, to address these challenges. Their method significantly outperformed classical algorithms, such as Haar cascades and background subtraction, by leveraging modern convolutional neural networks (CNNs). This approach enables the system to effectively identify pedestrians in diverse lighting conditions, weather environments, and dynamic scenarios, making it an ideal solution for real-time driver-assistive applications, particularly in high-traffic areas such as intersections and pedestrian zones.

[3]. The integration of vehicle detection and tracking technologies plays a crucial role in the development of autonomous driving systems. In a practical guide, Karunakaran (2020) outlined the methods employed in Udacity's Self-Driving Car Nanodegree program to develop robust vehicle detection and tracking algorithms. The primary focus of this work was on the implementation of real-time systems for identifying vehicles on the road and tracking their movement, an essential component for enabling self-driving cars to safely navigate dynamic traffic environments. By utilizing advanced computer vision techniques such as background subtraction, color histograms, and optical flow, Karunakaran demonstrated how these tools could be applied to detect and track moving vehicles across a variety of scenarios.

III. PROPOSED APPROACH

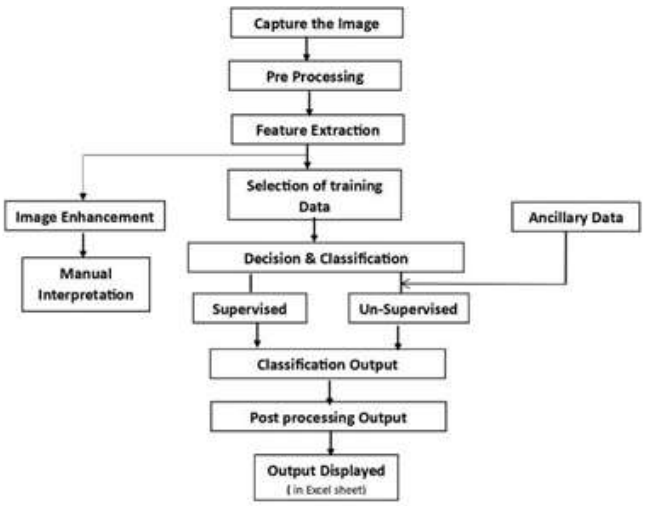

Figure 1: Architecture for proposed Approach

The flowchart outlines the process of image analysis and classification, showcasing how images are captured, pre-processed, and analysed. The process involves feature extraction, selection of training data, and decision-making for classification using supervised or unsupervised approaches. Ancillary data can also be incorporated to refine the analysis.

The classification output is post- processed, and the results are displayed, often in an Excel sheet for easy interpretation. Additionally, an alternative pathway for manual interpretation through image enhancement ensures flexibility and accuracy, particularly in cases requiring human expertise.

YOLO (You Only Look Once) is a real-time object detection algorithm that aligns with this flow. YOLO takes an image as input and, during feature extraction, processes it in a single forward pass through a neural network. Unlike traditional classification, YOLO performs classification and localization simultaneously, identifying multiple objects in real-time. YOLO's supervised training fits within this flowchart under "Selection of training data" and "Decision & Classification," leveraging labelled datasets to train the model for high-speed and accurate object detection. Its integration streamlines the classification process, making it highly efficient.
Hardware Set-up:

(Figure .1) illustrates the Architecture for the proposed Traffic management System.

1. High-Definition Cameras
To effectively capture real-time video feeds for vehicle detection and recognition, it's essential to use cameras with a resolution of 1080p or higher for clear image quality. Wide-angle lenses maximize coverage, and weatherproof casings ensure durability in outdoor environments. Infrared support enhances performance in low-light conditions. These features make the cameras reliable data sources for the detection system.

2. Local Storage
To store video feeds, processed data, and system logs, it's essential to utilize reliable storage solutions such as external hard disk drives (HDDs), solid-state drives (SSDs), or integrated storage systems. The required storage capacity should be at least 1 terabyte (TB) or more, depending on factors like video resolution, frame rate, the number of cameras, and the desired retention period. For instance, higher resolution and frame rates generate larger file sizes, necessitating greater storage capacity. Implementing such storage solutions ensures a dependable medium for maintaining all captured and processed data, supporting efficient retrieval and analysis when needed.

Software Requirements
Operating System Compatibility
The system must be compatible with modern operating systems, including Windows, Linux, and macOS, to ensure broad accessibility and ease of integration into existing infrastructure. This compatibility will enable seamless deployment across various platforms, catering to different user preferences and organizational needs. Additionally, cross-platform support ensures that the system can be easily integrated with diverse hardware configurations, maximizing its utility.

Programming Language Support
The software should be developed using robust and widely-supported programming languages such as Python or C++. These languages offer extensive libraries and frameworks suitable for real-time image processing and machine learning tasks integral to vehicle detection systems.

Database Management System (DBMS)
A reliable DBMS, such as PostgreSQL or MySQL, is required to manage and store large volumes of data, including vehicle logs, detection records, and user information, ensuring data integrity and efficient retrieval.

Security Protocols
The system must implement advanced security protocols, including multi-factor authentication and data encryption, to protect sensitive information and prevent unauthorized access, thereby maintaining the integrity and confidentiality of the data.

Real-Time Data Processing Capabilities
The system must support real-time data processing to ensure immediate detection and response to vehicle movements. This capability is crucial for applications such as traffic management and automated toll collection, where timely data processing is essential for operational efficiency.

Scalability and Modular Architecture
The software should be designed with a scalable and modular architecture to accommodate future expansions and integrations. This design approach allows for the addition of new features or integration with other systems without significant restructuring, ensuring long-term adaptability and maintainability.

IV. WORKFLOW

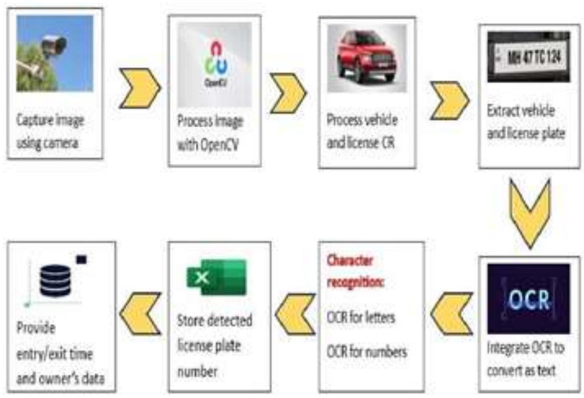

Figure 2: Architecture diagram for Workflow

1. Capturing image using Camera
The procedure starts with high-definition cameras taking pictures of vehicles instantaneously. These cameras come with attributes such as high resolution, weatherproofing, and infrared capabilities, enabling them to function effectively in various environmental situations. The images obtained act as the fundamental data for the next steps in the workflow, guaranteeing clear and accurate input for the system.

2. Process image with OpenCV
The captured images are then processed using OpenCV, a powerful computer vision library. This step includes pre-processing techniques such as noise reduction to enhance image clarity, grayscale conversion to simplify image data for faster processing, and edge detection to highlight key features like vehicle outlines and license plates. These optimizations ensure the images are suitable for accurate detection and recognition.

3. Process Vehicle and License Plate Detection
In this stage, the system uses YOLO (You Only Look Once) to detect vehicles and identify the regions containing license plates. YOLO efficiently draws bounding boxes around

International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 1, Jan-Feb-2025, ISSN (Online): 2395-566X

detected vehicles and license plates, isolating these regions of interest (ROI) for further processing. This step is crucial for distinguishing vehicles and preparing for detailed analysis of license plates.

4. Extract Vehicle and License Plate Details
After the license plate region is identified, the system extracts it for further processing. The region of interest is cropped and enhanced to improve clarity and ensure the alphanumeric characters on the license plate are legible. The extracted license plate image is then processed using optical character recognition (OCR) techniques to accurately identify the alphanumeric characters. These characters are parsed and stored for further validation, such as checking against databases for vehicle information or monitoring traffic patterns.

5. Integrate OCR to Convert as Text
The enhanced license plate region is processed using OCR (Optical Character Recognition) to convert the alphanumeric characters into text. OCR identifies both letters and numbers on the license plate, creating a text- based version of the plate details. This conversion is essential for further validation and logging. The extracted text can then be cross-referenced with vehicle registration databases to verify the authenticity of the license plate. Additionally, the converted text can be stored in a log file for real-time monitoring or future reference, ensuring accurate tracking of vehicle data.

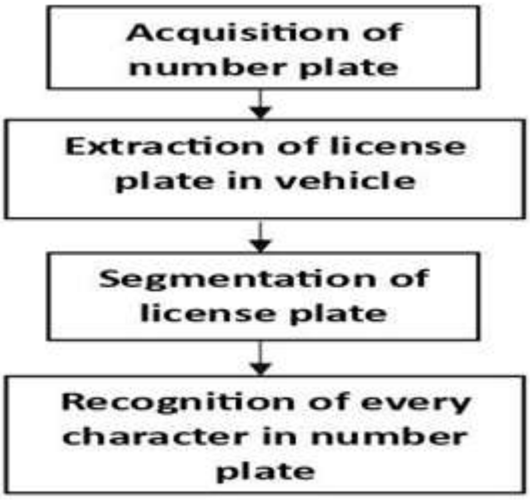

Figure 3. Automated Number Plate Recognition (ANPR) WorkFlow

6. Character Recognition for Letters and Numbers
OCR performs character recognition in two steps. First, it detects and extracts the alphabetic characters on the license plate, such as state codes. Second, it identifies and converts the numeric details, like vehicle registration numbers. This dual-stage recognition ensures the complete and accurate extraction of license plate information.

7. Store Detected License Plate Numbers
The recognized license plate data is stored in a structured format for future analysis. An Excel sheet or database is used to log the details, including the license plate number, entry and exit timestamps, and other relevant metadata. This organized storage allows for easy access and management of vehicle data. By maintaining a detailed and easily retrievable database, the system supports real-time decision-making and long-term planning for better vehicle management.

8. Provide Entry/Exit Time and Owner's Data
The final step involves generating detailed logs of vehicle movements. The system provides the entry and exit times of vehicles along with their corresponding license plate information. This data is essential for monitoring, reporting, and analyzing vehicle activity, enabling administrators to make data-driven decisions effectively.

V. OUTPUT

Figure 4. Vehicle Detection Dashboard

Advanced Vehicle Detection System designed for real-time monitoring and automated time recording using OpenCV technology. The system focuses on recognizing vehicle license plates, as demonstrated by the captured images of plates with distinct registration numbers (e.g., TN.47.AY.7554, TN.48AZ.6208, TN.48AZ.0401). It features options such as "Detect Image" and "Show Details," which likely facilitate the identification and retrieval of detailed information about the detected vehicles.

	A	B	C	D	E	F	G
1	Number Plate	Owner Details	Arrival Time	Depature Time	Bus No	Insurance Claim	Place
2	TN 47 AY 7554	Kongunadu Institutions	8:47 AM	5:10 PM	17	YES	THIRUPENJALAI
3	TN 48 AY 6208	Kongunadu Institutions	8:50 AM	5:14 PM	27	YES	SIVIYAMPALAYAM
4	TN 48 AZ 0401	Kongunadu Institutions	8:52 AM	5:16 PM	30	YES	PALAPALAYAM
5							
6							
7							
8							

Figure 5. Vehicle Data Sheet

This table showcases detailed data about vehicles detected by an advanced system, likely related to the buses of Kongunadu College of Engineering and Technology. It includes license plate numbers, the owner's details (PSK Periyasamy), arrival and departure times, bus numbers, insurance claim statuses, and corresponding locations (e.g., Thirupainjeli, Siviyampalayam, Palayapalayam). This organized data supports efficient tracking and management of vehicles, enhancing operational efficiency and monitoring in a structured manner.

This Excel sheet acts as a centralized repository for managing vehicle-related data, particularly for buses operating under Kongunadu College of Engineering and Technology. It records key information such as license plate numbers, bus ownership details, arrival and departure times, and assigned route locations. The inclusion of insurance claim status ensures readiness for financial and legal processes, while bus numbers and timings help in tracking schedules efficiently. The data also supports route analysis, enabling better planning and reduced delays. This well-structured format enhances accessibility and aids in seamless monitoring, ensuring smoother operation and better decision-making in daily transportation management.

VI. CONCLUSION

The Advanced Vehicle Detection System aims to tackle essential elements of vehicle safety and automated time tracking, offering a dependable and effective method for overseeing vehicle activity. As the demand for improved security and operational management grows, the system automates vehicle identification, license plate recognition, and time monitoring, guaranteeing that all vehicle entries and exits are precisely documented. By employing cutting-edge technologies like OpenCV for image preprocessing, YOLO for object detection, and OCR for recognizing license plates, the system ensures accurate and prompt surveillance.

This system greatly minimizes the chances of vehicle theft by keeping an extensive record of vehicle movements and cross-referencing license plate information with owner details. It guarantees that every vehicle entering or exiting the premises has permission, facilitating rapid detection of inconsistencies and unauthorized entrance. The automated time tracking function aids in monitoring vehicle movements and offers essential insights into traffic patterns and usage, facilitating improved resource management and access control.

In summary,, the system offers an additional benefit by protecting institutional property. By making sure that vehicle data is correctly recorded and aligned with a database of approved users in the excel sheet, the system removes the necessity for manual input, reduces mistakes, and improves overall operational efficiency. Its modular design and scalability allow it to be suited for different environments, ranging from small organizations to large university campuses.

REFERENCES

1. M. V. G. Aziz, H. Hindersah and A. S. Prihatmanto. (2021). Implementation of vehicle detection algorithm for self-driving car on toll road cipularang using Python language. 2017 4th International Conference on Electric Vehicular Technology (ICEVT). Sanur, 2021, pp. 149-153.doi:10.1109/ICEVT.2017.8323551.
2. Bush, I.J., Dimililer, K.(2022). Static and dynamic pedestrian detection algorithm for visual based driver assistive system. ITM Web Conf, 9(03002). doi: https://doi.org/10.1051/itmconf/20170903002.
3. Karunakaran, D. (2020, December 10). Vehicle Detection and Tracking: Udacity's Selfdriving Car Nanodegree, Medium.https://medium.com/intro-to- artificial-intelligence/vehicle-detection-and- trackingudacitys-self-driving-car- nanodegree- ca02330820ee.
4. Bradski, G. (2023). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
5. Hortovanyi, N. (2020, February 24). Vehicle Detection and Tracking. Towards Sata Science. https://towardsdatascience.com/vehicle-detection-
6. and-tracking-6665d6e1089b
7. Bush, I.J., Dimililer, K.(2023). Static and dynamic pedestrian detection algorithm for visual based driver assistive system. ITM Web Conf, 9(03002). doi: https://doi.org/10.1051/itmconf/2017090300.
8. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B. And Varoquaux, G. (2019). API Design For Machine Learning Software: Experiences From The Scikit-Learn Project, 108–122. https://arxiv.org/abs/1309.02388.
9. “OpenCV-Python Tutorials Documentation” Release 1,Alexander Mordvintsev & Abid K, Nov 05, 2019.
10. Huei-Yung Lin and Juang-Yu Wei, “A Street Scene Surveillance System for Moving Object Detection, Tracking and Classification”, Proceedings of the IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.
11. F. Bardet, et al., "Unifying real-time multi-vehicle tracking and categorization," in Intelligent Vehicles Symposium, 2009 IEEE, 2009, pp. 197-202.
12. Zhou, W., & Zhang, Z. (2022). Vehicle Detection and Tracking in Complex Traffic Environments Using Convolutional Neural Networks. IEEE Transactions on Intelligent Transportation Systems, 23(1), 88-101. doi: 10.1109/TITS.2021.3065520.

13. Huang, X., & Li, Y. (2023). Real-Time Vehicle Detection and Classification in Traffic Surveillance Using YOLOv5. International Journal of Intelligent Transportation Systems Research, 21(1), 45-56. doi: 10.1007/s13177-022-00465-6.
14. Li, J., & Wang, S. (2021). Multi-Stage Object Detection for Real-Time Vehicle Surveillance Systems. Sensors, 21(14), 4801. doi: 10.3390/s21144801.
15. Zhao, H., & Li, X. (2023). A Smart Traffic Monitoring System Based on Vehicle Detection and Counting for Urban Mobility. IEEE Access, 11, 22854-22864. doi: 10.1109/ACCESS.2023.3210624.
16. Gong, X., & Liu, X. (2022). Vehicle Detection and Classification Using Deep Learning Models for Automated Traffic Monitoring. Transportation Research Part C: Emerging Technologies, 129, 103271. doi: 10.1016/j.trc.2021.103271.