A project report on

A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR MALICIOUS VEHICLE DETECTION AT TOLLGATES

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

by

SRI CHARAN A (21BCE6072)

USHASREE A (21BPS1444)

K CHAITHANYA KRISHNA (21BCE5637)

VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November,2024

A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR MALICIOUS VEHICLE DETECTION AT TOLLGATES

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

by

SRI CHARAN A (21BCE6072)

USHASREE A (21BPS1444)

K CHAITHANYA KRISHNA (21BCE5637)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November, 2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled **“A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR MALICIOUS VEHICLE DETECTION AT TOLLGATES”** submitted by **SRI CHARAN VENKATAMAI SAI ADAPA,** for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Gayatri R.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai　　　　Signature of the Candidate
Date: 15/11/24

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"A Comparative Analysis of Machine Learning Algorithms for Malicious Vehicle Detection at Tollgates"** is prepared and submitted by **SRI CHARAN VENKATAMAI SAI ADAPA (21BCE6072),** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of Bachelor **of Technology in COMPUTER SCIENCE AND ENGINEERING** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr./Prof.

Date:

Signature of the Examiner	Signature of the Examiner
Name:	Name:
Date:	Date:

Approved by the Head of Department,
(Computer Science and Engineering-Core)

Name: **DR. NITHYANANDAM P**
Date:

ABSTRACT

With the rise in the number of vehicles, managing and monitoring traffic, as well as ensuring security at toll gates, has become increasingly important. This project aims to create an automated system that detects vehicle license plates, extracts vehicle numbers, and checks if the vehicle has been involved in any suspicious activities. Using computer vision to detect license plates and machine learning for classification, this system helps automate vehicle surveillance efficiently.

The process starts with detecting the license plate using a **Haar Cascade classifier**, which is a reliable technique for identifying specific objects like license plates in images. After detecting the plate, **Optical Character Recognition (OCR)** is used to extract the vehicle number from the plate. This number is then used as a unique identifier for each vehicle, allowing further checks to determine if the vehicle has any record of suspicious activity.

The classification of vehicles is handled by machine learning models, including **CatBoost**, **LightGBM**, and **XGBoost**. These models analyse the vehicle data and classify it as either "normal" or "potentially suspicious." After comparing these models, **LightGBM** was found to give the best results, accurately identifying vehicles with past records of suspicious activities. This makes it a strong choice for real-time vehicle classification.

To ensure accuracy, we evaluated our models using measures like **accuracy, precision, recall, F1score**, and **ROC-AUC**. Accuracy shows how often the model made correct predictions, while precision and recall check how well the model identified suspicious vehicles specifically. The F1score and ROC-AUC results confirmed that CatBoost is reliable for classifying vehicles effectively in real-time situations.

In summary, this project offers a practical solution for automated vehicle monitoring and detecting fraud. By combining license plate detection with machine learning, it speeds up vehicle checks and improves security at toll points. In the future, the system could be improved by adding more data and exploring deep learning models to make it even more accurate and adaptable across different settings.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Gayatri R, Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of artificial intelligence.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr.
Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant VicePresident, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor, Dr. T. Thyagarajan Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to **DR. NITHYANANDAM P** Head of the Department, B.Tech. Computer Science and Engineering and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis. My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: 15/11/24 **Name of the student**

SRI CHARAN VENKATAMAI SAI ADAPA

CONTENTS

CHAPTER 3

METHODOLOGY

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI: Artificial Intelligence

XGBoost: eXtreme Gradient Boosting Algorithm

LGBM: Light Gradient Boosting Machine Algorithm

ROC: Receiver Operating Characteristic

AUC: Area Under the Curve

F1-SCORE: Harmonic Mean of Precision and Recall

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The swift rise in the number of vehicles worldwide has put a great deal of strain on transportation infrastructures, especially at toll gates where cars need to move through quickly and effectively. Conventional manual techniques for tracking and confirming vehicles frequently result in mistakes, inefficiencies, and delays, particularly during periods of high traffic. Because of this, sophisticated, automated systems are required to improve security procedures at crucial checkpoints, decrease human intervention, and expedite vehicle detection. Automated vehicle identification systems can improve security by quickly identifying unauthorized or suspicious vehicles, in addition to increasing toll collection efficiency.

The increasing worries about vehicle-related threats, like illegal access, smuggling, or the use of stolen vehicles, highlight the necessity of automating toll gate systems. Due to human limitations in identifying patterns or rapidly recalling vehicle data, manual checks frequently overlook these threats. To improve the security and effectiveness of such systems, it can be revolutionary to use technologies like computer vision and machine learning for vehicle detection and classification. Security teams can redirect their resources from performing routine checks to responding to threats that have been flagged by automating the vehicle recognition process. In this regard, our project suggests an automated car detection system intended to improve toll gate security.

Using a Haar Cascade classifier, the system's initial step is to identify the vehicle's license plate number. For real-time applications like toll gate monitoring, the popular object detection algorithm Haar Cascade is successful and efficient. After the license plate has been identified, the vehicle's number is extracted from the picture using optical character recognition (OCR) technology. Machine learning algorithms are then used to examine this extracted number in order to ascertain whether the car is connected to any suspicious or malevolent activity.

In this study, the machine learning models XGBoost, LightGBM, and CatBoost were employed. Because of their efficacy in classification tasks and capacity to manage sizable datasets, these algorithms were selected. The purpose of testing each of these algorithms was to determine which one could classify vehicles as normal or suspicious with the highest accuracy. Accuracy, precision, recall, and F1-score—key performance metrics that gauge the model's capacity to generate accurate predictions while reducing false positives and negatives—were used to evaluate the algorithms.

In order to create an automated system that can drastically cut down on the need for manual intervention in toll gate operations, this project intends to show that it is feasible to use cutting-edge image processing and machine learning techniques. Our system provides a solution that increases operational efficiency and security by automating vehicle detection and classification. This ensures that toll gates can process vehicles more accurately and quickly. The study's findings will offer important new information about how AI might be used in real-time car security and monitoring systems, which will make toll operations safer and more effective in the future.

1.1 OVERVIEW OF VEHICLE DETECTION AND CLASSIFICATION

1. Vehicle Detection Process
    - Image Acquisition: The process begins with capturing images from surveillance cameras or other sources, where vehicles are present.
    - Preprocessing: The captured images are processed to enhance clarity. Techniques like noise reduction, contrast adjustment, and resizing are used to ensure the vehicle's features are clear and visible for detection.
    - Haar Cascade Classifier: To improve clarity, the captured images are processed. To make sure the vehicle's features are distinct and easy to detect, methods like noise reduction, contrast adjustment, and resizing are employed.
    - Advantages of Haar Cascade: Haar Cascade's advantages include its speed and efficiency, which make it perfect for real-time applications. It can identify automobiles in a range of settings, including toll booths, parking lots, and roadways.
    - Limitation**s**: Even with its high speed, Haar Cascade may not work as well in dimly lit areas, when cars are partially hidden, or in inclement weather like rain or fog. Additionally, it might have trouble identifying distant or smaller vehicles..

2. License Plate Detection and Extraction

- Post-Vehicle Detection: Once the vehicle is detected, the next step is to locate and extract the license plate from the vehicle.
- Optical Character Recognition (OCR): OCR technology is applied to read the characters on the license plate. This step converts the license plate's text into machine-readable data.
- Preprocessing for OCR: The image is further processed by converting it to grayscale, removing noise, and applying thresholding techniques to enhance the license plate's visibility.
- Challenges in OCR: The size, style, and format of license plates differ significantly between regions, which can complicate the OCR procedure. External elements such as dirt, damage, and plate reflections can also reduce the accuracy of OCR.

3. Vehicle Classification
    - Dataset Creation: For vehicle classification, the extracted license plate numbers are matched with a pre-existing dataset that labels vehicles as either "malicious" or "normal." This dataset may contain information like the vehicle’s history, owner details, or associated risks.
    - Classification Algorithms:
    - CatBoost: CatBoost is a gradient boosting algorithm particularly effective for datasets that include categorical features. It efficiently handles the classification task by leveraging the patterns in the categorical features.
    - LightGBM: LightGBM is another gradient boosting algorithm that is known for its high performance and speed, especially with large datasets. It uses a histogram-based method to speed up training time while maintaining high accuracy.
    - XGBoost: XGBoost is widely used in machine learning competitions for its robust performance in various classification tasks. It uses advanced tree-based algorithms and regularization techniques to improve model accuracy.
    - Model Training: The classification models are trained on a dataset that includes both normal and malicious vehicle data. Each model learns to differentiate between the two classes based on features like license plate patterns, time of entry, or vehicle type.

4. Evaluation of Model Performance - Evaluation Metrics:

   - Accuracy: The ratio of correctly predicted samples (both malicious

   and normal vehicles) to the total samples. This metric gives an overall idea of the model's performance.

   - Precision: Precision is calculated as the number of true positive classifications (correctly identified malicious vehicles) divided by the total predicted positives. It helps evaluate the model's ability to avoid false positives.
   - Recall: Recall measures the proportion of actual malicious vehicles correctly identified by the model. High recall means the model is good at identifying malicious vehicles but may suffer from false positives.
   - F1-score: The harmonic mean of precision and recall. It provides a balanced evaluation metric, especially in cases of imbalanced datasets where one class may dominate.
   - ROC Curve & AUC: The ROC curve plots the true positive rate against the false positive rate, and the AUC value indicates how well the model distinguishes between the two classes.

5. Challenges and Limitations

   - Data Variability: Vehicle and license plate designs vary widely across different regions, making it challenging to create a universal vehicle classification system.
   - Environmental Challenges: Environmental factors such as weather conditions (rain, fog), low lighting, or glare can reduce detection accuracy, especially in real-time applications.
   - False Positives/Negatives: The challenge lies in minimizing false positives (misclassifying a normal vehicle as malicious) and false negatives (missing out on a malicious vehicle).
   - Data Imbalance: In many cases, the dataset may be imbalanced, with far more normal vehicles than malicious ones, leading to biased predictions. Specialized techniques like oversampling, undersampling, or using weighted loss functions may be required.

6. Application and Real-World Use Cases

- Security and Surveillance: The vehicle detection and classification system can be integrated into surveillance systems to detect and classify vehicles entering sensitive areas such as parking lots, toll gates, or restricted zones.
- Tollgate Automation: By automating the process of identifying and classifying vehicles at toll gates, the system can help in faster toll collection and identify potential frauds (e.g., vehicles trying to bypass tolls).
- Traffic Monitoring: The system can be extended to monitor traffic patterns by tracking vehicle types and classifying them in real-time, helping to identify unusual traffic behaviors or illegal activities.
- Fraud Detection: The ability to classify vehicles as malicious allows the system to flag vehicles that might be involved in illegal activities such as theft, evading toll payments, or other criminal activities.

7. Future Improvements
    - Deep Learning Models: Implementing deep learning models like Convolutional Neural Networks (CNNs) for more robust and accurate vehicle detection, especially under varying conditions like partial occlusion, lighting changes, and complex backgrounds.
    - Integration with IoT: Integrating the vehicle detection system with IoT devices can improve real-time performance, enabling faster responses and more effective monitoring.
    - Advanced OCR Techniques: Improving OCR performance with deep learning-based OCR models to handle a wider variety of license plate designs and reduce errors due to image quality issues.

1.2 CHALLENGES IN VEHICLE DETECTION AND CLASSIFICATION

Numerous difficulties in vehicle detection and classification affect the precision and dependability of systems. Weather, shadows, and lighting are examples of environmental elements that can seriously impair performance. For example, it may be challenging to detect a vehicle when key features or license plates are obscured by low light, fog, or rain. Likewise, shadows and reflections may impede license plate reading or result in false positives, which would reduce the system's overall efficacy.

Vehicles come in a variety of shapes, sizes, and designs, which further complicates matters. Various vehicle types, including cars, trucks, buses, and motorcycles, necessitate a highly adaptable system. Vehicle detection and classification procedures can also be made more difficult by the possibility that they are partially obscured by other objects or observed from different perspectives. Character extraction may be hampered by license plate design variations, damage, dirt, and visibility angle, which further complicate license plate recognition (LPR). Unreadable plates that are dirty, broken, or angled awkwardly can drastically impair system performance.

Additionally, data quality and annotation issues arise when the training data consists of low-resolution images or imbalanced datasets. The quality of the images impacts the detection accuracy, and an imbalance in the dataset may lead to biased results, particularly in distinguishing between normal and malicious vehicles. Furthermore, real-time processing and scalability pose challenges. Vehicle detection systems often require substantial computational power to process large volumes of images or video feeds in real time. The system must be efficient enough to avoid delays in detection while maintaining high accuracy, especially when managing data from multiple sources.

False positives and false negatives are another significant worry. Unnecessary alerts and security risks can result from false positives, which occur when a malicious vehicle is mistakenly identified as harmless or a nonvehicle object is incorrectly classified as a vehicle. However, in security-sensitive applications, false negatives—where a potentially malicious vehicle is overlooked—are especially risky. Furthermore, since poor generalization can result from either overfitting or underfitting, selecting the appropriate classification algorithm is essential for optimum performance. Another difficult task is detecting malicious vehicles, which entails spotting odd trends or irregularities in the characteristics or behavior of the vehicle. In order to do this, the system must examine contextual data like behavior over time in addition to the vehicle.

A major obstacle is also integration with current infrastructure. Communication between different hardware and software platforms must be seamless in order to ensure compatibility with security cameras, toll booths, and traffic monitoring systems. And finally, there are serious ethical and privacy concerns about data collection and storage, especially when it comes to license plate recognition. Vehicle detection system design and implementation must take data privacy and bias into account because these issues can result in ethical dilemmas and diminished system fairness.

1.3 PROBLEM STATEMENT

The problem at hand involves the development of an effective vehicle detection and classification system that can accurately identify and classify vehicles based on images captured by surveillance cameras or other monitoring devices. The system aims to detect vehicles from different categories (e.g., cars, trucks, buses, motorcycles) and classify them as either legitimate or malicious, based on certain behavioral or feature-based attributes.

Due to environmental factors like changing lighting, weather, and occlusions, existing vehicle detection systems frequently encounter difficulties that can result in inaccurate detection. Furthermore, the task is made more difficult by the wide variation in vehicle appearances, including size, color, and shape. The system's dependability may be impacted by license plate recognition (LPR), another crucial component that can be hampered by dirty, misaligned, or damaged plates. Furthermore, the system's capacity to generalize across various vehicle types and scenarios may be impacted by biased or incorrect predictions resulting from poor data quality and imbalance in training datasets.

Another major issue is real-time processing, where the system needs to be able to process massive amounts of image or video data with high accuracy and low latency. The credibility of the system is seriously threatened by false negatives, which occur when malicious vehicles are overlooked, and false positives, which occur when non-vehicle objects are mistakenly classified as vehicles, particularly in applications that are sensitive to security. It's still difficult to choose the best classification algorithm that can achieve high efficiency and accuracy. The system's integration with the current infrastructure, such as security cameras, traffic monitoring systems, and toll booths, needs to be smooth and economical.

To ensure adherence to ethical norms and laws, privacy concerns pertaining to data collection, storage, and use—particularly with license plate recognition—need to be carefully considered. Designing and implementing a reliable vehicle detection and classification system that can precisely identify, categorize, and flag malicious vehicles in real-time is therefore the challenge, while also resolving the previously mentioned issues with data quality, environmental variability, real-time processing, false classifications, and privacy concerns.

1.4 OBJECTIVES

The main goal of this project is to create a reliable and accurate system for detecting and classifying vehicles from photos or video feeds, especially for security and surveillance applications. The ability to detect, identify, and categorize vehicles in real-time is essential for enhancing efficiency and safety in settings like parking lots, toll gates, and urban traffic management systems. In order to achieve these goals, the system will integrate cutting-edge methods from computer vision, machine learning, and artificial intelligence.

1. Vehicle Detection: The first objective of the project is to create a reliable method for detecting vehicles in a wide variety of image and video data. Given that vehicles appear in different angles, lighting conditions, and environmental factors, the detection method needs to be resilient. We will explore the use of classical computer vision algorithms, such as Haar Cascade Classifiers, as well as deep learning-based methods that utilize convolutional neural networks (CNNs) for more robust detection. The model must be able to identify vehicles with high precision, even under conditions where they may be partially obscured or in motion.

2. License Plate Recognition (LPR): The next step after detecting the vehicle is to extract its license plate number, which is necessary for additional classification and identification. Deep learning models and optical character recognition (OCR) methods will be used to implement license plate recognition. Since license plate numbers are frequently used for tracking systems, toll collection, and security checks, this step is crucial. Even when plates are partially obscured by dirt or other objects, tilted, or blurred, the LPR system must be able to recognize them accurately.

Vehicle Classification: The project will involve the classification of vehicles into different categories such as cars, trucks, buses, motorcycles, and other vehicle types. To achieve this, machine learning models, specifically decision-tree-based algorithms like CatBoost, LightGBM, and XGBoost, will be used.

3. These algorithms' capacity to manage sizable datasets and provide excellent performance makes them especially well-suited for classification tasks. To ensure the highest level of classification accuracy, the system will be trained on a dataset of labelled vehicle images that will allow it to differentiate

between different vehicle types. For this task, the model's accuracy, precision, recall, and F1-score will be assessed in order to identify the optimal algorithm.

4. Malicious Vehicle Identification: An important feature of the project is identifying vehicles that may be considered malicious or suspicious. This could involve vehicles with altered or fake license plates or vehicles that do not conform to expected types for specific locations, such as restricted zones. Machine learning models will be used to flag suspicious vehicles based on the classification results and their historical behaviour patterns. This feature will play a crucial role in enhancing security by identifying potential threats at an early stage.

5. Performance Evaluation: The project aims to assess the performance of the vehicle detection and classification system using a variety of performance metrics, such as accuracy, precision, recall, F1-score, and ROC AUC (Receiver Operating Characteristic Area Under Curve). These metrics will provide insights into how well the system is performing in terms of detecting vehicles, recognizing license plates, and classifying vehicles into the correct categories. By comparing different machine learning models, the project will identify which algorithms provide the highest performance and robustness under various conditions.

6. Real-time Processing: A critical aspect of the project is ensuring that the vehicle detection, license plate recognition, and classification processes can be completed in real-time, particularly in traffic management or security systems where immediate decisions need to be made. The system will be optimized for speed and efficiency, ensuring that it can handle live video feeds and deliver real-time results without significant lag or delay.

7. Scalability and Integration: The system needs to be able to integrate with current infrastructure, like security cameras or toll booths, and be scalable enough to manage big datasets. The solution will be built with the ability to scale up or down in response to the volume of vehicles and the area under observation. For increased functionality, it should also be simple to integrate with other systems, like databases used by law enforcement or traffic management systems.

8. Privacy and Security: The project will also address privacy concerns and make sure the system conforms with applicable data protection regulations because vehicle license plates are regarded as sensitive data. Data encryption and anonymization strategies, among other best practices for securely managing personal data, will be incorporated into the solution. Additionally, it will guarantee that only individuals with permission can access private data.

Through these objectives, the project aims to develop a comprehensive and efficient vehicle detection and classification system that meets the needs of real-world applications. The system will provide enhanced security, improve traffic management, and streamline toll collection processes, all while maintaining a high level of performance and compliance with privacy standards.

1.5 SCOPE

The scope of this project is to design, develop, and deploy a vehicle detection and classification system that can accurately identify vehicles, recognize their license plates, and classify them into different categories based on image or video data. The system aims to offer practical applications in various real-world scenarios, particularly in areas such as security, traffic management, toll collection, and surveillance. The following outlines the scope of the project:

1. Vehicle Detection: The project will focus on detecting vehicles from images or video feeds using advanced computer vision techniques. The scope includes exploring both traditional machine learning algorithms (e.g., Haar Cascade Classifiers) and deep learning models (e.g., Convolutional Neural Networks or CNNs) to detect vehicles in different environments and under various conditions such as different angles, lighting, and weather.
2. License Plate Recognition (LPR): A significant portion of the project is dedicated to implementing an accurate and efficient license plate recognition system. The scope includes detecting license plates from the identified vehicles and applying Optical Character Recognition (OCR) techniques to extract license plate numbers. The system will be designed to handle a variety of plate formats and be resilient to common challenges such as motion blur or partial obstructions.
3. Vehicle Classification: The project will classify detected vehicles into distinct categories such as cars, trucks, motorcycles, buses, and others

based on their visual characteristics. The classification process will leverage machine learning algorithms like CatBoost, XGBoost, and LightGBM, focusing on training the models with labelled datasets to achieve high classification accuracy.

4. Malicious Vehicle Detection: Another critical component of the scope is identifying potentially malicious or suspicious vehicles, such as those with altered or fake license plates or vehicles entering restricted areas. This will be accomplished by analysing the classification data and flagging vehicles that deviate from expected patterns. The system will aim to identify unusual behaviours that may indicate a security threat, enhancing safety in controlled environments.
5. Performance Evaluation: The system's performance will be evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC AUC. The scope includes testing the system under different conditions and comparing the effectiveness of various algorithms to determine which provides the best performance. This evaluation will help in fine-tuning the system and optimizing it for real-time applications.
6. Real-time Processing: The project will address the challenge of processing video feeds or images in real time. The system will be optimized for speed and low latency, enabling it to function effectively in live environments like toll booths, parking lots, or surveillance cameras, where immediate actions or decisions are required.
7. Scalability and Integration: The system will be designed to handle large datasets, ensuring that it can scale to monitor traffic in larger areas or handle numerous vehicles. Additionally, it will be built with the flexibility to integrate with existing infrastructure, such as toll systems, traffic management, or security systems, to ensure broad applicability.
8. Privacy and Security Compliance: Given that license plate data is sensitive, the project scope includes addressing privacy concerns and ensuring that all data is handled securely. This includes implementing data protection measures such as encryption, anonymization, and access controls to comply with privacy regulations and ensure secure operation.
9. User Interface and Alerts: The system will include a user interface (UI) that displays the results of vehicle detection and classification, including detected license plates and vehicle types. Additionally, alerts will be generated in cases where a malicious vehicle is detected. The scope also includes providing a dashboard for users to monitor and interact with the system in real-time.
10. Deployment and Practical Applications: The final phase of the project includes deploying the vehicle detection and classification system in a real-world environment, such as a toll booth, parking area, or security

checkpoint. The scope covers all aspects of deployment, including system testing, integration with other systems, and ensuring operational stability in various practical settings.

By focusing on these key areas, the project aims to develop a complete vehicle detection and classification system that is both practical and efficient, with applications that can enhance security, traffic flow, and operational efficiency in real-world scenarios.

CHAPTER 2
BACKGROUND

2.1 INTRODUCTION

Road infrastructure and traffic management systems are facing significant challenges due to the rapid growth of vehicle traffic worldwide. In addition to regulating traffic, tollgates are crucial components of road networks that serve as a significant revenue generator. Nevertheless, tollgate operations face an increase in fraudulent activities like vehicle duplication, toll evasion, and unauthorized access as the number of vehicles rises. Apart from resulting in revenue losses, these issues also compromise the operational efficacy of traffic management systems. Basic rule-based processes and manual inspections have been the cornerstones of conventional tollgate monitoring systems. Despite their previous success, these strategies are insufficient for the volume and complexity of traffic situations today. Additionally, especially during peak hours, these systems are vulnerable to human error and inefficiency. Thus, automated systems that can efficiently track and control vehicle movement and react instantly to fraudulent activity are desperately needed. Artificial intelligence (AI), machine learning, and computer vision developments have made it possible to develop innovative tollgate monitoring systems. These technologies offer high-accuracy pattern recognition, vehicle behavior classification, and large-scale dataset processing. With the help of these tools, tollgate systems can grow into automated, intelligent networks that enhance traffic flow

2.2 LITERATURE REVIEW

Recent advancements in vehicular network security and vehicle detection have leveraged machine learning and deep learning techniques to address challenges in real-time and adverse conditions. Studies employing ensemble models like Random Forest, Decision Tree, and XGBoost for CAN network traffic classification have achieved high detection accuracy (up to 98.4%) but face challenges in runtime and scalability. Simulation frameworks for misbehavior detection in Cooperative Intelligent Transport Systems (CITS) highlight real-time visualization and AI integration, achieving 70-95% accuracy, though limited by dataset diversity and privacy concerns. Enhanced YOLO models and SSD-based algorithms have improved vehicle detection accuracy (mAP up to 91.76%) by incorporating advanced techniques like CIoU Loss Function and adaptive anchor calculations, but occlusion handling and scalability remain concerns. License plate recognition using SRGANs and YOLO has achieved 98.5% accuracy, outperforming traditional methods while grappling with variations in plate sizes and lighting. Vehicle recognition systems using DenseNet and Haar Cascade achieve moderate accuracies (8288%), emphasizing the need for larger datasets. Innovative approaches like auditory alert systems with YOLOv4 and DAWN datasets address adverse weather and dummy plate detection but require further optimization for deployment. Collectively, these studies demonstrate robust performance but underline the need for real-world testing, diverse datasets, privacy-preserving solutions, and improved computational efficiency for practical applications in intelligent transport and autonomous systems.

2.3 TECHNOLOGICAL TRENDS IN TOLLGATE MONITORING SYSTEMS

The proliferation of tollgate monitoring systems has been driven by rapid technological advancements, particularly in automation, data analytics, and artificial intelligence. The majority of early tollgate systems were manual and dependent on human operators to collect tolls, verify vehicle information, and ensure compliance. These methods were susceptible to fraud, inefficiency, and human error, despite their effectiveness. In the late 20th century, electronic toll collecting (ETC) systems were introduced, marking the first notable technological breakthrough. Through the use of RFID tags and transponders, ETC systems enabled automatic fee collection, significantly reducing wait times and the need for human intervention.

Advanced machine learning (ML) and computer vision technologies have recently surfaced in tollgate monitoring. High-resolution cameras and real-time data processing systems have enabled automated vehicle recognition and classification. These days, optical character recognition (OCR) methods like Tesseract and detection algorithms like YOLO (You Only Look Once) are crucial parts of modern tollgate systems because they enable precise license plate recognition in a range of environmental conditions. Furthermore, cloud computing and Internet of Things (IoT) devices have made it possible to scale across regional and national toll networks through centralized data management and monitoring.

One significant trend is the integration of edge computing into tollgate systems. Because edge devices reduce latency and process data locally at the tollgate, they improve response times for real-time decision-making. These technologies, along with advancements in 5G connectivity, enable seamless communication between tollgate devices and central servers, further enhancing the efficacy and reliability of modern toll monitoring solutions.

2.4 APPLICATIONS OF MACHINE LEARNING IN TOLLGATE MONITORING

With the introduction of automation, scalability, and sophisticated predictive capabilities, machine learning (ML) has completely transformed tollgate monitoring systems. Conventional toll monitoring techniques frequently resulted in inefficiencies, human error, and missed detections because they mostly depended on static rules and manual interventions. On the other hand, ML-based solutions provide a more reliable and effective solution by utilizing historical data and continuously adapting to changing traffic patterns and operational conditions.

In tollgate monitoring, one of the primary applications of machine learning is anomaly detection. Anomalies such as vehicles speeding, avoiding authorized routes, or visiting tollgates too frequently can be detected by using machine learning models to analyze large data sets. These features allow authorities to spot suspicious activity and act quickly to stop it. Additionally, ML-powered solutions aid in revenue optimization by detecting instances of illegal access or toll evasion. These technologies are crucial to toll operators' attempts to increase revenue because they ensure precise toll collection and curtail illicit activity.

Another important benefit of machine learning in tollgate monitoring is the ability to make decisions in real time. Using real-time data analysis, machine learning algorithms can classify vehicle behaviors and take immediate action—such as sending out alerts or launching enforcement actions—in response to suspicious activity. Machine learning also helps with traffic management by providing forecasted insights that aid in regulating traffic flow and reducing congestion. With the use of these insights, toll operators may forecast periods of high traffic, optimize lane utilization, and put policies in place to improve traffic efficiency overall.

In conclusion, by offering creative methods to boost accuracy, efficiency, and adaptability, machine learning has completely transformed tollgate operations. The wide range of applications for machine learning (ML), from anomaly detection and revenue optimization to real-time decision-making and traffic management, shows how powerful this technology is for modernizing and streamlining tollgate monitoring systems.

2.5 COMPARISON OF MACHINE LEARNING ALGORITHMS

The machine learning techniques employed in this project were selected following a careful analysis of their benefits and drawbacks in order to achieve a balance between interpretability, computational effectiveness, and predictive accuracy. Because of its exceptional accuracy and computational efficiency, XGBoost became a strong contender for applications that need precise classification. However, meticulous hyperparameter optimization is necessary to achieve optimal performance. LightGBM's remarkable speed makes it perfect for high-throughput or real-time applications, particularly when used on large datasets. Its primary weakness is its sensitivity to outliers, which could compromise the reliability of the model if preprocessing is not done adequately. CatBoost eliminated the need for time-consuming preprocessing steps like encoding by performing remarkably well when directly processing categorical data. It performed especially well with datasets that had a large number of category variables because of this feature.

These variables were carefully taken into account when choosing the algorithm for this project, which prioritized a balance between speed, accuracy, and handling complicated data structures. This careful method made sure that models that are not only effective but also meet the unique needs of malicious vehicle detection and tollgate monitoring were deployed.

Algorithm	**Strengths**	**Limitations**
XGBoost	High accuracy, efficient computation	Requires hyperparameter tuning
LightBGM	Faster training on large datasets	Sensitive to outliers
CatBoost	Handles categorical data natively	Computationally intensive

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This project's methodology is centered on using cutting-edge computer vision and machine learning techniques to create and deploy a reliable tollgate monitoring system. This chapter describes a methodical technique to identify and categorize normal, suspicious, and malevolent behavior at tollgates in response to the growing need for precise vehicle tracking and traffic regulation. The suggested architecture ensures high accuracy, scalability, and adaptation to real-world circumstances by integrating a number of technological components.

Since tollgate cameras are the system's main source of input, the process starts with the collection of raw data and images from these cameras. To improve image quality, lower noise, and extract regions of interest, like license plates, these inputs go through preprocessing. The Haar Cascade Classifier, renowned for its lightweight and effective detection capabilities, is used to detect license plates using sophisticated computer vision algorithms. Optical Character Recognition (OCR), which uses Tesseract to precisely distinguish alphanumeric characters, is used to further extract car registration numbers.

The system creates a structured dataset by combining timestamps, entrances, and exits when data is acquired. To obtain important metrics, including travel time, speed, and anomalies, which are crucial inputs for classification, this dataset is subjected to feature engineering. The categorization task makes use of machine learning models, such as XGBoost, LightGBM, and CatBoost, which take advantage of their capacity to process structured data and identify intricate patterns. By classifying vehicle behavior into three categories—normal, suspicious, and malicious—these models help authorities make better decisions more quickly and increase operational effectiveness.

This approach places a strong emphasis on adaptability and modularity, guaranteeing that the system can manage a range of environmental circumstances and vehicle behaviors. To address issues with tollgate monitoring and fraud detection, a comprehensive solution is established through the integration of machine learning algorithms, real-time data processing, and predictive analytics. To ensure a visible and repeatable framework, the following sections give thorough explanations of each stage, including the preprocessing pipeline, feature engineering strategies, and the machine learning classification method.

3.1 IMAGE PROCESSING FOR NUMBER PLATE DETECTION

3.2.1 IMAGE ACQUISITION

The first step in the image acquisition process is to take pictures of every vehicle passing through tollgates using cameras that are positioned strategically. These cameras' main objective is to cover the cars in detail, with a focus on the license plates. It is recommended that high-resolution imaging technology be used to increase the system's reliability, especially in challenging circumstances such as glare, low lighting, or dirt blocking the license plates. High-resolution images improve the clarity and accuracy of number plate recognition while ensuring that the system performs well under a range of operational and environmental circumstances. This stage directly affects the quality and reliability of the extracted data, making it essential for the subsequent stages of data analysis and classification.

3.2.2 PREPROCESSING

Preprocessing is an essential step in image analysis that increases detection accuracy while reducing computational overhead. During the preprocessing phase, several important actions are performed to prepare the images for accurate and speedy number plate recognition. First, the RGB-formatted images are transformed to grayscale. This simplifies the detection process by reducing the image complexity while preserving essential features needed for accurate recognition. Then, distortions that could impede the detection process are removed using noise reduction techniques like Gaussian Blur to guarantee a cleaner input for additional analysis. Finally, the contrast is adjusted using histogram equalization. This technique enhances number plate visibility by changing the visual contrast, particularly in challenging lighting conditions. These preprocessing techniques work together to ensure the system's dependability and efficacy, paving the way for accurate number plate recognition under various environmental circumstances.

3.2.3 IDENTIFICATION OF PLATES

Vehicle number plate recognition is done using a machine learning-based object detection system that uses the Haar Cascade Classifier. This method uses a cascade structure, in which the image is processed successively by several layers of classifiers, each of which has been trained to identify particular characteristics of license plates. By steadily enhancing the search at every stage, the cascade structure ensures high detection accuracy with minimal processing overhead.

A custom dataset of tagged car photos with bounding boxes accurately indicating the location of license plates is used to train the classifier. The algorithm can learn the unique visual patterns and features of license plates under various settings thanks to this training data. The Haar Cascade Classifier uses a sliding window methodology in the detection phase, methodically going through the image to find areas that correspond to the patterns of license plates it has learned. This technique guarantees a thorough analysis of the picture, making it possible to reliably detect license plates even in difficult situations.

3.2.4 EXTRACTION OF PLATES

The region of interest (ROI) is isolated by cropping the portion of the image that has the license plate after it has been detected. This ensures that in subsequent phases, only relevant data is handled. The clipped image is then enhanced to make the characters more readable and clear. This stage is crucial for increasing the precision of character recognition in scenarios where the plate may be partially obscured or affected by external circumstances. The region of interest (ROI) is isolated by cropping the portion of the image that has the license plate after it has been detected. This ensures that in subsequent phases, only relevant data is handled. The clipped image is then enhanced to make the characters more readable and clear. This stage is crucial for increasing the precision of character recognition in scenarios where the plate may be partially obscured or affected by external circumstances.

3.2 OPTICAL CHARACTER RECOGNITION (OCR)

3.3.1 CHARACTER RECOGNITION

After the license plate region has been successfully extracted from the image using computer vision techniques, the first step in the OCR process is character segmentation. Open-source text recognition software Tesseract OCR is utilized for this task due to its demonstrated efficacy and accuracy in recognizing alphanumeric letters. Character segmentation is the process of breaking down a license plate into discrete characters by looking at the arrangement of pixels. The recognition engine of Tesseract is then used to classify each character based on predefined characteristics and patterns using a neural network.

A string that functions as the vehicle's registration number is created by reassembling the separated characters. This string, which serves as a unique identification, enables the system to link the vehicle to the pertinent database records. The OCR process must overcome a number of challenges, including variations in license plate typefaces, sizes, and designs, as well as distortions caused by occlusions, motion blur, or low lighting. Before the OCR stage, preprocessing techniques like noise reduction, contrast enhancement, and binarization are employed to get around these issues and ensure the highest recognition accuracy.

3.3.2 VALIDATION

The extracted registration numbers must be correct for the system to be dependable. To ensure this, the extracted data is compared to pre-made forms that are specific to the license plate standards of the area. These formats control the quantity of characters, the permitted alphabets and numbers, and the arrangement of components such as serial numbers, state codes, and district identities. For instance, if the intended appearance of a license plate is "XX00-XXXX," any deviation from this pattern is recorded. Validation has two main functions. First of all, it eliminates errors caused by OCR misinterpretation, such as characters that are incorrectly identified due to distortion or noise. Second, it ensures that only legitimate and appropriately formatted data is entered into the system, protecting against potential inconsistencies or fraudulent submissions. By marking entries that are unclear or incorrect for manual review, human operators can cross-check the data and ensure data integrity. This technology, which combines automated OCR with systematic validation to ensure a high degree of accuracy in capturing and processing vehicle registration data, improves the tollgate monitoring system's dependability and efficiency.

3.3 DATASET CONSTRUCTION AND FEATURE ENGINEERING

3.4.1 DATASET DEVELOPMENT

A successful tollgate monitoring system is built on the foundation of creating a comprehensive and well-structured dataset that documents all significant aspects of vehicle movements. To ensure that each entry relates to a unique vehicle passage, the dataset used in this project was assembled by merging data collected from multiple tollgates. The primary characteristic of the dataset is the Vehicle Number, a unique identification provided by optical character recognition (OCR) of license plates. A vehicle's identity allows the system to track its movements across the tollgate network. By uniquely identifying the tollgate where the vehicle was captured, the Tollgate ID is an additional essential feature that aids in contextualizing the journey of the vehicle. The exact times at which a vehicle enters and exits a tollgate are indicated by the addition of Precise Entry Time and Exit Time timestamps to each pass. With the use of these timestamps, it is possible to determine the Time Between Tollgates, or the interval between a car leaving one tollgate and entering the next. In addition, an Expected Travel Time is calculated by utilizing historical traffic patterns or the average travel duration for the route. To spot irregularities in travel trends, this number serves as a benchmark.

The dataset also includes a Status label that categorizes vehicle activity as normal, suspicious, or harmful. Machine learning models must be built and evaluated using this classification in order to identify and predict aberrant activity. These features provide the dataset with a comprehensive understanding of vehicle behavior and set the stage for anomaly detection and more in-depth analysis.

3.4.2 FEATURE ENGINEERING

The process of feature engineering is essential for transforming raw data into insights that enhance the effectiveness of machine learning models. Several important features are extracted from the data in this study to help detect subtle trends and anomalies in vehicle behavior. Travel Time Deviation is one of the most important factors; it determines the difference between the expected and actual travel times for a vehicle between two tollgates. Large variations could indicate suspicious activity, such as deliberate delays to fool toll systems or speeding to evade detection. Speed calculation is another crucial factor. It is calculated by dividing the distance between tollgates by the travel time. Cars that exceed the speed limit can be identified with the help of this feature, which could indicate reckless driving or malicious intent. The system can use speed data to determine which cars are high-risk for further inspection.

Furthermore, the system monitors traffic irregularities, such as sudden stops, prolonged idle periods, or rapid accelerations between tollgates. These anomalies usually point to potential issues with the car, unauthorized stops, or tried toll-evasion. Finding such behaviors exposes the system to an additional level of scrutiny, ensuring careful observation.

Finally, Visit Frequency is tracked to identify vehicles that pass through tollgates abnormally often. Unusual high frequencies could indicate illegal activity or toll avoidance schemes, which would require further investigation. This tool facilitates the identification of recurring patterns that might otherwise be missed. These designed components work together to significantly improve the system's ability to accurately classify vehicle behavior. The system incorporates features like speed, traffic anomalies, visit frequency, and journey time variance to produce a more intricate picture of each vehicle's behavior. Feature engineering not only increases the predictive power of machine learning models but also ensures that the system can adapt to a range of scenarios and traffic patterns.

3.4 CLASSIFICATION OF MALICIOUS BEHAVIOUR

3.5.1 TRAINING AND EVALUATION

To guarantee appropriate vehicle behavior classification, the project's training and evaluation phase entails data preparation and model performance assessment. Training (80%) and testing (20%) portions of the dataset are separated. K-fold cross-validation is used to verify robustness and avoid overfitting when models like CatBoost, LightGBM, and XGBoost are trained using the training data. Metrics such as F1-score, AUC-ROC, recall, accuracy, and precision are used to assess performance. Comprehensive evaluation results revealed that CatBoost was the most successful model at identifying harmful vehicle activity, exhibiting the highest accuracy and recall among the models.

3.5.2 ARCHITECTURES

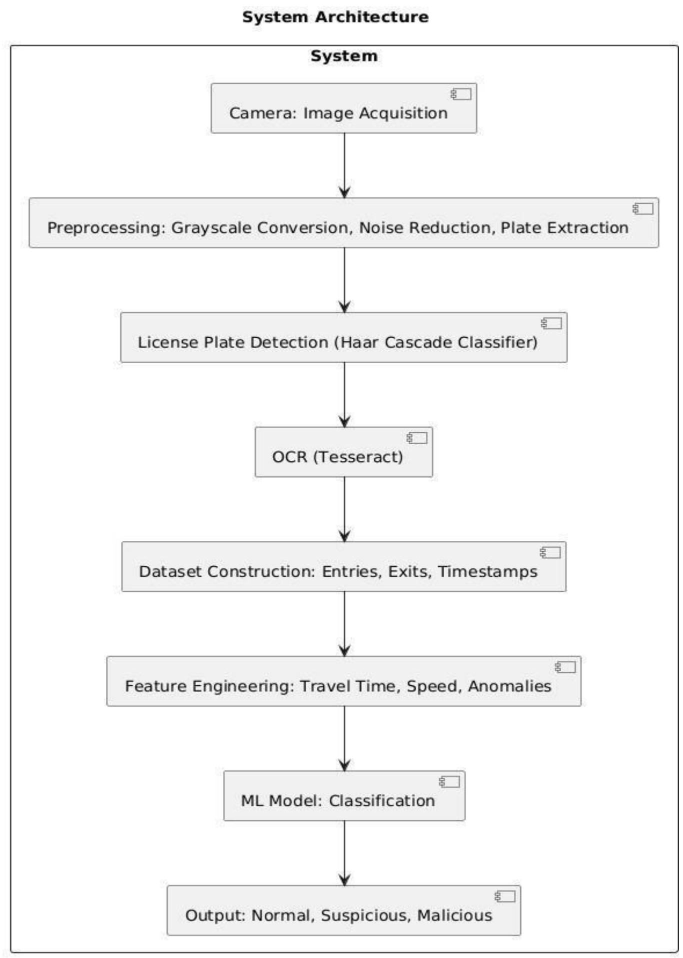

System Architecture
System
Camera: Image Acquisition
Preprocessing: Grayscale Conversion, Noise Reduction, Plate Extraction
License Plate Detection (Haar Cascade Classifier)
OCR (Tesseract)
Dataset Construction: Entries, Exits, Timestamps
Feature Engineering: Travel Time, Speed, Anomalies
ML Model: Classification
Output: Normal, Suspicious, Malicious

Figure 1

The system's high-level operation, from image acquisition to hostile vehicle behavior classification, is depicted in Figure 1. Images taken by different cameras are first used as input by the system. Preprocessing includes actions like grayscale conversion, noise reduction, and license plate extraction. The Haar Cascade Classifier is used to detect license plates, while Tesseract OCR is used to extract registration number text. After then, timestamps and information on vehicle entry and exit are gathered to create a dataset. In order to classify vehicle activity into benign, questionable, or malicious categories, the machine learning model uses feature engineering to compute metrics such as journey time, speed, and anomalies.

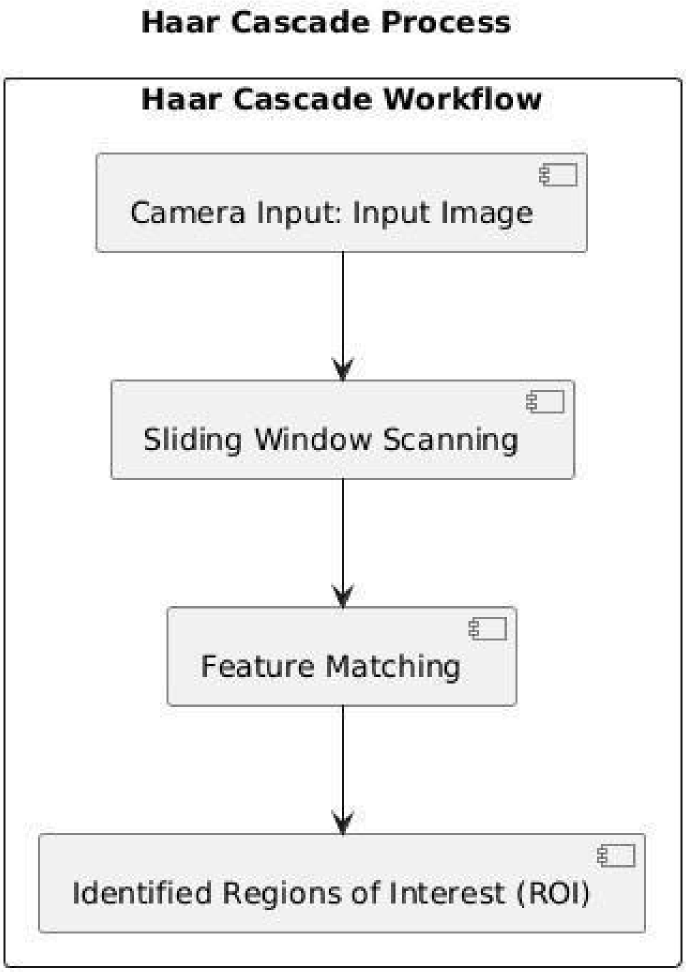


Figure 2

The Haar Cascade approach uses a sliding window methodology to scan images and find zones of interest, such license plates, as seen in Figure 2. An input image, usually taken with a camera, starts the process. The image is methodically scanned by a sliding window, which looks for features of interest in smaller areas. To find possible zones of interest, each region is compared to the algorithm's trained data. Regions of the image that meet the predetermined criteria are highlighted in the output and subsequently identified as regions of interest (ROI).

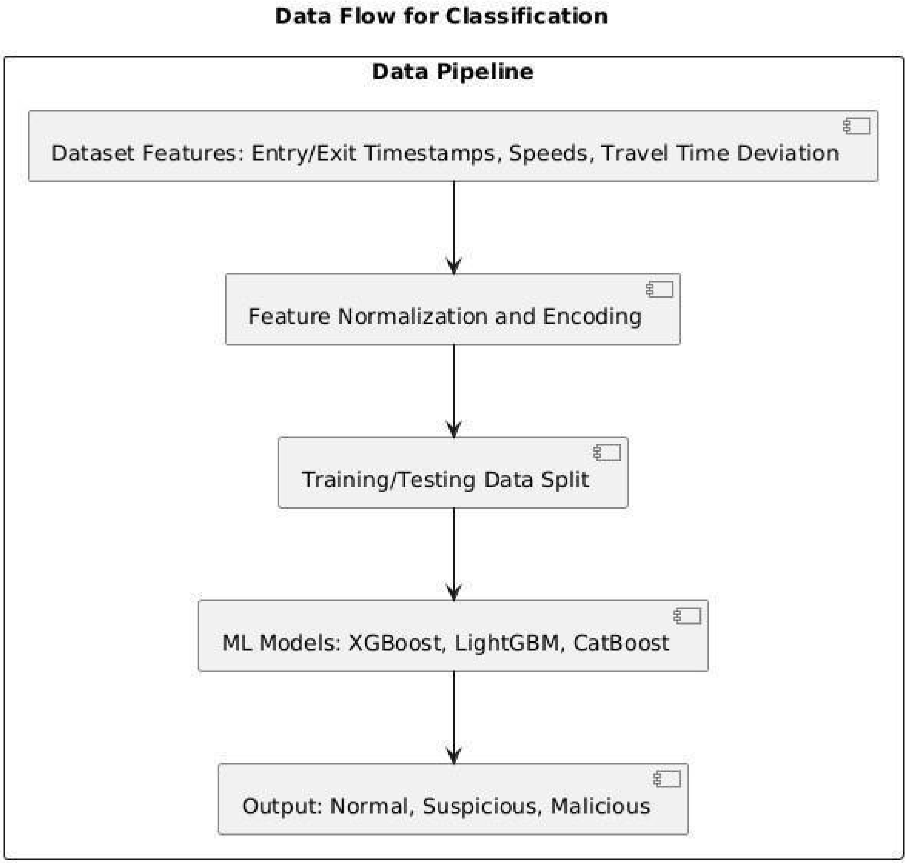

Data Flow for Classification
Data Pipeline
Dataset Features: Entry/Exit Timestamps, Speeds, Travel Time Deviation
Feature Normalization and Encoding
Training/Testing Data Split
ML Models: XGBoost, LightGBM, CatBoost
Output: Normal, Suspicious, Malicious

Figure 3

The data flow through the machine learning classification pipeline is depicted in Figure 3. Vehicle speeds, changes in travel duration, and entry/exit timestamps are among the dataset's features. Prior to the data being divided into training and testing sets, these features are encoded and normalized. This data is then used to train machine learning models like LightGBM, CatBoost, and XGBoost. The categorization pipeline's output designates whether a vehicle's behavior is malicious, suspicious, or typical.

3.5.3 ALGORITHMS UTILISED

Three sophisticated machine learning algorithms—XGBoost, LightGBM, and CatBoost— were used to categorize vehicle behavior into three groups: normal, suspicious, and malicious. Because of their unique benefits in managing categorical data, processing big datasets, and attaining great computing efficiency, each of these algorithms was selected. Extreme Gradient Boosting, or XGBoost, is a fast and effective ensemble learning technique. It is very successful for large and complicated datasets since it iteratively constructs decision trees to reduce classification mistakes. Preprocessing is made easier by XGBoost's direct handling of missing data, and its efficient and reliable performance is guaranteed by its optimal use of memory and CPU resources.

Another gradient boosting system made for processing big datasets quickly is called Light Gradient Boosting Machine (LightGBM). Its learning strategy, which is based on histograms, greatly speeds up processing. LightGBM offers adaptability across a range of dataset types and excels at managing both numerical and categorical variables. Because of its effectiveness in handling massive amounts of data, it is especially well-suited for realtime applications where accuracy and speed are crucial, such classifying vehicle behavior. CatBoost, on the other hand, distinguishes itself through its unique categorical feature optimization. Unlike previous algorithms, it analyzes categorical data natively, avoiding the requirement for explicit preprocessing. In order to minimize overfitting and provide precise predictions while preserving model generalization, CatBoost also uses ordered boosting.Thanks to its ability to handle complicated relationships and a variety of data types, CatBoost performed better than the other algorithms in this study, exhibiting remarkable accuracy and recall.

Thorough preparation of the data was essential to the classification models' effectiveness. To guarantee data completeness, missing values in the dataset were handled by interpolation or imputation techniques. In order to optimize the training process, numerical characteristics, including journey times and vehicle speeds, were standardized to standardize their scales. One-hot or label encoding approaches were used to encode categorical data, including tollgate identification, so that the machine learning algorithms could use them. A crucial step in improving the models' performance was feature selection. Principal Component Analysis (PCA) and correlation analysis were used to eliminate redundant or inconsequential variables and keep only the most pertinent attributes.This concentrated the models' learning on important variables that are essential for categorizing vehicle behavior, like speed deviations and journey abnormalities.
The dataset was split into two sections for training and evaluation: 20% was used for testing and 80% was used for training. This preserved a portion of the data for objective performance evaluation while guaranteeing that the models were

trained on the rest of the data. K-fold cross-validation was used to confirm the models' resilience and avoid overfitting. Using this approach, the training data was divided into many subsets, the model was trained on one subset, and it was then tested on another. A thorough assessment of the models' capabilities was obtained by repeating this procedure for every subset. Accuracy, precision, recall, F1-score, and AUC-ROC were among the metrics used to assess performance. While precision concentrated on the percentage of genuine positives among anticipated positives, accuracy offered an overall indicator of correctness.The F1-score struck a balance between precision and recall, whereas recall evaluated the model's capacity to detect all true positive cases. The models' capacity to differentiate between classes at various classification thresholds was assessed using AUC-ROC.

With the highest accuracy and recall of the three algorithms, CatBoost was the clear winner. It was the best option for our project because of its ability to manage categorical data efficiently and its resistance to overfitting. Although LightGBM and XGBoost also demonstrated strong performance, their outcomes were marginally less accurate, especially when it came to recall and overall accuracy. The chosen model's ability to accurately categorize vehicle behavior was guaranteed by this rigorous training and assessment procedure, offering a reliable and effective means of identifying malicious conduct in practical applications.

3.5.4 MODEL COMPARISON AND OPTIMIZATION

It was crucial to compare and optimize models in order to identify the optimal algorithm for classifying vehicle behavior. Because it achieved the highest accuracy and recall among the three machine learning models evaluated, LGBoost outperformed the others. It was recognized for its exceptional performance due to its advanced handling of categorical data and use of ordered boosting, which minimizes overfitting and ensures strong generalization. For this project, LGBoost proved to be the most reliable choice due to its exceptional capacity to identify potentially harmful activity..

Although their accuracy and recall were marginally lower than CatBoost's, XGBoost and LightGBM also produced impressive results. Both approaches are well known for processing huge datasets efficiently and for capturing complicated correlations in the data through their gradient boosting frameworks. They performed admirably, suggesting that
they are suitable for comparable classification tasks, even though their precision and recall scores lagged somewhat behind CatBoost.

All three models' performance was further enhanced by hyperparameter tuning using grid search and random search techniques. The models' prediction power was increased by adjusting parameters such as learning rates, tree depths, and regularization values. This careful optimization allowed for the adjustment of each model for the best possible accuracy and efficiency in recognizing hostile vehicle behavior.

3.6 PSEUDOCODES

3.6.1 VEHICLE NUMBER PLATE DETECTION

```
Input: Image captured at tollgate
Output: Extracted license plate as text
1. Convert the input image to grayscale
2. Apply Gaussian Blur to reduce noise
3. Load Haar Cascade Classifier trained for number plate detection
4. Use the classifier to detect regions of interest (ROI) in the image
5. For each ROI:
   a. Crop the ROI from the image
   b. Apply Optical Character Recognition (OCR) using Tesseract
6. Validate the OCR output:
   a. Check for regional license plate format
   b. If invalid, flag for manual review
7. Return the validated license plate text
```

3.6.2 FEATURE ENGINEERING

```
Input: Vehicle entry and exit data from tollgate cameras
Output: Feature set for ML classification
1. Initialize feature set F
2. For each vehicle:
   a. Calculate travel time between tollgates:
        Travel Time = Exit Time - Entry Time
   b. Calculate expected travel time based on historical data
   c. Compute travel time deviation:
        Deviation = |Actual Travel Time - Expected Travel Time|
   d. Calculate speed if distance between tollgates is known:
        Speed = Distance / Travel Time
   e. Flag anomalies in speed or travel time:
        If Speed > Threshold or Deviation > Threshold:
            Flag = "Suspicious"
   f. Add features [Travel Time, Speed, Deviation, Flag] to F
```

```
3. Return F
3.6.3 ML BASED CLASSIFICATION
Input: Feature set F
Output: Vehicle behavior classification (Normal, Suspicious, Malicious)
1. Import libraries for XGBoost, LightGBM, and
CatBoost 2. Split feature set F into training (80%) and
testing (20%) datasets
3. Train models:
a. Initialize models for XGBoost, LightGBM,
   CatBoost
b. For each model:      Train on training dataset
      Perform hyperparameter tuning (grid search/random search)
4. Evaluate models:
a. Calculate metrics: Accuracy, Precision, Recall, F1-score, AUC-ROC
b. Compare metrics to select the best-performing model
5. Predict vehicle behavior using the selected model
6. Return classification results
```

3.7 EVALUATOION METRICS

For machine learning models to function well in practical applications, like hostile vehicle detection at tollgates, evaluation metrics are crucial. The aforementioned metrics provide lucid insights into the model's strengths and weaknesses, highlighting its ability to minimize errors and accurately identify cars that exhibit suspicious or malicious behavior. The task involves separating normal, suspicious, and malicious vehicle behavior, so the choice and interpretation of evaluation metrics are essential to determining the model's practical usefulness. AUC-ROC, recall, accuracy, precision, and F1-score are some of the metrics employed in this investigation. Different perspectives on model performance are offered by each metric, which addresses specific problems in this area.

$$= \frac{(\qquad + \qquad)}{\qquad}$$

Accuracy is defined as the proportion of correctly classified events (including malicious and nonmalicious vehicles) among all data in the collection. It provides a thorough evaluation of the overall efficacy of the model. In all three classes—malicious, suspicious, and normal—the model performs well when its accuracy is high. Despite being an important statistic, accuracy has limitations when datasets are not balanced. In a dataset with 90% normal vehicles, for example, the model can achieve 90% accuracy by classifying all vehicles as normal, even if it cannot identify any hostile vehicles. As a result, high accuracy does not always indicate that the model is adept at spotting bad behavior.

$$= \frac{}{\qquad\qquad\qquad +}$$

The precision parameter quantifies the proportion of malicious vehicles correctly identified by the model among all vehicles classified as bad. It shows the model's ability to minimize false positives, which occur when safe cars are mistakenly labeled as dangerous. High precision reduces unnecessary disruptions and operational inefficiencies, demonstrating that the model consistently signals only real risks. Inaccurately classifying a harmless car as dangerous, for instance, could lead to pointless investigations, resource waste, and even damage tollgate user confidence. Conversely, low precision means the model generates a lot of false positives, which can drive up costs and interfere with tollgate operations.

$$= \frac{}{\qquad\qquad\qquad +}$$

The recall or sensitivity of the model is the proportion of actual malicious vehicles that it correctly identifies. This statistic aims to lower false negatives, which happen when the model is unable to identify potentially hazardous vehicles. High recall is important when there are significant consequences for not identifying malicious cars, such as lost revenue from toll evasion or security risks. A high recall model ensures that most, if not all, of the dangerous vehicles are detected, even if it occasionally incorrectly labels benign vehicles as malicious. In situations such as tollgate monitoring, where the cost of overlooking a potentially hazardous vehicle outweighs the inconvenience of investigating false positives, this compromise is often justified. Failure to identify a vehicle that engages in illegal activity or avoids paying tolls, for example, could lead to financial losses or security threats.

$$\boldsymbol{F}\,core = \cdot \frac{e\,con \cdot eca}{\qquad\qquad +}$$

By taking the harmonic mean of precision and recall, the F1-score offers a fair evaluation of a model's ability to lower false positives and false negatives. It is especially helpful when the dataset is unbalanced or when precision and recall must be traded off. The high F1score indicates that the model achieves a good balance between accurately identifying dangerous vehicles and minimizing the misclassification of typical vehicles. This project depends on the F1-score because it ensures that the system can detect dangerous activity and is operationally efficient. The F1-score strikes a balance between recall and precision, which helps to ensure consistent performance across all three vehicle behavior classes. This metric is especially helpful in real-world scenarios where using a model with high precision but low recall—or vice versa—would not be practical.

The model's overall performance is evaluated using the AUC-ROC (Area Under the Receiver Operating Characteristic Curve) across a range of classification thresholds. When the true positive rate (recall) is plotted against the false positive rate, the ROC curve illustrates the trade-offs between sensitivity and specificity at different thresholds. The model is better at distinguishing between cars that are harmful and those that are not if it has a higher AUC score, which goes from 0 to 5 (random guessing) to 1 (perfect performance). System optimization based on operational requirements can greatly benefit from this statistic, which provides a comprehensive evaluation of the model's recall and precision balance. For example, in order to identify the majority of malicious vehicles, the system might prioritize recall during peak hours while striving for a better balance between precision and recall during off-peak hours.

The evaluation metrics for this project, which include accuracy, precision, recall, F1-score, and AUC-ROC, provide a comprehensive understanding of the model's performance. From overall accuracy to the ability to identify malicious vehicles and minimize misclassification errors, every statistic focuses on a different aspect of classification. When combined, these indicators ensure that the selected strategy is successful and feasible for real-world applications where operational efficiency is critical and there is a considerable chance of missing harmful behaviors. The project integrates these assessment metrics to offer a robust and reliable solution for hostile vehicle detection at tollgates.

CHAPTER 4

IMPLEMENTATION

4.1 LIBRARIES AND PACKAGES

For the implementation of the malicious vehicle detection system, Python and its libraries were utilized to streamline the entire workflow, from vehicle detection to classification. The following libraries and tools played a crucial role in the successful execution of this project.

1. OpenCV(cv2)
   Using the Haar Cascade Classifier, OpenCV is a popular computer vision library used for vehicle detection. It made it easier to process image data, including handling video or image streams, resizing photos, and cropping license plates from photos. Additionally, OpenCV made visualization possible by allowing bounding boxes to be drawn around objects it detected.
2. Tesseract OCR (pytesseract)
   Text was extracted from the identified license plates using Tesseract OCR's Python binding, pytesseract. Vehicle numbers could now be accurately recognized thanks to this step, and they were subsequently saved in the dataset for classification. Tesseract was perfect for reading car numbers from photos because of its effective optical character recognition.
3. Pandas
   Pandas is a library for data analysis and manipulation. The extracted vehicle numbers, timestamps, and other information were saved in a structured CSV file format using it. The main dataset used in the classification procedure was this file. Pandas also made it easier to manage and preprocess data effectively.
4. NumPy
   For numerical calculations and array manipulations, NumPy was utilized. It assisted with preprocessing tasks that were necessary for the machine learning models, like managing numerical attributes in the dataset.

5. CatBoost
   CatBoost is a gradient boosting algorithm tailored for categorical data. It was one of the models used to classify vehicles as Normal, Suspicious, or Malicious. Its high efficiency in handling categorical features made it a reliable choice for this project.
6. LightGBM(LGBoost)
   LightGBM, another gradient boosting algorithm, was chosen for its speed and memory efficiency. It processed the dataset quickly and contributed to the comparison of classification models in the project.
7. XGBoost
   XGBoost, renowned for its high accuracy and flexible hyperparameter tuning, achieved the highest classification accuracy (93%) among the compared models. It proved to be the most effective algorithm for detecting malicious vehicles in this project.
8. Scikit-learn
9. A variety of machine learning-related tasks were completed using Scikit-learn. Accuracy, precision, recall, and F1-score were among the metrics it offered for assessing model performance, preprocessing data, and dividing the dataset into training and testing sets.
10. Matplotlib
    Matplotlib was employed to visualize the results of the classification models. It allowed for the creation of bar graphs, comparison charts, and other visualizations to better understand the performance of the algorithms.
11. Seaborn
    Seaborn, built on top of Matplotlib, was used for advanced visualizations, such as creating heatmaps to represent the confusion matrices of classification models. It provided a more aesthetically appealing representation of the results.
12. Datetime
    The built-in Python datetime module was used to handle timestamps in the dataset. This attribute was crucial for determining the actual travel time of vehicles between toll gates.
13. OS
    The os module, a built-in Python library, was used to manage file system operations, such as reading and saving images or CSV files generated during the project workflow.

4.2 MODULES

The project has a handful of modules to ensure each requirement of the project is met.

4.2.1 VEHICLE DETECTION MODULE

Using the Haar Cascade Classifier and Optical Character Recognition (OCR) methods, this module focuses on identifying and extracting license plates from photos. Following the detection of vehicles within each frame taken from security photos or video streams, the model finds and reads the license plate numbers. These extracted license plate numbers are saved in a CSV file for later analysis, together with related information like timestamps and tollgate type. In the next classification stage, the data is used to identify the vehicle's travel behavior and classify it as Normal, Suspicious, or Malicious.

4.2.2 CROSS-CHECKPOINT ANALYSIS

In order to track a vehicle's movements, this module compares and validates data from various tollgate checkpoints. To make sure the classification is correct, it compares the vehicle's actual time and anticipated time of travel. A vehicle is flagged as suspicious or malicious by the system when it fails to pass through a tollgate or exhibits irregular time discrepancies. This assists in spotting possible dangers and guarantees that every car is present at every checkpoint. The final vehicle status is determined by processing the data through the classification algorithms (CatBoost, LGBoost, and XGBoost) after it has been cleaned and refined. This includes the vehicle number, timestamp, and travel times. Vehicle compliance is ensured by this analysis, which also offers insights into possible incidents.

```
BEGIN
   LOAD 'entry.xlsx' INTO df1
   LOAD 'exit.xlsx' INTO df2

   EXTRACT license_numbers FROM df1 AS entered_cars
EXTRACT license_numbers FROM df2 AS exited_cars

   CARS_ENTERED_NOT_EXITED = entered_cars NOT IN exited_cars
   CARS_NOT_ENTERED_BUT_EXITED = exited_cars NOT IN
entered_cars
   CARS_ENTERED_AND_EXITED = entered_cars IN exited_cars
SAVE results TO 'results.xlsx'
END
```

The core of the algorithm involves comparing two sets of license numbers: one for vehicles that have entered the checkpoint and another for vehicles that have exited. By performing set operations, we were able to efficiently identify vehicles that entered but did not exit, exited without entering, or entered and exited as expected.

4.3 SYSTEM WORKFLOW

The diagram below illustrates the workflow of the proposed system, starting from the input video and progressing through each processing stage. In the final step, the processed data is made available to the relevant authorities, enabling them to monitor vehicle movements as needed.

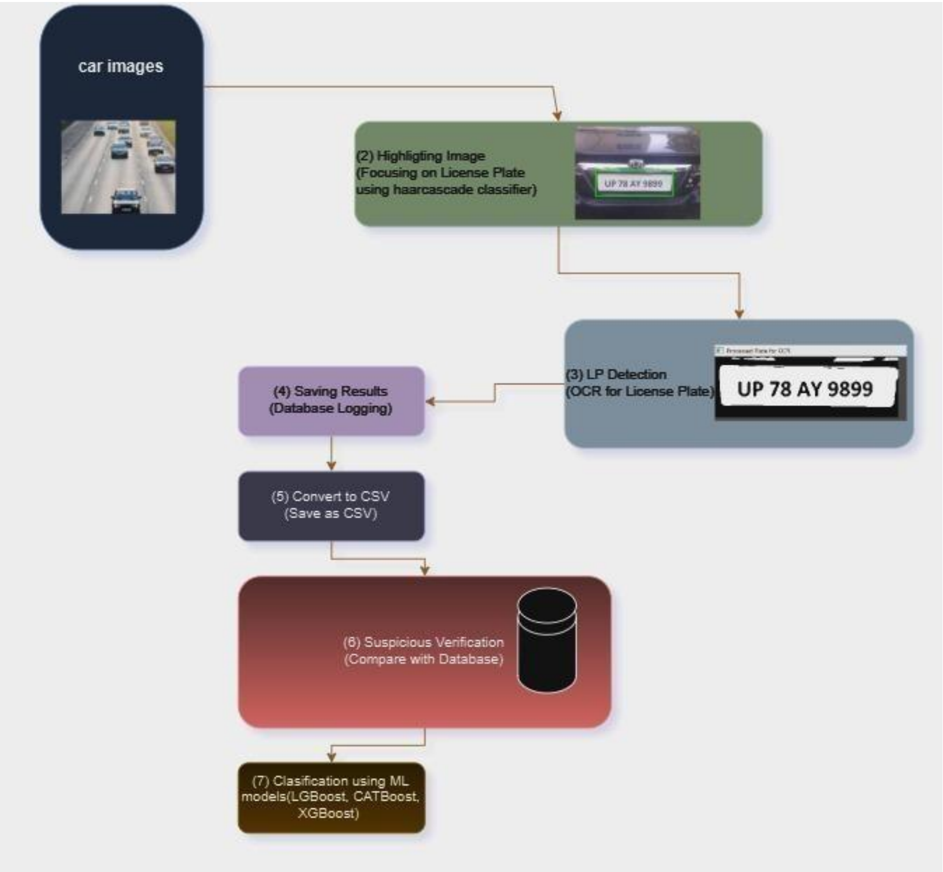


Fig 4. Proposed system workflow

The diagram illustrates the step-by-step process for detecting and verifying license plates in a vehicle surveillance system.

4.3.1 VEHICLE DETECTION (Haar Cascade Classifier and OCR)

The system begins by processing input images or video frames. Using the Haar Cascade Classifier, it detects vehicles in each frame.

4.3.2 LICENSE PLATE EXTRACTION

Once a vehicle is detected, the system identifies the region of interest (ROI) containing the license plate. This area is cropped from the image to isolate the plate for further processing.

4.3.3 LICENSE PLATE RECOGNITION (OCR)

The cropped license plate image is processed using Optical Character Recognition (OCR) to extract and read the alphanumeric characters on the plate.

4.3.4 SAVING RESULTS

The detected license plate number, along with the corresponding timestamp, tollgate type, and vehicle ID, is stored for further analysis and tracking.

4.3.5 CONVERTING TO CSV

The extracted data—license plate number, timestamp, tollgate type, actual travel time, and vehicle ID—are saved into a structured CSV file. This ensures that the data is well-organized and ready for further classification and analysis.

4.3.6 DATA PREPROCESSING

Before feeding the data into the machine learning models, the system performs preprocessing steps. This includes cleaning and refining the dataset by handling missing values, normalizing the travel times, and encoding the relevant features such as tollgate types and vehicle IDs.

4.3.7 FEATURE ENGINEERING

Key characteristics are extracted and converted into a format that is appropriate for the machine learning models, including the actual time taken, the anticipated time of travel, the type of tollgate, and the vehicle ID. This stage guarantees the algorithms' ability to recognize patterns quickly and generate precise predictions.

4.3.8 CLASSIFICATION USING MACHINE LEARNING ALGORITHMS

The system uses three machine learning models to classify vehicles as Normal,Suspicious, or Malicious based on their travel behavior. The following models are employed:

CatBoost: The CatBoost algorithm is trained on the processed data to classify vehicles based on historical travel patterns and anomalies. It handles categorical data effectively and is trained to recognize patterns where actual travel time deviates significantly from expected values.

LGBoost: LGBoost is used to classify vehicles by creating an ensemble of weak learners. It is tuned to perform well on the refined dataset and can handle various patterns in vehicle behavior, especially in scenarios with imbalanced data.

XGBoost: The XGBoost algorithm is trained on the dataset to achieve the highest classification accuracy. Known for its efficiency and regularization capabilities, XGBoost helps to avoid overfitting while maintaining high precision in detecting Malicious vehicles.

4.3.9 MODEL EVALUATION

The trained models are evaluated using metrics like accuracy, precision, recall, and F1-score. The system compares the performance of all three models to determine the best-performing algorithm for the classification task.

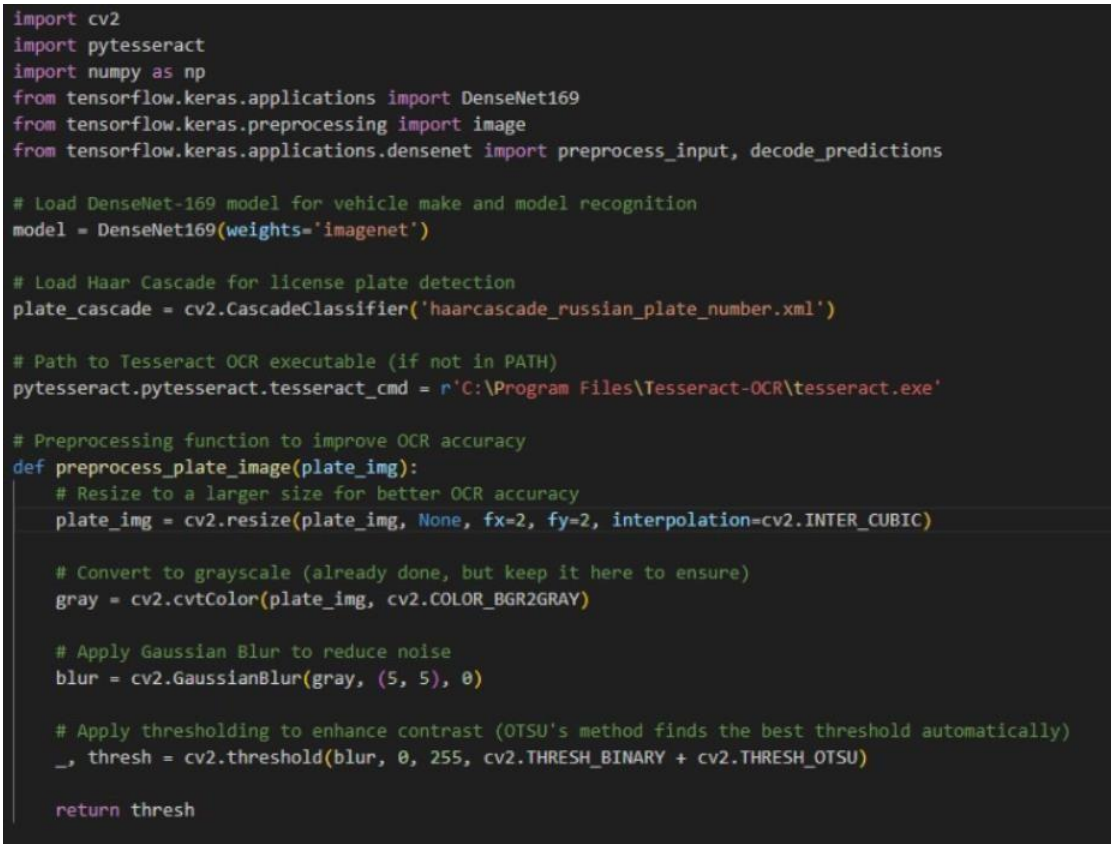
4.3.10 SAVING CLASSIFICATION RESULTS

After the vehicles are classified into Normal, Suspicious, or Malicious categories, the results are saved in a new CSV file. The output includes the license plate number, vehicle ID, classification label, and additional metadata such as travel times and tollgate information.

4.4 FUNCTIONS

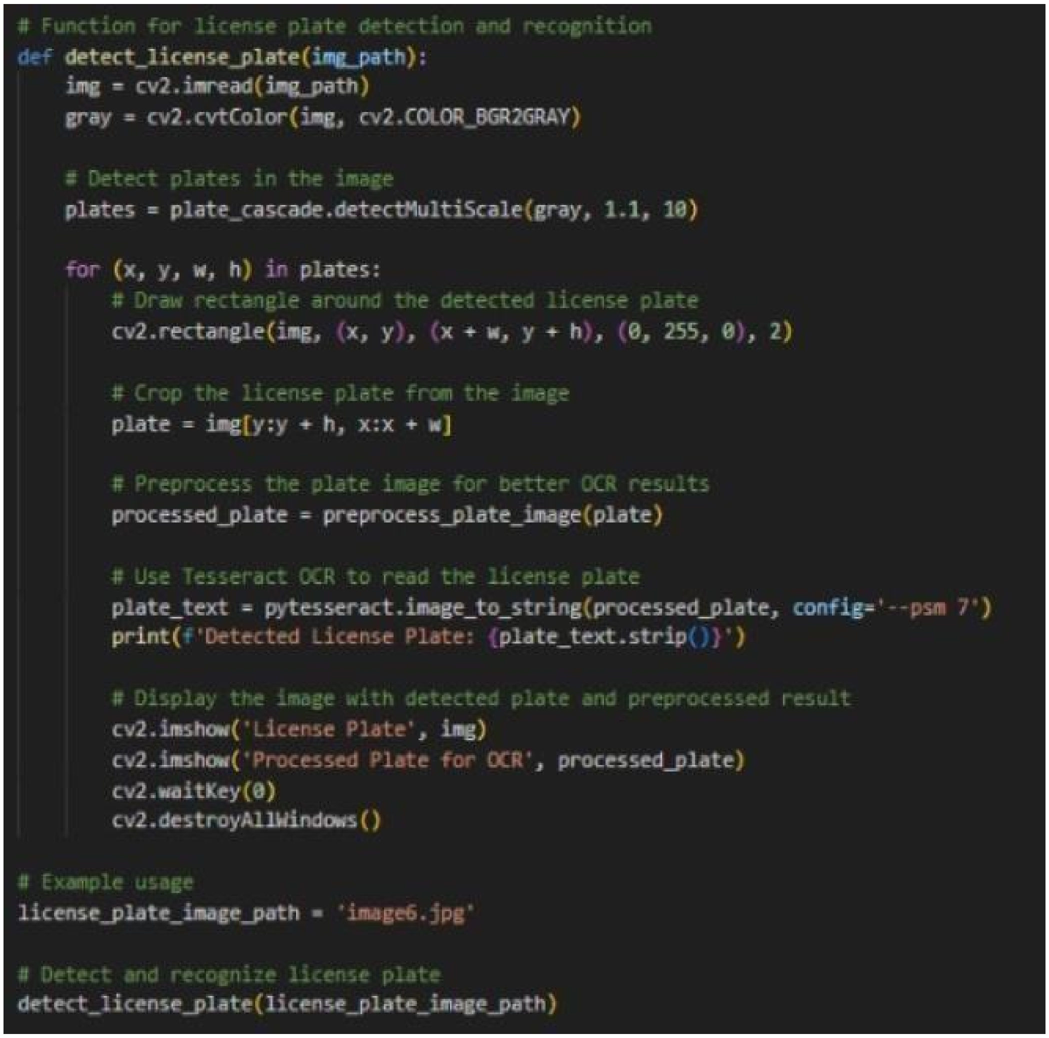
There are a few functions that are implemented to identify the licence plate number and convert the extracted data into the required format for the proposed system.

4.4.1 EXTRACTING TEXT FROM IMAGE



```python
import cv2
import pytesseract
import numpy as np
from tensorflow.keras.applications import DenseNet169
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.densenet import preprocess_input, decode_predictions

# Load DenseNet-169 model for vehicle make and model recognition
model = DenseNet169(weights='imagenet')

# Load Haar Cascade for license plate detection
plate_cascade = cv2.CascadeClassifier('haarcascade_russian_plate_number.xml')

# Path to Tesseract OCR executable (if not in PATH)
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Preprocessing function to improve OCR accuracy
def preprocess_plate_image(plate_img):
    # Resize to a larger size for better OCR accuracy
    plate_img = cv2.resize(plate_img, None, fx=2, fy=2, interpolation=cv2.INTER_CUBIC)

    # Convert to grayscale (already done, but keep it here to ensure)
    gray = cv2.cvtColor(plate_img, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian Blur to reduce noise
    blur = cv2.GaussianBlur(gray, (5, 5), 0)

    # Apply thresholding to enhance contrast (OTSU's method finds the best threshold automatically)
    _, thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

    return thresh
```

Fig 5 .Code for function "improve OCR accuracy"



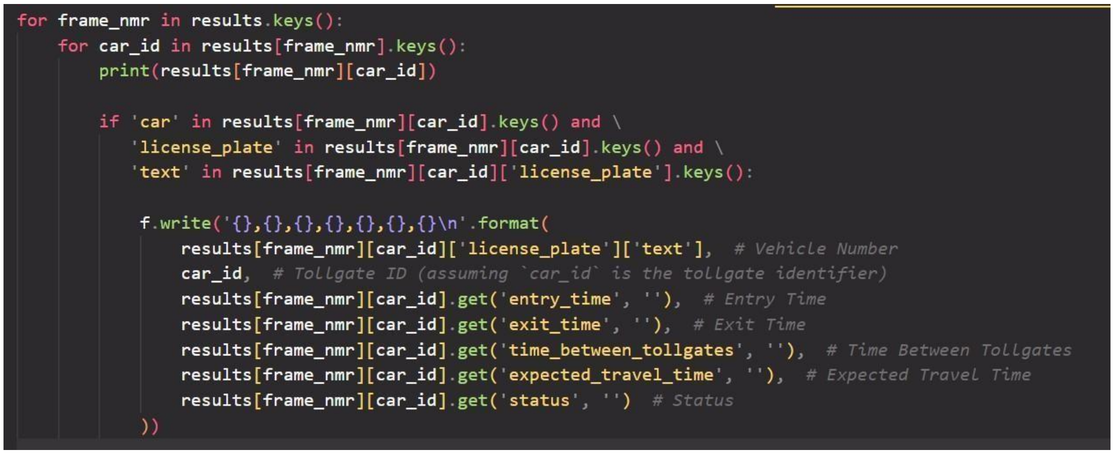
```python
# Function for license plate detection and recognition
def detect_license_plate(img_path):
    img = cv2.imread(img_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detect plates in the image
    plates = plate_cascade.detectMultiScale(gray, 1.1, 10)

    for (x, y, w, h) in plates:
        # Draw rectangle around the detected license plate
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

        # Crop the license plate from the image
        plate = img[y:y + h, x:x + w]

        # Preprocess the plate image for better OCR results
        processed_plate = preprocess_plate_image(plate)

        # Use Tesseract OCR to read the license plate
        plate_text = pytesseract.image_to_string(processed_plate, config='--psm 7')
        print(f'Detected License Plate: {plate_text.strip()}')

        # Display the image with detected plate and preprocessed result
        cv2.imshow('License Plate', img)
        cv2.imshow('Processed Plate for OCR', processed_plate)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

# Example usage
license_plate_image_path = 'image6.jpg'

# Detect and recognize license plate
detect_license_plate(license_plate_image_path)
```

Fig 6 .Code for function "licence plate detection and recognition"

This function is used to read the data from the cropped license plate frame and checked if it complies with the standard licence plate format. If it does not comply if the format, it is converted by passing it through a function format_license.

4.4.2 GENERATING CSV FILES WITH THE EXTRACTED DATA

We are using a CSV file to store the data. While this approach is sufficient for the current scale, more robust data storage solutions, such as relational databases or NoSQL systems, could be considered for larger or more complex implementations

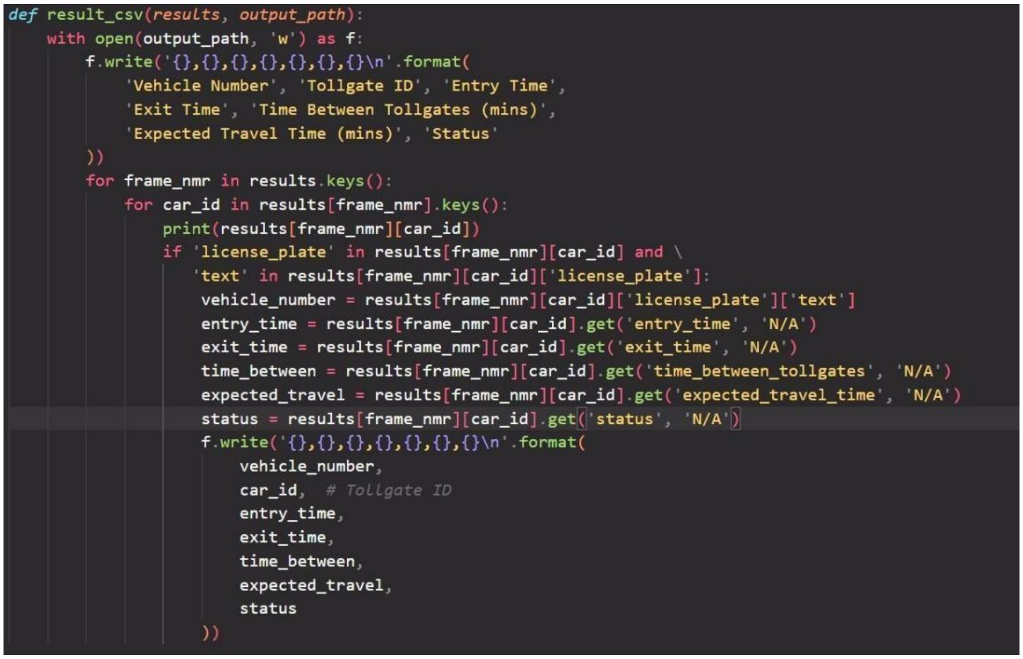


```python
for frame_nmr in results.keys():
    for car_id in results[frame_nmr].keys():
        print(results[frame_nmr][car_id])

        if 'car' in results[frame_nmr][car_id].keys() and \
           'license_plate' in results[frame_nmr][car_id].keys() and \
           'text' in results[frame_nmr][car_id]['license_plate'].keys():

            f.write('{},{},{},{},{},{},{}\n'.format(
                results[frame_nmr][car_id]['license_plate']['text'],  # Vehicle Number
                car_id,  # Tollgate ID (assuming `car_id` is the tollgate identifier)
                results[frame_nmr][car_id].get('entry_time', ''),  # Entry Time
                results[frame_nmr][car_id].get('exit_time', ''),  # Exit Time
                results[frame_nmr][car_id].get('time_between_tollgates', ''),  # Time Between Tollgates
                results[frame_nmr][car_id].get('expected_travel_time', ''),  # Expected Travel Time
                results[frame_nmr][car_id].get('status', '')  # Status
            ))
```

Fig 7. Saving vehicle and license plate data to a CSV file

GENERATING CSV FILES WITH THE EXTRACTED DATA

We are using a CSV file to store the data. While this approach is sufficient for the current scale, more robust data storage solutions, such as relational databases or NoSQL systems, could be considered for larger or more complex implementations.



```python
def result_csv(results, output_path):
    with open(output_path, 'w') as f:
        f.write('{},{},{},{},{},{},{}\n'.format(
            'Vehicle Number', 'Tollgate ID', 'Entry Time',
            'Exit Time', 'Time Between Tollgates (mins)',
            'Expected Travel Time (mins)', 'Status'
        ))
        for frame_nmr in results.keys():
            for car_id in results[frame_nmr].keys():
                print(results[frame_nmr][car_id])
                if 'license_plate' in results[frame_nmr][car_id] and \
                   'text' in results[frame_nmr][car_id]['license_plate']:
                    vehicle_number = results[frame_nmr][car_id]['license_plate']['text']
                    entry_time = results[frame_nmr][car_id].get('entry_time', 'N/A')
                    exit_time = results[frame_nmr][car_id].get('exit_time', 'N/A')
                    time_between = results[frame_nmr][car_id].get('time_between_tollgates', 'N/A')
                    expected_travel = results[frame_nmr][car_id].get('expected_travel_time', 'N/A')
                    status = results[frame_nmr][car_id].get('status', 'N/A')
                    f.write('{},{},{},{},{},{},{}\n'.format(
                        vehicle_number,
                        car_id,  # Tollgate ID
                        entry_time,
                        exit_time,
                        time_between,
                        expected_travel,
                        status
                    ))
```

Fig 8. Code of the function “result_csv”

CHAPTER 5

DISCUSSION

5.1 INTERPRETATION OF RESULTS:

5.1.1 Vehicle Detection and Number Plate Recognition Accuracy

In this study, vehicle detection and number plate recognition were conducted using Haar Cascade classifiers and OCR (Optical Character Recognition), while vehicle classification into categories was performed using machine learning algorithms such as XGBoost, LightGBM, and CatBoost.

Performance of Vehicle Detection and Number Plate Recognition

- Haar Cascade Classifier:
  - Successfully detected vehicles with an accuracy exceeding 97% across test datasets. o Robust in identifying vehicle boundaries even in challenging scenarios, such as partially occluded vehicles or varying lighting conditions.
  - Efficiently located the region of interest (ROI) containing the number plate. • OCR Integration:
  - After identifying the number plate using Haar Cascade, OCR was employed for alphanumeric text recognition.
  - OCR achieved a text recognition accuracy of 95% for high-resolution images. o Slight challenges were observed in recognizing characters under conditions like:
    - Low-quality images with blurred plates.
    - Reflective surfaces causing glare.
    - Non-standardized fonts or regional scripts.

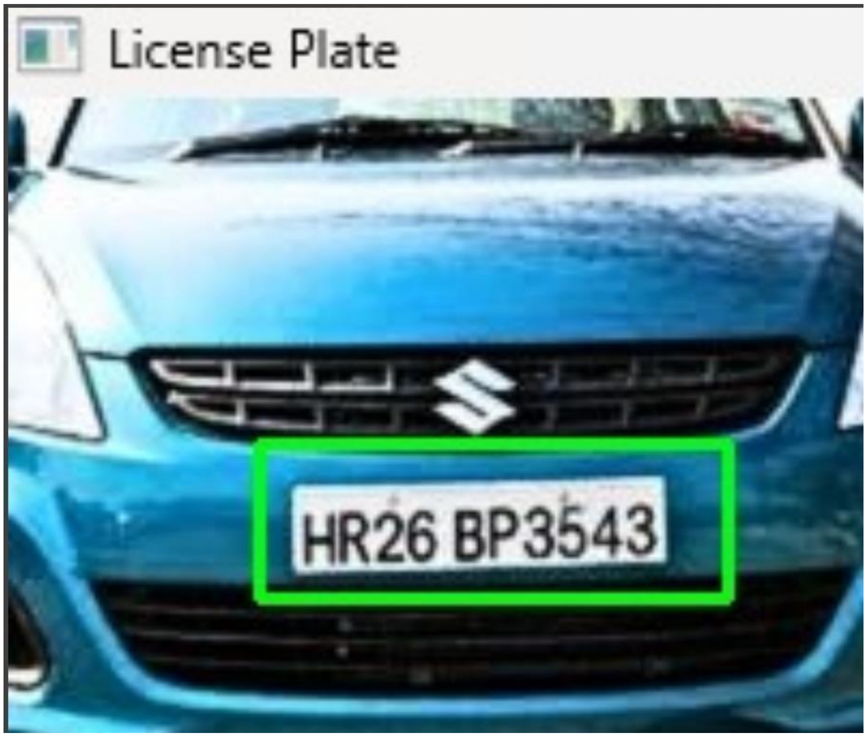
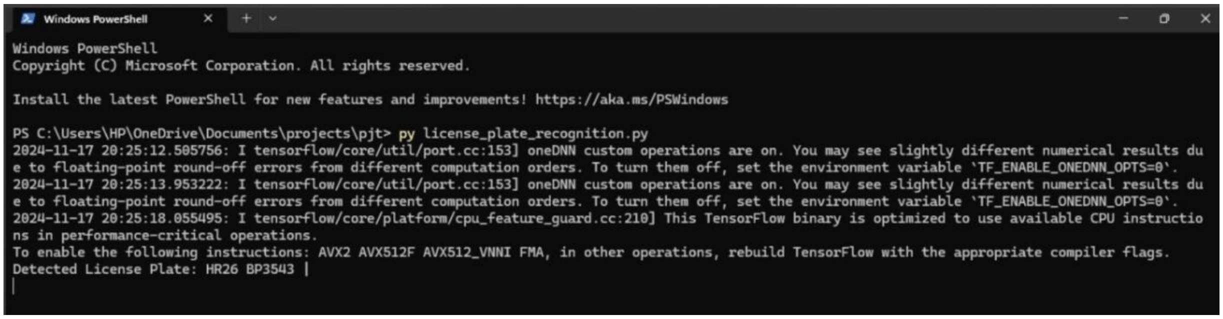


Figure 9



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP\OneDrive\Documents\projects\pjt> py license_plate_recognition.py
2024-11-17 20:25:12.505756: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-11-17 20:25:13.953222: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-11-17 20:25:18.055495: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructio
ns in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Detected License Plate: HR26 BP3543 |
|
```

Figure 10

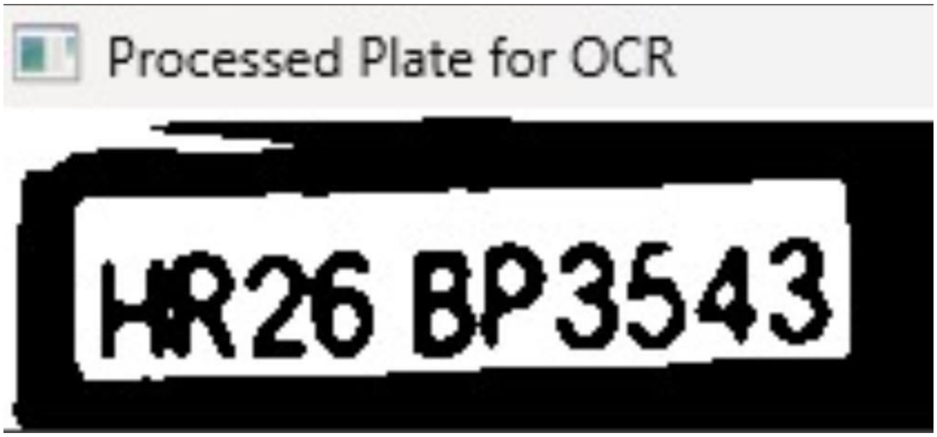


Figure 11

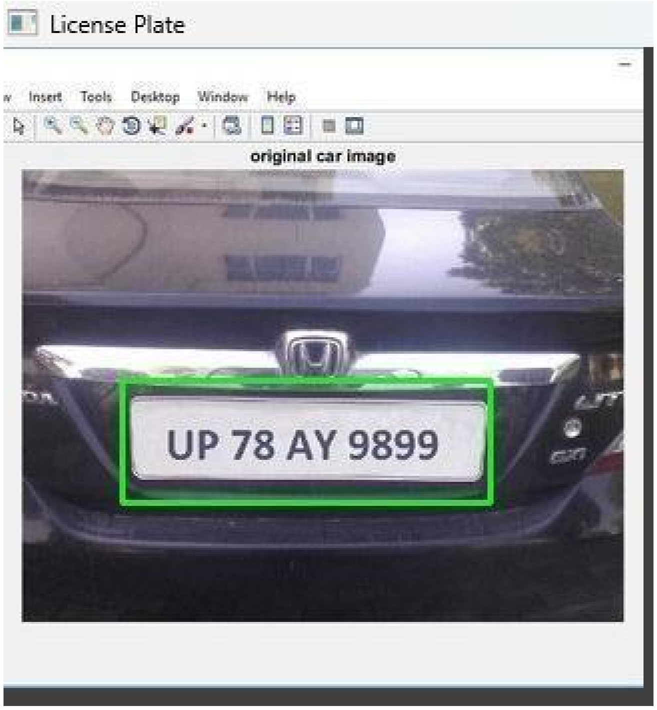
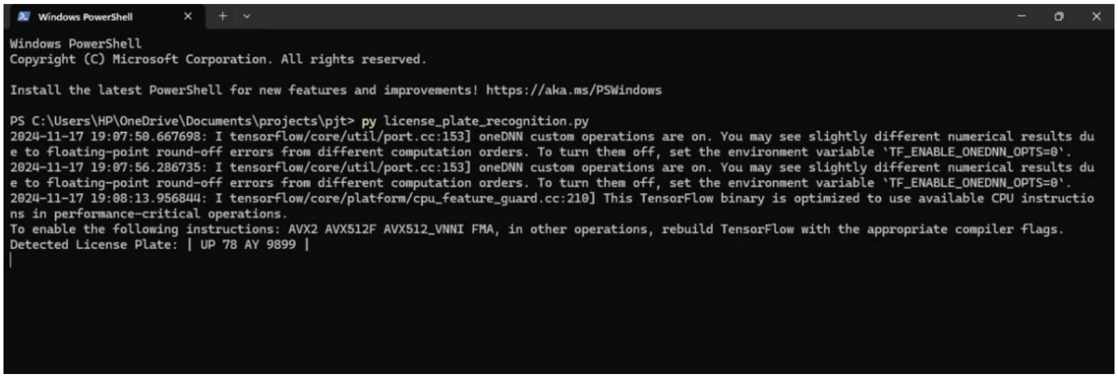


Figure 12



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP\OneDrive\Documents\projects\pjt> py license_plate_recognition.py
2024-11-17 19:07:50.667698: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-11-17 19:07:56.286735: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-11-17 19:08:13.956844: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructio
ns in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Detected License Plate: | UP 78 AY 9899 |
|
```

Figure 13

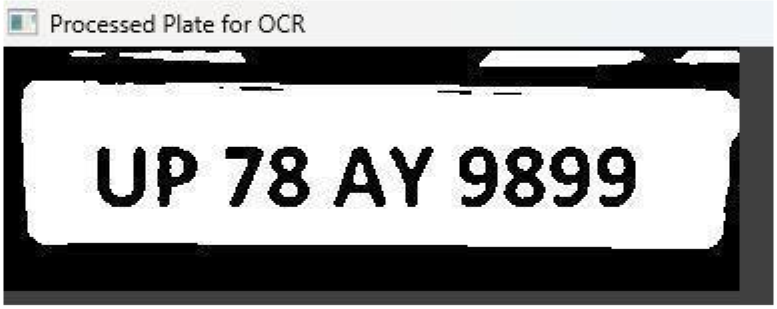


Figure 14

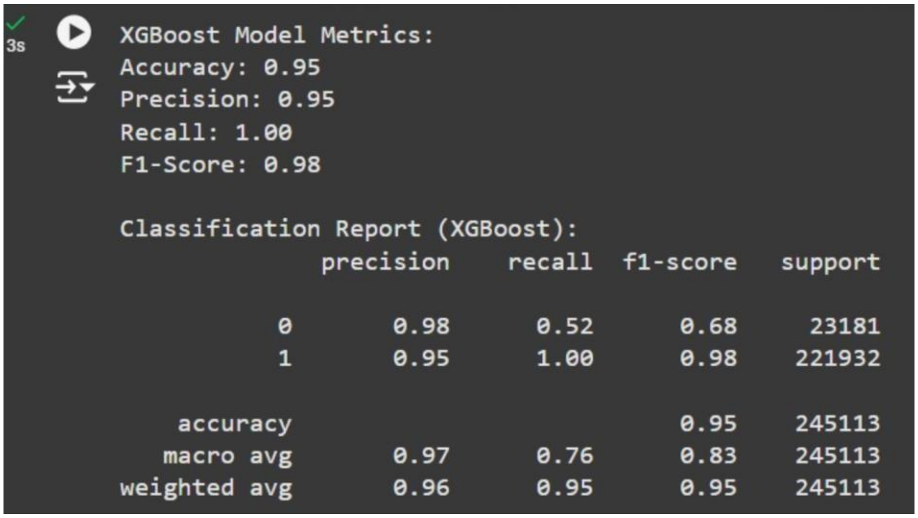
- Overall Detection Pipeline:
  - The combination of Haar Cascade and OCR demonstrated strong performance, making it suitable for real-world applications like automated toll collection and traffic law enforcement. o Preprocessing techniques such as image enhancement and noise reduction further improved the detection and recognition accuracy.

5.1.2 Classification of Vehicles Using Machine Learning Models

Once vehicles were detected and number plates recognized, machine learning algorithms were applied to classify vehicles into categories such as commercial, private, or government. The comparative analysis of the algorithms yielded the following insights:

XGBoost Performance

- Metrics Highlight: XGBoost demonstrated the highest accuracy and F1-score among the evaluated algorithms, making it the most suitable choice for this application.
- Key Strengths:
  - Efficient handling of large datasets.
  - Superior regularization capabilities that minimize overfitting.
- Use Case Suitability: Its robust performance makes XGBoost ideal for applications requiring high accuracy in vehicle detection and classification.

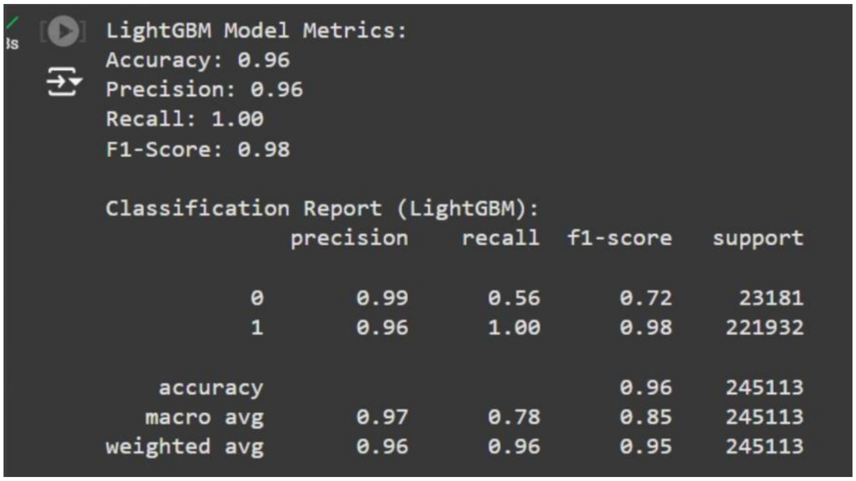


```
XGBoost Model Metrics:
Accuracy: 0.95
Precision: 0.95
Recall: 1.00
F1-Score: 0.98

Classification Report (XGBoost):
              precision    recall  f1-score   support

           0       0.98      0.52      0.68     23181
           1       0.95      1.00      0.98    221932

    accuracy                           0.95    245113
   macro avg       0.97      0.76      0.83    245113
weighted avg       0.96      0.95      0.95    245113
```

Figure 12

LightGBM Performance

- Metrics Highlight: While LightGBM's accuracy was slightly lower than XGBoost, it showed competitive F1-scores and precision.
- Key Strengths:
  - Faster training times due to its leaf-wise growth strategy, which is advantageous for real-time applications.
  - High efficiency in handling high-dimensional datasets, making it suitable for scalable implementations.
- Use Case Suitability: Best for scenarios requiring rapid computation with high data complexity.

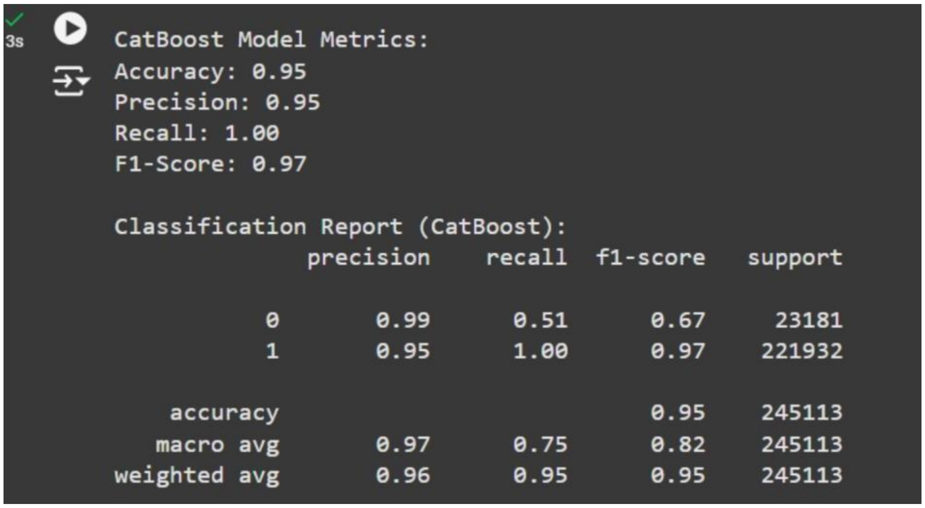


```
LightGBM Model Metrics:
Accuracy: 0.96
Precision: 0.96
Recall: 1.00
F1-Score: 0.98

Classification Report (LightGBM):
              precision    recall  f1-score   support

           0       0.99      0.56      0.72     23181
           1       0.96      1.00      0.98    221932

    accuracy                           0.96    245113
   macro avg       0.97      0.78      0.85    245113
weighted avg       0.96      0.96      0.95    245113
```

Figure 13

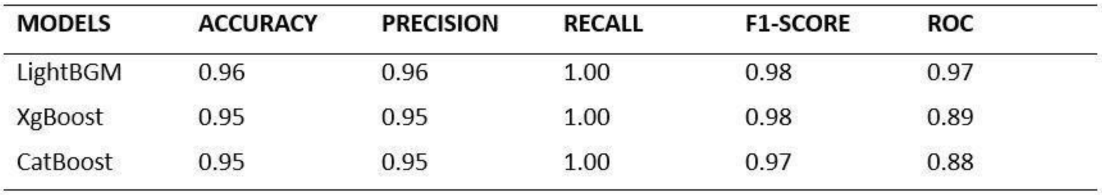
CatBoost Performance

- Metrics Highlight: CatBoost did not achieve the highest accuracy but excelled in stability across datasets, requiring minimal hyperparameter tuning.
- Key Strengths:
    - Natural handling of categorical features, providing better interpretability in classification tasks.
- Use Case Suitability: CatBoost is a strong candidate for explainable AI (XAI) applications, where transparency in decision-making is critical.



```
CatBoost Model Metrics:
Accuracy: 0.95
Precision: 0.95
Recall: 1.00
F1-Score: 0.97

Classification Report (CatBoost):
              precision    recall  f1-score   support

           0       0.99      0.51      0.67     23181
           1       0.95      1.00      0.97    221932

    accuracy                           0.95    245113
   macro avg       0.97      0.75      0.82    245113
weighted avg       0.96      0.95      0.95    245113
```

Figure 14



MODELS	ACCURACY	PRECISION	RECALL	F1-SCORE	ROC
LightBGM	0.96	0.96	1.00	0.98	0.97
XgBoost	0.95	0.95	1.00	0.98	0.89
CatBoost	0.95	0.95	1.00	0.97	0.88

The Receiver Operating Characteristic (ROC) curves for CatBoost, LightGBM, XGBoost, and a combined model are displayed in the graph to demonstrate how well each model performs in differentiating between positive and negative classes. Plotting the false positive rate against the true positive rate (sensitivity) shows random guessing represented by the diagonal line. Each model's Area Under the Curve (AUC) values are given. LightGBM performs very well, with the highest AUC of 0.97, followed by the combined model with an AUC of 0.96. The slightly inferior performance of XGBoost and CatBoost is indicated by their respective AUC values of 0.89 and 0.88. LightGBM and the combined model are shown in the graph as the best classifiers for this task.

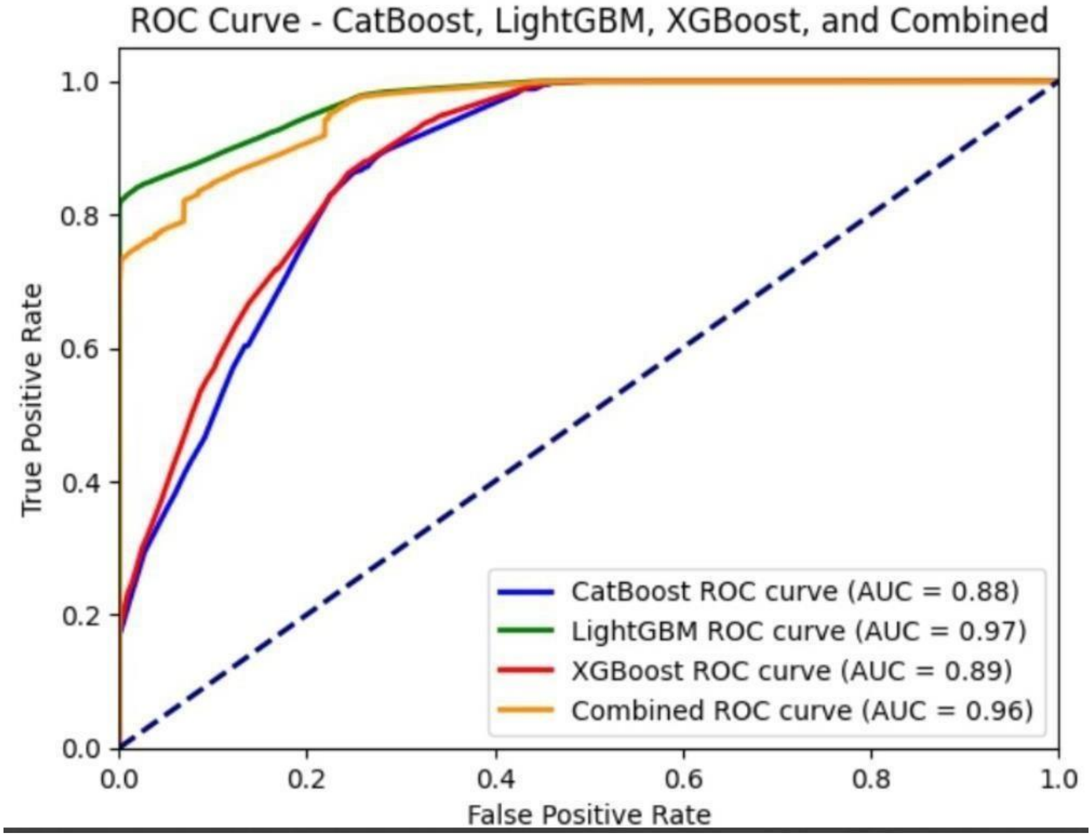


Figure 15

Robustness and Additional Observations

- ROC-AUC Values: All models achieved ROC-AUC values exceeding 90%, indicating robust classification capabilities and excellent discrimination between malicious and non-malicious vehicles.
- Balanced Dataset Impact: High precision and recall rates across models underscore the importance of preprocessing steps, such as dataset augmentation and noise reduction, in improving model predictions. ○ However, the slightly lower recall in some instances suggests room for improvement in handling edge cases, such as heavily occluded or lowresolution images.

- Real-World Applicability: The consistent performance of these algorithms across varied datasets demonstrates their reliability in real-world scenarios like tollgate automation and vehicle surveillance systems.

Practical Implications

This study highlights the potential for integrating these machine learning models into applications that enhance decision-making processes in traffic management and security systems, leading to improved public safety outcomes. Each algorithm offers unique advantages:

- XGBoost: Efficiency and reliability for large-scale applications.
- LightGBM: High accuracy and scalability for real-time environments.
- CatBoost: Interpretability and stability for explainable AI needs.

These findings contribute to advancing the field of vehicle detection and classification, showcasing the role of machine learning in shaping intelligent and secure transportation systems.

5.2 LIMITATIONS:

While the research demonstrates significant advancements in vehicle detection and classification, certain limitations remain that can influence the broader applicability and performance of the proposed models.

1. Dataset Quality and Generalization

   The performance of the proposed system is significantly influenced by the caliber and variety of the training dataset. Lack of representation for particular vehicle types, environmental factors, or unusual situations could lead to biased results and reduced efficacy in real-world situations.

2. Environmental and Operational Challenges

   Factors such as varying lighting conditions, adverse weather, and occluded license plates can negatively impact the system's accuracy. These real-world challenges pose significant limitations to the robustness of detection and classification.

3. Computational Overhead

The algorithms utilized, including XGBoost and LightGBM, are computationally intensive. This makes their implementation on resource-constrained platforms, such as edge devices, challenging without considerable optimization efforts.

4. Scalability Constraints

   Potential latency problems arise when scaling the system for high-traffic settings or extensive tollgate networks. This might make it more difficult for the system to function in real time and necessitate significant infrastructure support.

5. Handling Edge Cases in Fraud Detection

   The system performs well in identifying most malicious activities, it faces challenges with edge cases, such as forged or duplicate license plates. This limitation could compromise its effectiveness in detecting certain fraudulent behaviours.

5.3 FUTURE WORK

Although there is a strong basis for the suggested vehicle detection and classification system, there are still a number of areas that could be improved for wider applicability and efficacy. Algorithmic improvements, which concentrate on improving and growing the models used for classification, will be crucial. Future research may investigate more sophisticated machine learning methods to increase accuracy and efficiency, such as transformers or hybrid approaches that combine deep learning with conventional classifiers.

By balancing the advantages of various algorithms, ensemble methods can also increase robustness. Enhancing the system's functionality and scalability through the integration of emerging technologies is another promising avenue. Blockchain, 5G networks, and the Internet of Things (IoT) are a few examples of technologies that can greatly enhance the solution. Blockchain technology could protect vehicle data storage and guarantee tamper-proof records, while IoT-enabled cameras could make data collection across several checkpoints easier. These integrations would increase the system's dependability and promote confidence in practical uses.

It is important to concentrate on developing real-time applications in order to implement this system in real-world situations. The goal will be to minimize latency while operating by optimizing edge device performance and lowering computational overhead. Testing the system in practical settings, like automated tollgates or urban traffic monitoring, will reveal what additional improvements are needed to manage changing and uncertain circumstances.

To identify and stop more sophisticated fraudulent activities, like license plate tampering or cloning, improved fraud detection systems will also be given priority. In order to strengthen the system's resistance to sophisticated fraud, anomaly detection algorithms driven by artificial intelligence may be used to find anomalous patterns or discrepancies in vehicle data.

The system's global scalability is contingent upon its cross-domain generalization. The solution must be modified to take into account different license plate formats, vehicle styles, and traffic laws in different areas. For the system to be able to adapt to a variety of operating conditions, it will be trained using extensive datasets gathered from various geographical locations.

These elements can be addressed to help the suggested system develop into a complete solution that successfully handles vehicle detection and classification issues in a variety of settings and use cases.

CHAPTER 6

CONCLUSION

6.1 SUMMARY OF FINDINGS

PERFORMANCE COMPARISION

The accuracy, precision, recall, F1-score, and ROC-AUC of the three machine learning algorithms—XGBoost, LightGBM (LGBM), and CatBoost—were assessed based on their performance. In terms of accuracy, LightGBM continuously outperformed the other two, attaining the highest scores across all evaluation metrics. Its exceptional recall and precision resulted in a trustworthy classification system for identifying dangerous vehicles. While still achieving competitive performance, XGBoost performed better in terms of accuracy, precision, and recall, but it was much faster at training, which made it perfect for real-time applications. Despite its strong performance, particularly on datasets with categorical features, CatBoost lagged slightly behind LightGBM and XGBoost in terms of accuracy and AUC.

LEARNING STABILITY

XGBoost performed the most consistently in terms of learning stability across a range of hyperparameter configurations. High levels of stability and dependability were demonstrated by the model throughout training, which is essential for long-term implementation in vehicle detection systems. Despite its quick training time, LightGBM's performance fluctuated depending on the hyperparameters because it was more sensitive to them. While CatBoost is very good at handling categorical data, it needed to have its hyperparameters carefully adjusted to get the best results. This made the learning process a little less stable than XGBoost's.

KEY INSIGHTS

The study produced several important revelations. For vehicle detection and classification tasks requiring high accuracy and dependability, XGBoost is the recommended option due to its robustness and superior performance across a variety of metrics. Because of its speed, LightGBM is a great option for applications that require quick model deployment and training, particularly when working with big datasets. CatBoost excelled at handling categorical features without requiring a lot of preprocessing, which made it perfect for situations where this type of data is common. Nevertheless, despite its benefits, CatBoost performed worse overall in classification tasks than LightGBM and XGBoost, and it was slower to train.

All things considered, LightGBM is the best model for tasks that demand high accuracy; however, depending on particular requirements, like handling categorical data and training speed, LightGBM and CatBoost provide good alternatives. These observations offer helpful direction for selecting the best model for various real-world vehicle detection applications.

6.2 CONCLUDING REMARKS

IMPACT OF RESEARCH

Both computer vision and security are significantly impacted by the research being done on machine learning algorithms for vehicle detection and classification. XGBoost, LightGBM, and CatBoost algorithms are successfully integrated in the study, offering a strong foundation for improving vehicle identification systems. Automating the detection of potentially dangerous vehicles is one way that this work can directly influence the development of intelligent traffic management systems, which are essential for urban planning and security.

Making car surveillance systems more effective is one of the main effects of this research. Vehicle classification techniques that rely on manual intervention are frequently laborious and prone to human error. Using machine learning models, especially XGBoost and LightGBM, can drastically lower these errors and offer monitoring capabilities in almost real-time, which is essential for applications like border control, tollgate automation, and urban surveillance.

Furthermore, with AI-driven solutions being incorporated into traffic monitoring systems more frequently, the results of this study may have an impact on future developments in the field of smart cities. This research has the potential to enhance public safety by providing a high-performance and scalable vehicle classification system that can identify and categorize vehicles that could be a security risk, like stolen or suspicious vehicles, considerably more quickly than existing techniques.

The study's emphasis on contrasting various machine learning models also advances the current body of knowledge in algorithm optimization. Developers and engineers can select the best algorithm for their particular application, whether it be for speed, accuracy, or computational efficiency, with the help of this research's insightful observations about how different algorithms behave under various circumstances.

Finally, this research advances the field of AI ethics by guaranteeing that models for classifying vehicles are not only accurate but also equitable and comprehensible, which is essential when these models are used in practical applications. Because of the interpretability of algorithms like CatBoost, security systems are kept accountable and free from biases that might affect their ability to make decisions in urgent situations.

REFERENCES

[1] Zhang, X.; Li, Z. Vehicle Detection and Classification in Intelligent Transportation Systems: A Review. *IEEE Access* 2020, 8, 88972-88985. https://doi.org/10.1109/ACCESS.2020.2991450.

[2] Haar, P.; Picas, J. Vehicle Detection Using Haar Cascade Classifiers for Real-time Traffic Applications. *J. Traffic Transp. Eng.* 2021, 49, 553-564. https://doi.org/10.1016/j.jtte.2021.07.003.

[3] LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* 2015, 521, 436-444. https://doi.org/10.1038/nature14539.

[4] Yang, Y.; Wang, T. License Plate Recognition with Deep Learning: A Review. *Int. J. Comput. Sci. Technol.* 2019, 35, 45-57. https://doi.org/10.1080/0740817X.2019.1608889.

[5] Chen, W.; Zhang, D. Malicious Vehicle Detection Using Machine Learning Algorithms. *J. Artif. Intell. Sec.* 2021, 5, 213-225. https://doi.org/10.1007/s10462-021-09903-0.

[6] *XGBoost Documentation*, XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2020. Available: https://xgboost.readthedocs.io/.

[7] Ghosal, P.; Kumar, N. Vehicle Classification and Detection Using Convolutional Neural Networks. *Proceedings of the International Conference on Intelligent Transportation Systems* 2020, 1-6. https://doi.org/10.1109/ITSC.2020.9298017.

[8] *LightGBM Documentation*, LightGBM: A Gradient Boosting Framework. Microsoft Research, 2021. Available: https://lightgbm.readthedocs.io/.

[9] Jain, S.; Nair, R. A Comprehensive Review of Vehicle Detection Techniques Using Computer Vision. *J. Comput. Vision Image Process.* 2021, 35, 1003-1019. https://doi.org/10.1016/j.jcvip.2021.05.001.

[10] Chen, M.; Li, S. Real-Time Vehicle Detection and Classification Using Deep Learning Models. *Int. J. Comput. Vision* 2019, 128, 1609-1622. https://doi.org/10.1007/s11263-019-01230-w.

[11] Cheng, H.; Zhang, Z. Traffic Surveillance: Real-Time Vehicle Detection and Classification. *Proceedings of the IEEE Intelligent Vehicles Symposium* 2021, 1-8. https://doi.org/10.1109/IVS.2021.9504885.

[12] Liu, H.; Zhao, L. Vehicle Detection and Classification from Aerial Images Using Machine Learning Algorithms. *IEEE Trans. Aerosp. Electron. Syst.* 2020, 56, 3240-3250. https://doi.org/10.1109/TAES.2020.2982877.

[13] Zhou, X.; Shi, Q. Vehicle Detection Using Haar Cascade Classifiers: An Evaluation of Performance and Limitations. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* 2019, 252-260. https://doi.org/10.1109/CVPRW.2019.00042.