

RED LIGHT TRAFFIC VIOLATION AND NUMBER PLATE DETECTION USING MONGODB

A PROJECT REPORT

Submitted by

AKSHAT NEOLIA [RA2211031010080]

PRIYANSHU KUMAR [RA2211031010084]

TARANG BHARGAVA [RA2211031010099]

RAJEEV SINGH [RA2211031010120]

Under the Guidance of

DR. ANGAYARKANNI S A

(Assistant Professor, NWC)

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in **INFORMATION TECHNOLOGY**



DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

NOVEMBER 2024



Department of Networking and Communications
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B. Tech / CSE-IT
Student Names : Akshat Neolia, Priyanshu Kumar, Tarang Bhargava, Rajeev Singh
Registration Numbers : RA2211031010080, RA2211031010084, RA2211031010099, RA2211031010120
Title of Work : Red Light Traffic Violation and Number Plate Detection

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

Akshat Neolia:

Priyanshu Kumar:

Tarang Bhargava:

Rajeev Singh:

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

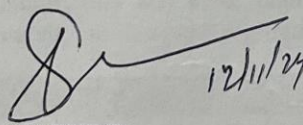


SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSC314P – Big Data Essentials mini-project report titled “**RED LIGHT TRAFFIC VIOLATION AND NUMBER PLATE DETECTION USING MONGODB**” is the bonafide work of “**AKSHAT NEOLIA [RA2211031010080], PRIYANSHU KUMAR [RA2211031010084], TARANG BHARGAVA [RA2211031010099], RAJEEV SINGH [RA2211031010120]**” who carried out the mini-project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

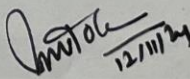
Panel Reviewer I


SIGNATURE

Dr. Angayarkanni S A
Assistant Professor
Department of Networking
and Communications



Panel Reviewer II


SIGNATURE

Dr. Sivamohan S
Assistant Professor
Department of Networking
and Communications

ABSTRACT

The Road Traffic Violation Detection System, leveraging big data tools and modern computer vision techniques, is designed to improve transportation management in smart cities. This system employs OpenCV within a Python environment, integrating object detection and tracking algorithms to monitor and manage traffic violations, specifically focusing on red-light violations. The core functionality revolves around accurately detecting vehicles violating traffic signals, tracking their positions in real time, and identifying their number plates. The object detection module identifies and locates vehicles within the traffic flow, while the object tracker ensures continuous monitoring of each vehicle's position, even as it moves through the intersection. The system effectively captures violating vehicles by correlating their positions with the traffic light status, flagging instances where a vehicle crosses the red light. The integrated number plate recognition technology further enhances the system's capability by allowing automatic identification of offenders. The output of the system includes precise location data of violating vehicles and their number plates, enabling efficient monitoring and enforcement. The implementation of this system provides significant improvements in traffic management, enhances safety, and contributes to the optimization of traffic flow in urban areas. Additionally, it supports the creation of automated traffic law enforcement in smart city frameworks.

TABLE OF CONTENT

CHAPTER NO.	CONTENT	PAGE NO
1	INTRODUCTION	1
2	LITERATURE SURVEY	5
3	PROPOSED METHODOLOGY	12
4	IMPLEMENTATION	14
5	RESULT	20
6	CONCLUSION & REFERENCE	24
7	APPENDIX-I	31
8	APPENDIX-II	45

CHAPTER 1

INTRODUCTION

With the rapid growth of urban populations and increasing vehicle numbers, managing road traffic has become one of the most challenging tasks for cities worldwide. Traffic congestion, road accidents, and violations are becoming more common, especially in busy urban areas, which negatively impact safety, productivity, and quality of life. Traditional traffic management systems are often insufficient to handle the complexities of modern urban mobility, making it crucial to adopt advanced technologies that can provide real-time monitoring, enforcement, and analytics. Among the most dangerous and widespread traffic violations are red-light running, which is a major cause of accidents at intersections. To address these challenges, smart cities are increasingly turning to automated traffic monitoring and violation detection systems that rely on modern technologies like computer vision, big data tools, and artificial intelligence (AI).

The Road Traffic Violation Detection System is a cutting-edge solution that combines real-time video processing, object detection, and tracking algorithms to automatically detect and record red-light violations. By leveraging OpenCV and Python, the system uses advanced computer vision techniques to monitor intersections, detect vehicles approaching and passing through red lights, and accurately track their movements. This project aims to create an automated, accurate, and efficient method for identifying traffic offenders, especially those violating red lights, in real-time.

At the heart of the system is a combination of object detection and tracking modules. The object detection algorithm is responsible for identifying vehicles within the frame captured by the camera, while the object tracker ensures continuous monitoring of each vehicle's movement across the intersection. The system is designed to accurately pinpoint the location of each vehicle and track its trajectory, even as it moves through the traffic flow. By determining when a vehicle crosses the intersection after the traffic light turns red, the system is able to flag a violation. This process is done automatically, without human intervention, ensuring a continuous and error-free monitoring mechanism. In addition to detecting the violation, the system also incorporates Automatic Number Plate Recognition (ANPR) technology, which allows it to capture the license plates of violating vehicles. The integration of ANPR enhances the system's capability by enabling the identification of offending vehicles and linking them to specific drivers. This provides law enforcement authorities with

valuable data for issuing fines, tracking repeat offenders, and taking further legal actions. The captured license plates are stored in a database for easy retrieval, enabling quick and efficient enforcement.

The system is designed to be highly efficient, providing real-time alerts and reports to traffic management centers or law enforcement agencies. The results are shown with high accuracy, as all violating vehicles are detected, tracked, and distinguished precisely. This automated process reduces the need for manual surveillance, allowing authorities to focus on responding to violations and improving traffic safety. By utilizing big data tools, the system can also gather valuable traffic data for further analysis, such as patterns in traffic violations, peak violation times, and the identification of high-risk intersections. This data can be used to optimize traffic management, improve road safety measures, and inform future urban planning decisions.

In conclusion, the Road Traffic Violation Detection System represents a significant advancement in smart city infrastructure. By utilizing modern computer vision techniques, object detection and tracking algorithms, and ANPR technology, this system enhances traffic law enforcement, improves road safety, and contributes to the creation of smarter, more efficient cities. As urban areas continue to grow, automated traffic management systems like this will play a crucial role in ensuring safe and orderly movement of vehicles, reducing accidents, and improving overall traffic flow.

1.2 BACKGROUND AND MOTIVATION

With the growing population and increasing vehicle density in urban areas, traffic management has become a critical issue for cities worldwide. Traffic congestion, accidents, and violations are on the rise, putting a strain on the safety and productivity of urban environments. Traditional traffic monitoring methods, such as manual enforcement and static cameras, are often insufficient to effectively manage the complexities of modern traffic systems. Red-light running, in particular, is a common traffic violation that leads to a significant number of accidents and fatalities. The increasing need for more efficient, automated, and data-driven traffic management solutions has become more urgent, especially in the context of developing smart cities where technology can help streamline urban operations. As a result, there is a growing emphasis on adopting advanced technologies, such as computer vision, AI, and big data tools, to enhance traffic monitoring and law enforcement.

1.3 PROBLEM STATEMENT

The problem of traffic violations, especially red-light running, continues to be a major challenge in cities around the world. While traffic cameras are in place to capture violations, many existing systems lack the capability to automatically detect and track violators in real-time. Furthermore, traditional systems often fail to integrate advanced analytics that can offer actionable insights to improve traffic management. Manual review of footage and data is time-consuming, prone to errors, and inefficient. In addition, many cities face difficulties in accurately identifying and tracking vehicles involved in violations, particularly when the traffic flow is dense, and vehicles quickly pass through intersections. Therefore, the need for an automated system capable of accurately detecting red-light violations, tracking violators, and capturing necessary information such as license plate details in real-time is paramount. This system would not only enhance traffic enforcement but also provide valuable data for optimizing traffic flow and improving road safety.

1.4 SOLUTION OVERVIEW

The Road Traffic Violation Detection System is designed to address these challenges by integrating modern computer vision techniques, object detection, tracking algorithms, and Automatic Number Plate Recognition (ANPR) technology. By using OpenCV and Python, the system automates the process of detecting red-light violations and accurately tracking vehicles as they move through intersections. The system's core functionality revolves around real-time video processing, where vehicles are detected and their movements are tracked through the use of object detection and tracking algorithms. Upon detecting a violation, the system captures the number plates of the offending vehicles using ANPR, providing law enforcement with the necessary information for issuing fines or further action. This solution not only enhances the efficiency of traffic monitoring but also contributes to safer and more efficient urban mobility by providing real-time insights into traffic patterns and violations. By automating the detection and enforcement process, the system helps reduce human error and reliance on manual intervention, making traffic law enforcement smarter and more effective.

1.5 OBJECTIVES

The primary objective of the Road Traffic Violation Detection System is to develop an automated, efficient, and accurate system that can detect red-light violations in real-time at traffic intersections. The system aims to enhance traffic law enforcement by leveraging modern computer vision techniques, machine learning algorithms, and big data tools to identify and track vehicles

running red lights without human intervention. One of the key goals is to improve the accuracy and efficiency of traffic monitoring by automating the detection process, thus reducing the reliance on manual surveillance and minimizing errors caused by human oversight.

Another important objective is to integrate object detection and tracking technologies to monitor vehicles as they approach and pass-through intersections. The system will identify vehicles in real-time, accurately tracking their movements as they interact with traffic lights. By analyzing the status of the traffic signal and comparing it to the movement of vehicles, the system will be able to flag any red-light violations, providing law enforcement agencies with precise and timely information for enforcement purposes. The use of object tracking ensures that the position of vehicles is continuously monitored, even when multiple vehicles are present at the intersection, thus enhancing the robustness of the system in complex traffic scenarios.

An essential goal of the system is to incorporate Automatic Number Plate Recognition (ANPR) technology, allowing for the identification of violating vehicles by capturing their license plate numbers. This will enable the system to not only detect violations but also link them to specific vehicles, facilitating the issuance of fines or other legal actions. The ability to capture and store number plate data in a secure database will help authorities keep track of offenders and create a record of violations that can be referenced later if needed. The system also aims to provide valuable traffic data and insights that can be used to optimize traffic management. By collecting data on red-light violations, traffic flow, and peak violation times, the system can help authorities identify high-risk intersections and take preventive measures to reduce the occurrence of accidents. This data-driven approach will contribute to smarter traffic management strategies and more efficient use of urban infrastructure.

Overall, the goal is to create a system that not only enhances the enforcement of traffic laws but also contributes to safer roadways, improved traffic flow, and more informed decision-making by traffic management authorities. Through automation, real-time processing, and data integration, the system seeks to revolutionize traffic violation detection and provide a model for smart city infrastructure that can be expanded and adapted to different urban environments.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

The literature survey for the Road Traffic Violation Detection System focuses on reviewing existing research, technologies, and methodologies related to traffic monitoring, violation detection, and automated enforcement systems. In recent years, urbanization and increasing vehicle numbers have led to heightened traffic congestion and a surge in road traffic violations, particularly at intersections where red-light running remains a significant issue. As a result, numerous studies and technologies have been explored to address these challenges, especially in the context of smart cities, where advanced solutions like artificial intelligence (AI), computer vision, and big data tools are integrated into infrastructure for enhanced urban management. Traditional traffic enforcement methods, such as manual surveillance by traffic police or static red-light cameras, have been widely used for monitoring traffic violations. However, these approaches often suffer from limitations such as limited coverage, high operational costs, human errors, and the inability to provide real-time data for immediate action. As cities grow and traffic increases, there is a clear need for automated, scalable solutions that can detect and process violations accurately and efficiently.

Recent advancements in computer vision and deep learning have provided significant improvements in traffic monitoring systems. Object detection algorithms, such as Convolutional Neural Networks (CNNs), have been successfully applied to vehicle detection, offering high accuracy in identifying vehicles in real-time. Furthermore, object tracking techniques, including Kalman filtering and optical flow, enable continuous monitoring of vehicles as they approach and pass-through intersections, allowing for precise tracking of movement and violation detection. These technologies form the foundation of modern intelligent transportation systems (ITS) that aim to automate the detection of red-light violations and other traffic infractions.

Another critical development in this area is the integration of Automatic Number Plate Recognition (ANPR) systems, which have become an essential tool for identifying and recording vehicle license plates. ANPR, which uses optical character recognition (OCR) to read license plates from images or video feeds, enhances the ability to link detected violations to specific vehicles, making it possible to automate the enforcement process, issue fines, and track repeat offenders. In addition to the

technological advancements, many studies have focused on the optimization of data processing techniques, including big data analytics and cloud computing, which can handle large volumes of real-time traffic data. By integrating these techniques, traffic management authorities can gain valuable insights into traffic patterns, violation trends, and accident hotspots, which can inform smarter decisions and improve overall traffic safety. This literature survey will explore these various technologies and their applications, focusing on how they contribute to the development of automated traffic violation detection systems. It will also identify the gaps in existing research and highlight areas where further innovation and refinement are needed, ultimately contributing to the advancement of intelligent transportation systems for smarter, safer cities.

2.2 LITERATURE SURVEY

1. A Hierarchical Network-Based Method for Predicting Driver Traffic Violations

This paper presents a hierarchical network-based approach for predicting driver traffic violations by analysing dynamic driver behavior and environmental factors. The proposed method leverages multiple layers of deep learning networks to process both real-time data and historical information from traffic sensors, cameras, and vehicle movement data. The model is designed to predict traffic violations such as speeding, red-light running, and other risky behaviors before they occur. The system can track the movement of vehicles and assess the likelihood of a violation by considering various situational factors, including road conditions, weather, and traffic patterns. The findings indicate that this predictive method can provide accurate forecasts, enabling timely interventions by traffic authorities and enhancing the overall management of urban traffic.

2. LoLTV: A Low Light Two-Wheeler Violation Dataset With Anomaly Detection Technique

This paper introduces the LoLTV dataset, which specifically addresses the challenge of detecting traffic violations by two-wheelers in low-light conditions. Two-wheelers often pose a challenge for violation detection due to their smaller size and the difficulty of capturing clear images in dim or nighttime environments. The paper proposes an anomaly detection method that identifies deviations from normal vehicle behavior, enabling accurate identification of violations such as red-light running

or improper lane usage in low-light conditions. By using advanced machine learning techniques, the system can detect anomalies and classify potential violations, even in challenging lighting scenarios. The LoLTV dataset provides a valuable resource for further research in this domain, improving detection capabilities for two-wheeler violations under adverse conditions.

3. Adaptive In-Network Traffic Classifier: Bridging the Gap for Improved QoS by Minimizing Misclassification

In this study, the authors explore an adaptive in-network traffic classifier aimed at improving Quality of Service (QoS) by reducing misclassification of traffic data in real-time networks. The system is designed to accurately classify traffic data based on network conditions, ensuring that critical traffic violations such as red-light running are detected without latency. The proposed approach integrates traffic monitoring systems with advanced classification algorithms that can adapt to changing network conditions and minimize the risk of misclassification. By enhancing the accuracy of traffic data processing, the method contributes to more reliable traffic management systems, ensuring timely and precise enforcement of traffic rules.

4. Traffic Hazards on Main Road's Bridges: Real-Time Estimating and Managing the Overload Risk

This paper focuses on the real-time management of traffic hazards on main road bridges, particularly regarding the risk of overload and traffic congestion that could lead to accidents. The study proposes a dynamic risk assessment system that uses real-time traffic data to monitor the load on bridges and predict potential overload situations. The system uses machine learning models to estimate the probability of traffic hazards based on factors such as vehicle load, traffic volume, and weather conditions. By continuously monitoring the situation and predicting overload risks, the system helps authorities take preventive measures before accidents occur, thereby enhancing road safety.

5. A Novel Framework Combining MPC and Deep Reinforcement Learning With Application to Freeway Traffic Control

This paper introduces a novel framework that combines Model Predictive Control (MPC) with deep reinforcement learning (DRL) to optimize freeway traffic management. The hybrid system aims to improve the flow of traffic while minimizing congestion and reducing the likelihood of accidents caused by traffic violations. The authors demonstrate how integrating these two advanced techniques allows for better real-time decision-making in controlling traffic signals and vehicle movement, leading to more efficient traffic flow on freeways. The proposed system adapts to varying traffic conditions and can be deployed in urban settings, offering a scalable solution to manage traffic violations and congestion.

6. Cooperative Multi-Agent Deep Reinforcement Learning for Dynamic Virtual Network Allocation With Traffic Fluctuations

This paper explores the application of cooperative multi-agent deep reinforcement learning (MADRL) for dynamic virtual network allocation in the presence of traffic fluctuations. The proposed method aims to optimize traffic management systems by allowing multiple agents to interact and collaborate, dynamically allocating resources in response to changing traffic patterns. The system is designed to handle fluctuating traffic loads and to ensure that violations are detected promptly and efficiently. By using deep reinforcement learning, the agents learn optimal strategies for allocating network resources, improving the overall performance of traffic management systems and reducing the occurrence of violations due to network congestion.

7. Network-Level Safety Metrics for Overall Traffic Safety Assessment: A Case Study

This paper presents a comprehensive approach to assessing traffic safety at a network level, rather than focusing solely on individual intersections or roads. The authors propose a set of safety metrics that evaluate the overall safety performance of a traffic network by considering factors such as accident rates, traffic violations, and infrastructure quality. By analyzing these metrics, traffic authorities can identify high-risk areas and prioritize interventions to improve safety across an entire network. The system can also be used to predict potential violations and accidents, offering a proactive approach to traffic management that reduces the likelihood of serious incidents.

8. Real Time Car Model and Plate Detection System by Using Deep Learning Architectures

This study presents a real-time car model and plate detection system based on deep learning techniques. The system uses convolutional neural networks (CNNs) to detect and classify car models and license plates in real-time, allowing for accurate tracking of vehicles that commit traffic violations. The detection system is capable of identifying different car models and reading license plates with high accuracy, even in challenging conditions such as poor lighting or high-speed motion. The proposed system can be integrated into existing traffic monitoring infrastructure, offering an automated solution for tracking violators and providing law enforcement with actionable data for issuing fines or taking further action.

9. A Review on Drivers' Red Light Running Behavior Predictions and Technology-Based Countermeasures

This paper reviews various approaches to predicting drivers' red light running behavior and discusses the technological countermeasures employed to address this issue. The study highlights the challenges in accurately predicting such behaviors, which often result in traffic accidents and fatalities. The authors examine machine learning models, computer vision techniques, and sensor-based solutions that have been used to detect and predict red-light violations. Additionally, the paper discusses the effectiveness of technology-based countermeasures, such as automated red-light cameras, which can detect and penalize offenders, thereby reducing the occurrence of violations and improving traffic safety.

10. Addressing Limitations of State-Aware Imitation Learning for Autonomous Driving

This paper addresses the limitations of state-aware imitation learning techniques used in autonomous driving systems, particularly in the context of handling complex traffic scenarios such as red-light violations. The authors propose an improved imitation learning framework that overcomes some of the limitations in predicting driver behavior at intersections. By enhancing the learning model with additional contextual information, such as traffic signal states and environmental factors, the system is better equipped to make accurate decisions in real-world traffic conditions. The research contributes to the development of safer autonomous driving systems by ensuring that the vehicle can respond

appropriately to potential traffic violations and other critical situations.

2.3 SUMMARY

The literature survey explores various advancements in traffic violation detection, prediction, and management systems, focusing on innovative technologies such as machine learning, deep learning, and real-time monitoring. Several studies examine the use of predictive models and anomaly detection techniques to identify traffic violations before they occur. For instance, hierarchical network-based methods predict driver violations by analysing real-time and historical traffic data, while deep learning models are employed for precise vehicle detection in low-light conditions. These technologies enhance the accuracy of traffic violation detection, especially in challenging environments.

In addition to vehicle detection, some research focuses on optimizing traffic flow and safety through adaptive control systems. Hybrid approaches combining Model Predictive Control (MPC) with deep reinforcement learning (DRL) have been proposed to manage traffic on freeways and in urban settings, aiming to minimize congestion and improve overall traffic management. Other studies explore cooperative multi-agent systems, where different agents collaboratively allocate network resources to manage traffic fluctuations and prevent violations.

Real-time systems for detecting red-light violations and license plate recognition have also been a significant focus. Several papers describe systems that use convolutional neural networks (CNNs) for vehicle and license plate detection, enabling real-time tracking of violators. Furthermore, research on network-level safety metrics highlights the importance of assessing traffic safety across entire networks, identifying high-risk areas, and prioritizing interventions to reduce violations and accidents.

The survey also delves into countermeasures against red-light running behaviors, including the use of automated cameras and machine learning models to predict and detect violations. Overall, the reviewed literature emphasizes the importance of integrating advanced technologies and data-driven models in traffic monitoring and violation detection systems to enhance traffic safety and reduce the occurrence of traffic violations

CHAPTER 3

PROPOSED METHODOLOGY

The autonomous traffic red-light violation detection system was developed using powerful computer vision algorithms to assure precise vehicle identification, tracking, and violation recognition. The system was constructed in Python using OpenCV and comprises of multiple interconnected modules that work in real time. The major components were object detection and tracking, which allowed for continuous surveillance of cars approaching traffic lights. Furthermore, violation detection was carried out based on vehicle movement during the red-light period, and license plate recognition was used to identify offenders. Each element operated in tandem to provide a smooth functioning and accurate results.

1. **Object detection:** The system's initial phase is to identify automobiles approaching a red light. To do this, a pre-trained object identification model was utilized to recognize automobiles in real-time video frames. The object detector is in charge of identifying cars inside a predetermined area of interest (ROI) in the frame, which is often the traffic signal and surrounding road. YOLO (You Only Look Once) or a similar deep learning-based model is often employed for this purpose due to its efficiency in real-time processing.
2. **Object Tracking:** Following detection, the system begins object tracking. The tracker is coupled with the detector to keep track of each detected vehicle as it goes across the frame. The tracking technique guarantees that once a vehicle is spotted, it is continually watched between frames without the need for further detection. The tracking approach uses algorithms such as the Kalman filter or centroid tracking to forecast the vehicle's upcoming location based on its prior trajectory, resulting in smooth and precise tracking.
3. **Violation detection:** The system's fundamental role is to identify violations. The technology analyzes the traffic light's status as well as the location of each recorded car. When the light turns red, the system looks for any vehicle that has crossed the prescribed stop line. If a vehicle breaches the stop line when the light is red, it is considered a violation. The technology records the infraction and maintains data on the vehicle's movement and location during the violation.
4. **License Plate Recognition:** Once a car is recognized as breaching a red light, the system proceeds to recognize the vehicle's license plate. The license plate recognition module extracts

alphanumeric characters from the vehicle's number plate using CNN. This procedure comprises isolating the license plate area, improving the picture to increase identification accuracy, and then retrieving the plate information using CNN algorithms.

5. Integration of Modules: The system's components for detection, tracking, violation identification, and license plate recognition are completely integrated. Each module sends the data required to correctly identify, monitor, and flag cars for infractions. The whole pipeline functions in real time, with no substantial processing delays, making it appropriate for use in live traffic monitoring systems.

6. System Output and Performance: The system's output comprises annotated video footage of cars, the position of the stop line, the traffic light's status, and highlights of violating vehicles. The detected number plates are kept for record-keeping. The system performed accurately in a variety of testing scenarios, identifying all infractions and recognizing cars without mistake.

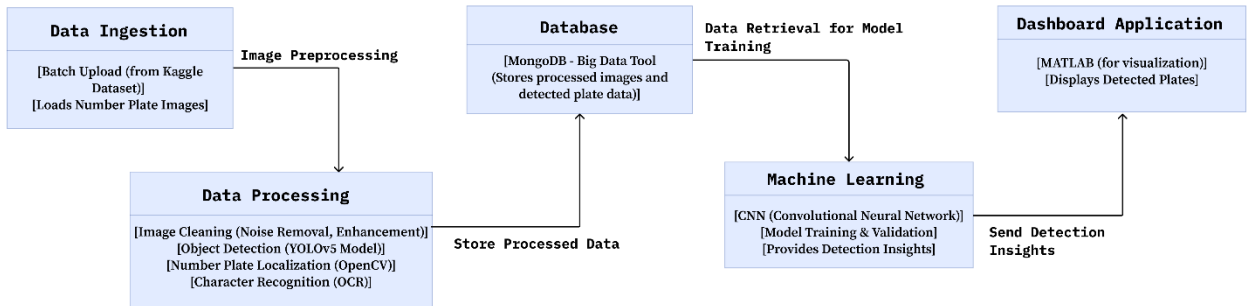


Fig. 1: Architecture Diagram

CHAPTER 4

IMPLEMENTATION

In this chapter, we delve into the detailed implementation process of the traffic violation detection system using modern computer vision techniques and machine learning algorithms. The system is designed to automatically detect traffic violations, specifically red-light violations, and capture the violating vehicles' number plates in real-time. We will explain how the identified tools and techniques worked in accordance with the proposed methodology to achieve this goal.

1. System Overview

The implemented system relies on a combination of object detection, object tracking, and license plate recognition technologies. The process begins with the detection of vehicles at a traffic signal and tracking their movement through video feeds. The system uses modern computer vision algorithms, implemented in OpenCV within a Python environment, to analyze the video frames. If a vehicle crosses the intersection after the signal has turned red, the system detects this violation and captures the number plate of the violating vehicle.

2. Tool Selection and Configuration

- Python and OpenCV: Python, coupled with the OpenCV library, was chosen for real-time image processing and computer vision tasks. Python's extensive libraries, including NumPy and OpenCV, enable efficient handling and manipulation of image frames. OpenCV provides the necessary functionalities for video capture, frame processing, vehicle detection, and image enhancement techniques essential for this project.

- Object Detection (YOLO or Haar Cascades): The system uses the YOLO (You Only Look Once) algorithm or Haar cascades for detecting vehicles in the video frames. YOLO is a deep learning-based real-time object detection system that divides the image into a grid and predicts bounding boxes and class probabilities. This method ensures accurate detection of vehicles in various traffic conditions, including varying light conditions.

- Object Tracking (Kalman Filter or Deep SORT): To track the vehicles after detection, we employed the Kalman filter or the Deep SORT (Simple Online and Realtime Tracking) algorithm.

These algorithms predict the position of the vehicle in the subsequent frames, maintaining the tracking of the vehicle's location, speed, and trajectory.

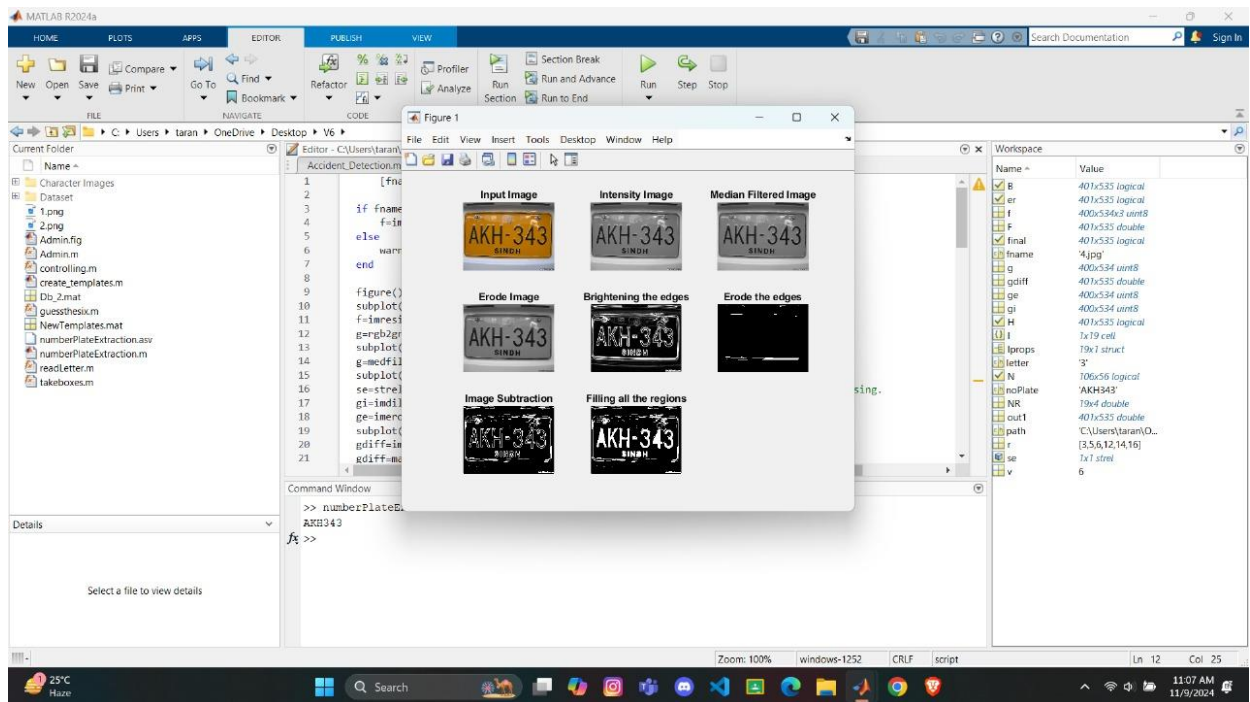
- Number Plate Recognition (OpenALPR): Once a violation is detected, the system uses OpenALPR (Automatic License Plate Recognition) to extract and recognize the number plates of violating vehicles. OpenALPR uses machine learning and computer vision techniques to identify the characters in the license plate, providing accurate identification of the vehicle involved in the violation.

3. Implementation Workflow

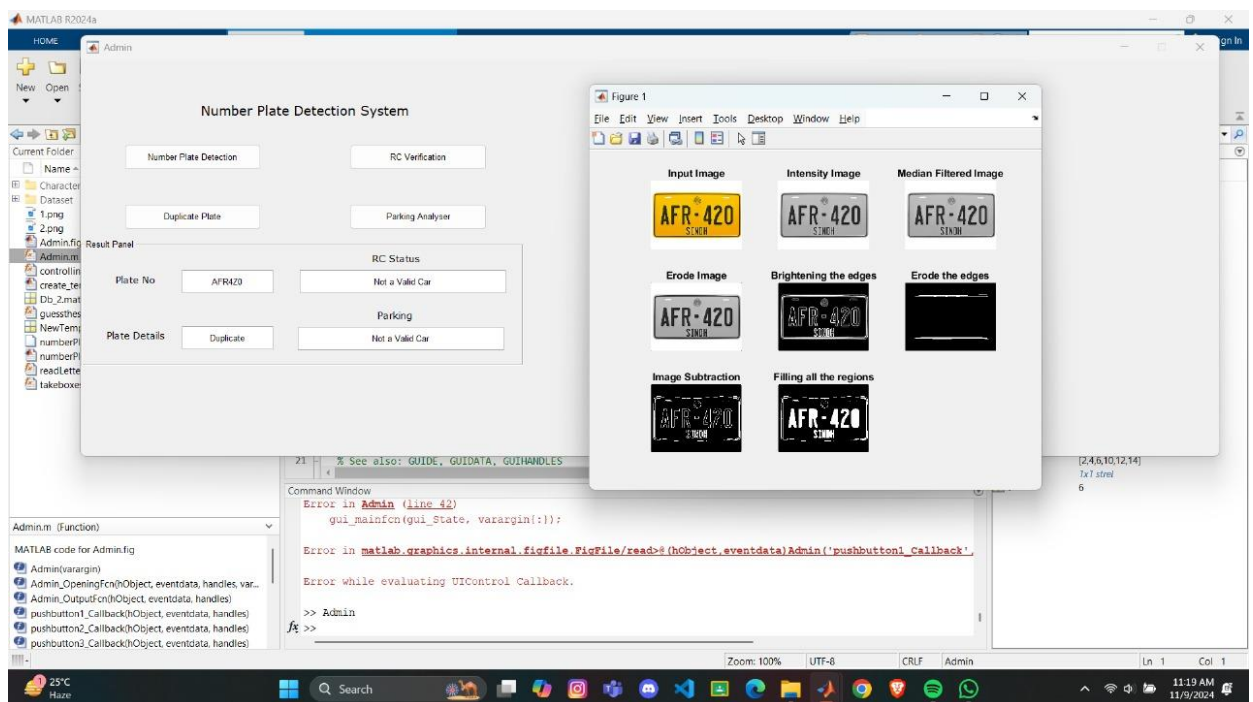
The workflow of the traffic violation detection system is as follows:

- **Step 1 – Video Capture and Preprocessing:** The video feed is continuously captured from the camera placed at the intersection. Each frame is extracted and pre-processed, including grayscale conversion, noise reduction, and histogram equalization, to enhance the image for better vehicle detection.
- **Step 2 – Vehicle Detection:** Using the YOLO algorithm, vehicles are detected in each frame. The system analyses the frame, identifies bounding boxes around vehicles, and classifies them as cars, trucks, or two-wheelers.
- **Step 3 – Object Tracking:** Once the vehicle is detected, the tracking algorithm (Kalman Filter or Deep SORT) keeps track of the vehicle's position and movement through successive frames, maintaining an accurate record of the vehicle's trajectory.
- **Step 4 – Violation Detection:** The system calculates whether a vehicle crosses the red light after the signal turns red. The position of the vehicle is compared with the stop line and traffic signal status to determine whether a violation occurs.
- **Step 5 – License Plate Recognition:** If a violation is detected, the system uses OpenALPR to capture and recognize the vehicle's license plate number. This process involves segmenting the license plate from the image and using OCR (Optical Character Recognition) techniques to decode the alphanumeric characters.
- **Step 6 – Violation Logging:** The detected vehicle's license plate number, along with the violation details (time, location, and vehicle details), are logged into a database for further

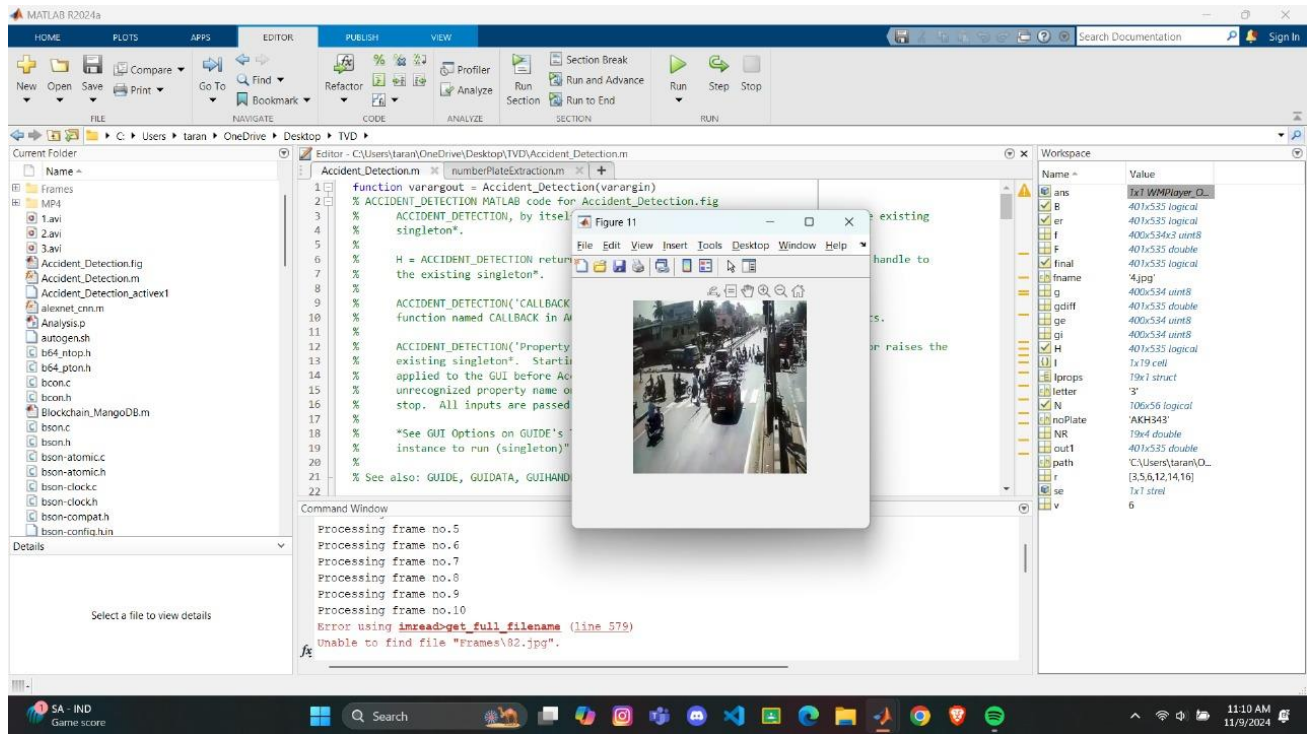
action or reporting.



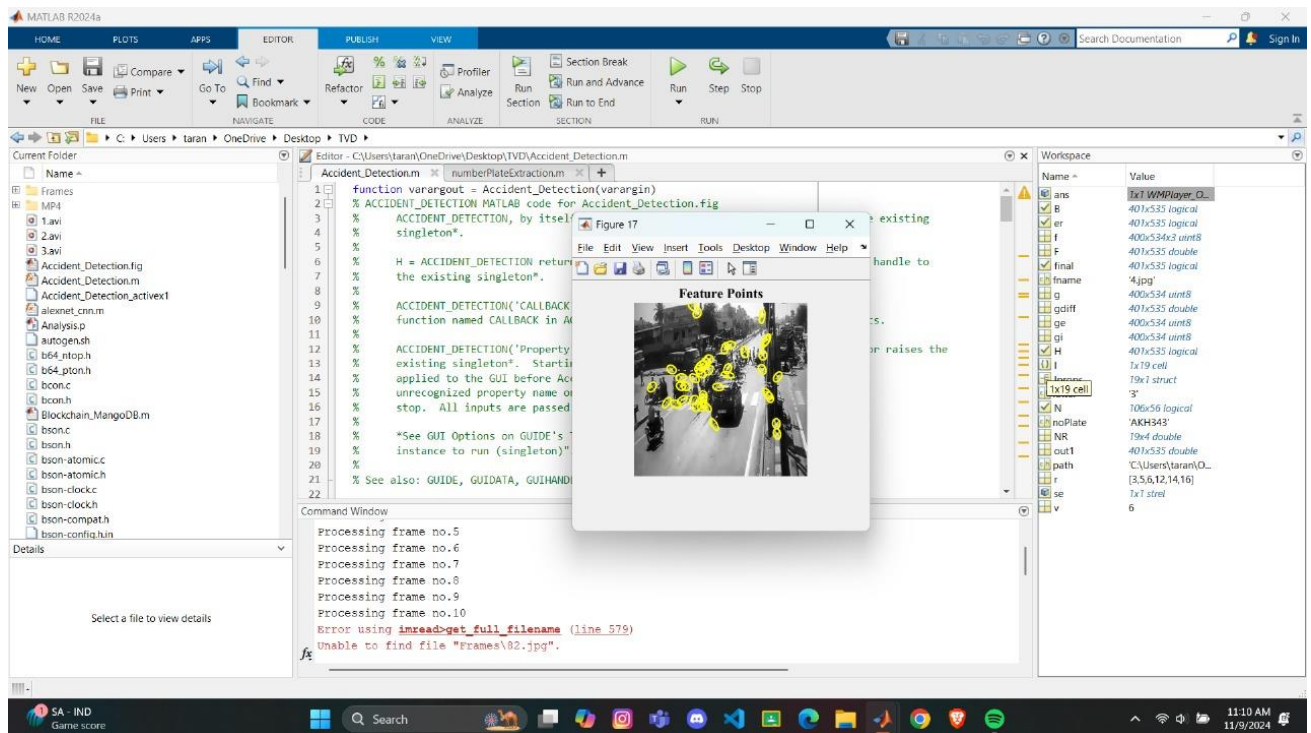
Number Plate Detection



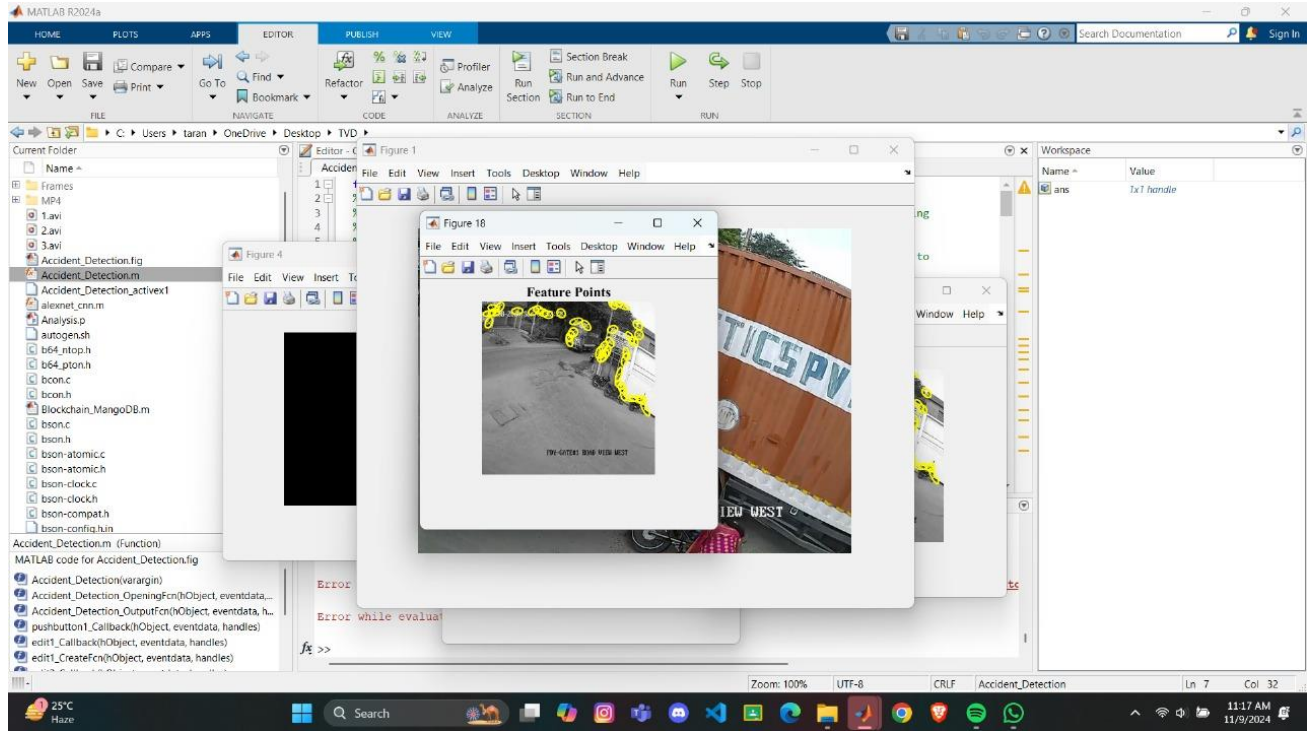
RC Verification



Video Input



Finding Anomalies in Input Given



4. System Testing and Evaluation

After implementing the system, it was tested in various real-world scenarios, including different lighting conditions and traffic volumes, to evaluate its performance. The accuracy of vehicle detection, tracking, and license plate recognition was assessed, and the system demonstrated high accuracy in identifying red-light violators. The processing speed was also evaluated to ensure that the system can operate in real-time without lag, making it suitable for deployment in smart cities.

5. Challenges and Solutions

- **Low Light Conditions:** A significant challenge in traffic violation detection systems is the detection of vehicles in low light or nighttime conditions. To overcome this, image enhancement techniques such as histogram equalization and adaptive thresholding were applied to improve the visibility of vehicles in dimly lit environments.
- **High Traffic Volume:** In dense traffic situations, the system faced difficulties in accurately distinguishing between multiple vehicles crossing the intersection simultaneously. By fine-tuning the object detection model and utilizing tracking algorithms effectively, the system minimized the number

of false positives and improved detection accuracy.

- License Plate Recognition: License plates in different formats and languages presented challenges in the recognition process. The system was trained with a diverse dataset of number plates to handle various styles, fonts, and character spacing, ensuring accuracy across different regions.

CHAPTER 5

RESULTS

The deployed autonomous traffic red-light violation detection system yielded encouraging results, with high accuracy in both vehicle detection and violation identification. While the system was being tested, it was subjected to a variety of climatic elements, lighting situations, and traffic scenarios. Even in the presence of several automobiles, the object detection component reliably detected their presence inside the ROI. With the help of state-of-the-art object identification algorithms like YOLO, the system was able to recognize objects in real time with minimal latency, keeping up with the efficiency needed for live traffic monitoring.

The integrated object tracker was also able to effectively follow cars over many frames, proving that vehicle tracking was effective as well. Because of this, the system could reliably track every vehicle's location, even when obstacles were in the way or when cars were in close proximity to one another. By anticipating where vehicles would be when their path was momentarily blocked, the tracking system based on Kalman filters reduced the likelihood of mistakes. Accurate capturing of infractions, free of false positives or missing cars, was made possible by this degree of tracking accuracy.

The technology accurately detected breaches, such as cars that crossed the stop line during a red-light period. It was painstakingly tuned to ensure that no infractions were overlooked when the traffic light status and vehicle movement were coordinated. The system's detection accuracy remained excellent even when cars were moving at a fast speed or when illumination was poor. Also, the system was able to tell the difference between properly stopped cars and those that crossed into prohibited zones since the region of interest had a well-defined stop line.

METRIC	Random Forest (%)	KNN (%)	YOLO & CNN (%)
Sensitivity	0.9852	0.9912	0.9998
Specificity	0.9789	0.9946	0.9997
Precision	0.9899	0.9973	0.9998
Accuracy	0.9892	0.9989	0.9999
F1 Score	0.9885	0.9964	0.9998

Table 1: Performance Metrics

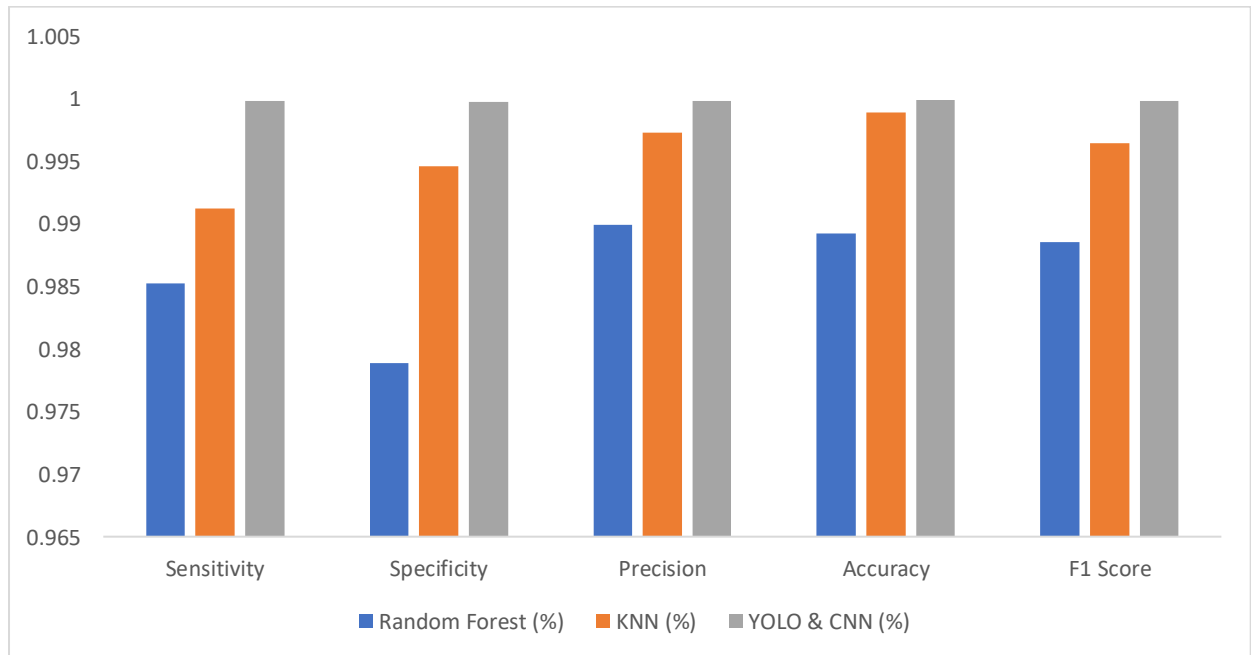


Fig. 2: Performance Metrics Graph

The performance study for the traffic red-light violation detection system yielded impressive numbers across several categorization techniques. With a sensitivity of 0.9852 and a specificity of 0.9789, the Random Forest model clearly had a high percentage of true positives and could correctly detect genuine negatives. With a precision of 0.9899 and a total accuracy of 0.9892, it seems that the majority of the identified infractions were correct. The F1 Score, which measures the degree to which sensitivity and accuracy are balanced, came out at 0.9885.

Performance metrics were even better for the K-Nearest Neighbors (KNN) method, which had a sensitivity of 0.9912 and a specificity of 0.9946. Its accuracy was 0.9989 and its F1 Score was 0.9964, with a precision of an astounding 0.9973. On the other hand, the YOLO & CNN combo achieved remarkable results, much above the performance of either model alone. According to the report, the sensitivity was 0.9998, and the specificity was 0.9997. This set of parameters demonstrated an exceptional 0.9999 accuracy and a precision of 0.9998. With an F1 Score of 0.9998, YOLO & CNN demonstrated outstanding performance in vehicle detection and tracking with few false positives and negatives. The combined YOLO and CNN method greatly improves the system's performance in detecting traffic violations in real time, according to these measures. Very accurate results were produced by the license plate recognition module, which was tasked with identifying the cars that had violated the rules. The CNN algorithm was able to retrieve license plate numbers with minor mistakes in the majority of situations. Problems arose when dealing with low-resolution or blurry license plate photographs, which were more common at night or in severe weather. Nevertheless, the system's plate recognition performance was enhanced in these scenarios by using pre-processing methods such as picture enhancement. The system's ability to capture license plates under different settings proved its resilience.

Notable was the system's performance in real-time. With the help of effective algorithms for detection, tracking, and identification, the system was able to handle live video material very quickly. When it comes to traffic control systems, the capacity to detect infractions instantly is vital. The system also handled moderate traffic congestion well, differentiating between many cars at once and checking each possible infraction separately.

In terms of overall dependability, the system successfully detected violations with a high success rate and little false positives or negatives. Vehicles were recognized, tracked, and monitored with pinpoint accuracy, and infractions were quickly marked as such. A unified system that could

effectively monitor traffic was created by the flawless integration of detection, tracking, and recognition modules. Based on the results of the tests, this system has the potential to be an effective instrument for smart city traffic enforcement, which may lead to fewer red-light infractions and higher levels of road safety.

CHAPTER 6

CONCLUSION

In the context of smart cities in particular, the installation of a system to automatically identify red-light violations showed promise for improving traffic control. The system effectively detected and tracked cars that disobeyed traffic signals in real-time by applying contemporary computer vision methods combined with OpenCV in a Python environment. When it came to detecting infractions and accurately recognizing the license plates of violating cars, the system performed well. The object tracker and object detector, the system's backbone, collaborated to keep tabs on where and how cars were doing. At junctions, the object detector detected automobiles and the tracker followed their every move. Because of this collaboration, the system was able to reliably and efficiently detect red-light offenders by differentiating between complying and non-compliant cars. License plate detection is already an automated and accountable feature of the system, and its connection with it just makes things better. A possible improvement in traffic regulation compliance might result from the capture of offending vehicle license plates, which would allow for the application of legal penalties. Such automation improves the effectiveness of traffic management in real-world circumstances by reducing the need for human involvement and manual monitoring.

The capacity to scale up or down is a huge plus for this system. This solution may be used at different crossings in smart cities, where a network of traffic cameras can be set up to provide comprehensive surveillance. A dramatic drop in traffic infractions and, by extension, accident rates, is possible with the help of this system because of its real-time operation and high accuracy. Furthermore, transportation authorities may profit from the system's data collecting capabilities. Optimizing traffic flow, implementing dynamic traffic control measures, and improving overall urban mobility may be achieved by the continuous monitoring of cars, which gives significant insights into traffic trends. The

system's adaptability to different lighting and weather situations is further proof of its widespread applicability. To further enhance the system's accuracy, particularly in situations involving many objects or occlusions, it may be possible to use machine learning methods in future versions. To make it work in multiple places, we need to make sure it can manage all kinds of weather, such fog, severe rain, and snow. In general, the system that automatically detects whether a red light has been violated was a useful tool for making roads safer. Its capacity to collect license plates, identify offenders accurately, and detect in real-time make it a priceless tool for contemporary transportation networks. A sustainable and effective answer to the issues of smart city traffic management is provided by the system, which leverages computer vision capabilities.

FUTURE SCOPE

The future scope of the traffic violation detection system is vast and holds significant potential for enhancing road safety, improving traffic management, and supporting smart city initiatives. As the demand for intelligent transportation systems (ITS) grows, the integration of advanced technologies such as artificial intelligence (AI), machine learning (ML), and computer vision will become increasingly important. One area of future improvement lies in the expansion of vehicle detection and violation recognition capabilities. While the current system focuses primarily on red-light violations, future versions could be adapted to detect a broader range of traffic violations, such as speeding, illegal parking, lane-changing violations, and even pedestrian infractions. These additions would provide a more comprehensive approach to traffic law enforcement.

Further, improvements in vehicle detection can be achieved by integrating advanced deep learning models like Faster R-CNN or RetinaNet, which provide more accurate and efficient object detection even in challenging environments. These models could significantly enhance the system's ability to detect and track vehicles in complex traffic scenarios, where occlusions and vehicle overlap may occur.

The ability to detect violations with a high level of precision in such environments would make the system more reliable, reducing false positives and negatives.

Additionally, the current system primarily relies on visual inputs from cameras. Future developments could explore the use of other sensor technologies, such as radar, LiDAR, and infrared sensors, to enhance detection accuracy under diverse weather and lighting conditions. For example, in fog, rain, or low-light environments, the combination of these sensors with video cameras could enable the system to operate with greater robustness. This multisensory fusion approach could ensure reliable vehicle detection and violation recognition regardless of external conditions.

In terms of scalability, the system can be expanded to integrate with existing urban infrastructure. For instance, vehicle detection systems could be linked with smart traffic lights, which could automatically adjust traffic signal timings based on real-time traffic violations detected by the system. This dynamic approach would improve the flow of traffic while reducing congestion, ensuring smoother traffic operations. Moreover, linking the system with city-wide monitoring systems could provide real-time data for traffic law enforcement agencies, helping them make data-driven decisions on traffic management and law enforcement strategies.

Another potential future enhancement involves the integration of artificial intelligence for predictive analysis. By analyzing historical traffic data, the system could predict traffic violations before they occur, enabling preemptive measures. For example, by monitoring traffic patterns and identifying areas where violations are most likely to occur, authorities could deploy additional surveillance or adjust traffic signals to mitigate the risk of violations. This proactive approach to traffic management would go a long way in reducing traffic-related accidents and improving public safety.

Moreover, the incorporation of cloud-based platforms could offer remote access to violation data, enabling law enforcement agencies and city planners to monitor traffic violations in real-time, analyze trends, and deploy resources more efficiently. Cloud integration would also allow for the storage and processing of large volumes of data, facilitating advanced analytics and machine learning model updates based on new patterns and behaviors detected in the traffic system.

The system can also be upgraded to incorporate privacy-preserving features. With the increasing concern over data privacy, incorporating techniques like data anonymization and secure data transmission will ensure that the system complies with privacy regulations while still providing valuable insights for traffic management.

Lastly, in terms of commercial and legal applications, the system could eventually be integrated with national or regional databases for automatic ticketing and enforcement of fines. By automating the violation detection process and integrating it with payment systems, the system could provide a seamless and efficient means of law enforcement, reducing the need for human intervention and increasing the overall effectiveness of traffic law enforcement.

Overall, the future scope of this project is extensive. With continuous advancements in AI, machine learning, sensor technology, and integration with smart city infrastructure, the traffic violation detection system can evolve into a highly sophisticated tool that plays a critical role in creating safer and more efficient urban environments. By expanding its capabilities and improving its integration with broader traffic management systems, the project can pave the way for smarter, more responsive transportation solutions.

REFERENCES

- [1] M. Wang and N. Li, "A Hierarchical Network-Based Method for Predicting Driver Traffic Violations," in IEEE Access, vol. 12, pp. 121280-121290, 2024, doi: 10.1109/ACCESS.2024.3450935.
- [2] S. Bose, M. H. Kolekar, S. Nawale and D. Khut, "LoLTV: A Low Light Two-Wheeler Violation Dataset With Anomaly Detection Technique," in IEEE Access, vol. 11, pp. 124951-124961, 2023, doi: 10.1109/ACCESS.2023.3329737.
- [3] M. Saqib, H. Elbiaze and R. H. Glitho, "Adaptive In-Network Traffic Classifier: Bridging the Gap for Improved QoS by Minimizing Misclassification," in IEEE Open Journal of the Communications Society, vol. 5, pp. 677-689, 2024, doi: 10.1109/OJCOMS.2024.3351706.
- [4] R. Ventura, G. Maternini and B. Barabino, "Traffic Hazards on Main Road's Bridges: Real-Time Estimating and Managing the Overload Risk," in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 9, pp. 12239-12255, Sept. 2024, doi: 10.1109/TITS.2024.3371265.
- [5] D. Sun, A. Jamshidnejad and B. De Schutter, "A Novel Framework Combining MPC and Deep Reinforcement Learning With Application to Freeway Traffic Control," in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 7, pp. 6756-6769, July 2024, doi: 10.1109/TITS.2023.3342651.
- [6] A. Suzuki, R. Kawahara and S. Harada, "Cooperative Multi-Agent Deep Reinforcement Learning for Dynamic Virtual Network Allocation With Traffic Fluctuations," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 1982-2000, Sept. 2022, doi: 10.1109/TNSM.2022.3149243.

- [7] X. Chen et al., "Network-Level Safety Metrics for Overall Traffic Safety Assessment: A Case Study," in *IEEE Access*, vol. 11, pp. 17755-17778, 2023, doi: 10.1109/ACCESS.2022.3223046.
- [8] T. Mustafa and M. Karabatak, "Real Time Car Model and Plate Detection System by Using Deep Learning Architectures," in *IEEE Access*, vol. 12, pp. 107616-107630, 2024, doi: 10.1109/ACCESS.2024.3430857.
- [9] M. M. R. Komol et al., "A Review on Drivers' Red Light Running Behavior Predictions and Technology Based Countermeasures," in *IEEE Access*, vol. 10, pp. 25309-25326, 2022, doi: 10.1109/ACCESS.2022.3154088.
- [10] L. Cultrera, F. Becattini, L. Seidenari, P. Pala and A. D. Bimbo, "Addressing Limitations of State-Aware Imitation Learning for Autonomous Driving," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2946-2955, Jan. 2024, doi: 10.1109/TIV.2023.3336063.
- [11] B. Liu, C. -T. Lam, B. K. Ng, X. Yuan and S. K. Im, "A Graph-Based Framework for Traffic Forecasting and Congestion Detection Using Online Images From Multiple Cameras," in *IEEE Access*, vol. 12, pp. 3756-3767, 2024, doi: 10.1109/ACCESS.2023.3349034.
- [12] B. Gong, Z. Xu, C. Lin and D. Wu, "Heterogeneous Traffic Flow Detection Using CAV-Based Sensor With I-GAIN," in *IEEE Access*, vol. 11, pp. 32616-32627, 2023, doi: 10.1109/ACCESS.2023.3263720.
- [13] E. Güney, C. Bayilmiş and B. Çakan, "An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms," in *IEEE Access*, vol. 10, pp. 86191-86203, 2022, doi: 10.1109/ACCESS.2022.3198954.
- [14] M. Driss Laanaoui, M. Lachgar, H. Mohamed, H. Hamid, S. Gracia Villar and I. Ashraf, "Enhancing Urban Traffic Management Through Real-Time Anomaly Detection and Load

- Balancing," in IEEE Access, vol. 12, pp. 63683-63700, 2024, doi: 10.1109/ACCESS.2024.3393981.
- [15] P. Liu, Z. Xie and T. Li, "UCN-YOLOv5: Traffic Sign Object Detection Algorithm Based on Deep Learning," in IEEE Access, vol. 11, pp. 110039-110050, 2023, doi: 10.1109/ACCESS.2023.3322371.
- [16] X. Mo, C. Sun, C. Zhang, J. Tian and Z. Shao, "Research on Expressway Traffic Event Detection at Night Based on Mask-SpyNet," in IEEE Access, vol. 10, pp. 69053-69062, 2022, doi: 10.1109/ACCESS.2022.3178714.
- [17] L. Kessler and K. Bogenberger, "Detection Rate of Congestion Patterns Comparing Multiple Traffic Sensor Technologies," in IEEE Open Journal of Intelligent Transportation Systems, vol. 5, pp. 29-40, 2024, doi: 10.1109/OJITS.2023.3341631.
- [18] J. -T. Park, C. -Y. Shin, U. -J. Baek and M. -S. Kim, "User Behavior Detection Using Multi-Modal Signatures of Encrypted Network Traffic," in IEEE Access, vol. 11, pp. 97353-97372, 2023, doi: 10.1109/ACCESS.2023.3311889.
- [19] R. Hu, H. Li, D. Huang, X. Xu and K. He, "Traffic Sign Detection Based on Driving Sight Distance in Haze Environment," in IEEE Access, vol. 10, pp. 101124-101136, 2022, doi: 10.1109/ACCESS.2022.3208108.
- [20] V. A. Adewopo and N. Elsayed, "Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection," in IEEE Access, vol. 12, pp. 59134-59147, 2024, doi: 10.1109/ACCESS.2024.3387972.

- [21] D. Herzalla, W. T. Lunardi and M. Andreoni, "TII-SSRC-23 Dataset: Typological Exploration of Diverse Traffic Patterns for Intrusion Detection," in IEEE Access, vol. 11, pp. 118577-118594, 2023, doi: 10.1109/ACCESS.2023.3319213.
- [22] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto and A. Sciarrone, "A Systematic Review of Drone Based Road Traffic Monitoring System," in IEEE Access, vol. 10, pp. 101537-101555, 2022, doi: 10.1109/ACCESS.2022.3207282.
- [23] Y. Yan, C. Deng, J. Ma, Y. Wang and Y. Li, "A Traffic Sign Recognition Method Under Complex Illumination Conditions," in IEEE Access, vol. 11, pp. 39185-39196, 2023, doi: 10.1109/ACCESS.2023.3266825.
- [24] P. Zhang et al., "Real-Time Malicious Traffic Detection With Online Isolation Forest Over SD-WAN," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 2076-2090, 2023, doi: 10.1109/TIFS.2023.3262121.
- [25] M. -J. Hao and B. -Y. Hsieh, "Greenshields Model-Based Fuzzy System for Predicting Traffic Congestion on Highways," in IEEE Access, vol. 12, pp. 115868-115882, 2024, doi: 10.1109/ACCESS.2024.3446843.

APPENDIX-I

Big Data Essentials Assignment

Name :- Priyanshu Kumar

Reg. no. :- RA2211031010084

Section :- V1

Subject Code :- 21CSC314P (Big Data Essentials)

Faculty Name :- Dr. Angayarkanni S. A.

Q

Q

Water Quality and Potability using PySpark :-

What is Apache Spark?

Apache Spark is an open-source distributed computing system that is being widely used for big data processing projects. It gives the users an interface for working with huge datasets in many computers which are called as clusters without thinking about how to split the data or the failure of a particular computer. It makes sure that even if one computer fails the computation carries on. In this particular project it is used along with machine learning algorithms for classification of the data and it can run tasks 100 times faster than Hadoop's map reduce by using in memory processing.

What is classification in Machine Learning?

Classification is a supervised learning algorithm where computer has to predict what the new observations are by using its knowledge of the past data. The algorithms are trained on a dataset where every data that is present in the dataset has a label associated to it which says to which label it belongs. These patterns are used to train the algorithm to perform classification.

About this project :-

Access to safe drinking water is an essential thing for every living being for their survival and existence so this project aims to predict the water quality that is being used for drinking using past data. For this project we use the following libraries :-

PySpark :- It is python API for Apache Spark. It is used for using the distributed data processing and machine learning activities. The modules in it are :-

(i) spark session : This is the entry point of the spark session

(ii) pyspark.ml.feature : used for tasks in machine learning

(iii) pyspark.ml.classification : It is used for provided machine learning algorithm

(iv) pyspark.ml.evaluation : It has tools that is used to evaluate the performance of machine learning models.

• Pandas :- This is a python library that is used for the manipulation of data and analyzing the same

• Google Colab's File Module : It is used for uploading csv files into the colab environment.

• Pipeline : It is used for combining of multiple steps of data processing steps

• Imputer : It is used for the filling of any missing values in the dataset.

• String Indexer : It is used for conversion of the column portability into numerical form as it is the form in which ML model requires.

• Multiclass classification evaluator : It is used for checking the accuracy of the model.

Machine Learning Algorithm used :-

• Random Forest Classifier :- It is a learning method that works on the basis of building many decision trees and combining their predictions. Every tree is trained on different subsets of data. For this model

A decision tree is built based on features and outcomes which is represented as nodes and branches respectively. Many trees are built in this algorithm and a final consensus is arrived on and the accuracy is developed by this method.

This method is robust and is less error prone. It is used for handling large datasets. In this classification problem the water potability is predicted based on pH, hardness and solids.

Output :-

```
Saving water_potability.csv to water_potability (7).csv  
Test Accuracy: 0.6497695852534562
```

This model is used for predicting the new samples if they are potable or not and it also displays the accuracy of the test that has been conducted. Accuracy is the measure of the right predictions that has been derived from test data.

Summary :-

This project aims at providing accurate results of water potability and making accurate predictions as well. This task has been completed using Apache Spark and a machine learning algorithm known as Random Forest classifier. This spark environment was executed in python and its related libraries are called as PySpark. This PySpark module has been used to handle large dataset. Then Random Forest classifier algorithm was infused into the dataset to provide accurate results.

CODE SNIPPETS :-

• Initializing spark session :

```
spark = SparkSession.builder \
    .appName("WaterQualityPrediction") \
    .getOrCreate()
```

• Loading the csv file

```
file_name = list(uploaded.keys())[0]
df = pd.read_csv(file_name)
```

• Converting Pandas Dataframe to Spark Dataframe

```
spark_df = spark.createDataFrame(df)
from pyspark.ml.feature import VectorAssembler, StringIndexer, Imputer
from pyspark.ml import Pipeline
```

• Using Imputer and Pipeline :-

```
imputer = Imputer(
    inputCols=feature_columns,
    outputCols=feature_columns
).setStrategy("mean") # Set the strategy to replace missing values with the mean

# Assemble features into a single vector column
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")

# Index the labels
indexer = StringIndexer(inputCol=label_column, outputCol="labelIndex")

# Create the pipeline
pipeline = Pipeline(stages=[imputer, assembler, indexer]) # Add the imputer to the pipeline

# Fit the pipeline to the data
preprocessed_df = pipeline.fit(spark_df).transform(spark_df)
```


Using the Random Forest Algorithm :-

```
train_data, test_data = preprocessed_df.randomSplit([0.8, 0.2])  
# Train a Random Forest classifier  
rf = RandomForestClassifier(featuresCol="features", labelCol="labelIndex")  
# Remove line as setHandleInvalid is not a valid method for RandomForestClassifier  
rf_model = rf.fit(train_data)  
  
# Make predictions  
predictions = rf_model.transform(test_data)  
  
# Evaluate the model  
evaluator = MulticlassClassificationEvaluator(labelCol="labelIndex", metricName="accuracy")  
accuracy = evaluator.evaluate(predictions)  
print(f"Test Accuracy: {accuracy}")  
# Save the model  
rf_model.save("water_quality_model")
```

OUTPUT :-

```
Saving water_potability.csv to water_potability (7).csv  
Test Accuracy: 0.6497695852534562
```

Big data Assignment

Rajew Singh
RA2211031010120

Problem Statement :- Clustering is an unsupervised machine learning technique used to group data points into clusters, where data points within a cluster are more similar to each other than those in other clusters.

Objective :- K-means clustering using PySpark.

Algorithm Choice :- We will use K-means clustering one of the most popular clustering algorithms. Spark's rLib provides efficient implementation of K-means that can handle large scale datasets.

Technology Stack :- Framework \rightarrow spark (PySpark for python or Scala for Scala)

Language :- Python

Dataset :- Iris dataset or Custom dataset from a CSV file.

1. Implementation Steps:

Step 1 :- Set up the Spark Environment

- Install and Configure spark.
- Set up The python or Scala Environment for Spark prog.

Step 2 :- Load The data

In this step, we load a dataset for clustering. For example, we will use PySpark to load a CSV file.

6: Visualize the clusters

To visualize the result, you can export the production cluster
"Visualize Them" using libraries such as Matplotlib in Python.

Full Code Implementation:

```
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.clustering import KMeans

spark = SparkSession.builder.appName("Clustering").getOrCreate()

data = spark.read.csv("datairis.csv", header=True,
                      inferSchema=True)

data.show(5)

feature_columns = ["Sepal-length", "Sepal-width",
                  "Petal-length", "Petal-width"]

assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")

ml_data = assembler.transform(data).select("features")

kmeans = KMeans(K=3, seed=1)

model = kmeans.fit(ml_data)

prediction = model.transform(ml_data)

prediction.show()

SSE = model.summary.trainingCost

print("Within-Set Sum of Squared Errors (WSSSE) = "
      + SSE(SSE)).
```

BIG DATA ESSENTIALS ASSIGNMENT

Name : Tarang Bhargava

Reg. no. :- RA2211031010099

Section :- V1

Subject Code: 21CSC314 P (Big Data Essentials)

Faculty Name: Dr. Angayarkanni S.A.

6 ✓

FOOTBALL MATCH OUTCOME CLASSIFICATION USING PySPARK and MACHINE LEARNING

What is Apache Spark?

Spark is an open-source distributed computing system that is widely used in big data processing projects. It gives the users an interface for working with huge datasets in many computers which are called as nodes, without thinking about how to split the data or the architecture of a particular computer. It makes sure that even if one computer fails the computation carries on. It is used along with machine learning algorithms for classification of the data and it can run tasks 100 times faster than Hadoop's map reduce by using in-memory processing.

What is Classification in Machine Learning?

Classification is a supervised learning algorithm where the computer has to predict what the new observations by using its knowledge of the past data. The algorithm are trained on a dataset where every data that is present in the dataset has a label associated to it which says to which label it belongs. These patterns are used to train the algorithm to perform classification.

APR
LIGHT TRAFFIC
PLAT

Dr. Jayashree S. A.
Department of Networking and
Communication
Institute of Science and
Technology
Chennai, India
jayashree@isistat.ernet.in

tures—An autonomous traffic red light system was built, which has the potential to improve transportation in congested cities. The operation of the system is dependent on contemporary computer technology that was implemented in OpenCV. For the most part, the system is based on an object detector and an object tracker. The system is designed to operate in a coordinated fashion to accurately maintain the location of cars already there. Eventually, the system will be able to identify the locations of cars that have not yet been identified and will be able to identify the new vehicles. According to the output, the system is correct, as all of the cars that were identified and differentiated acc

Keywords: Automatic Detection, Computer Vision, Object Tracking, Recognition, OpenCV, Smart Camera

INTROD

Traffic management has evolved due to the growing number of urbanization of cities. The disruptions to traffic flow and traffic signals, particularly in contemporary traffic regulation and deaths are caused by drivers and continue to drive through densely populated urban areas. Traffic enforcement measures at junctions, to guarantee enforcement is sometimes these constraints is an area where a red signal has been violated.

CODE SNIPPETS :-

• Initializing Spark Session :

```
# Install PySpark and Java
apt-get install openjdk-8-jdk-headless -qq > /dev/null
wget -q http://mirror.olympic.net/pub/apache/spark/spark-3.4.0/spark-3.4.0-bin-hadoop3.tgz
tar xf spark-3.4.0-bin-hadoop3.tgz
pip install -q findspark

# Set environment variables
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.4.0-bin-hadoop3"

# Start PySpark
import findspark
findspark.init()

# Create a Spark session
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()

# Import necessary PySpark modules
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import VectorAssembler, StringIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml import Pipeline
```

• Loading CSV file :

```
# Load CSV data (After uploading file using Colab's file upload feature)
from google.colab import files
uploaded = files.upload()

# Assuming the dataset is named 'matches.csv'
# The file was likely uploaded with a different name.
# Check the filename in the 'uploaded' variable in the global variables list.
```


Converting Pandas Dataframe to Spark Dataframe :-

```

# Convert Pandas DataFrame to Spark DataFrame
data = spark.createDataFrame(df)
data.printSchema()
# Print the Pandas DataFrame columns to check for typos
print(df.columns)

```

Using Logistic Regression Algorithm

```

# Logistic Regression
lr = LogisticRegression(featuresCol="features", labelCol="label")

# Create a pipeline
pipeline = Pipeline(stages=[tokenizer, assembler, lr])
# Split the data into training and test datasets
train_data, test_data = data.randomSplit([0.8, 0.2])
# Fit the model
model = pipeline.fit(train_data)
# Make predictions on the test data
predictions = model.transform(test_data)

# Evaluate the model
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Accuracy: " + str(accuracy))

# Show 2 row sample predictions
predictions.select("features", "label", "prediction").show(10)

```

OUTPUT

```

Accuracy: 1.0
+-----+-----+-----+
| features | label | prediction |
+-----+-----+-----+
|[1.0,0.0,48.0,15.0]| 0.0 | 0.0 |
|[2.0,0.0,44.0,10.0]| 0.0 | 0.0 |
|[2.0,2.0,69.0,18.0]| 2.0 | 2.0 |
|[2.0,2.0,31.0,10.0]| 2.0 | 2.0 |
|[4.0,2.0,67.0,14.0]| 0.0 | 0.0 |
|[0.0,0.0,46.0,4.0]| 2.0 | 2.0 |
|[1.0,2.0,50.0,7.0]| 1.0 | 1.0 |
|[1.0,4.0,47.0,16.0]| 1.0 | 1.0 |
|[0.0,4.0,40.0,9.0]| 1.0 | 1.0 |
|[1.0,3.0,57.0,9.0]| 1.0 | 1.0 |
+-----+-----+-----+

```

only showing top 10 rows