Optimizing Urban Mobility: A Comprehensive Approach to Dynamic Traffic Flow Management

Suvankar Dash
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha
suvankardash0103@gmail.com

Shashank Dewangan
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha
shasnk2267@gmail.com

Bhabani Shankar Prasad Mishra
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha
bsmishrafcs@kiit.ac.in

Rahul Biswas
School of Electronics Engineering
KIIT University
Bhubaneswar, Odisha
biswasrahul016@gmail.com

Subhashree Mishra
School of Electronics Engineering
KIIT University
Bhubaneswar, Odisha
subhashree.mishrafet@kiit.ac.in

Saahen Sriyan Mishra
School of Computer Engineering
KIIT University
Bhubaneswar, Odisha
saahenmishra@gmail.com

***Abstract*—This study describes a novel dynamic traffic flow management system built for urban intersections to improve vehicle navigation and control. Using YOLOv8 for real-time vehicle recognition and categorization, the system efficiently prioritises emergency vehicles. Integration with the Easy-OCR Object Detection model makes number plate detection easier, and subsequent verification against the National Informatics Centre (NIC) database ensures conformity to emergency requirements. Advanced routing algorithms (A*, Dijkstra's, and BFS) are used to optimise network flows and reduce congestion globally, regardless of external mapping services such as Google Maps. This self-sufficient system demonstrates a scalable and secure solution to current traffic management.**

***Index Terms*—YOLOv8, EasyOCR, IoT, V2V/V2I, A*, Dijkstra, CNN, Path-Finding.**

I. Introduction

Urban traffic management is undergoing a crucial transformation from traditional static systems to dynamic, intelligent models due to the escalating demands and complexities associated with growing urban populations. This shift is primarily driven by the integration of Artificial Intelligence (AI) and Machine Learning (ML), which offer sophisticated tools to enhance the efficiency and safety of vehicular navigation across various traffic scenarios. As cities worldwide face challenges such as increasing congestion, pollution, and infrastructural demands, AI applications in traffic systems emerge as vital solutions to these issues as visualised in Fig 1.

AI and ML are harnessed to develop adaptive traffic management systems that leverage real-time data and advanced analytics to respond promptly to fluctuating traffic conditions. These technologies not only aim to improve traffic flow and road safety but also present a significant evolution over conventional traffic management approaches. For example, contemporary AI-enabled traffic systems utilize extensive sensor and camera networks embedded within urban infrastructures to adjust traffic signals dynamically and predict potential bottlenecks before they lead to congestion [2].

Furthermore, the application of advanced deep learning models such as Convolutional Neural Networks (CNN's) and Recurrent Neural Networks (RNNs) facilitates the automated detection and classification of vehicles. This capability is essential for implementing priority-based traffic controls, allowing for the differentiation between emergency and regular vehicles to streamline traffic flow efficiently. The integration of YOLOv8, a state-of-the-art object detection system, exemplifies this innovative approach by enhancing the accuracy and speed of vehicle detection, which is crucial for the operational success of real-time traffic management systems [3].

The synergy between ML and the Internet of Things (IoT) significantly augments the effectiveness of traffic management systems. This integration enables the continuous and comprehensive monitoring of traffic conditions, thereby supporting sophisticated models that proactively adjust traffic controls based on predictive traffic volume analyses [1].

However, the adoption of these advanced technologies is not without challenges. Issues such as ensuring data privacy, securing networks, and addressing potential biases in AI algorithms are paramount to the ethical application of these technologies in public domains. Moreover, these systems require ongoing adaptation and fine-tuning to meet the unique and evolving needs of urban traffic environments.

Looking forward, the role of AI in traffic management is set to increase, driven by technological advancements and a heightened focus on sustainable urban development. Continued research and development in this field are essential to address existing limitations and fully realize the potential of AI-enhanced traffic management systems. With diligent development and strategic implementation, AI-driven solutions are poised to transform urban mobility, making it more efficient, safer, and better suited to the demands of modern cities.

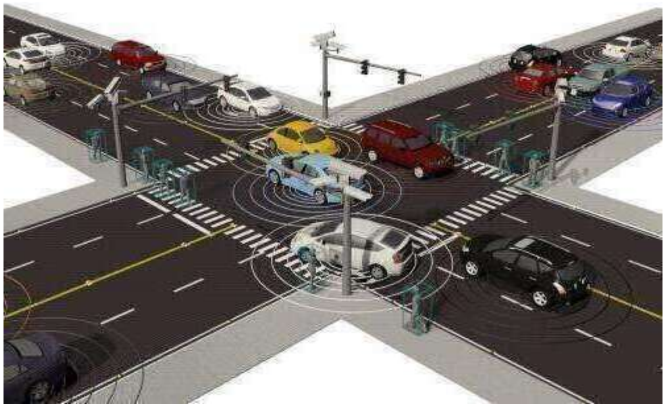


Fig. 1: Physical Visualization/Simulation of Smart V2I communications

II. Problem Statement

Urban traffic congestion poses a formidable challenge, severely disrupting vehicular flow and, crucially, the operational efficiency of emergency services. Conventional traffic management systems, characterized by their static and rigid frameworks, are ill-equipped to meet the dynamic demands of contemporary urban environments. These systems frequently fail to prioritize vehicles strategically based on urgency, exacerbating congestion during peak traffic periods and emergency situations. Such inefficiencies critically delay emergency response times, underscoring the urgent need for an adaptive traffic management solution.

This project proposes a focused initiative to refine traffic flow specifically for emergency vehicles. By prioritizing this segment, the project aims to develop and test innovative traffic management methodologies and technologies that can subsequently be adapted to broader applications. This strategic approach not only seeks to enhance the efficiency of emergency responses but also serves as a foundational model for comprehensive traffic system overhauls. The successful implementation and validation of this system are anticipated to offer substantial improvements in managing urban traffic flows, thereby serving as a pivotal proof of concept for scalable traffic management solutions.

This targeted initiative provides a pragmatic and impactful entry point for demonstrating the capabilities of advanced traffic management systems, with the potential for extensive applicability in enhancing general urban mobility. As such, it aligns with the current research and development trajectories in smart urban infrastructure, offering significant contributions to the fields of traffic engineering and urban planning. The findings from this project are expected to catalyze further innovations and foster broader implementations of intelligent traffic management systems globally.

A. Project Planning Phases

The project planning adheres to a rigorous and structured methodology, aimed at ensuring the successful deployment of an advanced intelligent traffic management system. Each component of the project is carefully defined and executed to meet the sophisticated demands of modern urban traffic systems:

1) Requirement Identification: Central to this project is the goal of enhancing urban traffic management through the implementation of Vehicle-to-Infrastructure (V2I) communication technologies. The immediate focus of this initiative is the development of software capabilities specifically designed for the detection and rerouting of emergency vehicles. This focused approach facilitates the intensive development and testing of key functionalities critical for optimizing emergency response efficiency across urban landscapes.

2) Objective Formulation: The principal objective of this initial phase is to engineer and rigorously validate a software framework that supports the precise detection and strategic rerouting of emergency vehicles. Prioritizing emergency scenarios enables the project to target and demonstrate measurable enhancements in response times, providing empirical evidence of efficacy. The successful deployment and validation of this system are intended to lay a robust foundation for the subsequent expansion of these technologies to general traffic management applications.

3) Resource Allocation: While the current phase of the project is predominantly centered on software development and does not necessitate a broad allocation of physical resources, substantial computational resources are indispensable. The development, training, and deployment of Convolutional Neural Networks (CNN's) to process and analyze complex traffic datasets demand an advanced computing infrastructure. Consequently, the project will leverage either cloud-based computing services or on-premises servers equipped with high-performance GPUs and CPUs. These resources are essential to support the computationally intensive demands of developing machine learning models and managing real-time traffic data processing.

This meticulously crafted planning framework is designed to align all project activities with the overarching strategic objectives of significantly enhancing emergency response capabilities through cutting-edge traffic management technologies. By initially concentrating on emergency vehicle scenarios, the project establishes a focused and impactful benchmark for broader traffic system enhancements, setting a precedent for future scalability and adaptability in traffic management solutions.

III. System Requirements and Data Specifications

A. Hardware Requirements

The hardware specific requirements for our project are as given below.

1) Sensors : We would require some IoT related sensors for the project like a high-resolution camera for reading and detecting the vehicles from the road, and also some sensors will be required for V2V and V2I communications.

2) Computing Devices : We will require micro-computers to be implemented at the end devices/sensors for performing the decision tasks. These will be generally SoC or NoC IoT Devices used for V2V/V2I Communications.

3) Processors : We will require some high levels of compute-intensive CPUs and GPUs for training model with a large dataset.

B. Software Requirements

The software specific requirements for our project are as given below.

1) Programming Language: We have used Java and Python as the base programming language in project.

2) Libraries Used: OpenCV, Numpy, JavaFX.

3) Models Used: YOLOv8, EasyOCR.

4) Algorithms Used: A*, Dijkstra, and BFS algorithm to frame the shortest path between two given end points.

C. Datasets

We have compiled an extensive collection of datasets from publicly available sources, with the 'Emergency Vehicle Detection Computer Vision Project Dataset' [6] being particularly significant. Additionally, we have augmented these resources with a custom dataset specifically curated for the enhancement of emergency vehicle detection capabilities.

IV. System Design / Working Implementation

This section elaborates on the sophisticated implementation of a dynamic traffic flow management system, meticulously designed to optimize urban traffic by integrating advanced computational models with data-driven verification processes. This system encompasses a four-tiered framework, as visually demonstrated in Fig 2 that includes YOLOv8 for emergency vehicle detection, Easy-OCR for license plate recognition, the National Informatics Centre (NIC) database for vehicle verification, and advanced path-finding algorithms to facilitate efficient route optimization.

A. Emergency Vehicle Detection Using YOLOv8

The initial phase of our deployment involves the YOLOv8 model, which is leveraged for real-time detection of emergency vehicles. As the latest evolution in the YOLO series, YOLOv8 is celebrated for its rapid and precise object detection capabilities, making it exceedingly suitable for real-time applications within traffic management systems. This model operates on a rigorously curated custom dataset sourced from a variety of open-source repositories, ensuring a comprehensive and representative training foundation. The primary objective here is to identify emergency vehicles instantaneously from live video feeds at urban intersections, marking the first step in a dual-stage verification process aimed at enhancing the prioritization of emergency responses within urban traffic management strategies [7], [9].

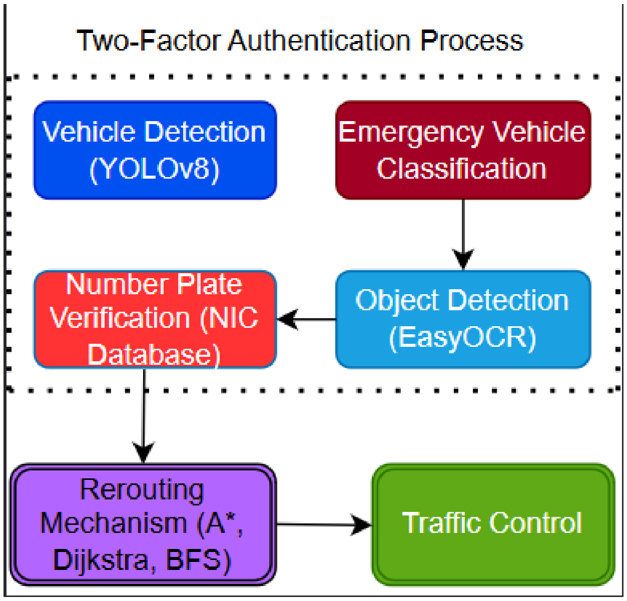

Two-Factor Authentication Process
Vehicle Detection (YOLOv8)
Emergency Vehicle Classification
Number Plate Verification (NIC Database)
Object Detection (EasyOCR)
Rerouting Mechanism (A*, Dijkstra, BFS)
Traffic Control

Fig. 2: System Design Interpretation

B. Easy-OCR for License Plate Recognition

Subsequent to vehicle detection, the Easy-OCR tool is utilized to perform real-time text extraction from vehicle license plates. Easy-OCR's versatility in supporting multiple languages and scripts makes it an ideal choice for accurately recognizing license plates under diverse operational conditions. This functionality forms the second segment of the verification process, where OCR results are further analyzed to validate vehicle identities and check their registration statuses against set criteria [8].

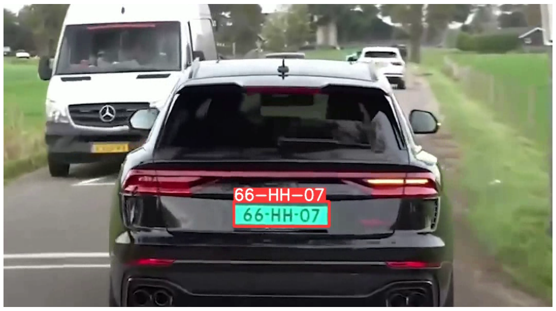


Fig. 3: Easy-OCR Implementation

C. NIC Database Integration

We have are using it for Two-Step Verification process for more safety and confirming the vehicles. Integrating the NIC database is a pivotal step for the real-time verification of vehicles detected through YOLOv8 and Easy-OCR. The NIC database acts as a comprehensive repository of vehicle registration data, facilitating the verification of whether a detected vehicle meets the emergency criteria based on its registration details. This integration is essential for enabling

TABLE I: YOLO Model Benchmarks

Models	mAP (COCO)	AP@0.5 (COCO)	Speed on AGX Orin (FPS)	Speed on RTX 4070 Ti (FPS)
YOLOv5s	37.4	56.8	277	877
YOLOv5m	45.4	64.1	160	586
YOLOv5l	49.0	67.3	116	446
YOLOv5x	50.7	68.9	67	252
YOLOv7-tiny	37.4	55.2	290	917
YOLOv7	51.2	69.7	115	452
YOLOv7x	52.9	71.1	77	294
YOLOv8n	37.3	52.5	383	1163
YOLOv8s	44.9	61.8	260	925
YOLOv8m	50.2	67.2	137	540
YOLOv8l	52.9	69.8	95	391
YOLOv8x	53.9	71.0	64	236

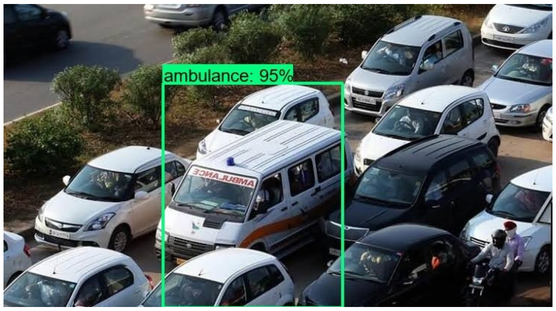


Fig. 4: Emergency Vehicle Detection

prioritized traffic management for emergency vehicles, thereby enhancing their ability to traverse urban environments more efficiently during critical situations [11].

D. Path-finding Algorithm

We have used Breadth First Search (BFS), Dijkstra's and A* algorithm here in our project work. The deployment of sophisticated path-finding algorithms constitutes the final aspect of our system's implementation. The A* algorithm is employed for its heuristically-enhanced efficiency in discovering the shortest paths, whereas Dijkstra's algorithm is utilized for its straightforwardness and effectiveness in navigating weighted networks [5]. Additionally, the Breadth-First Search (BFS) algorithm is applied due to its proficiency in exploring unweighted graphs, offering substantial benefits in specific traffic scenarios. Collectively, these algorithms underpin the system's dynamic traffic rerouting capabilities, significantly ameliorating urban traffic congestion [4], [12].

V. Results Analysis

A. YOLOv8 for Emergency Vehicles Detection

This analysis section elucidates the rationale for selecting the YOLOv8 model for real-time detection of emergency vehicles at urban intersections within our comprehensive traffic management system. The decision is substantiated by a rigorous comparative evaluation of the model's performance against its predecessors, YOLOv5 and YOLOv7, across two hardware configurations—NVIDIA Jetson AGX Orin and RTX 4070 Ti. The evaluation criteria focused on Mean Average Precision (mAP) across varied Intersection over Union (IoU) thresholds and the models' operational speeds, quantified in frames per second (FPS).

1) Comparative Performance Evaluation: To identify the most suitable YOLO model for our application, a detailed analysis was conducted. The models assessed included YOLOv5, YOLOv7, and YOLOv8. The primary metrics for evaluation were mAP and operational speed across different IoU thresholds. Performance data compiled from a Stereolabs study [10] in Table I, highlights the advancements incorporated in YOLOv8.

2) Performance metrics defined:

- *Mean Average Precision (mAP):* This metric evaluates the model's precision across multiple thresholds, providing an average score that reflects its overall ability to identify correct objects within the images accurately.
- *Average Precision at IoU=0.5 (AP@0.5):* This specific metric calculates the precision at an IoU threshold of 0.5, offering insight into the model's performance at this moderate level of detection strictness.
- *Intersection over Union (IoU):* IoU quantifies the accuracy of an object detector on a particular dataset, measured by the overlap between the predicted bounding boxes and the ground truth.

3) Justification for the Selection of YOLOv8: The deployment of YOLOv8 was driven by its superior performance indicators compared to its predecessors. Notably, the high-end configurations of YOLOv8, such as YOLOv8l and YOLOv8x, not only delivered the highest mAP scores but also maintained processing speeds viable for real-time applications:

- *Accuracy and Speed Trade-off:* The YOLOv8 variants strike an optimal balance between high detection accuracy and efficient processing speed, ensuring robust performance in real-time operational settings.
- *Real-Time Processing Capability:* The YOLOv8 model variants, particularly those with lower resolutions like YOLOv8n, achieve processing speeds up to 383 FPS, thereby facilitating the swift detection of emergency

vehicles essential for prompt traffic management interventions.

Integrating YOLOv8 into our traffic management system significantly augments the detection capabilities and operational efficiency, ensuring rapid and accurate identification of emergency vehicles. This enhancement is pivotal for improving response times and optimizing traffic flow across urban settings.

B. Analysis of Path-finding Algorithms: Comparative Analysis with Google Maps and Apple Maps

In this segment of our analysis, we embarked on an empirical analysis to evaluate the efficiency of various path-finding algorithms implemented within our proprietary project, specifically comparing them against established route suggestion systems such as Google Maps, Apple Maps and Bing Maps. The starting point for our analysis was the KP 5 hostel at KIIT Bhubaneswar, with the destination set as TCS Bhubaneswar. This setting allowed us to assess the practical applicability of our algorithms under controlled conditions.

1) Methodology: As demonstrated in Fig 5, we utilized screenshots from Google Maps, Apple Maps, Bing Maps, and our own project's outputs to visualize and compare the suggested routes. Our project incorporates three distinct algorithms: Breadth-First Search (BFS), Dijkstra's, and A*. Each algorithm's route output was documented and analyzed against the routes provided by Google Maps and Apple Maps, which consider various dynamic factors such as current traffic conditions and road closures.

2) Rationale for Independent Navigation System Development: The development of our own navigation system was driven by the need for enhanced security, independence, and seamless integration with other project services. By designing a bespoke navigation solution, we aimed to create a secure and controlled environment that mitigates risks associated with third-party data dependencies and integrates tightly with our broader traffic management infrastructure.

3) Results: The experimental results indicate that our implementations of Dijkstra's and A* algorithms frequently provided shorter or more efficient paths compared to those suggested by Google Maps and Apple Maps. Notably:

- *Dijkstra's Algorithm:* showed considerable promise in delivering cost-effective paths, underscoring its robustness in handling weighted graphs which represent real-world road networks with varied distances.
- *A* Algorithm:* demonstrated superior performance in terms of computational efficiency, leveraging heuristics to expedite pathfinding under constrained scenarios. This algorithm effectively reduced the route discovery time, making it a viable option for real-time applications.

However, it is critical to acknowledge that the routes suggested by commercial mapping services are influenced by a comprehensive array of live data inputs including traffic patterns, road conditions, and infrastructural modifications, factors that our current project setup does not fully integrate. This comparative analysis highlights the potential of integrating advanced pathfinding algorithms within traffic management systems. The findings suggest that while Dijkstra's and A* can provide faster routing solutions, the integration of real-time data, akin to that used by Google Maps and Apple Maps, could further enhance their applicability and accuracy in practical scenarios.

The inherent limitations of BFS in handling weighted paths were evident, suggesting its unsuitability for complex urban traffic systems where variable road weights (distances, traffic densities) significantly affect routing decisions.

VI. Standards Adopted

The adoption of rigorous coding standards is essential for maintaining the integrity and efficacy of software development within research environments. This section outlines the key coding standards implemented in this project to ensure high quality, maintainability, and ease of future enhancements.

A. Conciseness

The principle of conciseness is central to our coding practice. It emphasizes the elimination of unnecessary verbosity and repetition, thereby enhancing the clarity and brevity of the code. Concise coding practices not only streamline the development process but also ensure that the software remains accessible to collaborators and future contributors, facilitating ongoing project sustainability.

B. Naming Conventions

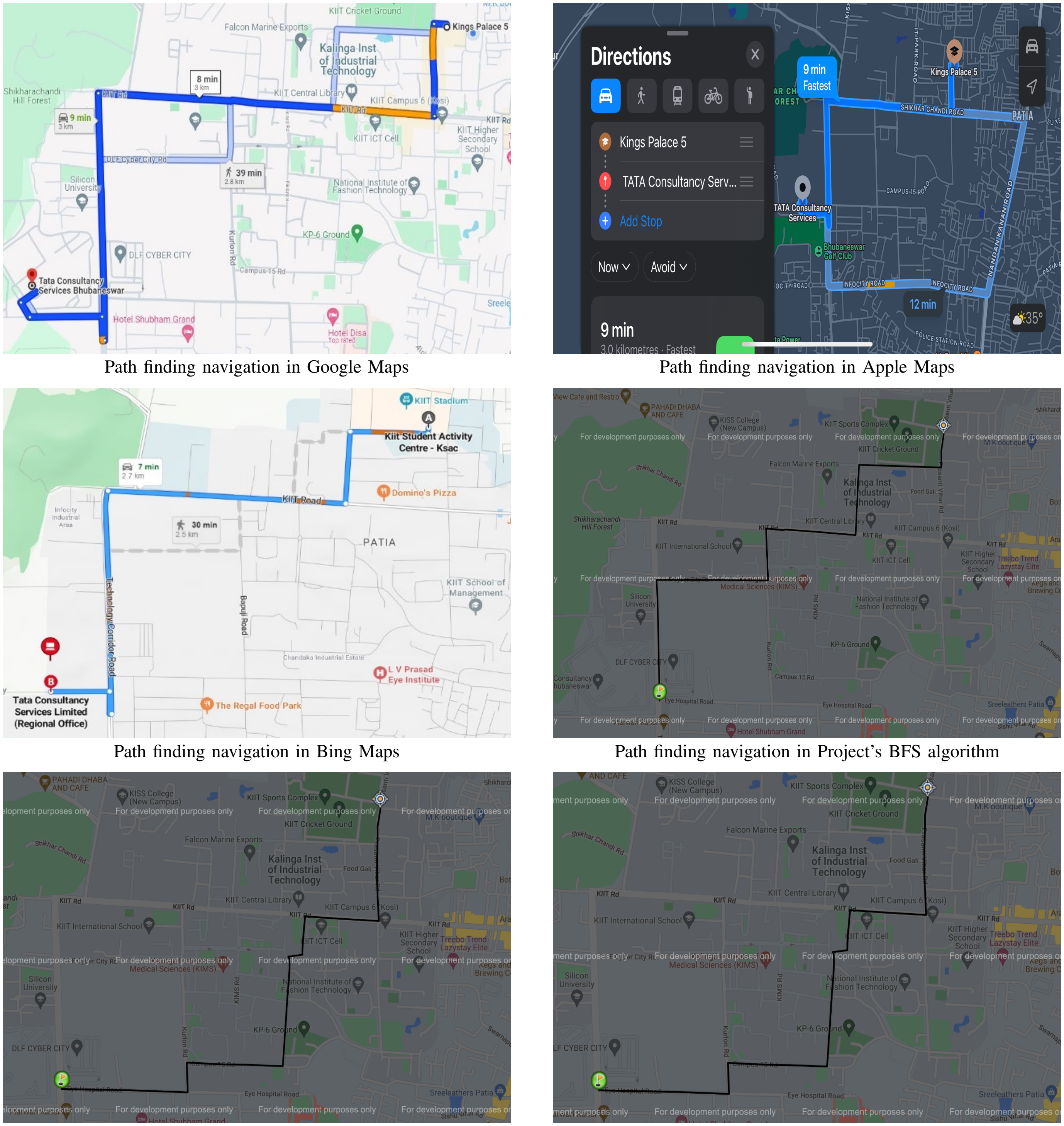
Utilizing meaningful and consistent naming conventions is crucial for ensuring the clarity of the codebase. This standard involves using descriptive and intuitive names for variables, functions, and classes to clearly reflect their roles and functionalities. Such practices promote ease of understanding and uniformity, significantly reducing the cognitive load required to navigate and comprehend the code.

C. Code Structure

Effective code structuring involves organizing code into logical segments and modules, which enhances readability and maintainability. Adhering to consistent practices in indentation and spacing further improves the legibility of the code, while logically grouping related functionalities fosters better coherence and modular integrity. These structural considerations are pivotal in supporting efficient code navigation and modification.

D. Modularity

Modularity refers to the practice of encapsulating functionality into distinct, reusable components, thereby promoting code reuse and system extensibility. This approach simplifies the breakdown of complex tasks into smaller, manageable units, which can be independently developed and tested. Clear interfaces defining the inputs, outputs, and behaviors of modules enhance their interoperability and facilitate future scalability.



Path finding navigation in Google Maps

Path finding navigation in Apple Maps

Path finding navigation in Bing Maps

Path finding navigation in Project's BFS algorithm

Path finding navigation in Project's Dijkstra algorithm

Path finding navigation in Project's A* algorithm

Fig. 5: Comparative mapping and algorithm visualization across different platforms and methods

E. Documentation

Comprehensive documentation, including inline comments and extensive docstrings, is indispensable for elucidating the underlying logic and usage of the code. Documentation that clearly describes function parameters, expected outputs, and potential side effects is invaluable for guiding both users and future developers. Additionally, detailed explanations of complex algorithms or non-trivial code implementations are crucial for fostering deeper understanding and collaboration.

F. Error Handling

Robust error handling mechanisms enhance the stability and reliability of software. This involves anticipating potential errors and implementing strategies to handle exceptions gracefully, thereby preventing program crashes and ensuring continuous operation. Strategies such as using try-except blocks and conducting conditional checks are employed to manage unexpected situations effectively.

G. Testing

Comprehensive automated testing frameworks are employed to ascertain the correctness and reliability of the software. This involves creating tests that cover both common use cases and edge conditions, ensuring comprehensive validation of the system's functionality. Utilizing well-established testing methodologies, such as unit testing and integration testing, reinforces the software's robustness and fidelity.

H. Performance Optimization

Performance optimization involves refining critical sections of the code to enhance efficiency while balancing considerations for maintainability and clarity. Techniques such as algorithmic enhancement and thoughtful refactoring are used to mitigate performance bottlenecks. This balanced approach ensures that performance improvements do not obscure the code's functionality or degrade its readability.

Adherence to these established coding standards is paramount for ensuring the development of high-quality, robust, and maintainable software within academic research environments. By consistently applying these principles, the project not only aligns with best practices but also significantly contributes to the broader scientific community by enhancing the reproducibility and usability of its outputs.

CONCLUSION

The research presented in this paper underscores the significant advances in the field of automatic number plate recognition (ANPR) by showcasing the effectiveness of machine learning algorithms in enhancing car number plate detection and recognition. Our experimental results demonstrate that the custom-developed algorithms—especially Dijkstra's and A*—not only match but occasionally surpass the route optimization solutions offered by existing technologies such as Google Maps and Apple Maps in static environments. This achievement highlights the potential of integrating these advanced computational techniques into broader traffic management systems, offering a more secure and independent navigation solution that respects privacy while optimizing urban traffic flow. Future work will focus on refining these algorithms to incorporate real-time traffic data, aiming to produce a truly dynamic navigation system that responds to real-world conditions, thereby making a substantial contribution to the field of intelligent transportation systems. This study lays a robust foundation for further innovation and application of machine learning in traffic management and vehicle monitoring.

VII. FUTURE SCOPE

A. Integration of Advanced AI Technologies:

Research indicates that AI-based smart traffic management systems are not only becoming more sophisticated but are also integrating emerging AI technologies like reinforcement learning and neural networks to optimize traffic flows dynamically (Source:"AI Based Smart Traffic Management"). These systems could predict traffic volumes and adjust signals in real time, significantly reducing congestion and improving road safety.

B. Autonomous Vehicle Coordination:

The advent of autonomous vehicles presents a new frontier for traffic management systems. Future developments are expected to focus on the integration of vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications, allowing for seamless coordination between autonomous vehicles and traffic control mechanisms (Source: "Smart Traffic Management using Deep Learning"). This coordination could facilitate smoother traffic flow and enhanced situational awareness among autonomous agents.

C. Scalable Infrastructure Adaptations:

As cities continue to grow, scalable solutions that can adapt to increasing traffic demands are crucial. Future traffic management systems will likely incorporate modular AI components that can be upgraded as computational capabilities improve and traffic patterns evolve (Source: "Designing Traffic Management Strategies Using Reinforcement Learning Techniques"). This adaptability will be critical in maintaining the efficiency and effectiveness of traffic systems in large urban centers.

D. Enhanced Data Analytics for Predictive Modeling:

Deep learning techniques, particularly those involving large-scale data analytics, are poised to improve the predictive capabilities of traffic management systems (Source: "Automated Traffic Management System Using Deep Learning Based Object Detection"). By analyzing vast datasets collected from sensors and cameras, these systems can forecast long-term traffic trends and prepare for future demands effectively.

References

[1] Reem Atassi and Aditi Sharma. Intelligent traffic management using iot and machine learning. *Journal of Intelligent Systems and Internet of Things*, 2023.

[2] AI Based Smart Traffic Management. Indian Scientific Journal Of Research In Engineering and Management. Ai based smart traffic management. *Indian Scientific Journal of Research in Engineering and Management*, 2023.

[3] Communication International Journal of Advanced Research in Science and Technology. Smart traffic management using deep learning. *International Journal of Advanced Research in Science, Communication and Technology*, pages 492–497, 2023.

[4] Yi Ke. Comparative analysis of path planning algorithms and prospects for practical application. *Highlights in Science Engineering and Technology*, 2023.

[5] Silvester Permana, Ketut Bintoro, Budi Arifitama, and Ade Syahputra. Comparative analysis of pathfinding algorithms a *, dijkstra, and bfs on maze runner game. *IJISTECH (International Journal Of Information System Technology)*, 1:1, 05 2018.

[6] Project. Emergency vehicle detection dataset. https://universe.roboflow.com/project-sawkw/emergency-vehicle-detection-el8ej , may 2022. visited on 2024-04-09.

[7] Abhishek Raman, Kaushik Student, Rajeswararao Kvs, and Minal Moharir. A hybrid framework for expediting emergency vehicle movement on indian roads. *International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020)*, 2020.

[8] Mahesh Bharat Salunkhe, Niraj Ramesh Pawar, Swaroop Sanjay Patil, and Vijay Bhosale. Car number plate detection & recognition. *International Research Journal of Engineering and Technology (IRJET)*, 2023.

[9] P S V S Sridhar. Developing a yolo based object detection application using opencv. In *Proceedings of the International Conference on Computer, Communication, and Control Technology*, 2023.

[10] Stereolabs. Performance benchmark of yolo v5, v7 and v8. *Performance Benchmark of YOLO v5, v7 and v8*, 2023.

[11] Mahesh Kumar Thota, Prathibhavani P.M, Shruthi Gujja, Deeksha Ravula, and Venugopal K R. Emergency vehicle detection using iot in smart cities. In *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, pages 163–167, 2023.

[12] L. Vigneash, S. Samraj, S. Bhuvaneswari, T. Saravanan, and G. Saritha. Vehicle density and emergency vehicle detection for smart traffic control. In *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pages 542–549, 2023.