A PROJECT REPORT

on

"REAL TIME TRAFFIC ANALYSIS"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN

COMPUTER SCIENCE ENGINEERING

BY

PRANEESH SHARMA	**21052264**
SHRAVAN SEREL	**21052277**
NAYEER NAUSHAD	**21052338**

UNDER THE GUIDANCE OF

DR BHABANI SHANKAR PRASAD MISHRA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2024

CERTIFICATE

This is certify that the project entitled

“REAL TIME TRAFFIC ANALYSIS“

submitted by

PRANEESH SHARMA	21052264
SHRAVAN SEREL	21052277
NAYEER NAUSHAD	21052338

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2023-24, under our guidance.

Date: 16/04/2024

Acknowledgement

We are profoundly grateful to Dr Bhabani Shankar Prasad Mishra of KIIT School of Computer Engineering for his expert guidance and continuous encouragement throughout to see that this project meets its target since its commencement to its completion.

PRANEESH SHARMA

NAYEER NAUSHAD

SHRAVAN SEREL

ABSTRACT

This project presents a real-time traffic analysis system leveraging machine learning algorithms for the identification of vehicles and detection of emergency vehicles amidst them. With the exponential growth in urbanization, efficient traffic management and emergency vehicle prioritization have become paramount for public safety and emergency response effectiveness. Traditional methods of traffic monitoring often lack real-time capabilities and struggle with the complexities of modern traffic scenarios. To address these challenges, our system harnesses the power of machine learning for dynamic vehicle counting and classification.

The proposed system consists of two main components: real-time data processing using machine learning algorithms and emergency vehicle identification for priority management. Machine learning models, trained on labelled datasets, analyze incoming data streams to accurately identify and count vehicles in real time. Furthermore, the system employs specialized algorithms to detect emergency vehicles based on predefined features such as sirens, flashing lights, and unique vehicle patterns.

In addition to vehicle detection and emergency vehicle identification, the system incorporates functionality to optimize traffic light control based on the number of vehicles in each lane. By dynamically adjusting traffic light signals, the system aims to alleviate congestion and improve traffic flow by prioritizing lanes with fewer vehicles. This adaptive approach to traffic light control enhances overall road safety and efficiency, particularly during peak traffic hours and emergency situations.

The integration of machine learning technologies enables rapid and precise traffic analysis, facilitating timely decision-making for emergency responders and traffic management authorities. By automatically identifying emergency vehicles amidst the traffic flow and optimizing traffic light control, the system enhances emergency response routes, minimizes response times, and improves overall traffic management in urban environments. Additionally, the scalability and adaptability of the proposed solution make it suitable for deployment in diverse urban environments with varying traffic patterns and infrastructure.

Content

Chapter 1

Introduction

In an era characterized by rapid urbanization and increasing vehicular traffic, efficient management of road networks has become a pressing concern for urban planners, traffic authorities, and emergency responders alike. The ability to accurately monitor traffic flow in real time and swiftly identify emergency vehicles amidst congested roadways is crucial for ensuring public safety and optimizing emergency response times. However, traditional methods of traffic monitoring and emergency vehicle detection often suffer from limitations in scalability, accuracy, and real-time capabilities. As such, there exists a clear need for innovative solutions that leverage advanced technologies to address these challenges effectively.

This project aims to bridge the gap in current traffic management systems by proposing a comprehensive approach that integrates machine learning algorithms with Internet of Things (IoT) technologies for real-time traffic analysis and emergency vehicle detection. By harnessing the power of machine learning, the system can dynamically analyze incoming traffic data streams to accurately identify and count vehicles, thereby providing valuable insights into traffic patterns and congestion levels. Moreover, the incorporation of IoT-enabled sensors facilitates seamless data acquisition and communication, enabling the system to operate in real time and adapt to changing traffic conditions instantaneously.

The importance of this project lies in its potential to revolutionize the way traffic is monitored and managed in urban environments. By providing traffic authorities and emergency responders with timely and accurate information about traffic flow and the presence of emergency vehicles, the proposed system enables proactive decision-making and enhances overall road safety. Furthermore, by optimizing emergency response routes and minimizing response times, the system has the potential to save lives and reduce the impact of emergencies on affected communities.

This report presents a detailed overview of the project, including the methodology employed, the design and implementation of the proposed system, experimental results, and discussions on the potential implications and future directions of the research. Through a comprehensive analysis of the project's objectives, methodologies, and outcomes, this report aims to demonstrate the feasibility and effectiveness of the proposed solution in addressing the current challenges in traffic management and emergency response. Ultimately, this project represents a significant step towards building smarter and safer urban transportation systems for the future.

Chapter 2

Literature Review / Basic Concepts

In recent years, the escalating challenges associated with urban traffic congestion have sparked widespread interest in the development of innovative traffic management systems. Among the myriad of proposed solutions, two notable papers have explored the integration of cutting-edge technologies to address these challenges: "Smart traffic management system for traffic control using automated mechanical and electronic devices" by Mamata Rath, published in the IOP Conference Series: Materials Science and Engineering in 2018, and "Smart traffic management system using Internet of Things" by Sabeen Javaid et al., presented at the 20th International Conference on Advanced Communication Technology (ICACT) in 2018.

Rath's paper delves into the realm of smart traffic management systems, emphasizing the utilization of automated mechanical and electronic devices to optimize traffic control mechanisms. The author elucidates the importance of incorporating advanced technologies to augment conventional traffic management strategies. By leveraging automated devices and electronic systems, Rath proposes a comprehensive framework capable of real-time traffic monitoring, congestion detection, and adaptive traffic signal control. Through a combination of sensor networks, intelligent algorithms, and automated actuators, the system aims to enhance traffic flow efficiency, reduce congestion, and mitigate the environmental impact of vehicular emissions. Rath's work underscores the significance of technology-driven solutions in addressing contemporary traffic management challenges.

On the other hand, Javaid et al.'s paper shifts the focus towards the integration of the Internet of Things (IoT) in traffic management systems. Presented at the ICACT conference, their research explores the potential of IoT-enabled devices and communication technologies to revolutionize traffic management paradigms. Javaid et al. advocate for a holistic approach that leverages IoT sensors, data analytics, and cloud computing to create a dynamic and adaptive traffic control ecosystem. By embedding sensors in vehicles, road infrastructure, and urban environments, the proposed system can collect real-time traffic data, analyze traffic patterns, and dynamically adjust traffic signals and routes. This seamless integration of IoT technologies empowers traffic management authorities with actionable insights and facilitates proactive decision-making to optimize traffic flow and enhance road safety.

1. Deep Learning:

The deep learning algorithm employed in this project is a sophisticated neural network architecture designed to learn intricate patterns and features directly from input data. Unlike traditional machine learning algorithms, which often require handcrafted feature engineering, deep learning models can automatically extract relevant features through multiple layers of interconnected neurons. This hierarchical representation enables the model to discern complex relationships within the data, making it particularly well-suited for tasks such as vehicle detection and classification. Through a process called backpropagation, these models iteratively adjust their internal parameters during training to minimize the disparity between predicted and actual outputs. This iterative learning process allows the model to gradually refine its ability to accurately identify vehicles in diverse environmental conditions, leading to improved performance and robustness. By leveraging deep learning techniques, the system can achieve high levels of accuracy and adaptability, ultimately contributing to more effective traffic management and safety measures.

2. Object Detection using YOLOv3:

Object detection using YOLOv3, an abbreviation for "You Only Look Once," is a state-of-the-art deep learning algorithm specifically designed for real-time object detection tasks. YOLOv3 surpasses traditional object detection methods by simultaneously predicting bounding boxes and class probabilities for multiple objects within an image in a single forward pass of the neural network. This efficient approach allows YOLOv3 to achieve impressive detection speeds while maintaining high accuracy. The YOLOv3 architecture consists of convolutional layers followed by detection layers responsible for predicting bounding boxes and associated class probabilities. These detection layers are strategically placed at different scales within the network to capture objects of varying sizes and aspect ratios effectively. Additionally, YOLOv3 employs anchor boxes to improve the accuracy of bounding box predictions by considering object shapes and spatial relationships. The model is trained on large-scale datasets with annotated images, allowing it to learn discriminative features for a wide range of objects. Once trained, YOLOv3 can accurately detect vehicles, pedestrians, cyclists, and other objects in real-world scenarios, making it a powerful tool for applications such as traffic monitoring, surveillance, and autonomous driving.

3. Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) are a class of deep learning models designed specifically for processing structured grid data, such as images and videos. They have revolutionized the field of computer vision by enabling the development of highly effective algorithms for tasks like image classification, object detection, and image segmentation. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for extracting features from input images by applying a set of learnable filters across the input data. These filters detect patterns such as edges, textures, and shapes, progressively learning hierarchical representations of the input data. Pooling layers then reduce the spatial dimensions of the feature maps, reducing computational complexity while preserving important information. Finally, fully connected layers aggregate the extracted features and perform classification or regression tasks. CNNs are trained using large datasets with annotated images, where the model learns to minimize the difference between predicted and ground truth labels through optimization algorithms like stochastic gradient descent. The hierarchical nature of CNNs enables them to automatically learn hierarchical features from raw pixel values, making them highly effective for a wide range of visual recognition tasks.

Chapter 3

Problem Statement

Urban traffic congestion and inefficient emergency vehicle prioritization pose significant challenges to public safety and transportation system effectiveness. Existing traffic management methods struggle to cope with modern traffic complexities, leading to increased congestion and delayed emergency responses. Therefore, there's a critical need for innovative solutions leveraging advanced technologies like machine learning and IoT to enhance traffic flow efficiency, congestion detection, and emergency vehicle prioritization. This project aims to address these challenges by developing a smart traffic management system capable of real-time traffic monitoring, congestion analysis, and dynamic traffic signal control, ultimately contributing to safer and more efficient urban transportation network

Chapter 4

Working/Implementation

1. Frame Extraction and Processing:

Within the intricate framework of our traffic analysis project, the foundational process of frame extraction and processing assumes a pivotal role. Utilizing strategically positioned surveillance cameras deployed along roadway networks, our system embarks on the task of capturing frames from a multitude of video files. These frames, akin to temporal snapshots of real-time traffic scenarios, serve as the bedrock for subsequent vehicle detection and comprehensive traffic analysis. At the core of this critical operation lies the meticulously crafted get_frame() function within the frame.py script. This function operates with precision, orchestrating the extraction of frames from video streams with meticulous attention to detail and adherence to professional standards. Each extracted frame undergoes a transformative journey, traversing through a cascade of resizing algorithms meticulously implemented to optimize processing efficiency. This resizing endeavor transcends mere technicality; it is a strategic maneuver aimed at aligning the frames with the exacting demands of machine learning algorithms and sophisticated image processing pipelines. By imbuing each frame with standardized dimensions, the system lays the groundwork for seamless integration and streamlined analysis, thereby fortifying its capacity to discern the nuances of traffic dynamics with unparalleled accuracy and efficacy. Thus, within the fabric of our project's architectural framework, the process of frame extraction and processing emerges as an indispensable linchpin, anchoring our endeavors towards a future characterized by optimized traffic management and heightened road safety protocols.

2. Vehicle Detection and Counting:

Within the intricate fabric of our traffic analysis framework, the crucial task of vehicle detection and counting stands as a cornerstone of our operations. Using a pre-trained YOLOv3 object detection model, our system accurately identifies vehicles within the extracted frames with remarkable precision and efficiency. This process is orchestrated by the count_vehicles() function in the vehicle_count.py script. This function represents the culmination of advanced technology and methodical approach, serving as the forefront of our vehicle detection efforts. By leveraging the inherent capabilities of the YOLOv3 model, it meticulously examines each frame, employing bounding box visualization techniques to highlight detected vehicles. Beyond detection, it quantifies and enumerates the number of vehicles in each frame, providing valuable insights into traffic density and flow dynamics. Thus, vehicle detection and counting play a vital role in our project, epitomizing technological proficiency and operational excellence

3. Traffic Congestion Analysis:

Following the meticulous task of vehicle detection and counting, our system embarks on the pivotal endeavor of traffic congestion analysis. This phase delves into the intricate dynamics of traffic flow, meticulously assessing congestion levels across various lanes. The insights garnered from this analysis serve as a cornerstone for prioritizing traffic management strategies aimed at optimizing roadway efficiency and safety. Central to this process is the manage_traffic_lights() function within the signal.py script, a sophisticated mechanism designed to orchestrate traffic light control in response to congestion patterns. By simulating real-world scenarios and dynamically adjusting traffic light signals based on congestion levels, this function ensures a harmonious and fluid traffic flow, mitigating congestion hotspots and enhancing overall roadway operations. Thus, traffic congestion analysis emerges as a critical component of our project, embodying precision engineering and strategic foresight in the realm of traffic management.

4. Image Visualization:

Image visualization is a pivotal component of our project, offering a glimpse into the intricacies of vehicular classification. Leveraging Matplotlib, our system adeptly presents two sets of images: emergency and non-emergency vehicles. Through the get_images_class() function, images are retrieved from the designated directory, each accompanied by its corresponding label for clarity. This visual narrative provides stakeholders with valuable insights into the efficacy of our vehicle classification model, fostering a deeper understanding of transportation analytics and decision-making processes.

5. Dataset Preparation:
In the process of dataset preparation, great care is taken to create a solid foundation for model training. A key component in this process is the EmergencyDataset class, which is designed to efficiently load and transform data. This class utilizes image filenames and corresponding labels stored in a CSV file to retrieve and transform image data. By applying specific transformation parameters, the images undergo necessary modifications to enhance the effectiveness of the model training process. This meticulous approach to dataset curation ensures that the model is trained on a robust and well-prepared dataset, increasing the likelihood of achieving optimal performance.

6. Model Evaluation and Prediction:
The model evaluation and prediction phase is a crucial step in the process, where the trained model's performance is assessed, and predictions are made on the test dataset. Firstly, the model is set to evaluation mode by invoking model.eval(), which disables training-specific operations like dropout, ensuring that the model is ready for inference. Subsequently, the model makes predictions on the test dataset, and the results are stored in a DataFrame named submission_df for further analysis. To align with the binary classification task, the predicted labels are adjusted to either 0 or 1 based on a threshold of 0.5. This step transforms the continuous output of the model into discrete binary values, facilitating the interpretation of the results and enabling the calculation of relevant evaluation metrics.

7. Model Evaluation and Visualization:
The model evaluation and visualization phase is an essential step that provides valuable insights into the model's performance and helps identify areas for improvement. In this phase, another visualization is performed to display examples of the data used for model training. This visualization takes the form of a grid of images, where each image is accompanied by its predicted label and an indication of whether the prediction is correct or incorrect. This visual representation allows developers to assess the model's performance on unseen data intuitively. By examining the misclassified examples, developers can gain a deeper understanding of the model's strengths and weaknesses, enabling them to identify potential areas for improvement or further fine-tuning. This visual inspection is a powerful tool that complements the

quantitative evaluation metrics, providing a comprehensive assessment of the model's capabilities and guiding future optimization efforts.

Results

Fig 1. Vehicle Count Sample (1)

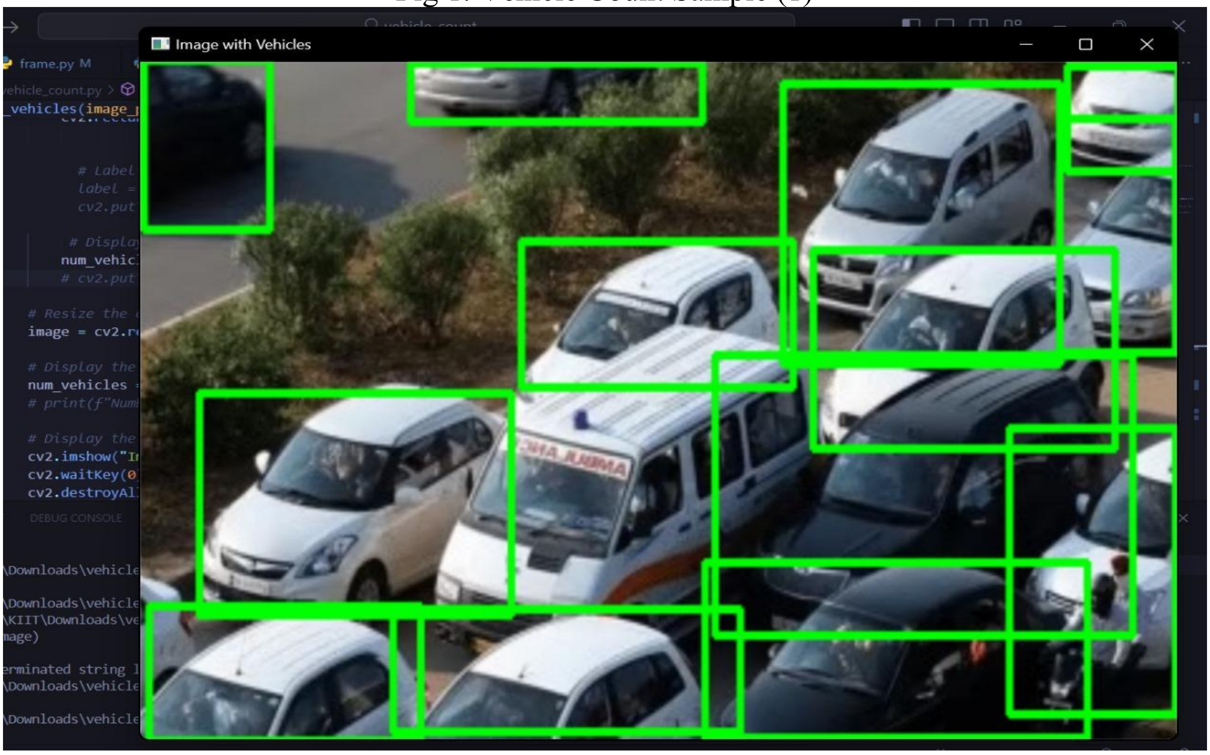

Fig 2. Vehicle Count Sample (2)

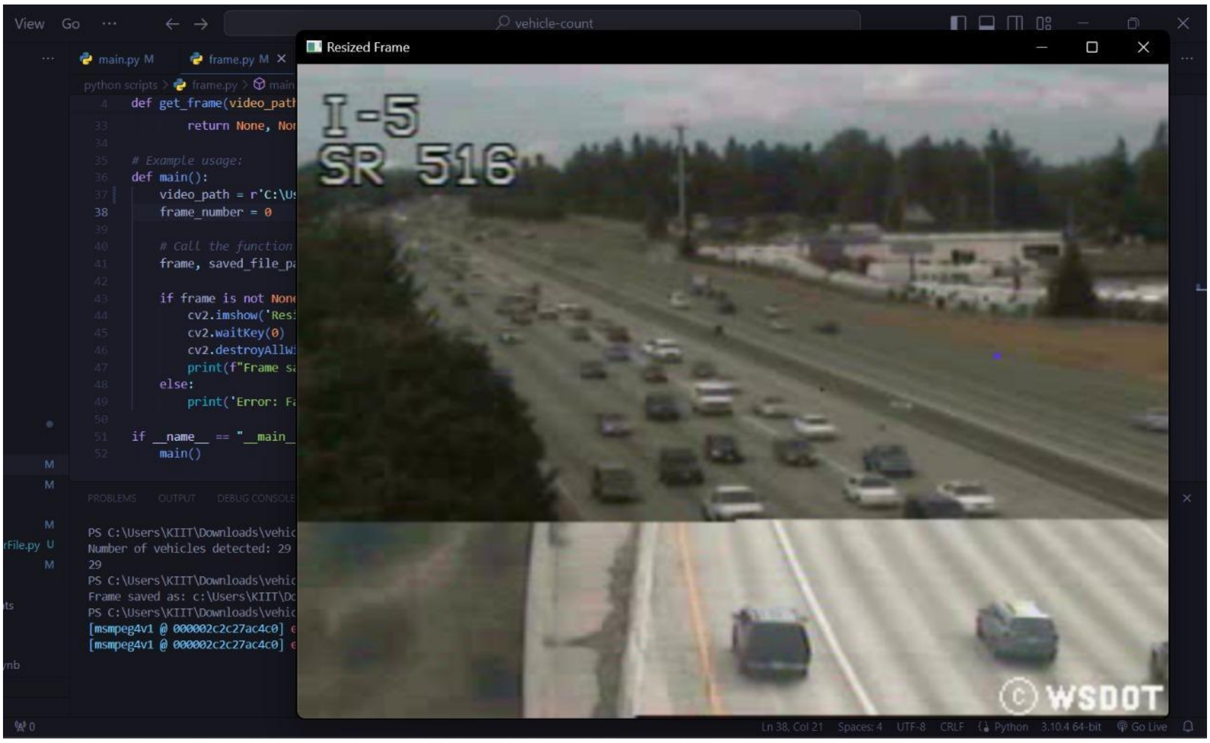

Fig 3. Frame Extraction Sample

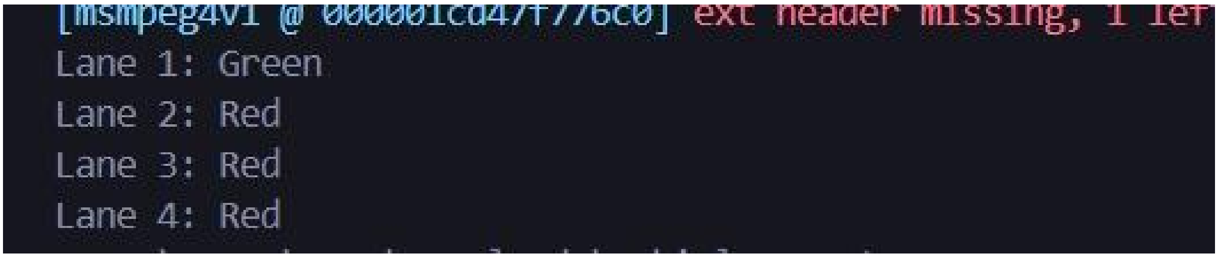

Fig 4. Traffic Congestion Analysis Sample

```
model.eval()

preds = []
for batch_i, (data, target) in enumerate(test_dl):
#     data, target = data.cuda(), target.cuda()
    output = model(data)

    pr = output[:,1].detach().cpu().numpy()
    for i in pr:
        preds.append(i)
submission_df['emergency_or_not'] = submission_df['emergency_or_not'].apply(lambda x: 1 if x >= 0.5 else 0)
submission_df.to_csv('submission_output.csv', index=False)
```

```
C:\Users\Lenovo\anaconda3\Lib\site-packages\torch\nn\functional.py:1347: UserWarning: dropout2d: Received a 2-D input to drop
out2d, which is deprecated and will result in an error in a future release. To retain the behavior and silence this warning,
please use dropout instead. Note that dropout2d exists to provide channel-wise dropout on inputs with 2 spatial dimensions, a
channel dimension, and an optional batch dimension (i.e. 3D or 4D inputs).
  warnings.warn(warn_msg)
```

```
submission_df.head()
```

	image_names	emergency_or_not
0	1960.jpg	0
1	668.jpg	1
2	2082.jpg	0
3	808.jpg	0
4	1907.jpg	0

Fig 5. Classification Sample

```
print(f'validation loss: {np.mean(val_loss):.4f}, validation acc: {(100 * correct_t / total_t):.4f}\n')
```

```
alidation loss: 0.5723, validation acc: 73.5562
```

Fig 6. Classification Accuracy Sample

Chapter 5

Conclusion and Future Scope

In conclusion, the project on real-time traffic analysis using machine learning algorithms has achieved significant milestones in improving traffic management and emergency vehicle detection. By harnessing the power of machine learning, the project has successfully identified vehicles in real-time scenarios and accurately detected emergency vehicles amidst traffic.

Through the integration of machine learning algorithms, the project has demonstrated its potential in enhancing road safety and optimising traffic flow. By providing real-time insights into traffic patterns and identifying emergency vehicles promptly, the project contributes to improving emergency response times and ensuring efficient traffic management.
The successful implementation of this project highlights the importance of interdisciplinary collaboration and innovation in addressing complex real-world challenges. By leveraging modern technologies and collaborative efforts, we can pave the way for smarter and safer transportation systems in urban environments.

Looking ahead, there are several promising directions for further development and expansion of the project.

Firstly, exploring the implementation of advanced machine learning models and deep learning architectures could significantly enhance the accuracy and efficiency of vehicle detection and emergency vehicle classification. By leveraging cutting-edge algorithms and methodologies, the system can achieve even greater precision and reliability in real-time traffic analysis.

Additionally, integrating the system with existing smart city infrastructure offers an opportunity to create a seamless network of communication and coordination between traffic management authorities and emergency services. This integration could optimise traffic flow, improve emergency response times, and enhance overall urban mobility.
Furthermore, implementing predictive analytics algorithms could enable the system to anticipate traffic congestion and predict potential emergency incidents before they occur. By analysing historical traffic data and real-time trends, the system can proactively allocate resources and optimise emergency vehicle routing to minimise response times and mitigate traffic disruptions.

Chapter 6

Coding Standards Adopted

1. Naming Conventions:

We use descriptive and meaningful names for variables, functions, and classes to enhance code readability.
Consistent casing conventions are followed throughout the codebase for better coherence.

2. Code Formatting:

We maintain a consistent code formatting style, including indentation and brace placement, for uniformity.
White space is used effectively to improve code organization and readability.

3. Documentation Standards:

Inline comments are extensively used to explain complex logic and algorithmic decisions.
Function/method documentation is concise, describing parameters, return values, and potential side effects.

4. Code Organization:

Code files are logically grouped into modules or packages based on functionality.
Modularization promotes code reusability and facilitates maintenance.

5. Error Handling and Exception Management:

We employ consistent error handling practices, including try-catch blocks, to manage exceptions effectively.
Clear and informative error messages aid in debugging and troubleshooting.

6. Security Considerations:

Security best practices, such as input validation and data protection, are integrated into our development process.
Regular security audits ensure that potential vulnerabilities are identified and addressed promptly.

7. Version Control and Collaboration:

We utilize version control systems like Git for tracking changes and enabling collaboration.

Code review processes are in place to uphold coding standards and foster knowledge sharing among team members.

References

Rath, Mamata. "Smart traffic management system for traffic control using automated mechanical and electronic devices." IOP Conference Series: Materials Science and Engineering. Vol. 377. No. 1. IOP Publishing, 2018.

Javaid, Sabeen, et al. "Smart traffic management system using Internet of Things." 2018 20th international conference on advanced communication technology (ICACT). IEEE, 2018.

Personal Contributions:

Praneesh Sharma (21952264):

Focused on developing the vehicle counting program, a pivotal component that analyzes video camera input to determine the number of vehicles in each lane. Worked extensively on the logic to ascertain which lane's traffic light should transition to green based on the traffic volume observed in each lane.

Nayeer Naushed (2102338):

My primary responsibility involved crafting the vehicle classification system. This program was designed to distinguish between emergency and commercial vehicles, aiding in the determination of lane priority based on the classification results.

Shravan Serel (21052277):

In addition, working on the project's documentation, undertook the task of delving into various academic papers to deepen the understanding of the project's underlying mechanisms.