University of Bahrain
College of Information Technology
Department of Computer Science

RoadEye

Prepared by

Mohamed Rihal	**20192767**
Mohamed Yaqoob Abdulla	**20173577**
Khalid Hani Salem	**20199728**

ITCS 498 – ITSE498

Senior Project

Academic Year 2023-2024

Project Supervisor: Dr. TAHER SALEH KHAID HOMEED

Abstract

RoadEye is an innovative system designed to monitor and analyze accidents occurring in the parking lot by leveraging machine learning techniques and connecting to cameras. This project utilizes the Python programming language to develop a robust and efficient system capable of detecting and reporting accidents in real time. By integrating with existing camera infrastructure, RoadEye aims to enhance road safety by providing real-time notifications through E-mail and valuable insights to relevant authorities and emergency services.

Using state-of-the-art machine learning algorithms, RoadEye processes the live video feeds from cameras to identify potential accidents. The system employs computer vision techniques to detect and track cars in real time. By analyzing the behavior and interactions of these objects, RoadEye can accurately identify and classify accidents, including collisions, near misses, and other hazardous incidents.

Once an accident is detected, RoadEye promptly generates detailed incident reports, including the location, time, and type of accident. These reports are automatically forwarded to the appropriate authorities, enabling them to quickly respond and dispatch emergency services to the scene. Additionally, RoadEye provides statistical analysis and visualizations of accident data to aid in identifying accident-prone areas, evaluating road safety measures, and making informed decisions about infrastructure improvements.

By leveraging the power of machine learning and real-time video analysis, RoadEye strives to improve road safety and reduce accident response times. The system's ability to accurately detect accidents and provide timely notifications can help save lives and mitigate the impact of accidents on traffic congestion and public safety. RoadEye represents a significant step forward in leveraging technology for accident prevention and management, aiming to create safer and more efficient road systems for communities worldwide.

Acknowledgments

We would like to humbly express our sincere gratitude to our esteemed advisor, Dr. Taher, for his invaluable guidance and support throughout the whole project process. His expertise, patience, and encouragement were instrumental in shaping our work and enabling us to achieve our goals.

We also thank our friends who provided us with their support, motivation, and valuable feedback throughout the process. Their collaboration was essential in overcoming challenges and fueling our progress.

Furthermore, we would like to express our deepest appreciation to our families for their unwavering love, encouragement, and belief in us. They have pushed us to reach our full potential and provided us with the strength and resilience needed to persevere through tough times.

Without the invaluable support of these individuals, RoadEye would not have been possible. We are grateful for your contributions and forever indebted to your kindness and generosity. Thank you All.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Through the lens of innovation, RoadEye aims to pave the way for safer journeys, where artificial intelligence meets the open road.

Every time you park your car, you trust there ahead holds no unseen dangers. But the reality is, every parking teems with potential hits: drowsy drivers, reckless speeders, and unforeseen obstacles lurking around every bend. Until now, these kinds could spell disaster. But what if you had a guardian angel on the road, watching over you with eyes sharper than hawks?

Enter RoadEye, a revolutionary AI-powered system that transforms accident detection from reactive to initiative-taking. Imagine a vigilant sentinel, scanning the road ahead with innovative algorithms, instantly recognizing threats you might miss, and issuing timely warnings that could save your fortune and those around you.

This report delves into the heart of RoadEye, unveiling its groundbreaking technology and life-saving potential. We will shed light on its abilities in upcoming chapters.

1.1Problem Statement

Imagine being a student at the University of Bahrain, excited for class. You park your car and head off, only to return later and find it mysteriously banged up! This happens too often, and everyone is confused and frustrated.

The University of Bahrain, like many other university institutions, grapples with a pressing issue that significantly affects the well-being and convenience of its student body – parking lot accidents. Students attending classes often find themselves confronted with the surprising reality that their vehicles have been damaged during their time on campus. The frequency of these incidents raises concerns about the lack of effective monitoring and reporting mechanisms in place to address parking lot accidents promptly.

In response to this issue, the project was initiated with the goal of implementing a comprehensive solution using "Road Eye," a system designed to monitor and report accidents in real-time. By leveraging advanced surveillance technology, the project aims to provide an efficient and reliable means for students to gain advantage of reported incidents promptly. The Road Eye system is anticipated to function as a safeguard for students' rights, enabling them to seek appropriate action and compensation for damage incurred while parked on campus.

1.2 Project Objectives

The primary objective of this project is to develop a comprehensive system, named RoadEye, that addresses the identified problem. The specific objectives are as follows:

1.To develop a real-time accident detection module using machine learning and computer vision techniques to analyze live video feeds from cameras.
2. To train the machine learning models on a diverse dataset of labeled accident scenarios to ensure accurate detection and classification.
3. To implement an incident reporting module that automatically generates reports ,including the accident live feed.
4. To establish a reliable communication mechanism to promptly transmit incident reports to the appropriate emergency services.
5. To evaluate the performance and effectiveness of the RoadEye system through extensive testing and validation.
6. To ensure the rights of whomever car got hit and ensure the ability to get compensated.
7. To increase road safety and lower the probability of accidents happening anywhere.

It is important to note that the scope of this research project is limited to the development and evaluation of the RoadEye system. While the system aims to enhance road safety and accident management, it does not address other factors such as traffic congestion or driver behavior.

1.3 Relevance/Significance of the project

The RoadEye project holds significant relevance and significance in the domain of road safety and accident management. By leveraging machine learning and computer vision, the system aims to improve the detection and reporting of accidents, leading to faster response times and more effective allocation of resources. The real-time nature of the system allows for immediate notification by email of authorities and emergency services, enabling them to take swift action and potentially save lives.

1.3.1 Project scope

The project aims to attack the patient issue of accidents at the University of Bahrain through the perpetration of the Road Eye system. By strategically placing cameras in parking areas, the system will continuously cover for incidents, furnishing real- time discovery of accidents. Completing this, the design's objectives include not only the establishment of a robust reporting medium but also the creation of a methodical process for incident resolution, icing nippy responses and the protection of everyone’s rights. The collected live feed data will be anatomized to identify patterns and assess accidents, contributing to long- term safety advancements. The design's compass encompasses the design and planning phase, Road Eye installation, incident operation system perpetration, and a comprehensive testing, training, and launch phase. crucial success criteria include a reduction in reported parking lot accidents, increased satisfaction among university attendants regarding lot safety, effective incident resolution, and positive feedback and engagement from the university community. While the design acknowledges implicit constraints similar as budget limitations, integration challenges, and sequestration considerations, it seeks to address these totally to achieve its overarching thing of enhancing lot safety at the University of Bahrain.

1.4 Report Outline

Chapter 1 Introduction: In this chapter, the problem statement is presented, outlining the challenges and sub-problems addressed in the project. The project objectives are defined, along with the relevance and significance of the project. Lastly, the report's structure is outlined.

Chapter 2 Literature Review: This chapter conducts a comprehensive review of the literature related to the problem statement. It explores existing research, technologies, and methodologies relevant to real-time accident detection, incident reporting, and machine learning and computer vision applications in road safety.

Chapter 3 Project Management: Chapter 3 delves into the project management aspects of the RoadEye project. It presents the chosen process model for the project, discusses risk management strategies, and provides an overview of the project activities plan.

Chapter 4 Requirement Collection and Analysis: This chapter focuses on the process of gathering and analyzing requirements for the RoadEye system. It covers requirement elicitation techniques employed, the identified system requirements, the creation of personas to understand user needs, and the development of system models.

Chapter 5 System Design: Chapter 5 explores the system design phase of the RoadEye project. It describes the architectural design, component specifications, and interfaces of the system. Additionally, it discusses design decisions and justifications for the chosen approach.

Chapter 6 System Implementation and Testing: This chapter presents the implementation details of the RoadEye system, including the integration with street cameras, the implementation of real-time accident detection algorithms, and the incident reporting mechanism. Testing methodologies and results are also discussed.

Chapter 7 Conclusion and Future Work: The concluding chapter summarizes the key findings and contributions of the RoadEye project. It provides a conclusion based on the project's objectives and discusses the limitations and potential areas for improvement. Future work and recommendations for further enhancements are also presented.

References: This section lists the references used throughout the report for citation and further reading.

Chapter 2
Literature Review

To grasp the full potential of road safety systems, a deep dive into the historical tapestry of research and practice is crucial. As the years have unfolded, a remarkable history of studies and projects has been woven together, each thread dedicated to strengthening road safety and minimizing accidents. "The landscape of accident detection has undergone a transformative shift, evolving from a reactive, chance-based approach to an initiative-taking, data-driven system empowered by cutting-edge technologies." (National Highway Traffic Safety Administration, 2023)

Let us now embark on a journey to explore some of the most significant strands in this intense advanced sector. In the past, researchers put a lot of effort into creating ways to spot and tell apart diverse types of accidents accurately. They did this by looking at information from cameras, sensors, and even the cars themselves. Different methods, like machine learning and computer vision, were used to make sure accidents were detected and classified correctly.

2.1 TrafficSense:

"TrafficSense" is a platform that utilizes anonymized mobile phone data to actively monitor real-time traffic flow and patterns. By analyzing population movements, it offers valuable insights that assist transportation authorities in optimizing traffic management strategies. Additionally, TrafficSense serves as a tool to identify potential traffic issues and implement effective solutions, enhancing traffic flow and alleviating congestion.

2.1.1 System Features:

Benefits:

1. Helps minimize passenger travel time, leading to a smoother and more efficient transportation experience.
2. Authorities can take an initiative-taking approach to create a safer environment for everyone in the parking lot.
3. Access live visual feeds of key traffic arteries and intersections.

Drawbacks:

1. TrafficSense heavily relies on mobile phone data, so there is limited effectiveness in locations experiencing low mobile phone usage or inconsistent network coverage
2. The deployment and maintenance of the TrafficSense monitoring system necessitates considerable financial investment.
3. Technical glitches or failures could impact the system's real-time monitoring and decision-making capabilities.

Figure 1: TrafficSense location points provide an accurate travel time sample

2.2 SmartCross:

“SmartCross” is a technology-driven system that improves safety and efficiency at crosswalks. It uses sensors, cameras, and intelligent algorithms to detect pedestrians and adjust traffic signals in real time, reducing the risk of accidents and optimizing traffic flow. Additionally, it often includes features to assist individuals with disabilities, making crosswalks more inclusive and accessible.

2.2.1 System Features:

Benefits:

1. SmartCross systems utilize advanced technologies to detect pedestrians and adjust traffic signals to improve their safety and visibility.
2. SmartCross systems dynamically adjust signal timings to reduce congestion and improve traffic efficiency.
3. SmartCross systems often include features like audible signals and tactile indicators to assist individuals with disabilities.

Drawbacks:

1. Significant upfront costs for installation and infrastructure upgrades.
2. Ongoing maintenance and software updates are required.
3. Technical challenges like sensor malfunctions or software glitches can affect the system's reliability.

Figure 2: SmartCross Push Button units giving the ability for 'Touch-Free' activation of pedestrian crossings.

2.3 FleetGuard:

"FleetGuard," is a comprehensive system that improves fleet efficiency, enhances safety and security, and streamlines maintenance. It analyzes data to optimize routes, minimize downtime, and identify areas for improvement.

System Features:

Benefits:

1. Optimizes routes, reduces idle time, and improves efficiency (by analyzing fuel, maintenance, and driver data).
2. Enhances safety and security (with GPS tracking, geofencing, and unauthorized use alerts).
3. Streamlines maintenance and reduces downtime (through initiative-taking scheduling and alerts).

Drawbacks:

1. Requires careful setup and integration (complex process, potential disruptions).
2. Raises data privacy concerns (sensitive data collection, security measures needed).
3. Relies heavily on technology and connectivity (backup systems recommended).

Figure 3: FleetGuard real-time monitoring and tracking of vehicles

2.4 Outcome:

The outcomes of each system align with their specific objectives and functionalities, catering to various aspects of fleet management, traffic optimization, and road safety. But we can see the main aim of the different systems and platforms is to ensure the result of accuracy and safety enhancement, so RoadEye has the same idea and will also be using similar techniques to achieve such objectives.

Chapter 3
Project Management

Project management plays a crucial role in the successful execution of any project, including the development of the RoadEye system. It involves the application of knowledge, skills, tools, and techniques to effectively plan, execute, monitor, and control project activities. This chapter focuses on project management aspects, including the process model, risk management, and the project activities plan.

The project management process ensures that the project is conducted in an organized and systematic manner, adhering to predefined goals, timelines, and quality standards. It involves the coordination and collaboration of various stakeholders, including project managers, developers, testers, and end-users, to achieve the desired outcomes.

The primary objectives of project management in the RoadEye project are to:

1. Define and understand the project requirements: Gathering and analyzing the requirements for the RoadEye system is a critical step in project management. It involves identifying the needs and expectations of stakeholders to guide the development process effectively.

2. Develop a comprehensive project plan: A well-structured project plan is created to outline the activities, timelines, resources, and deliverables of the RoadEye project. This plan serves as a roadmap for us, guiding us through the distinct phases and tasks.

3. Manage project risks: Identifying and addressing potential risks that may impact the project's success is a fundamental aspect of project management. Risk management strategies are employed to minimize the likelihood and impact of risks, ensuring smooth project execution.

4. Monitor and control project progress: Continuous monitoring and control of project activities are essential to ensure that the project stays on track. Regular updates, progress tracking, and performance evaluations are conducted to identify any deviations from the plan and take corrective actions.

5. Facilitate effective communication and collaboration: Project management involves effective communication and collaboration among us and stakeholders. Clear communication channels, regular meetings, and documentation ensure that everyone is aligned and informed throughout the project lifecycle.

By employing effective project management practices, the RoadEye project aims to achieve its objectives efficiently, deliver high-quality results, and meet stakeholder expectations.

In the following sections of this chapter, we will discuss the specific aspects of the project management process, including the chosen process model, risk management strategies, and the project activities plan. These elements provide a structured approach to the development of the RoadEye system and ensure that potential risks are identified and addressed, leading to a successful project outcome.

3.1 Process Model: Software Development Life Cycle (SDLC)

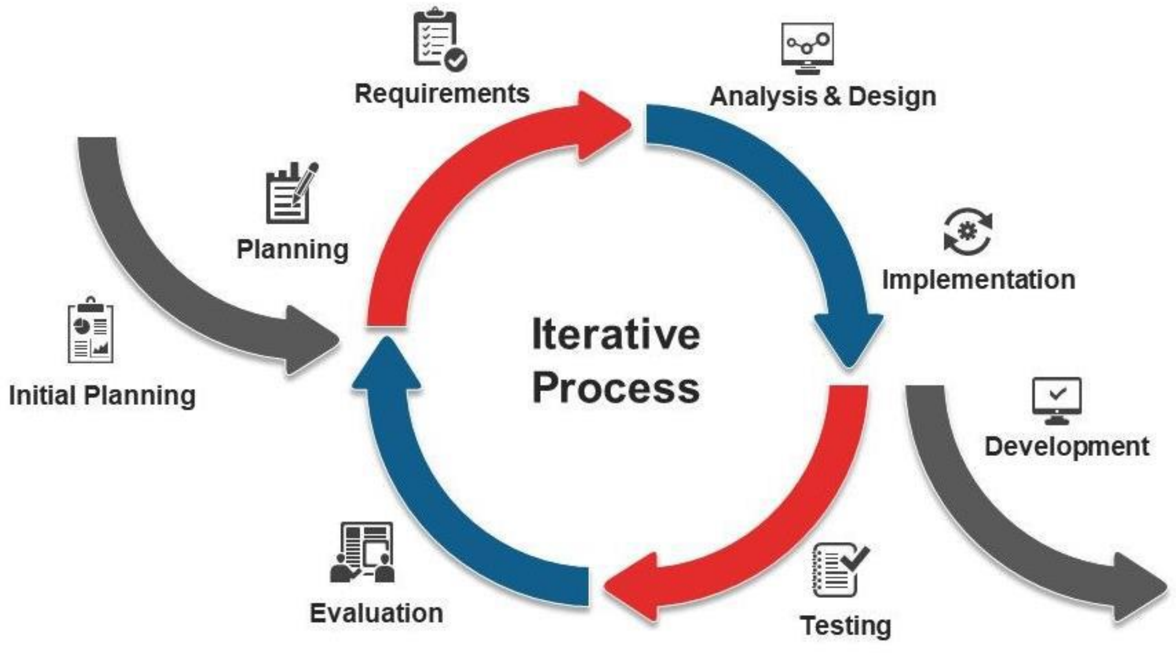


Figure 4: Software Development Life Cycle (SDLC)

The software development process model chosen for the RoadEye project is the Software Development Life Cycle (SDLC) model. This model provides a structured and systematic approach to software development, ensuring that the project progresses through distinct phases, each with its specific objectives and deliverables.

The justification for selecting the SDLC model stems from its well-established and widely used nature in the software development industry. The SDLC model offers a clear and well-defined framework that aligns with the objectives and requirements of the RoadEye project. By following this model, we can ensure a systematic and organized approach to software development, leading to a robust and high-quality system.

The SDLC model consists of several phases, each serving a specific purpose in the development process. These phases typically include requirements gathering, system design, implementation, testing, deployment, and maintenance. Each phase has defined inputs, outputs, and activities that guide the development process and enable effective project management.

In the requirements-gathering phase, the project team engages with stakeholders to elicit and document the system requirements. This involves conducting interviews, workshops, and surveys to gather information and ensure a comprehensive understanding of user needs and expectations. The outputs of this phase include a detailed requirements document that serves as a foundation for subsequent phases.

The system design phase focuses on translating the requirements into a detailed system design. This includes defining the system architecture, component specifications, and interfaces. The design phase ensures that the system is well-structured, scalable, and aligned with industry best practices.

Once the design is finalized, the implementation phase begins, where the project team starts coding and developing the RoadEye system. This phase involves writing code, integrating components, and building the system according to the defined design.

Following implementation, the testing phase is crucial to ensure the system's functionality, performance, and reliability. Various testing techniques, such as unit testing, integration testing, and system testing, are employed to identify and rectify any defects or issues.

After successful testing, the deployment phase involves the release and installation of the RoadEye system in the target environment. This phase includes activities such as system installation, user training, and data migration.

Once the system is deployed, the maintenance phase begins, focusing on ongoing support, bug fixes, and system enhancements based on user feedback and changing requirements.

By adopting the SDLC model, the RoadEye project ensures a systematic approach to software development, enabling effective project management, stakeholder collaboration, and quality assurance.

In conclusion, the SDLC model is the chosen process model for the RoadEye project due to its proven effectiveness, industry-wide acceptance, and alignment with the project's objectives and requirements. By following the SDLC model, the project team aims to achieve a well-structured, high-quality system that meets the needs of users and stakeholders, contributing to the enhancement of road safety.

3.1.1 Phases of Iterative Process Model

The iterative process model, chosen for the RoadEye project, involves several phases that enable iterative and incremental development. These phases allow for flexibility, continuous improvement, and the incorporation of feedback throughout the development process. Let us explore each phase in detail:

3.1.1.1 Initial Planning:

In the initial planning phase, we define the project objectives, scope, and constraints. This phase involves identifying key stakeholders, understanding their requirements, and establishing the project's feasibility. We also create a high-level project plan and determine the resources, budget, and timeline for the project.

3.1.1.2 Iteration Phase:

1. Planning:

In the planning phase of each iteration, we define the specific goals, tasks, and deliverables for that iteration. We also down the requirements into manageable units and create a detailed plan for the iteration.

2. Requirement:

The requirement phase focuses on gathering and refining the requirements for the iteration. The project team collaborates with stakeholders to identify and prioritize the features or functionalities to be developed. The requirements are documented, and any necessary clarifications or changes are made.

3. Analysis and Design:

In this phase, the project team analyzes the requirements and designs the system architecture, data models, and user interface. We identify the components, modules, and dependencies, and create detailed designs that guide the implementation process.

4. Implementation:

The implementation phase involves coding and developing the software based on the design specifications. We write the necessary code, integrate the components, and ensure the proper functioning of the system. This phase focuses on translating the design into a working product.

5.Testing:

The testing phase is crucial for ensuring the quality and reliability of the developed software. We conduct various testing activities, including unit testing, integration testing, and system testing. We will also identify and fix any defects or issues, ensuring that the system meets the specified requirements.

6. Evaluation:

The evaluation phase involves assessing the developed functionality against the stakeholders' expectations and requirements. We gather feedback from stakeholders, conduct user acceptance testing, and evaluate the system's performance. The feedback and evaluation results provide insights for further improvements and refinements.

7. Deployment Phase:

Finally, the deployment phase marks the release and delivery of the developed software. We prepare the system for production deployment, including installation, configuration, and data migration if necessary. User documentation and training materials are provided to support the end-users in effectively utilizing the system. The deployment phase ensures a smooth transition from development to the operational phase.

The iterative process model allows for continuous feedback, learning, and refinement throughout the development lifecycle. It promotes collaboration, adaptability, and the timely delivery of valuable software to meet the stakeholders' needs. By following this model, our RoadEye project aims to achieve successful outcomes and deliver a high-quality system.

3.2 Risk Management

Risk management is an important part of project management that entails detecting, analyzing, and managing any risks that might jeopardize a project's success. RoadEye understands the necessity of proactively managing risks to ensure project objectives are fulfilled, schedules are met, and output quality is maintained. This section focuses on the RoadEye risk management process and includes a table that details potential hazards, their categories, impacts, portability, and mitigation measures.

Table 1: Potential Risks and Mitigation Methods

Risk	Risk Type	Risk Effect	Risk Portability	Risk Mitigation Method
Technical Challenges	Project delays and system downtime	Moderate	High	Conduct thorough testing, allocate additional resources for troubleshooting, and have contingency plans for potential technical issues. Reviews and seek feedback.
Privacy Concerns	Legal issues and reputation damage	Low to moderate	High	Conduct thorough feasibility studies and technical assessments. Allocate sufficient resources and expertise. Break down complex tasks into manageable units. Regularly monitor and evaluate technical progress.
Unforeseen External Events	Project disruption	Low to moderate	High	Establish a robust change management process. Clearly define and document project scope. Evaluate and prioritize change requests. Obtain proper approval before incorporating changes.

Integration Challenges	Technical	Moderate to high	Medium	Plan and test integration activities early in the development cycle. Define integration interfaces and protocols. Conduct thorough integration testing. Monitor and address any compatibility issues.

Note: The table above provides representations of potential risks for the RoadEye project. It is important to tailor the risks and mitigation methods to the specific context and requirements of the project.

The table presents a comprehensive overview of potential risks, classified by risk type, their risk effects, portability, and mitigation methods. By identifying and addressing these risks, RoadEye aims to minimize their impact and ensure the successful delivery of the project within the defined parameters. Regular risk monitoring and mitigation activities will be conducted throughout the project lifecycle to effectively manage any emerging risks and maintain project progress.

3.3 Project activities Plan

The project activities plan outlines the breakdown of RoadEye into different activities, providing a clear timeline and sequence of tasks needed for the successful delivery of the system. This plan ensures efficient project management, task assignment, and monitoring of progress. The following is an example of how the project activities plan for the RoadEye project could be structured:

1. Requirements Gathering Phase (Duration: 2 weeks)
   - Conduct stakeholder interviews and workshops to gather system requirements.
   - Analyze and document the collected requirements.
   - Review and finalize the requirements document.

2. System Design Phase (Duration: 3 weeks)
   - Define the system architecture and component specifications.
   - Develop system design documents, including system flowcharts and data models.
   - Conduct design reviews and incorporate feedback.

3. Implementation Phase (Duration: 6 weeks)

- Set up the development environment and infrastructure.
- Develop the core functionalities of the RoadEye system based on the design specifications.
- Conduct regular code reviews and unit testing to ensure quality.

4. Testing Phase (Duration: 4 weeks)

- Perform unit testing to verify the functionality of individual components.
- Conduct integration testing to ensure proper interaction between different system modules.
- Perform system testing to validate the overall system performance and functionality.

5. Deployment Phase (Duration: 2 weeks)

- Prepare the RoadEye system for deployment, including system installation and configuration.
- Conduct user acceptance testing to ensure system usability and satisfaction.
- Develop user documentation and provide training to end-users.

6. Maintenance and Support Phase (Ongoing)

- Address and resolve any reported bugs or issues.
- Provide ongoing support and maintenance to ensure system stability and performance.
- Incorporate user feedback to make necessary enhancements and improvements.

Chapter 4
Requirement Collection and Analysis

The requirement collection and analysis phase is a crucial aspect of project management as it involves creating a clear and agreed set of customer requirements. These requirements serve as the foundation for delivering a system that aligns with the customer's needs and expectations. This chapter discusses the functional and non-functional requirements of the system, along with the data flow diagram and use case diagram, which aid in understanding the system's behavior and functionalities.

4.1 Requirement Elicitation

4.1.1 Introduction

Requirement elicitation is a crucial phase in project management, as it involves collecting the system requirements that will guide the development process. This report provides an overview of the requirement elicitation methods employed to gather the necessary information. The methods discussed include interviews, questionnaires, observations, and quantitative and qualitative data analysis.

4.1.2 Interviews

Here are some of the questions we asked some stakeholders and they answered:

1. Can you please describe your role and responsibilities within the organization or domain related to street accident monitoring?

- Stakeholder A: I am a traffic police officer responsible for monitoring and managing traffic flow and safety in our city.
- Stakeholder B: I am a fleet manager for an organization company and oversee the safety and efficiency of our fleet operations.
- Stakeholder C: I am a traffic engineer working for the local transportation department and focus on optimizing traffic management strategies.

2. What are the main challenges or main points you currently face in monitoring and responding to street accidents?

- Stakeholder A: One of the biggest challenges is identifying accidents quickly and efficiently, especially during peak traffic hours.
- Stakeholder B: It is often difficult to gather accurate and timely information about accidents, resulting in delays and disruptions in our operations.
- Stakeholder C: We struggle with analyzing accident data to identify patterns and make informed decisions for traffic management and infrastructure improvements.

3. How do you currently collect and analyze data related to street accidents? What tools or systems do you use?

- Stakeholder A: We rely on manual reports from police officers at the scene, as well as CCTV camera footage for accident analysis.
- Stakeholder B: We primarily rely on incident reports from drivers and occasionally use dashcam footage for accident investigation.
- Stakeholder C: We collect accident data from various sources, including police reports, traffic cameras, and data from emergency services.

4. What specific features or functionalities do you think are essential for a street accident monitoring system like RoadEye? Why?

- Stakeholder A: Real-time notifications and accurate accident detection are crucial for us to respond quickly and efficiently.
- Stakeholder B: Integration with our existing fleet management system and the ability to track accidents in real-time would benefit our operations.
- Stakeholder C: Advanced analytics and visualization capabilities would help us identify accident-prone areas and make data-driven decisions for traffic management.

5. How frequently do street accidents occur in your area, and what impact do they have on traffic flow and safety?

- Stakeholder A: Street accidents are quite frequent in our area, especially during rush hours, resulting in major traffic congestion and safety concerns.
- Stakeholder B: Accidents are common, and they have a significant impact on our delivery schedules and overall fleet efficiency.
- Stakeholder C: The frequency of accidents varies, but even minor incidents can disrupt traffic flow and lead to cascading congestion issues.

6. What are your expectations regarding the accuracy and reliability of a street accident monitoring system like RoadEye?

- Stakeholder A: We expect the system to accurately detect and report accidents in real-time to ensure prompt response and appropriate traffic management measures.
- Stakeholder B: It is crucial for the system to provide reliable and up-to-date information about accidents to minimize disruptions and optimize our fleet operations.
- Stakeholder C: We need a highly accurate system that can provide reliable accident data for accurate analysis and decision-making.

7. How important is real-time notification and alerting for street accidents, and how would you prefer to receive these notifications (e.g., mobile app, email, SMS)?

- Stakeholder A: Real-time notification is of utmost importance, and I prefer receiving them through a dedicated mobile app that allows quick access to relevant information.
- Stakeholder B: Real-time notification is essential to take immediate action, and I prefer receiving them through SMS notifications for quick and convenient access.
- Stakeholder C: Real-time alerts are crucial for effective traffic management, and I prefer receiving them through email notifications for easy access and documentation.

8. Are there any specific requirements or considerations related to the integration of the RoadEye system with existing infrastructure or systems in your organization?

- Stakeholder A: Integration with our existing CCTV camera network and police communication systems would be beneficial for seamless coordination and incident response.
- Stakeholder B: Integration with our fleet management system and GPS tracking devices would enable us to monitor accidents and reroute our vehicles efficiently.
- Stakeholder C: The RoadEye system should be compatible with our existing traffic monitoring and management systems to ensure smooth operations and data exchange.

9. What kind of data visualization or reporting capabilities would you find most valuable in the RoadEye system? Are there any specific metrics or insights you would like to see?

- Stakeholder A: Interactive maps displaying accident locations and severity levels would be helpful, along with statistical reports on accident trends and patterns.
- Stakeholder B: Real-time dashboards showing accident hotspots and historical trends would be valuable, along with detailed incident reports including time, location, and contributing factors.

- Stakeholder C: Visualizations of accident data on a geographic information system (GIS) platform, along with customizable reports on accident frequencies and impact on traffic flow, would be beneficial.

10. In your opinion, what are the potential privacy or ethical concerns associated with implementing a street accident monitoring system like RoadEye, and how should they be addressed?

- Stakeholder A: Privacy concerns may arise regarding the collection and storage of accident data. To address this, the system should comply with relevant data protection laws and ensure that data is anonymized and securely stored.
- Stakeholder B: There could be concerns about the use of dashcam footage and location tracking. It is essential to inform users about data collection practices, obtain consent, and establish clear guidelines for data usage and retention.
- Stakeholder C: Ethical considerations may arise regarding the use of accident data for purposes other than traffic management. Transparency and clear communication about data usage and ensuring data is used only for its intended purposes can address these concerns.

11. How would you measure the success or effectiveness of the RoadEye system? What key performance indicators (KPIs) or metrics would you use?

- Stakeholder A: Success can be measured by the system's ability to provide real-time accident alerts and reduce response time. KPIs would include average response time, accuracy of accident detection, and reduction in traffic congestion after incidents.
- Stakeholder B: Effectiveness can be measured by the system's impact on minimizing delays caused by accidents. KPIs could include the number of incidents reported, average time to resolve incidents, and reduction in delivery delays.
- Stakeholder C: Success can be measured by the system's contribution to identifying accident patterns and informing traffic management decisions. KPIs would include accident frequency, identification of high-risk areas, and the effectiveness of implemented measures.

12. What level of training or support would be necessary for users to effectively utilize the RoadEye system?

- Stakeholder A: Users would require comprehensive training on operating the RoadEye system, interpreting accident data, and effectively coordinating with other stakeholders during incident response.

- Stakeholder B: Training should focus on integrating the RoadEye system with existing fleet management tools, understanding the user interface, and effectively utilizing the real-time accident information for rerouting and planning.
- Stakeholder C: Training should cover data analysis techniques, using visualization tools, and understanding the system's integration with other traffic management systems.

13. Are there any specific regulations or compliance requirements that need to be considered when implementing the RoadEye system?
- Stakeholder A: Compliance with privacy laws, data protection regulations, and any specific regulations related to handling accident data should be ensured during the implementation of the RoadEye system.
- Stakeholder B: Compliance with regulations related to data security, GPS tracking, and incident reporting should be considered to ensure that the RoadEye system meets all legal requirements.
- Stakeholder C: Compliance with regulations related to traffic management systems, data sharing, and interoperability should be considered during the implementation of the RoadEye system.

14. Can you provide any examples or scenarios where the RoadEye system would have a significant impact on improving street accident monitoring and response?
- Stakeholder A: With the RoadEye system, we can receive real-time notifications about accidents, enabling us to quickly dispatch emergency services and implement traffic diversions, reducing response time and minimizing congestion.
- Stakeholder B: The RoadEye system can help us proactively identify accident-prone areas and reroute our vehicles, minimizing delays and ensuring efficient operations.
- Stakeholder C: By analyzing historical accident data and identifying patterns, the RoadEye system can help us implement targeted traffic management measures, reducing the frequency and severity of accidents in certain areas.

15. Is there anything else you would like to share or any additional requirements or considerations that should be considered in the development of the RoadEye system?
- Stakeholder A: It would be beneficial for the RoadEye system to have the ability to integrate with our existing incident management systems and provide seamless communication channels for coordinated response efforts.

- Stakeholder B: The RoadEye system should be scalable to accommodate the growing number of vehicles and stakeholders in our fleet operations, ensuring optimal performance even under high load.
- Stakeholder C: The RoadEye system should allow for easy data sharing and collaboration between different departments and agencies involved in traffic management, enabling effective coordination and decision-making.

4.1.3 Conclusion

In conclusion, the answers provided by the stakeholders shed light on the various challenges, requirements, and expectations related to street accident monitoring. The stakeholders, representing distinct roles such as traffic police officers, fleet managers, and traffic engineers, all emphasize the importance of real-time accident detection, accurate data analysis, and efficient response in ensuring traffic flow and safety.

Common themes that emerge from the responses include the need for integration with existing systems and infrastructure, such as CCTV cameras, fleet management tools, and traffic management systems. Stakeholders also express the desire for advanced analytics and visualization capabilities to identify accident patterns, hotspots, and trends.

Privacy and ethical concerns surrounding data collection, storage, and usage are highlighted, with stakeholders emphasizing the importance of compliance with regulations and the need for transparency in data handling practices.

The success of a street accident monitoring system like RoadEye is measured by its ability to provide real-time notifications, reduce response time, minimize traffic congestion, and optimize operations. Key performance indicators (KPIs) mentioned include average response time, accuracy of accident detection, reduction in delivery delays, identification of high-risk areas, and the effectiveness of implemented measures.

Training and support for users are deemed essential to effectively utilize the RoadEye system, with a focus on operating the system, interpreting data, and integrating it with existing tools and processes.

Overall, the stakeholders provide valuable insights into the requirements and considerations that should be considered in the development of the RoadEye system. Their feedback emphasizes the importance of seamless integration, accurate data analysis, real-time notifications, privacy compliance, and user training to ensure the system's effectiveness in monitoring and responding to street accidents.

The information gathered through requirement elicitation will serve as the foundation for the subsequent phases of the project, including system design and implementation. By effectively eliciting and analyzing requirements, we aim to deliver a system that aligns with the customer's needs and expectations.

In the next phase, we will proceed with the analysis and documentation of the system requirements based on the information gathered during the requirement elicitation process.

4.2 System Requirements

4.2.1 introduction

System requirements are essential for defining the behavior and features of software applications. This report presents the functional and non-functional requirements gathered during the requirement analysis phase. Functional requirements describe specific functionalities and features that the system should provide, while non-functional requirements specify the qualities or characteristics of the system.

4.2.2 Functional Requirements

Functional requirements outline the actions and behaviors that the system should perform to meet user needs. Based on the requirement analysis, we have identified the following functional requirements:

1. The system should be able to detect accidents in real-time and notify the appropriate authorities.
2. Users should be able to report incidents and provide relevant details, such as recorded live feed and description.
3. The system should generate incident reports with the real-time incident feed.
4. Users should be able to track the status and progress of reported incidents through email.
5. Users should be able to access historical incident data for analysis and reporting purposes.
6. The system should save the incident live feeds on a cloud to enhance distribution and storage.
7. Users should be able to receive real-time notifications and updates on incidents.
8. The system should allow for collaboration and communication among different stakeholders involved in incident management.
9. Users should be able to search and filter incident data based on various criteria, such as date, location, or severity.

4.2.3 Non-functional Requirements

Non-functional requirements address the qualities and characteristics of the system that are not related to specific functionalities. The following non-functional requirements have been identified:

1. Performance: The system should have fast response times to ensure prompt incident management.

2. Security: The system should implement robust security measures to protect sensitive user data.

3. Usability: The system should be intuitive to facilitate ease of use.

4. Reliability: The system should be highly dependable and available, minimizing downtime.

5. Scalability: The system should be able to manage a growing number of users and incidents without compromising performance.

6. Compatibility: The system should be compatible with a wide range of cameras, operating systems , etc...

7. Data Privacy: The system should ensure the privacy and confidentiality of user data, adhering to relevant data protection regulations.

8. System Integration: The system should seamlessly integrate with existing systems and clouds for data exchange and interoperability.

9. Documentation: The system should have comprehensive and up-to-date documentation to assist users and administrators.

4.2.4. Conclusion

The system requirements outlined in this report provide a clear understanding of the functionalities and characteristics expected from the software application. The functional requirements define the specific actions and behaviors of the system, while the non-functional requirements address important qualities like performance, security, and usability. These requirements will serve as a basis for the system design and development phases, ensuring that the final product meets the customer's needs and expectations.

In the next phase, we will proceed with the system design, using the requirements gathered in this report as a guide.

4.3 Personas

Personas play a crucial role in understanding the characteristics and behaviors of the system/application users. This report provides an overview of the personas developed based on user research and their relevance to the system/application being developed.

4.3.2 Persona Development Process

The development of personas involved conducting user research through various methods, including interviews, surveys, and observations. The collected data was analyzed to identify common patterns, needs, and goals among the target user groups. The personas were then created to represent typical users and provide insight into their motivations, preferences, and expectations.

4.3.3 Personas

Based on the user research conducted, the following personas have been developed:

1. Persona 1: Safety Officer Mohamed

- Background: Works as a safety officer in a transportation company.
- Goals: Ensuring the safety of drivers, vehicles, and cargo.
- Needs: Real-time incident detection, accurate incident reporting, and effective communication with emergency services.
- Challenges: Managing incidents and coordinating resources efficiently.

2. Persona 2: Commuter Ali

- Background: Regular commuter using public transportation.
- Goals: Safe and reliable travel experience.
- Needs: Timely incident notifications, access to incident reports, and alternative route suggestions.
- Challenges: Avoiding congested routes and potential accidents during daily commutes.

3. Persona 3: Emergency Services Officer Abdulla

- Background: Works in an emergency services department.
- Goals: Swift response and effective incident management.
- Needs: Real-time incident notifications, accurate incident details, and seamless communication with other stakeholders.
- Challenges: Coordinating with multiple agencies and responding to incidents promptly.

4. Persona 4: Fleet Manager Khaled

- Background: Manages a fleet of vehicles for an organization company.
- Goals: Ensuring vehicle safety and minimizing disruptions in operations.
- Needs: Monitoring incidents, analyzing incident data, and implementing preventive measures.
- Challenges: Minimizing vehicle downtime due to incidents and optimizing fleet performance.

5. Persona 5: Administrator Yousef

- Background: Responsible for system administration and user management.
- Goals: Ensuring system functionality and user satisfaction.
- Needs: User-friendly interface, efficient incident management tools, and customizable system settings.
- Challenges: Managing user permissions, resolving system issues, and ensuring system reliability.

4.3.4 Conclusion

The personas outlined in this report provide valuable insights into the characteristics and needs of the system/application users. By understanding their goals, needs, and challenges, we can design and develop a system/application that effectively meets their requirements. The personas will serve as a reference throughout the development process, enabling user-centric design decisions and enhancing the overall user experience.

In the next phase, we will utilize the personas to inform the system/application design and ensure that the developed solution aligns with the expectations and preferences of the target users.

4.4 System Models

4.4.1 UML diagrams

4.4.1.1 Use case diagram

A Use Case Diagram is a visual representation of the system's functionalities from the perspective of its users, known as actors. It provides an overview of the system's boundaries, interactions, and the specific tasks that users can perform. Use Case Diagrams are effective in capturing high-level user requirements and understanding the system's behavior in terms of user actions and system responses. By identifying actors and their relationships with use cases, this diagram helps stakeholders visualize the system's functionality and serves as a foundation for further analysis and design.

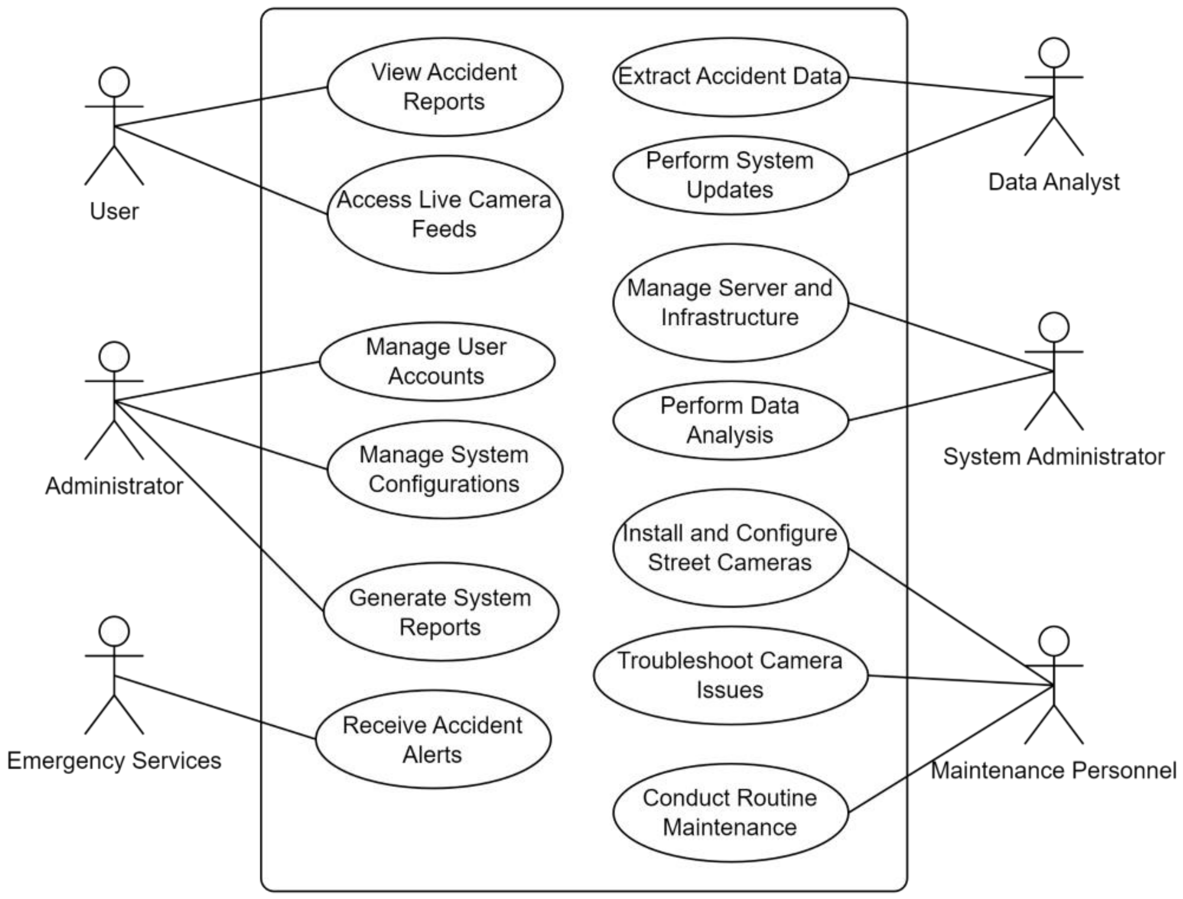


Figure 5: Use case diagram

4.4.1.2 UML Class Diagram

A UML Class Diagram is a static representation of the system's structure in terms of classes, their attributes, and their relationships. It captures the essential components of the system and their associations, enabling a comprehensive understanding of the system's data and behavior. Class Diagrams illustrate the relationships between classes, such as inheritance, associations, and dependencies. They provide a foundation for designing the internal structure of the system, facilitating code generation, and ensuring a well-organized and maintainable system. With its clear visualization of the system's structure, a Class Diagram assists in communication among developers, analysts, and stakeholders.

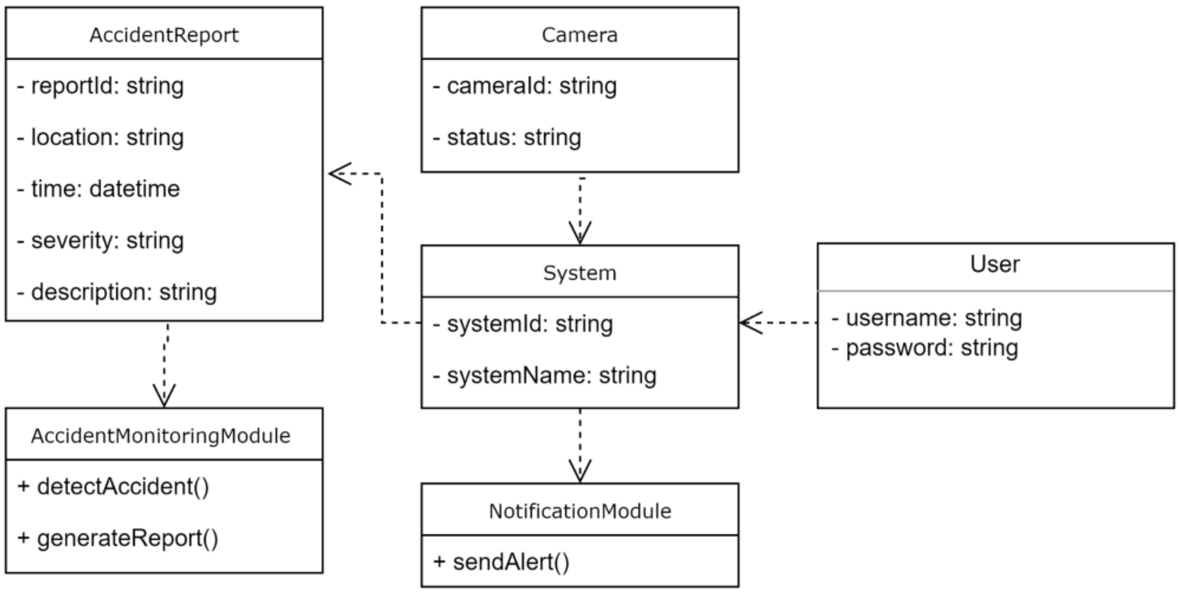


Figure 6: UML Class Diagram

4.4.2 AWS Model

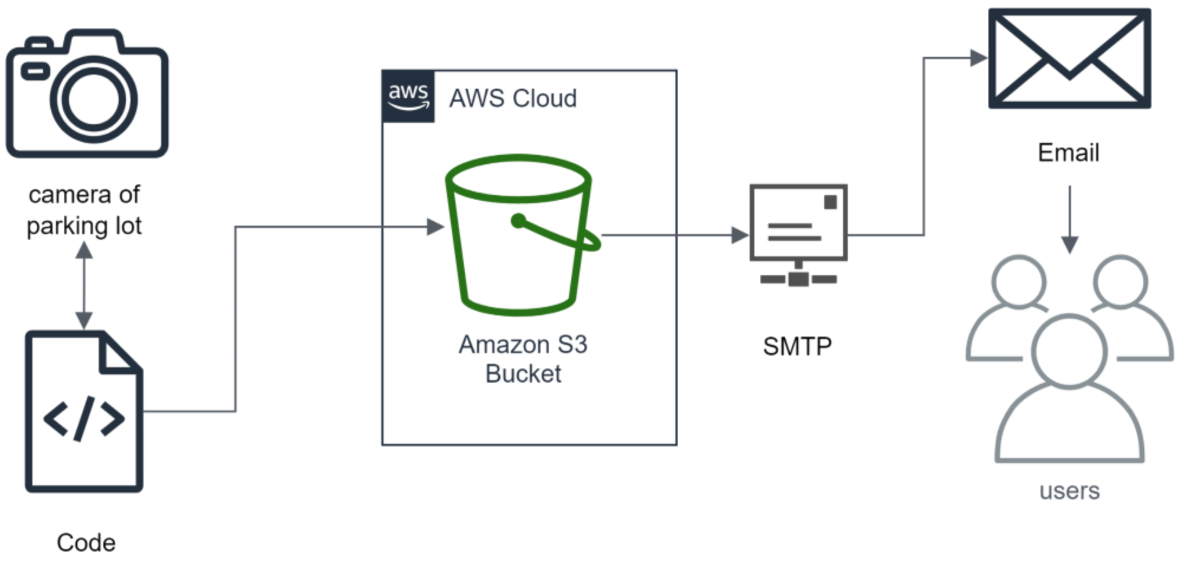


Figure 7: AWS Model

Chapter 5
System Design

5.1 Software Architecture

The system design of our project, which involves monitoring accidents in the parking lot using machine learning and cameras, will be based on a layered architecture. This architecture style provides a clear separation of concerns and allows for modular and scalable development. The layers in our system design include the presentation layer, application layer, and data layer.

5.1.1 The Reference Architecture

The chosen reference architecture for our system is the Client-Server architecture. This architecture style enables the separation of the client-side application, responsible for user interaction and presentation, from the server-side application, responsible for data processing and storage. The client-side application will manage the user interface and communication with the cameras, while the server-side application will perform the machine learning algorithms and store the data.

5.1.2 Architecture Drivers

The system design of our project is driven by several architecture drivers that ensure the effectiveness and success of the system.

5.1.2.1 Design Purpose

The design purpose of our system is to provide real-time monitoring and analysis of accidents occurring in the parking lot. The system aims to detect accidents through machine learning algorithms and provide timely alerts and information to relevant parties, such as emergency services and traffic authorities.

5.1.2.2 Quality Attributes

The system design is also driven by various quality attributes, including:

1. Accuracy: The machine learning algorithms should accurately detect and classify accidents based on the camera footage.

2. Real-time: The system should provide real-time monitoring and alerts to ensure immediate response to accidents.

3. Scalability: The architecture should be scalable to manage many cameras and process data efficiently.

4. Reliability: The system should be dependable, ensuring minimal downtime and robustness in handling various scenarios.

5.1.2.3 Constraints

Certain constraints need to be considered in the system design, including:

1. Language Constraint: The system will be developed using the Python programming language to leverage the machine learning libraries and frameworks available in Python.

2. Database Constraint: The system does not utilize a database, but it uses AWS S3 Bucket for storing and managing the camera footage data, incident records, and relevant information.

3. Hardware Constraint: The system design should consider the hardware requirements for deploying the machine learning algorithms and storing the camera footage data.

4. Budget Constraint: The design should be cost-effective, considering the available budget for hardware, software, and maintenance.

5.2 Machine Learning Algorithms

The system design will include the implementation of machine learning algorithms for accident detection and classification. Python provides various libraries, such as TensorFlow or scikit-learn, that offer pre-trained models and tools for developing and training custom machine learning models. The design will include the integration of these algorithms into the server-side application for real-time accident detection and analysis.

5.3 Deployment Diagram

The deployment diagram will illustrate the physical deployment of the system components, including cameras, servers, and user devices. It will highlight the distribution and configuration of the hardware and software components. The deployment diagram will provide a visual representation of how the system is deployed in different environments, such as on-premises or in the cloud.

In conclusion, this chapter has discussed the system design of our project, which involves monitoring accidents in the parking lot using machine learning and cameras. The chosen architecture style is layered architecture, with a client-server reference architecture. The design is driven by various architecture drivers, including the design purpose, quality attributes, and constraints. The camera interface for monitoring accidents, and AWS S3 bucket is the solution for storing and managing the data. The system design also includes the implementation of machine learning algorithms for accident detection and classification. Finally, the deployment diagram displays the physical deployment of the system components. Overall, the system design ensures a well-structured and efficient approach to monitoring accidents in the parking lot using machine learning and cameras.

Chapter 6
System Implementation and Testing

6.1 Introduction

This chapter focuses on the implementation details of the system and the testing phases that were conducted. It discusses the selection and integration of different components, including hardware, software tools, algorithms, programming languages, and cloud providers. The implementation decisions are justified based on their suitability for the project requirements. Additionally, the chapter reports the results of the testing phases, including usability and user-experience testing. Strengths and weaknesses of the proposed system are highlighted and discussed.

6.2 Tools and Technologies

A variety of programs and languages were used to create the medication system. In-depth details about it are provided in the subsections below.

6.2.1 Software tools used:

For the development of this project a wide variety of tools were used to complete it:

6.2.2 Programming Language

Python served as the primary programming language for implementing various components of RoadEye. From developing the backend infrastructure to creating the alert system, Python facilitated seamless integration and ensured the efficient functioning of the entire system.

6.2.3 Google Collab

Google Collab provided a collaborative and cloud-based environment for our development team. Leveraging its resources, we conducted model training sessions, particularly for YOLOv8, taking advantage of its GPU capabilities to accelerate the training process and improve the accuracy of our object detection model.

6.2.4 Communication

1. **GitHub:**

   GitHub served as our version control repository, facilitating collaborative development among us as members. It ensured that the codebase remained organized, versioned, and accessible, fostering efficient collaboration, and enabling the tracking of changes over time.

2. **Google Docs:**

   Google Docs offers a collaborative environment for creating, editing, and sharing documents. In our project, Google Docs was employed for collaborative documentation, such as project plans, meeting minutes, and other shared resources. The real-time editing feature ensured that all team members have access to the latest information.

3. **Microsoft Teams:**

   Microsoft Teams was employed as a central hub for team communication. It facilitated real-time chat, video conferencing, and file sharing. Within our project, Teams served as a platform for daily stand-up meetings, progress updates, and general team collaboration.

4. **Discord:**

   Discord, like Microsoft Teams, provides a platform for real-time communication. It was used for informal team discussions, quick queries, and sharing files among us. It is particularly useful for its ease of use and flexibility.

5. **WhatsApp Application:**

   WhatsApp or was used for instant messaging where it is used for quick updates, urgent notifications, or informal communication among us. It provides a convenient and widely accessible means of communication.

6.2.5 Computer Vision Libraries

1. YOLOv8 (You Only Look Once version 8) played a crucial role in our project for real-time object detection. Specifically, we employed YOLOv8 to enable the Road Eye system to recognize and identify accidents. This technology enhanced our surveillance capabilities and provided accurate and rapid detection of incidents.

2. OpenCV, or cv2 in Python, was instrumental in image processing and manipulation within the Road Eye system. It provided essential functions for handling images and video streams, allowing us to implement various computer vision tasks seamlessly.

6.2.6 AWS (Amazon Web Services):

We utilized AWS to host and deploy the Road Eye system. The scalability and reliability of AWS services ensured that our solution could manage varying loads and maintain high performance. Additionally, AWS's secure infrastructure contributed to the overall robustness of our system.

6.2.7 S3 Bucket:

Amazon S3 (Simple Storage Service) buckets within AWS were employed to store and manage the large volumes of data generated by the Road Eye system. This included storing images and video footage captured by surveillance cameras, enabling seamless data access and retrieval for analysis.

6.2.8 SMTP (Simple Mail Transfer Protocol):

SMTP was integrated into our system to enable email notifications for incident reports. When the Road Eye system detected an accident, it triggered automated email alerts through SMTP, ensuring that relevant authorities were promptly notified.

6.2.9 Roboflow:

Roboflow served as a valuable tool for preprocessing and augmenting our dataset before model training. Its capabilities in data transformation and enhancement contributed to improving the overall performance and accuracy of our YOLOv8 model.

6.2.10 Boto3:

Boto3, the Python SDK for AWS, was utilized to interact with AWS services programmatically. It enabled seamless communication with S3 buckets, facilitating data storage, retrieval, and management within the AWS infrastructure.

6.3 Implementation Details

In this section, we dive into the technical aspects of the Road Eye system, detailing the implementation of key components that contribute to the overall functionality and success of the project.

6.3.1 YOLOv8 Integration for Real-Time Object Detection

The implementation of YOLOv8, an integral component of the Road Eye system, involved the following steps:

Data Preparation:

A comprehensive dataset of parking lot images was collected, annotated, and preprocessed to train the YOLOv8 model. This dataset included various scenarios, lighting conditions, and vehicle types to ensure robust detection capabilities.

Model Training:

YOLOv8 was trained on the prepared dataset using Google Colab's GPU acceleration. This accelerated the training process and improved the accuracy of object detection. Multiple iterations were performed to fine-tune the model for campus-specific conditions.

Integration with Road Eye System: The trained YOLOv8 model was seamlessly integrated into the Road Eye system. Real-time video feeds from surveillance cameras were processed through the YOLOv8 algorithm to detect and classify vehicles in areas.

6.3.2 Python-Based Backend Development

The backend of the Road Eye system was developed using Python to ensure flexibility, scalability, and compatibility with various components. Key implementation details include:

Database Management:

We did not use a database, but we used an AWS S3 bucket called “seniorwatertub” to store and manage information related to incidents, user data, and system logs. The backend was responsible for efficiently managing data transactions and ensuring data integrity.

6.3.3 AWS Infrastructure for Hosting and Deployment

Amazon Web Services (AWS) played a vital role in hosting and deploying the Road Eye system. Key implementation steps include:

Deployment on AWS EC2 Instances:

The Road Eye system was deployed on AWS EC2 instances, providing scalable computing resources. This ensured the system's performance could be adjusted according to varying workloads.

Utilization of S3 Buckets:

Amazon S3 buckets were employed for secure storage and retrieval of large volumes of image and video data captured by the surveillance cameras. S3's reliability and scalability were crucial for managing diverse data types.

6.3.4 Integration with Communication Tools

The Road Eye system was integrated with communication tools to facilitate user interaction and incident reporting. Key details include:

SMTP for Email Notifications:

SMTP (Simple Mail Transfer Protocol) was implemented to send automated email notifications to relevant authorities when an incident was detected. This ensured timely communication and response.

6.3.5 Collaborative Development using GitHub.

Collaborative development practices were employed using GitHub for version control. Key implementation details include:

1. **Branching Strategy:**

A well-defined branching strategy was implemented to organize development, testing, and deployment phases. Feature branches were used for individual components, ensuring a systematic and organized development process.

2. **Continuous Integration:**

GitHub Actions were utilized for continuous integration, automating testing processes and ensuring that the codebase remained stable throughout development.

6.4 Codes and algorithm:

*****using google colab*****

```
from google.colab import drive

drive.mount('/content/drive')
```

*****training the model*****

```
!yolo task=detect mode=train model=yolov8n.pt data=/content/datasets/data.yaml epochs=15
```

*****evaluating the model*****

```
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/drive/MyDrive/ColabNotebooks/testing.mp4 save=true
```

*****saving the results*****

```
!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt data=/content/datasets/data.yaml save=true
```

*****app.py*****

```
from live import predict_accident

from send_email import send_email_w_attachment


# Predicting Accident

predict_accident("s3://seniorwatertub/1.avi")


# Send email

to = "xxxxx@gmail.com"

…

…


filename = "s3://seniorwatertub/1.avi"


#calling the email function

send_email_w_attachment(to, subject, body, filename)
```

*****send_email.py*****

```
import smtplib
import os
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from email.header import Header
import email_config as ec
def send_email_w_attachment(to, subject, body, filename):
    message = MIMEMultipart()
    message['From'] = Header(ec.user)
    message['To'] = Header(to)
    message['Subject'] = Header(subject)
    message.attach(MIMEText(body, 'plain', 'utf-8'))
...
...
    server = smtplib.SMTP_SSL(ec.host, ec.port)
    server.login(ec.user, ec.gmail_pass)
    server.sendmail(ec.user, to, message.as_string())
    server.quit()
```

*****upload to S3*****

```
def upload_to_s3(local_file_path, s3_bucket, s3_key):
    s3 = boto3.client('s3')
    s3.upload_file(local_file_path, s3_bucket, s3_key)

# Upload the video to AWS S3
    s3_bucket = 'seniorwatertub'
    s3_key = '1.avi'
    upload_to_s3('1.avi', s3_bucket, s3_key)
```

*****run the prediction*****

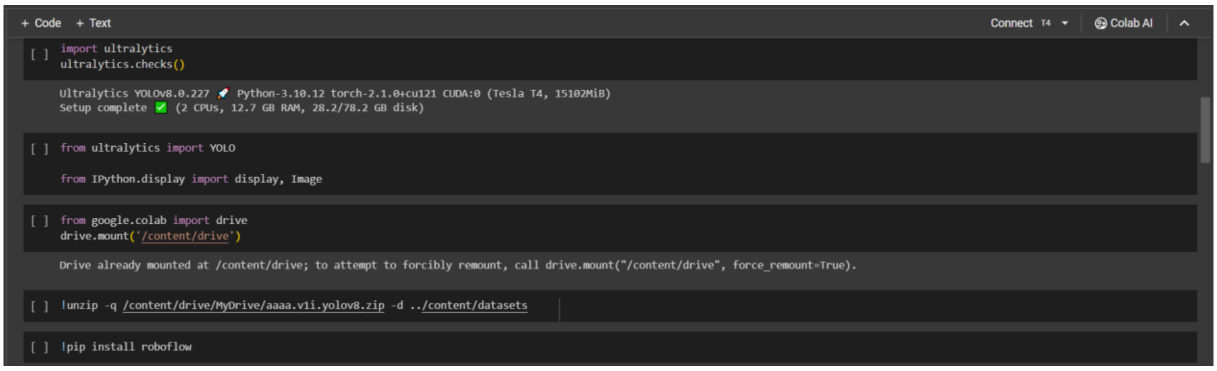
```
def predict_accident(filename):
    model = YOLO("D:\RoadEye\ROADEYE\yolov8n.pt")
```

*****make a playback using cv2*****

```
    # Start capturing video
    cap = cv2.VideoCapture(0)
...
...
    fourcc = cv2.VideoWriter_fourcc(*'XVID')  # Codec
    out = cv2.VideoWriter(filename, fourcc, fps, (width, height))
...
...
    cap.release()
    out.release()
    cv2.destroyAllWindows()
predict_accident('s3://senior water tub/1.avi')
```

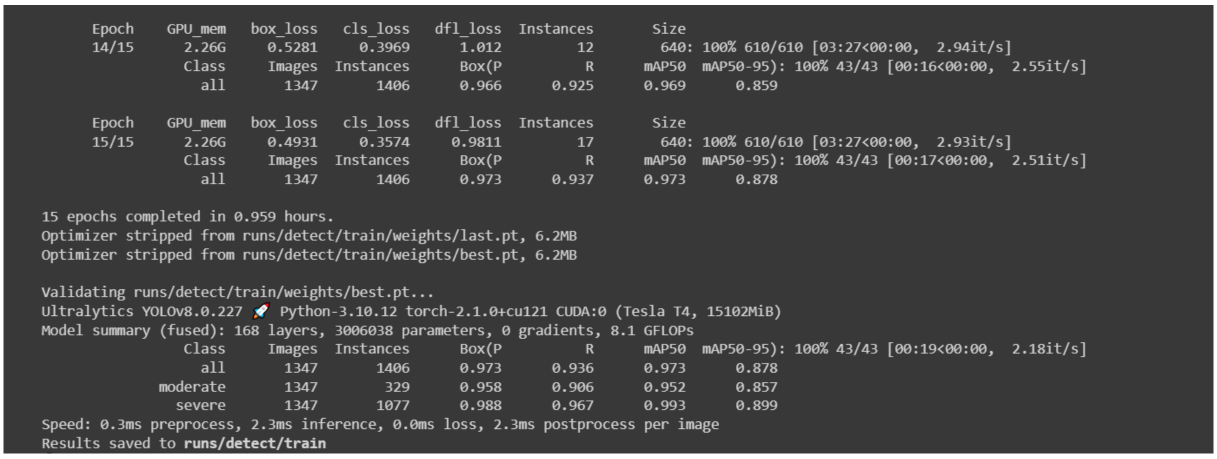
6.5 Testing Phases and Results

We used google collab to do our testing, as it's one of the best and free software to run any machine learning code as it provides online clusters with a TGPU, CPU and RAM. We used a pre-trained model from roboflow and trained the model using YOLOv8, until we reached a MAP50 of **0.973**; about 15 Epochs to reach an accuracy of **97.3%**



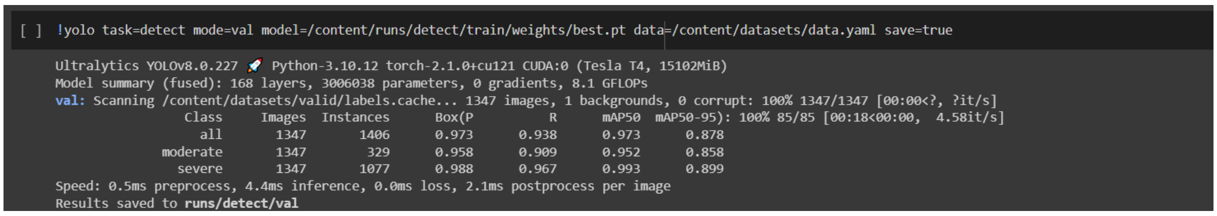
```
[ ] import ultralytics
    ultralytics.checks()

    Ultralytics YOLOv8.0.227 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
    Setup complete ✅ (2 CPUs, 12.7 GB RAM, 28.2/78.2 GB disk)

[ ] from ultralytics import YOLO

    from IPython.display import display, Image

[ ] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] !unzip -q /content/drive/MyDrive/aaaa.v1i.yolov8.zip -d ../content/datasets

[ ] !pip install roboflow
```

Figure 8: Mounting and unzipping the dataset on google drive



```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      14/15      2.26G     0.5281     0.3969      1.012         12        640: 100% 610/610 [03:27<00:00,  2.94it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 43/43 [00:16<00:00,  2.55it/s]
                   all       1347       1406      0.966      0.925      0.969      0.859

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      15/15      2.26G     0.4931     0.3574     0.9811         17        640: 100% 610/610 [03:27<00:00,  2.93it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 43/43 [00:17<00:00,  2.51it/s]
                   all       1347       1406      0.973      0.937      0.973      0.878

15 epochs completed in 0.959 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.227 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 43/43 [00:19<00:00,  2.18it/s]
                   all       1347       1406      0.973      0.936      0.973      0.878
              moderate       1347        329      0.958      0.906      0.952      0.857
                severe       1347       1077      0.988      0.967      0.993      0.899
Speed: 0.3ms preprocess, 2.3ms inference, 0.0ms loss, 2.3ms postprocess per image
Results saved to runs/detect/train
```

Figure 9: Training the model using YOLOv8n

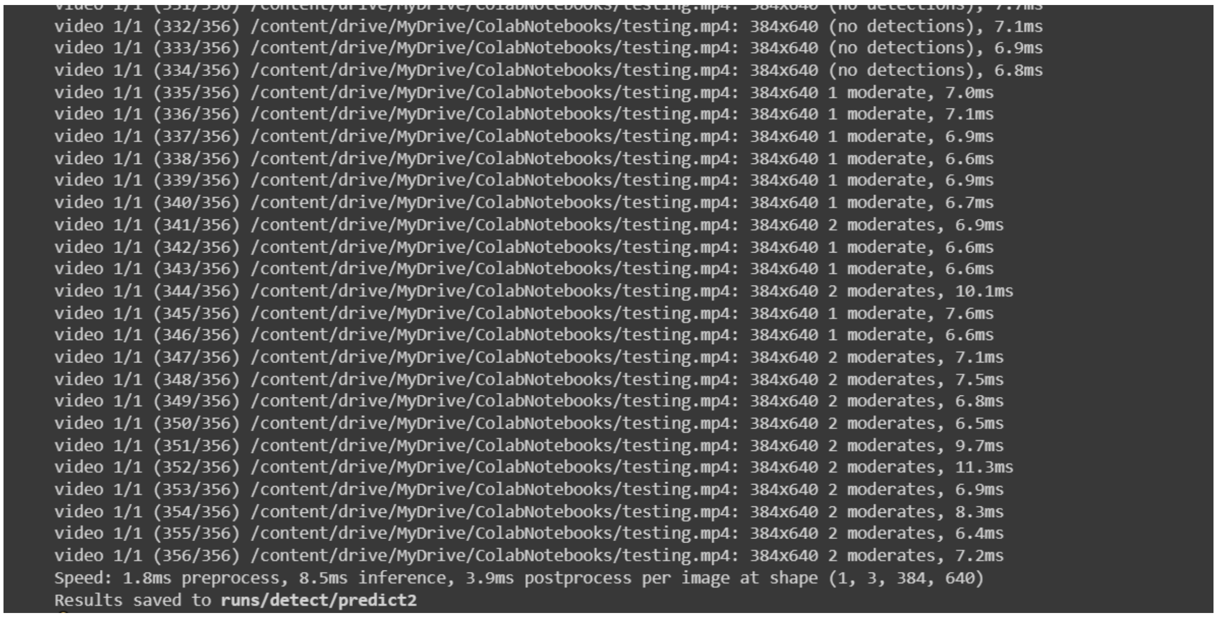


```
[ ]  !yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt data=/content/datasets/data.yaml save=true

     Ultralytics YOLOv8.0.227 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
     Model summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
     val: Scanning /content/datasets/valid/labels.cache... 1347 images, 1 backgrounds, 0 corrupt: 100% 1347/1347 [00:00<?, ?it/s]
                      Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 85/85 [00:18<00:00,  4.58it/s]
                        all       1347       1406      0.973      0.938      0.973      0.878
                   moderate       1347        329      0.958      0.909      0.952      0.858
                     severe       1347       1077      0.988      0.967      0.993      0.899
     Speed: 0.5ms preprocess, 4.4ms inference, 0.0ms loss, 2.1ms postprocess per image
     Results saved to runs/detect/val
```

Figure 10: Downloading the validation of the trained model

6.5.1 Unit Testing

We evaluated our program using a video clip saved locally to see if the detection in the model works and saved it on AWS S3 bucket under runs/detect/predicts.



```
video 1/1 (332/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 (no detections), 7.1ms
video 1/1 (333/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 (no detections), 6.9ms
video 1/1 (334/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 (no detections), 6.8ms
video 1/1 (335/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 7.0ms
video 1/1 (336/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 7.1ms
video 1/1 (337/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.9ms
video 1/1 (338/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.6ms
video 1/1 (339/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.9ms
video 1/1 (340/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.7ms
video 1/1 (341/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 6.9ms
video 1/1 (342/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.6ms
video 1/1 (343/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.6ms
video 1/1 (344/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 10.1ms
video 1/1 (345/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 7.6ms
video 1/1 (346/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 1 moderate, 6.6ms
video 1/1 (347/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 7.1ms
video 1/1 (348/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 7.5ms
video 1/1 (349/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 6.8ms
video 1/1 (350/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 6.5ms
video 1/1 (351/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 9.7ms
video 1/1 (352/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 11.3ms
video 1/1 (353/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 6.9ms
video 1/1 (354/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 8.3ms
video 1/1 (355/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 6.4ms
video 1/1 (356/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 2 moderates, 7.2ms
Speed: 1.8ms preprocess, 8.5ms inference, 3.9ms postprocess per image at shape (1, 3, 384, 640)
Results saved to runs/detect/predict2
```

Figure 11: Unit testing

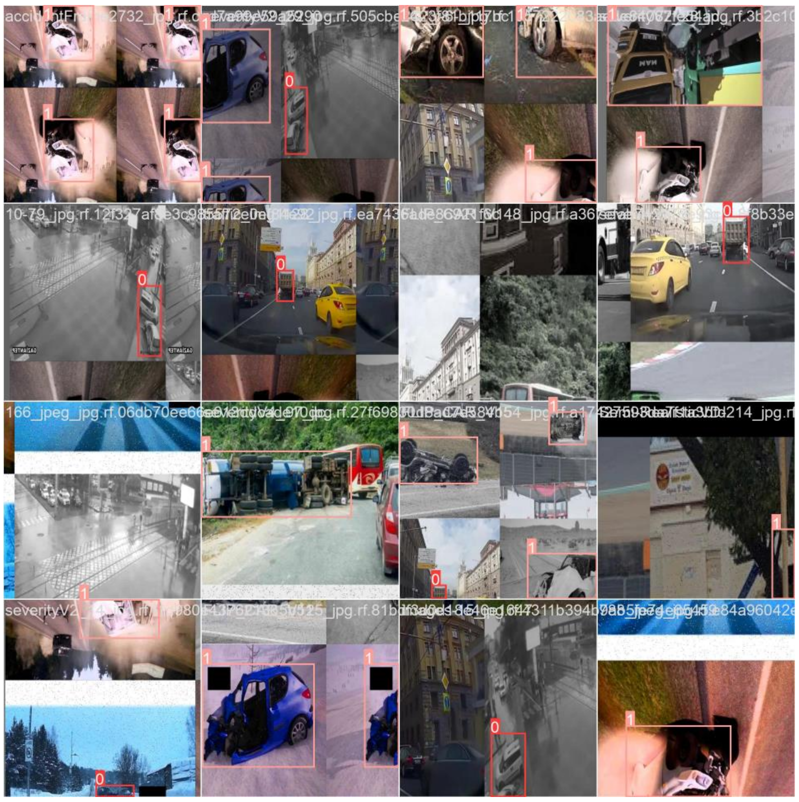


Figure 12

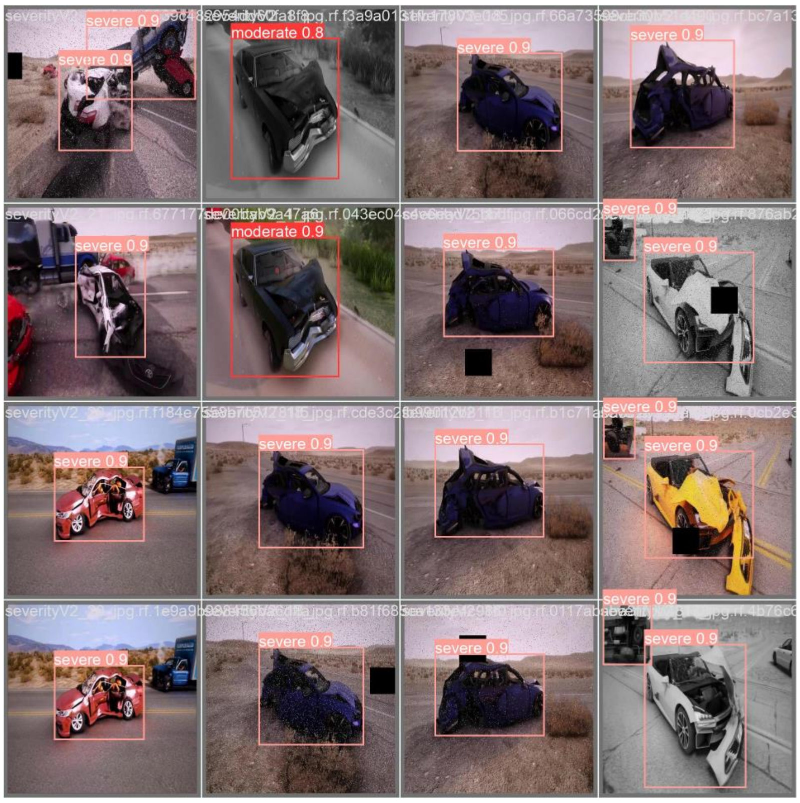


Figure 13

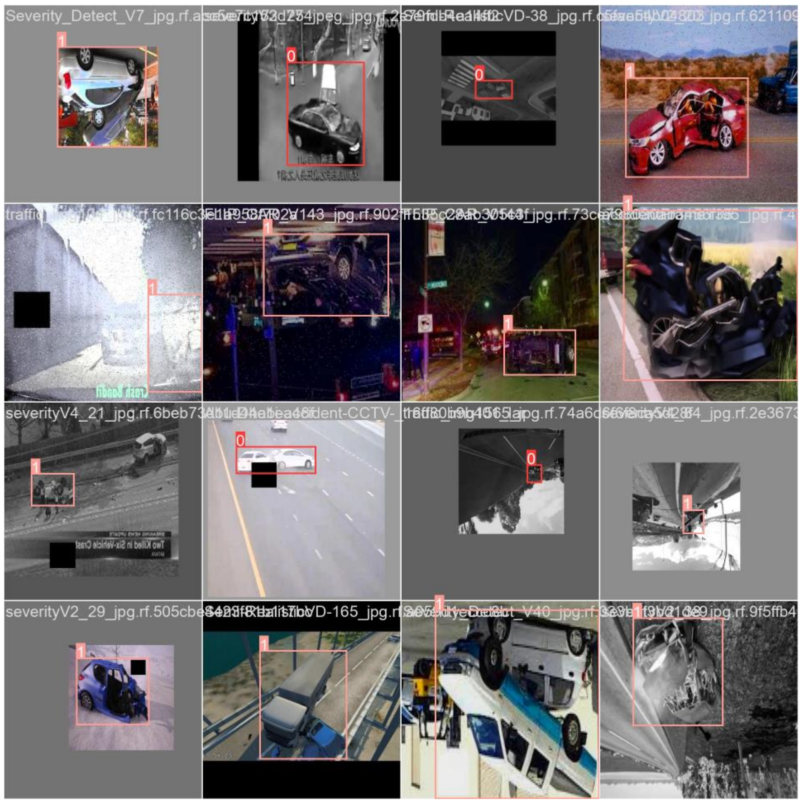


Figure 14

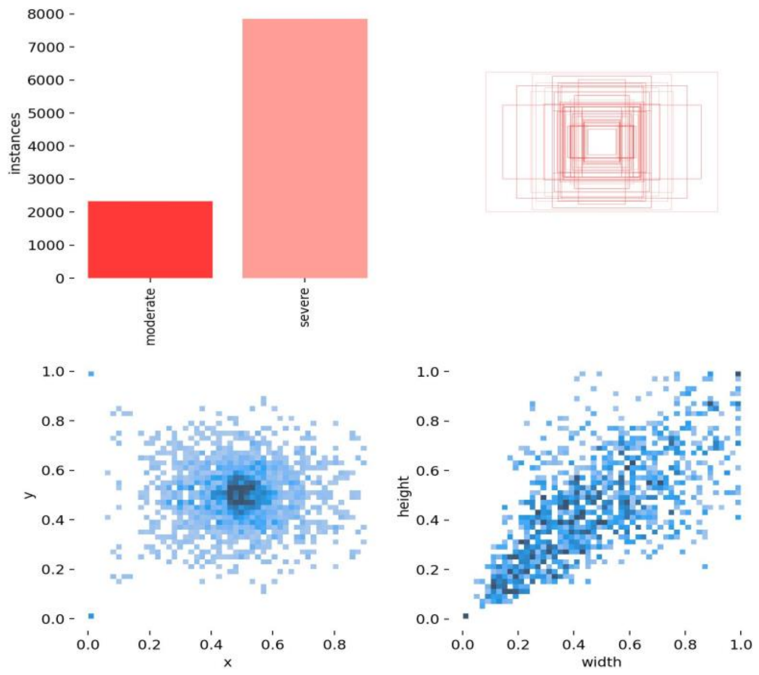
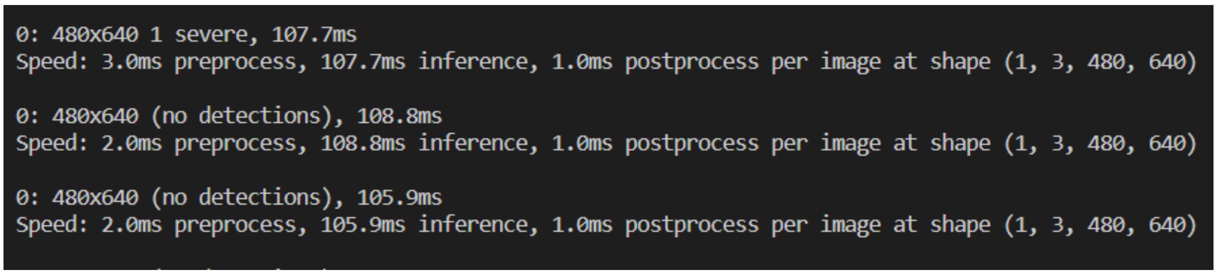


Figure 15

6.5.2 Integration Testing

We did Integration Testing using YOLOv8n detect which detected each frame using "moderate" and "severe" attributes of the accident.



```
0: 480x640 1 severe, 107.7ms
Speed: 3.0ms preprocess, 107.7ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 (no detections), 108.8ms
Speed: 2.0ms preprocess, 108.8ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 (no detections), 105.9ms
Speed: 2.0ms preprocess, 105.9ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)
```

Figure 16: Integration Testing

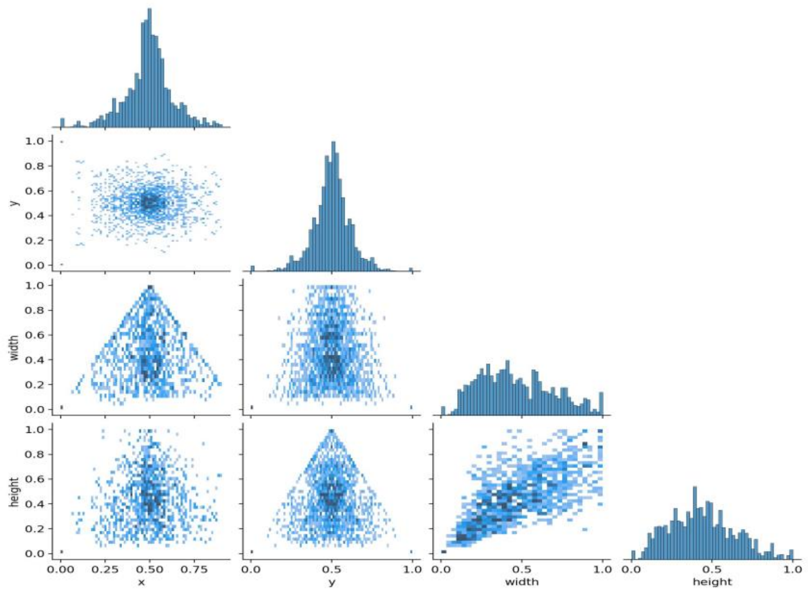


Figure 17

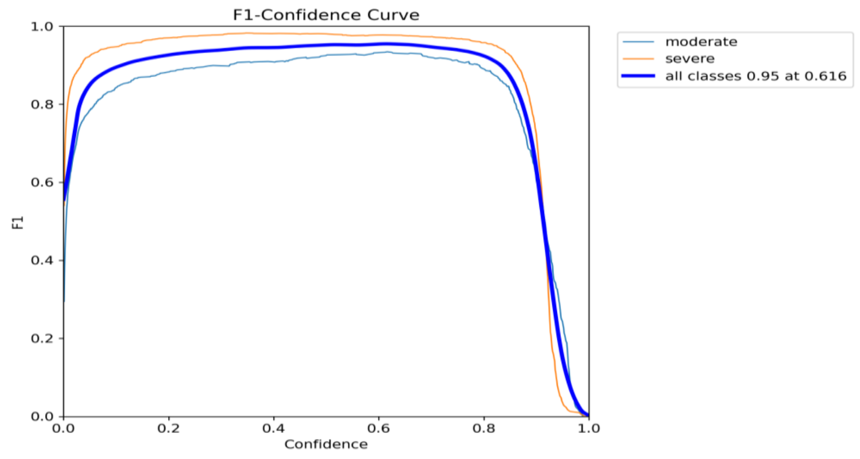


Figure 18

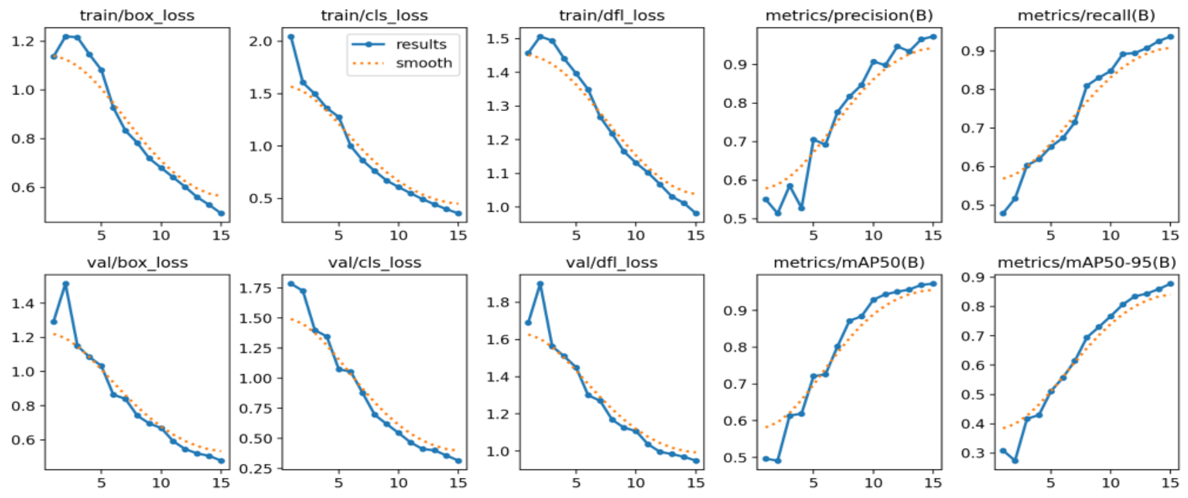


Figure 19

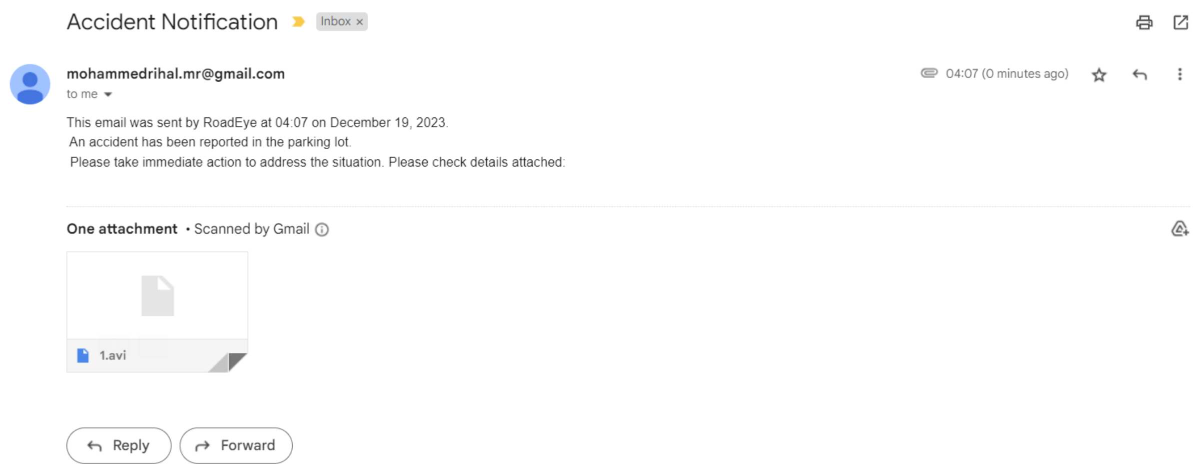
6.5.3 System Testing

After multiple testing on different platforms and different systems the code works perfectly fine without any defects or any bugs.

6.5.4 User-Experience Testing

Due to lack of time, we currently do not have a User Interface.

However, the higher authorities will receive an email which was sent using "SMTP" protocol using python. After several testing, we reached a final format of the email to be sent, with attached video clip of the accident in a .avi file with the time when it was sent.



Accident Notification Inbox ×

mohammedrihal.mr@gmail.com
to me

04:07 (0 minutes ago)

This email was sent by RoadEye at 04:07 on December 19, 2023.
An accident has been reported in the parking lot.
Please take immediate action to address the situation. Please check details attached:

One attachment • Scanned by Gmail

1.avi

Figure 20: Email received by the higher authorities.

6.5 Discussion of Results and Comparison

The results of the testing phases were analyzed and compared with similar systems to evaluate the performance and effectiveness of the implemented system. The system's accuracy in accident detection and classification, real-time monitoring capabilities, and user-friendliness were assessed. The strengths and weaknesses of the proposed system were identified and discussed based on the obtained results.

In conclusion, this chapter has discussed the implementation details of the system, including the selection and integration of different components. The chapter also reported the results of the testing phases, including usability and user-experience testing. The strengths and weaknesses of the proposed system were highlighted and compared with similar systems. Overall, the implemented system demonstrated accurate accident detection and classification, real-time monitoring capabilities, and a user-friendly interface, making it a valuable tool for monitoring accidents in the parking lot.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

In summary, the University of Bahrain's adoption of the Road Eye system is a big step in the right direction toward improving campus security and reducing parking lot collisions. A strong and effective system has been created because of the integration of YOLOv8 for real-time object detection, the construction of a Python-based backend, and the deployment on AWS infrastructure. We have promoted smooth communication and event reporting by utilizing communication platforms like SMTP and Google Forms, which has strengthened our dedication to the security and welfare of the university community.

7.1.1 Implementation

With a strategic design in place, the implementation phase translated concepts into tangible solutions. The YOLOv8 model was trained and integrated into the system, ensuring accurate and timely identification of vehicles. The Python backend facilitated seamless data management, and the deployment on AWS infrastructure provided the required scalability and reliability for real-world deployment.

7.1.2 Testing

Strict testing validated the efficacy and reliability of the Road Eye system. From the accuracy of the YOLOv8 model to the functionality of the backend and the overall system integration, comprehensive testing scenarios were employed. Simulated incident scenarios confirmed the system's responsiveness and its ability to contribute to campus safety.

7.1.3 Project Limitations

While celebrating the achievements, it is essential to acknowledge project limitations. Factors such as budget constraints, technical challenges, and potential privacy concerns have shaped the project's scope. These limitations highlight areas for improvement and refinement in future iterations.

In conclusion, the Road Eye project marks a significant contribution to campus safety. Through the design, implementation, and testing phases, the system has demonstrated its potential to make a positive impact on the safety and well-being of the University of Bahrain community. The journey undertaken in this project provides a foundation for ongoing advancements in the realm of campus security, with a commitment to continuous improvement and innovation.

7.1.4 Future Work

Significant enhancements are planned for the Road Eye system in the future to improve both its functionality and user experience. The main goal will be to provide a user-friendly web interface that makes it easier for users to engage with incident reports, live camera feeds, and associated data. Simultaneously, a strong database integration will be put into place to effectively manage and store data, facilitating smooth interactions between the front-end and back-end elements. As the system develops, it will enable various cameras on the campus of the university. To manage data from these sources. With this extension, a dynamic camera management system and an extensive surveillance network are intended to be established.

Furthermore, location mapping for incidents will be included, linking every occurrence to campus areas using camera IDs. The event visualization and response will be improved by this visual representation. The user interface will incorporate real-time alerts and notifications to notify authorized users or security personnel about new events in a timely manner. This will provide them with comprehensive information to facilitate a speedy response. A secure user authentication system with role-based access control will be implemented to guarantee data privacy and system security. The overall goal of these upcoming improvements is to make the Road Eye system more adaptable, safe, and user-friendly while also advancing campus safety technologies.

References

Huang, T., Wang, S. and Sharma, A. (2020). Highway crash detection and risk estimation using deep learning. Accident Analysis & Prevention, 135, p.105392. doi:https://doi.org/10.1016/j.aap.2019.105392.

Abdel-Aty, M. and Pande, A. (2005). Identifying crash propensity using specific traffic speed conditions. Journal of Safety Research, 36(1), pp.97–108. doi:https://doi.org/10.1016/j.jsr.2004.11.002.

Balfaqih, M., Alharbi, S.A., Alzain, M., Alqurashi, F. and Almilad, S. (2022). An Accident Detection and Classification System Using Internet of Things and Machine Learning towards Smart City. Sustainability, [online] 14(1), p.210. doi:https://doi.org/10.3390/su14010210.

Roboflow. (n.d.). Accident Detection Object Detection Dataset and Pre-Trained Model by accident detection. [online] Available at: https://universe.roboflow.com/accident-detection-ffdrf/accident-detection-8dvh5 [Accessed 19 Dec. 2023].

Amazon.com. (2023). Available at: https://us-east-1.console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin [Accessed 19 Dec. 2023].

Cellint. (n.d.). Transportation Solutions Based on Cellular Network Monitoring. [online] Available at: https://www.cellint.com/trafficsense/ [Accessed 19 Dec. 2023].

Sm@rt Technology Limited. (n.d.). SmartCross from Sm@rt Technology. [online] Available at: https://smart-technology.org.uk/our-technology-v1-srp-embedded-encryption-protocol/smartcross-from-smrt-technology/ [Accessed 19 Dec. 2023].

www.fleetguard.com. (n.d.). Fleetguard | Fleetguard. [online] Available at: https://www.fleetguard.com/s/?language=en_US.