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-Semester 1

Project Supervisor: Dr. TAHER SALEH KHAID HOMEED

Abstract

RoadEye is a revolutionary machine designed to reveal and analyze accidents happening within the parking zone by leveraging machine learning strategies and connecting to cameras. This undertaking makes use of the Python programming language to broaden a strong and efficient system capable of detecting and reporting injuries in actual time. By integrating with current camera infrastructure, RoadEye aims to enhance road protection by presenting real-time notifications through E-mail and valuable insights to relevant authorities and emergency services.

Using the latest machine studying algorithms, RoadEye processes the stay video feeds from cameras to become aware of potential accidents. The gadget employs pc imaginative and prescient strategies to detect and music motors in actual time. By studying the behavior and interactions of these objects, RoadEye can appropriately identify and classify injuries, such as collisions, near misses, and other risky incidents.

Once an accident is detected, RoadEye promptly generates unique incident reviews, which include the place, time, and type of accident. These reports are mechanically forwarded to the proper government, enabling them to quickly reply and dispatch emergency offerings to the scene. Additionally, RoadEye offers statistical evaluation and visualizations of coincidence statistics to aid in figuring out coincidence-susceptible areas, evaluating avenue safety measures, and making informed selections about infrastructure enhancements.

By leveraging the energy of device learning and actual-time video evaluation, RoadEye strives to enhance avenue safety and decrease coincidence reaction times. The system's capacity to appropriately come across injuries and provide well-timed notifications can assist save lives and mitigate the effect of injuries on site visitors congestion and public safety. RoadEye represents a good sized leap forward in leveraging technology for twist of fate prevention and management, aiming to create safer and greater efficient street systems for communities internationally.

Acknowledgments

We would really like to express our honest gratitude to our esteemed advisor, Dr. Taher, for his invaluable steering and aid in the course of the whole project system. His expertise, patience, and encouragement were instrumental in shaping our work and permitting us to reap our desires.

We also amplify our heartfelt thanks to our buddies who supplied us with their help, motivation, and valuable remarks all through the system. Their collaboration becomes essential in overcoming challenges and fueling our development.

We are immensely grateful to our families for their constant love, motivation, and faith in us. Their support has fueled our growth and provided the strength to overcome challenges. Without their backing, RoadEye could have faltered. We acknowledge and are indebted for their benevolence and support. Thank You all.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Through the lens of innovation, RoadEye targets to pave the way for safer journeys, where synthetic intelligence meets the open street.

Every time you park your car, you believe there in advance holds no unseen risks. But the fact is, every parking lot teems with capacity hits: drowsy drivers, reckless speeders, and unforeseen boundaries lurking around every bend. Until now, these types may want to spell disaster. But what if you had a guardian angel on the street, looking over you with eyes sharper than hawks?

Enter RoadEye, an innovative AI-powered system that transforms accident detection from reactive to initiative-taking. Imagine a vigilant sentinel, scanning the street beforehand with evolutionary algorithms, instantly recognizing threats you might leave out, and issuing timely warnings that might save your fortune and those around you.

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

1.1 Problem Statement

Imagine being a student at the University of Bahrain, excited for class. You park your car and head off, best to go back later and discover it mysteriously banged up! This occurs too often, and absolutely everyone is harassed and pissed off.

The University of Bahrain, like many other college establishments, grapples with a pressing issue that drastically influences the well-being and comfort of its pupil frame – automobile parking space incidents. Students attending lessons often discover themselves faced with the unexpected fact that their automobiles had been damaged during their time on campus.

The frequency of those incidents raises issues about the lack of powerful tracking and reporting mechanisms in the vicinity to address parking zone injuries right away.

In reaction to this problem, the task was initiated with the target of enforcing a comprehensive solution to the usage of "RoadEye" , a device designed to monitor and record accidents in real-time. By leveraging superior surveillance era, the assignment objectives to offer an efficient and reliable means for college kids to gain from mentioned incidents promptly. The Road Eye gadget is anticipated to feature as a guard for students' rights, allowing them to seek appropriate movement and reimbursement for damage incurred even as parked on campus.

1.2 Project Objectives

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

1.To develop an actual-time accident detection module the usage of machine getting to know and computer vision strategies to investigate stay video feeds from cameras.
2. To train the machine mastering models on a diverse dataset of categorized coincidence eventualities to make certain accurate detection and category.
3. To put in force an incident reporting module that routinely generates reviews ,inclusive of the accident live feed.
4. To establish a reliable communique mechanism to right away transmit incident reviews to the proper emergency services.
5. To evaluate the overall performance and effectiveness of the RoadEye system through appropriate testing and verification.
6. To make sure the rights of whomever vehicle were given hit and make sure the ability to get compensated.
7. To boom avenue safety and lower the probability of injuries occurring everywhere.

It is essential to notice that the scope of this research venture is limited to the improvement and assessment of the RoadEye system. While the gadget aims to decorate road protection and coincidence control, it does not address other factors which include traffic congestion or driver conduct.

1.3 Relevance/Significance of the project

The RoadEye's relevance and importance is in the area of safety. By using machine learning we are making it easier, the system enhances detection and reports accidents, leading to faster reaction times and greater effective allocation of resources. The real-time of the gadget lets in for fast notification with the help of email of the authorities and emergency services, letting them know and saving lives.

1.3.1 Project scope

The purpose of this task is to cope with the continued issue of injuries at the University of Bahrain with the aid of implementing the Road Eye device. The Road Eye device pursuits to address this problem by way of strategically putting cameras in parking areas, continuously monitoring for incidents and imparting real-time detection of injuries. The goals of this design pass past imparting a reliable reporting mechanism; it also ambitions to set up a scientific method to incident management, making sure speedy responses and safeguarding the rights of all people concerned. By collecting and studying the live feed information, styles may be recognized, and accidents may be determined, leading to lengthy-time period improvements in safety. The scope of the design encompasses numerous levels, together with layout and making plans, installation of the Road Eye device, implementation of the incident management machine, and a complete testing, schooling, and release segment. The key indicators of achievement for this venture encompass a discount in mentioned automobile parking space accidents, elevated pleasure among university attendees concerning parking zone protection, green resolution of incidents, and high-quality comments and engagement from the university community. While the design recognizes sure inherent constraints inclusive of price range obstacles, integration demanding situations, and privateness worries, it goals to address those problems successfully that allows you to achieve its overarching intention of enhancing automobile parking space safety at the University of Bahrain.

1.4 Report Outline

Chapter 1; Introduction: In this chapter, we state our problem statement, talking about the challenges in the project. We also mention the project agenda of the report along with the relevance of this project "RoadEye".

Chapter 2; Literature Review: In this chapter, we talk about and explain what we wrote in our literature review which is related to the problem statement. It explains and compares existing research and methodologies related to live accident detection.

Chapter 3; Managing the Project; In this chapter we delve into the project management aspect of our RoadEye project. We discuss the model we selected for the project, outline our risk management strategies and provide a view of the project.

Chapter 4; Gathering and Analyzing Requirements: This chapter focuses on the process of collecting and analyzing all the elements for the RoadEye system. We cover the requirements, approaches to implementing them and how we developed them.

Chapter 5; Designing the System; Here we explore the system design phase of the RoadEye project. We showcase our design using a use case scenario, a flow chart and highlight components of the system.

Chapter 6; Implementing and Testing the System: In this chapter we discuss how we implemented the RoadEye system. We touch upon software aspects such as code, S3 Bucket integration, Python usage as hardware components like webcams. Additionally we present our testing procedures along with tested values and recommendations for selecting an optimal detection model for accidents.

Chapter 7; Conclusion and Future Work: In this chapter, we give a conclusion of our report of "RoadEye", showcasing how good of a project RoadEye is. We also discuss the future work to be implemented for the betterment of RoadEye.

References: In this session, we have listed down all the references we used for this report and it is cited using IEEE Style.

Chapter 2
Literature Review

Digging deep into the historical fabric of studies and practices is crucial when examining the overall capabilities of road security precautions. A significant archive of studies and initiatives has been established over time, all aimed at improving traffic safety and lowering injury rates. “The landscape of fate discovery has undergone a transformational shift, evolving from a risk-based reactive approach to an initiative-taking, information-driven machine empowered with modern technology.”(National Highway Traffic Safety Administration, 2023), below we will talk about the major three systems we took our insights from.

2.1 TrafficSense:

"TrafficSense", a system that identifies motion in real-time traffic using mobile phone logs. It offers important insights that assist transportation authorities.

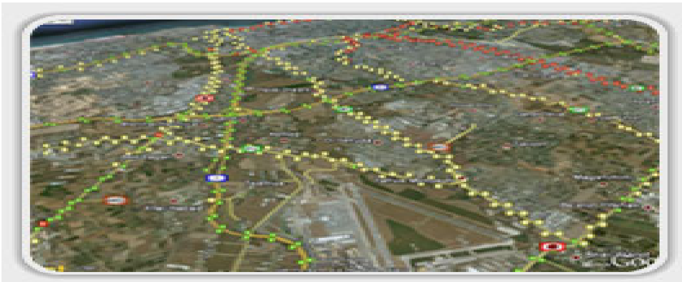
2.1.1 System Features:

Benefits:

1. It helps reduce passenger journey time, for a smoother and greater green transportation.
2. The authorities can take an initiative-taking method to create a more secure surroundings for each person within the car parking zone.
3. It shows live traffic and lets us access them.

Drawbacks:

1. TrafficSense relies heavily on mobile smartphone realities, so there may be limited effectiveness in locations with low mobile usage or inconsistent network security. Deploying and protecting a TrafficSense monitor requires significant cash funding. Technical system defects or errors should affect the system's tracking and selection talents in real time.
2. Deploying and protecting a TrafficSense monitor requires significant cash funding.
3. Technical system defects or errors must affect the system's ability to track in real time and for any selection.



2.2 SmartCross:

"SmartCross" is a technology-based system that has safety and efficiency benefits for pedestrian tasks. Anticipating devices, cameras and intelligent algorithms to target pedestrians and adjust visitor sensor alerts in real time, which will count and improve site visitor expectations. In addition, often including functions to assist individuals with special needs, allowing them to develop more comprehensively and accessible.

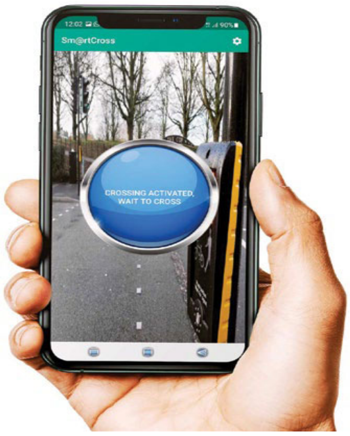
2.2.1 System Features:

Benefits:

1. SmartCross systems leverage highly advanced technologies to detect people and facilitate traffic alerts to enhance their protection and visibility.
2. SmartCross systems dynamically adjust signal timing to reduce congestion and improve site visitor efficiency.
3. SmartCross structures often include abilities such as audio alerts and cues to assist people having disabilities.

Drawbacks:

1. Substantial upfront costs for enhancements to the infrastructure and implementation.
2. Technically difficult conditions, such as software bugs or malfunctioning sensors, can affect the reliability of the device.



2.3 FleetGuard:

"FleetGuard" is a complete tool that improves fleet performance, enhances safety and security, and simplifies maintenance. It analyzes statistics to optimize routes, reduce downtime, and identify areas that need improvement.

System Features:

Benefits:

1. Optimizes routes, reduces idle time, and improves efficiency
2. Increases security and safety (with geofencing, GPS tracking, and alarms for unauthorized use).
3. simplifies remodeling and lowers the amount of unavailability (with adaptive planning and signs).

Drawbacks:

1. Demands cautious installation along with integration (difficult system, power issues).
2. Highlights concerns about analytics privacy (access to sensitive data, security measures required).
3. heavily depends on network and energy (backup systems advised).

2.4 Outcome:

The outcome of researching these systems was a great help to our RoadEye, as it benefited a lot. RoadEye has the identical idea and will also be using comparable strategies to achieve such goals.

Chapter 3
Project Management

Project management performs a critical role within the a hit execution of any assignment, including the development of the RoadEye device. It entails the software of know-how, skills, equipment, and strategies to efficiently plan, execute, display, and control venture activities. This chapter makes a speciality of venture management aspects, which include the process model, danger management, and the challenge activities plan.

The undertaking control system guarantees that the assignment is carried out in an organized and systematic manner, adhering to predefined dreams, timelines, and best standards.

3.1 Process Model: Software Development Life Cycle (SDLC)

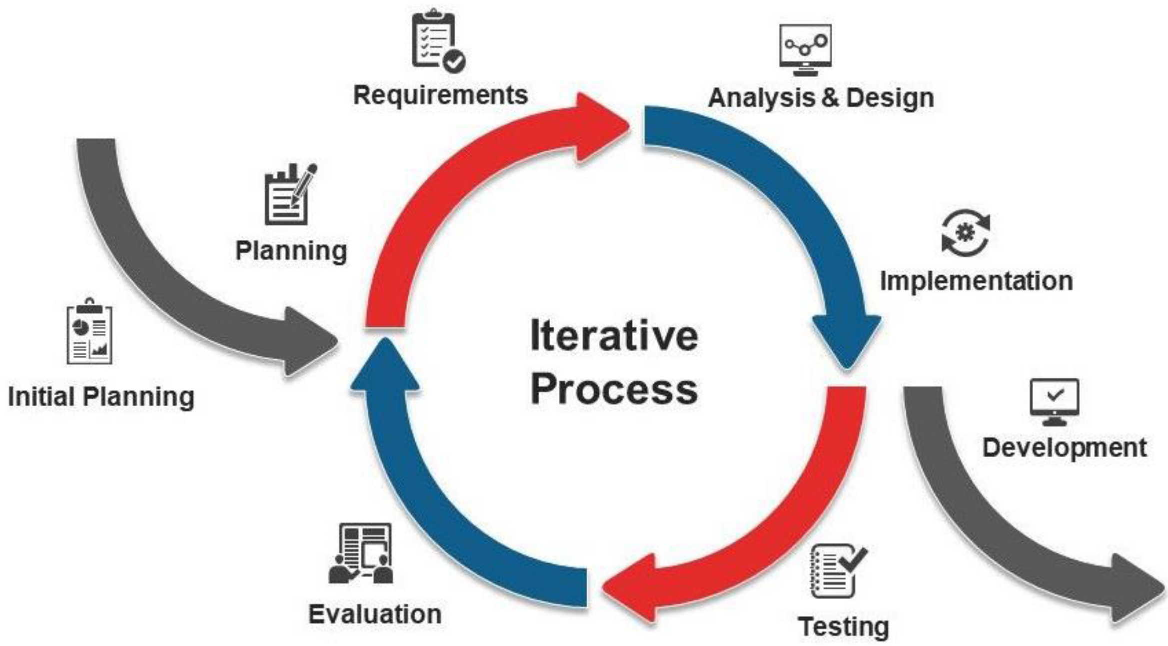


Figure 4: Software Development Life Cycle (SDLC)

The software Process model selected for the RoadEye task is the Software Development Life Cycle (SDLC) model. This version affords a established and systematic approach to software program improvement, ensuring that the task progresses thru wonderful stages, every with its precise targets and deliverables.

The justification for deciding on the SDLC version stems from its nicely-mounted and widely used nature inside the software improvement enterprise. The SDLC model gives a clean and nicely-defined framework that aligns with the goals and necessities of the RoadEye mission. By following this model, we will make certain a scientific and organized method to software program improvement, leading to a robust and excellent system.

The SDLC version includes numerous levels, every serving a selected purpose in the development manner. These stages commonly include requirements accumulating, machine layout, implementation, testing, deployment, and maintenance. Each phase has described inputs, outputs, and sports that manual the development technique and enable effective venture management.

During the phase of gathering necessities the team responsible for the task interacts with stakeholders to gather and document the requirements for the gadget. This involves conducting interviews, workshops and surveys to collect information and gain an understanding of what consumers want and expect. The outcomes of this phase include a document outlining all the requirements, which serves as a basis for stages.

The focus of the system design phase is to transform those requirements into a defined design for the gadget. This encompasses specifying interfaces, technical specifications and device architecture. The Planning Department ensures that the system is robust, adaptable and in line with industry standards.

Once the design is finalized our project team shifts their attention to developing and programming the RoadEye system during the implementation phase. This entails writing code integrating plugins and constructing the device according to the approved layout.

The inspection stage plays a role in guaranteeing that the device performs well. Maintains reliability after its implementation.

3.1.1 Phases of Iterative Process Model

For the RoadEye project, an iterative process model with multiple stages that allow for gradual iterative improvement was selected. These tiers facilitate adaptability, ongoing enhancement, and the incorporation of input at every phase of the model.

3.1.1.1 Initial Planning:

Initial Planning, first thought of an idea. It's the backbone of the project. Helps up to think forward and know what the next stage is.

3.1.1.2 Iteration Phase:

1. Planning:

The second and most challenging phase of our endeavor is planning. It is regarded as one of the project's most crucial phases. While the technical portion of building the project is not included in this phase, it does offer a path of activities and direction, as well as the channels that must be followed in order to advance to the next and ultimately the crucial phase.

2. Requirement:

The third stage of our project is requirements identification and collection. It is regarded as one of the project's key phases. We will identify the most crucial needs that the system must meet in collaboration with the appropriate authorities.

3. Analysis and Design:

During the analysis and design stage, we look at the requirements, design models, and user interfaces. We have a thorough understanding of plugins, modules, and dependencies and produce original concepts that direct implementation.

4. Implementation:

Building the system from scratch is the foundation of the implementation phase. During this stage, we write the essential code and combine the requirements to turn the designs into an operational solution.

5. Testing:

To guarantee that the code produced is reliable and acceptable validation is an essential step. We participate in a variety of testing activities, including machine, integration, and unit testing. Additionally, we are able to identify and detect any flaws or problems, guaranteeing that the system satisfies the necessary specifications.

6. Evaluation:

The assessment segment involves assessing the developed functionality against the stakeholders' expectations and necessities. We collect remarks from stakeholders, behavior user recognition trying out, and compare the machine's overall performance. The feedback and evaluation consequences offer insights for in addition enhancements and refinements.

7. Deployment Phase:

Deployment Phase, We put together the device for production deployment, including set up, configuration, and facts migration if necessary. User documentation and training materials are provided to assist the stop-customers in efficiently using the machine. The deployment section ensures a smooth transition from development to the operational phase.

The iterative system model lets in for continuous remarks, learning, and refinement all through the development lifecycle. It promotes collaboration, adaptability, and the timely transport of valuable software to fulfill the stakeholders' desires. By following this version, our RoadEye venture targets to gain successful effects and supply a exquisite device.

3.2 Risk Management

Risk management is a vital element of challenge control that includes identifying, assessing, and mitigating capacity risks that may impact the a success execution of a mission. RoadEye recognizes the significance of proactively addressing risks to ensure assignment objectives are met, timelines are adhered to, and the pleasant of deliverables is maintained. This section specializes in the danger management technique for RoadEye and affords a desk outlining ability dangers, their types, consequences, portability, and mitigation techniques.

Table 1: Potential Risks and Mitigation Methods

Risk	Risk Type	Risk Effect	Risk Portability	Risk Mitigation Method
Technical Challenges	Project delays and system downtime	Medium	High	Conduct thorough trying out, allocate extra sources for

				troubleshooting, and feature contingency plans for capacity technical issues. Reviews and seek remarks.
Privacy Concerns	Legal issues	Medium	High	Establish a strong change management process. Clearly define and file mission scope. Evaluate and prioritize change requests. Obtain proper approval earlier than incorporating changes.
Unexpected external events	Project disruption	Medium	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	High	Medium	Plan and take a look at integration sports early inside the improvement cycle. Define integration interfaces and protocols. Conduct thorough integration trying out. Monitor and cope with any compatibility problems.

Note: The table above affords representations of capability dangers for the RoadEye venture. It is vital to tailor the dangers and mitigation techniques to the specific context and necessities of the task.

3.3 Project activities Plan

The challenge activities plan outlines the breakdown of RoadEye into extraordinary activity, supplying a clean timeline and sequence of responsibilities wanted for the successful transport of the machine. This plan ensures efficient project management, task mission, and monitoring of development. The following is an instance of the way the undertaking activities plan for the RoadEye venture may be established:

1. Requirements Gathering Phase (Duration: 2 weeks)
   - Conduct stakeholder interviews and workshops to collect machine necessities.
   - Analyze and report the amassed necessities.
   - Review and finalize the requirements report.

2. System Design Phase (Duration: 3 weeks)
   - Define the gadget architecture and element specifications.
   - Develop gadget design documents, including device flowcharts and facts models.
   - Conduct design opinions and comprise remarks.

3. Implementation Phase (Duration: 6 weeks)
   - Set up the development environment and infrastructure.
   - Develop the core functionalities of the RoadEye device primarily based on the layout specifications.
   - Conduct normal code opinions and unit testing to make sure of quality.

4. Testing Phase (Duration: 4 weeks)
   - Perform unit checking to confirm the capability of individual components.
   - Conduct integration checking to make sure the right interaction between unique machine modules.
   - Perform machine checking to validate the general gadget performance and functionality.

5. Deployment Phase (Duration: 2 weeks)
   - Prepare the RoadEye gadget for deployment, including machine set up and configuration.
   - Conduct user recognition by trying out certain machine usability and pleasure.
   - Develop consumer documentation and provide education to stop-customers.

6. Maintenance and Support Phase (Ongoing)
   - Address and resolve any said insects or troubles.
   - Provide ongoing aid and maintenance to ensure system balance and overall performance.
   - Incorporate user remarks to make important upgrades and upgrades.

Chapter 4
Requirement Collection and Analysis

The requirement collection and analysis phase is a vital element of challenge management as it involves growing a clean and agreed set of patron necessities. These necessities serve as the inspiration for delivering a gadget that aligns with the patron's needs and expectancies. This chapter discusses the practical and non-functional requirements of the system, together with the facts, waft diagram and use case diagram, which resource in knowledge the device's behavior and functionalities.

4.1 Requirement Elicitation

4.1.1 Introduction

Requirement elicitation is a critical section in undertaking management, because it includes collecting the device necessities so that it will manual the development method. This record gives a top level view of the requirement elicitation techniques employed to accumulate the vital statistics. The techniques mentioned include interviews, questionnaires, observations, and quantitative and qualitative records evaluation.

4.1.2 Interviews

Here are a number of the questions we requested some stakeholders and they answered:

1. Can you please describe your function and duties within the company or area related to avenue accident tracking?

- 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 primary demanding situations or most important factors you currently face in monitoring and responding to road injuries?

- 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 presently acquire and examine information associated with avenue accidents? What equipment 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 functions or functionalities do you suspect are vital for a road accident monitoring gadget 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 injuries occur to your location, and what effect do they have on traffic float 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 expectancies concerning the accuracy and reliability of a road coincidence tracking device 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 critical is actual-time notification and alerting for road injuries, and the way might you choose to acquire those notifications (e.G., cell app, e-mail, 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 precise necessities or issues related to the integration of the RoadEye system with present infrastructure or systems on your employer?

- 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 type of statistics visualization or reporting abilities might you discover maximum treasured within the RoadEye machine? Are there any particular metrics or insights you would like to peer?

- 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 ability privacy or ethical issues related to enforcing a street coincidence monitoring system like RoadEye, and the way have to 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 might you degree the achievement or effectiveness of the RoadEye device? What key overall performance signs (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 stage of training or assist would be vital for customers to efficiently utilize the RoadEye gadget?

- 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 particular guidelines or compliance requirements that want to be considered when imposing the RoadEye device?

- 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 offer any examples or situations wherein the RoadEye machine would have a sizeable effect on enhancing street coincidence tracking 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 something else you would like to proportion or any extra requirements or issues that must be considered within the improvement 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.

4.2 System Requirements

4.2.1 Introduction

System requirements are critical for outlining the behavior and functions of software program applications. This record presents the functional and non-practical necessities accrued in the course of the requirement analysis section. Functional necessities describe unique functionalities and functions that the gadget ought to provide, whilst non-practical necessities specify the qualities or characteristics of the system.

4.2.2 Functional Requirements

Functional requirements outline the actions and behaviors that the device needs to carry out to satisfy user desires. Based on the requirement evaluation, we have recognized the subsequent purposeful necessities:

1. The gadget ought to be able to stumble on injuries in real-time and notify the appropriate authorities.
2. Users should be capable of document incidents and provide relevant information, inclusive of recorded stay feed and description.
3. The gadget has to generate incident reviews with the actual-time incident feed.
4. Users have to be capable of tuning the reputation and development of reported incidents through email.
5. Users ought to be able to get entry to historical incident facts for evaluation and reporting purposes.
6. The device should save the incident live feeds on a cloud.

7. Users ought to be capable of obtaining actual-time notifications and updates on incidents.
8. The device needs to allow for collaboration and communication amongst different stakeholders worried in incident management.
9. Users need to be capable of seeking and clearing out incident information based on various criteria, consisting of date, area, or severity.

4.2.3 Non-functional Requirements

Non-functional requirements deal with the features and traits of the machine that are not related to precise functionalities.
The following non-functional requirements were identified:
1. Performance: The machine needs to have rapid reaction times to make certain prompt incident control.
2. Security: The gadget has to put into effect strong security features to guard touchy user facts.
3. Usability: The gadget must be intuitive to facilitate ease of use.
4. Reliability: The machine has to be notably dependable and to be had, minimizing downtime.
5. Scalability: The gadget has to be able to control a growing wide variety of users and incidents without compromising performance.
6. Compatibility: The device must be compatible with a wide range of cameras, working structures , and so on...
7. Data Privacy: The machine should make certain the privacy and confidentiality of personal data, adhering to applicable facts protection regulations.
8. System Integration: The device needs to seamlessly combine with present systems and clouds for statistics trade and interoperability.
9. Documentation: The system should have complete and up to date documentation to assist users and directors.

4.3 Personas

Personas play a vital function in data the traits and behaviors of the tool/utility clients. This document offers a top level view of the personas advanced based totally on consumer research and their relevance to the gadget/software program being evolved.

4.3.1 Persona Development Process

The development of personas is carried out in various methods, interviews, surveys, and observations. The information will become analyzed in styles, wishes, and dreams of many of the goal person groups. The personas were then created to symbolize widespread customers and offer insight into their expectancies.

4.3.2 Personas

Based on the user research, these are the personas:

Ali

COMMUTER

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.

Abdulla

EMERGENCY SERVICES OFFICER

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.

Khaled

FLEET MANAGER

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.

Yousef

ADMINISTRATOR

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.

4.4 System Models

4.4.1 UML diagrams

4.4.1.1 Use case diagram

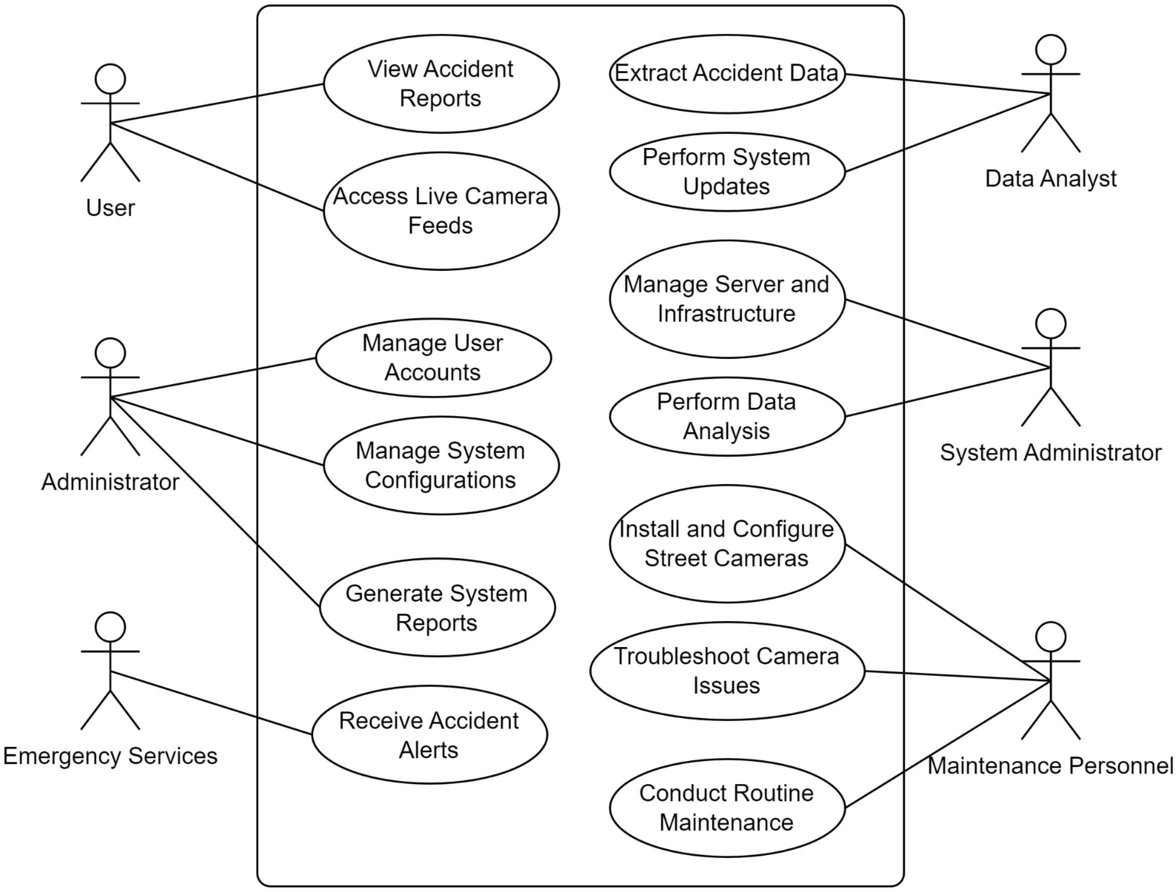


Figure 5: Use case diagram

4.4.1.2 UML Class Diagram

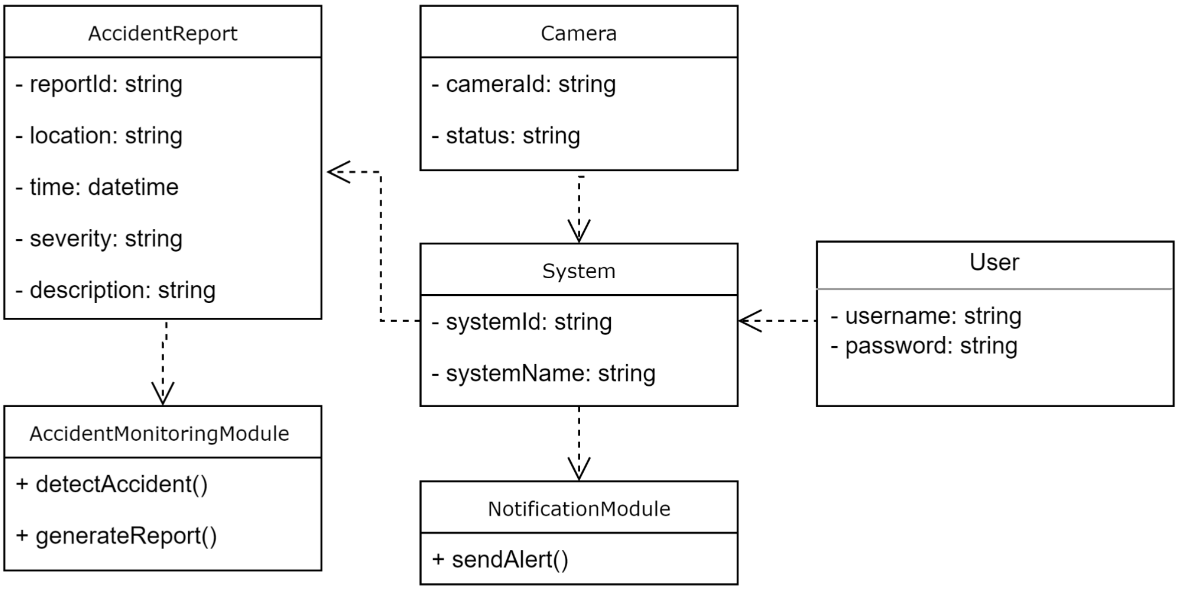


Figure 6: UML Class Diagram

4.4.2 AWS Model

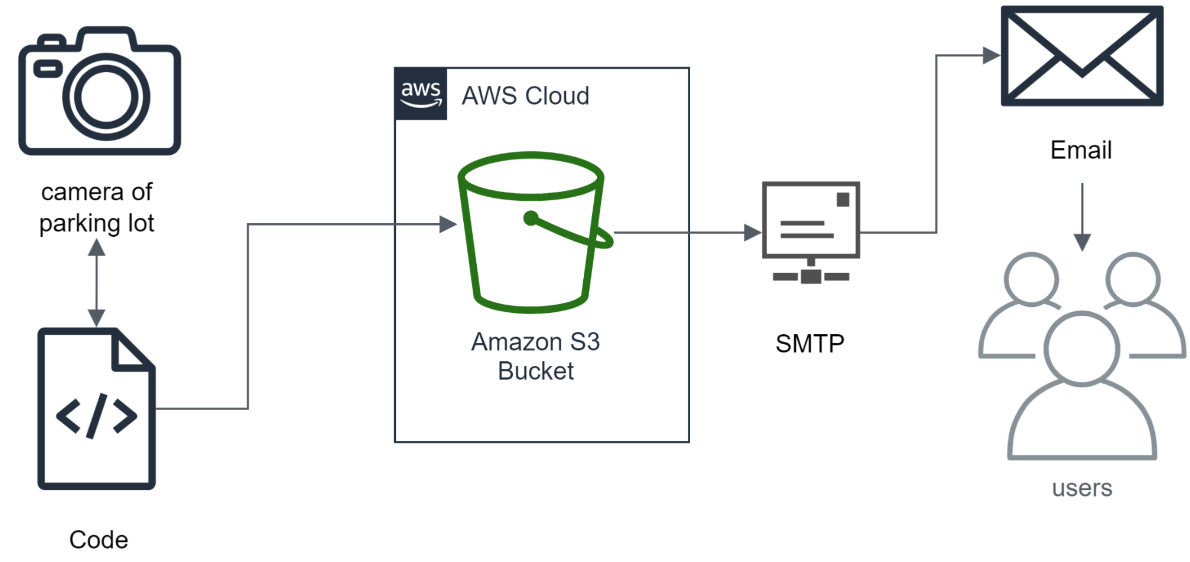


Figure 7: AWS Model

Chapter 5
System Design

5.1 Software Architecture

The system design of our assignment, which includes monitoring accidents within the parking zone, the usage of machine getting to know and cameras, can be based totally on a layered structure. This architecture affords a clean separation of worries and permits for modular and scalable development. The layers in our gadget layout consist of the presentation layer, utility layer, and facts layer.

5.1.1 The Reference Architecture

The chosen reference architecture for our machine is the Client-Server architecture. This structure style allows the separation of the client-aspect utility, liable for consumer interaction and presentation, from the server-aspect application, answerable for information processing and garage. The purchaser-side application will manage the user interface and verbal exchange with the cameras, even as the server-facet utility will perform the device getting to know algorithms and store the information.

5.1.2 Architecture Drivers

The device layout of our task is driven with the aid of several architecture drivers that make certain the effectiveness and fulfillment of the machine.

5.1.2.1 Design Purpose

The layout motive of our gadget is to provide actual-time monitoring and analysis of accidents taking place within the car parking zone. The gadget aims to detect accidents thru device mastering algorithms and provide timely signals and information to relevant parties, together with emergency services and visitors authorities.

5.1.2.2 Quality Attributes

The machine design is likewise pushed via various fine attributes, such as:

1. Accuracy: The system studying algorithms should as it should discover and classify accidents primarily based on the camera pictures.

2. Real-time: The system should offer real-time monitoring and signals to make sure immediate reaction to accidents.

3. Scalability: The architecture needs to be scalable to manage many cameras and manner facts efficiently.

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

5.1.2.3 Constraints

Certain constraints want to be taken into consideration within the system layout, which includes:

1. Language Constraint: The gadget might be developed using the Python programming language to leverage the gadget mastering libraries and frameworks available in Python.

2. Database Constraint: The gadget does not make use of a database, however it uses AWS S3 Bucket for storing and handling the digicam footage information, incident facts, and applicable information.

3. Hardware Constraint: The gadget design must keep in mind the hardware requirements for deploying the gadget, getting to know algorithms and storing the digicam footage statistics.

4. Budget Constraint: The design needs to be price-powerful, thinking about the available price range for hardware, software program, and maintenance.

5.2 AWS

AWS (Amazon Web Services) as a subsidiary of its determined enterprise Amazon, affords on-call for cloud computing platforms and APIs to diverse customers which unfold throughout people, agencies, and governments. Along with a large number of services inclusive of laptop, garage, databases, analytics, networking, AWS additionally offers cloud services to deal with excessive web visitors and excessive amounts of facts that are being generated daily on programs and/or websites. The AWS cloud architecture ensures that its clients follow the best available practices, guidance, and recommendations to not only develop but also maintain cloud solutions. Some of the basic AWS services are Route 53, Elastic Load Balancer, EC2, security groups, CloudFront, and Amazon S3 bucket, which we will be looking more in depth at. The Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, high availability of data, security, and quality performance. Clients spanning all sizes and industries using S3 Bucket can store and protect any amount of data for any use case, which include data lakes, cloud-native applications, and mobile applications. By using S3 (Simple Storage Service), companies can very easily store and retrieve several types of data using Application Programming Interface calls. Using Amazon S3 Bucket has the feature of memory management: this makes it amazingly fast and resilient for clients to host websites and get affirmative results on time. One can keep statistics throughout Amazon S3 storage lessons to reduce fees without in advance funding or hardware refresh cycles.

5.3 Machine Learning Algorithms

The machine learning algorithm we used was the YOLOv8 model, where we trained our model with pictures of the accident and using the detection, we could detect accidents. We used multiple YOLOv8 models and found out the nano model (YOLOv8n) was the best fit.

5.4 Deployment Diagram

Deployment Diagram for RoadEye, after processing and saving files inside the S3 bucket, a dedicated node is brought to represent the email notification machine. This node allows communication with an external electronic mail service, symbolizing the functionality of RoadEye to inform better authorities about vast activities or updates. A direct association line connects the RoadEye application to the e-mail notification node, indicating the seamless integration of the e-mail functionality into the software's workflow. This guarantees that, upon finishing document processing and storage, the RoadEye app triggers an occasion to send relevant notifications to better authorities through email. The inclusion of this email notification device complements the general usability of RoadEye, presenting a streamlined conversation channel for key stakeholders within the choice-making method.

Chapter 6
System Implementation and Testing

6.1 Introduction

This chapter talks about the device and the testing levels that we used. It also talks about the choice of different components, such as hardware, software, programming languages, and the cloud. The choices are based totally on their suitability for the task requirements. The chapter also showcases the results of our phases.

6.2 Tools and Technologies

A type of packages and languages had been used to create the accident machine.
In-depth information about it is supplied within the subsections underneath.

6.2.1 Software tools used:

For the improvement of this project a huge type of gear had been used to finish it:

6.2.2 Programming Language

Python, a programming language for imposing diverse components of RoadEye. From growing the backend infrastructure to developing the alert machine, Python facilitated seamless integration and ensured the efficient functioning of the entire machine.

6.2.3 Google Collab

Google Collab, a collaborative and cloud-based totally surroundings for our improvement crew. Leveraging its resources, we performed model schooling sessions, specially for YOLOv8, taking advantage of its GPU skills to boost up the schooling procedure and improve the accuracy of our object detection model.

6.2.4 Communication

1. **GitHub:**

   GitHub served as our model manipulation repository, facilitating collaborative improvement among us as individuals. It ensured that the codebase remained organized, versioned, and handy, fostering efficient collaboration, and enabling the monitoring of modifications over the years.

2. **Google Docs:**

   Google Docs offers a collaborative environment for creating, modifying, and sharing files. In our assignment, Google Docs changed into employed for collaborative documentation, inclusive of task plans, meeting minutes, and other shared sources. The actual-time editing characteristic ensured that each one crew participant got an entry to the latest information.

3. **Microsoft Teams:**

   Microsoft Teams turned into employed as a crucial hub for team communique. It facilitated actual-time chat, video conferencing, and record sharing. Within our project, Teams served as a platform for daily stand-up meetings, progress updates, and popular crew 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 become used for immediate messaging in which it's miles used for brief updates, urgent notifications, or informal verbal exchange amongst us. It affords a handy and widely handy manner of communique.

6.2.5 Computer Vision Libraries

1. YOLOv8 (You Only Look Once model eight) played an essential function in our task for actual-time object detection. Specifically, we hired YOLOv8 to enable the Road Eye gadget to recognize and perceive accidents. This era improved our surveillance skills and furnished accurate and speedy detection of incidents.

2. OpenCV, or cv2 in Python, was instrumental in photograph processing and manipulation inside the Road Eye system. It furnished essential features for dealing with pix and video streams, permitting us to put in force diverse pc vision responsibilities seamlessly.

6.2.6 AWS (Amazon Web Services):

We utilized AWS to host and set up the Road Eye gadget. The scalability and reliability of AWS offerings ensured that our answer ought to manipulate various masses and maintain excessive overall performance. Additionally, AWS's secure infrastructure contributed to the overall robustness of our device.

6.2.7 S3 Bucket:

Amazon S3 (Simple Storage Service) buckets inside AWS have been hired to keep and manage the big volumes of data generated by the Road Eye machine. This protected storing photos and video pictures captured by way of surveillance cameras, enabling seamless records get entry to and retrieval for analysis.

6.2.8 SMTP (Simple Mail Transfer Protocol):

SMTP was integrated into our gadget to allow email notifications for incident reviews. When the Road Eye machine detected an accident, it precipitated automated electronic mail alerts via SMTP, ensuring that applicable governments had been directly notified.

6.2.9 Roboflow:

Roboflow served as a valuable tool for preprocessing and augmenting our dataset earlier than model training. Its talents in fact transformation and enhancement contributed to enhancing the overall performance and accuracy of our YOLOv8 model.

6.2.10 Boto3:

Boto3, the Python SDK for AWS, changed to have interaction with AWS services programmatically. It enabled seamless communique with S3 buckets, facilitating information storage, retrieval, and management in the AWS infrastructure.

6.3 Implementation Details

In this section, we dive into the technical aspects of the Road Eye device, detailing the implementation of key components that contribute to the general functionality and achievement of the assignment.

6.3.1 YOLOv8 Integration for Real-Time Object Detection

The implementation of YOLOv8, an fundamental aspect of the Road Eye gadget, worried the subsequent steps:

Data Preparation:

A complete dataset of parking zone pictures turned into collected, annotated, and preprocessed to train the YOLOv8 model. This dataset protected various situations, lighting situations, and automobile sorts to make certain sturdy detection skills.

Model Training:

YOLOv8 organized a dataset of the usage of Google Colab's GPU acceleration. This accelerated the education manner and advanced the accuracy of item detection. Multiple iterations had been executed to exceptional-track the version for campus-particular situations.

Integration with Road Eye System: The educated YOLOv8 model seamlessly integrates into the Road Eye gadget. Real-time video feeds from surveillance cameras have been processed through the YOLOv8 algorithm to hit upon and classify vehicles in regions.

6.3.2 Python-Based Backend Development

We used python for its abundance of its libraries and the smooth live feature which helped up being the main platform to work on. Sending email, storing the file on S3 bucket etc.

6.3.3 AWS Infrastructure for Hosting and Deployment

Amazon Web Services (AWS) performed a crucial position in web hosting and deploying the Road Eye machine. Key implementation steps encompass:

Deployment on AWS EC2 Instances:

The Road Eye device was deployed on AWS EC2 times, providing scalable computing assets. This ensured the gadget's overall performance might be adjusted consistent with various workloads.

Utilization of S3 Buckets:

Amazon S3 buckets had been employed for steady garage and retrieval of massive volumes of photograph and video data captured by using the surveillance cameras. S3's reliability and scalability had been important for managing numerous statistics sorts.

6.3.4 Integration with Communication Tools

The Road Eye machine was incorporated with verbal exchange equipment to facilitate user interplay and incident reporting. Key information encompass:

SMTP for Email Notifications:

SMTP (Simple Mail Transfer Protocol) changed into applied to send automated email notifications to applicable authorities whilst an incident changed into detected. This ensured timely verbal exchange and reaction.

6.3.5 Collaborative Development using GitHub.

Collaborative improvement practices have been adopted using GitHub for version management. Key implementation info encompass:

1. **Branching Strategy:**

A well-defined branching strategy changed into applied to arrange development, checking out, and deployment stages. Feature branches had been used for character components, ensuring a systematic and organized improvement method.

2. **Continuous Integration:**

GitHub Actions were applied for non-stop integration, automating testing strategies and ensuring that the codebase remained stable throughout improvement.

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")
```

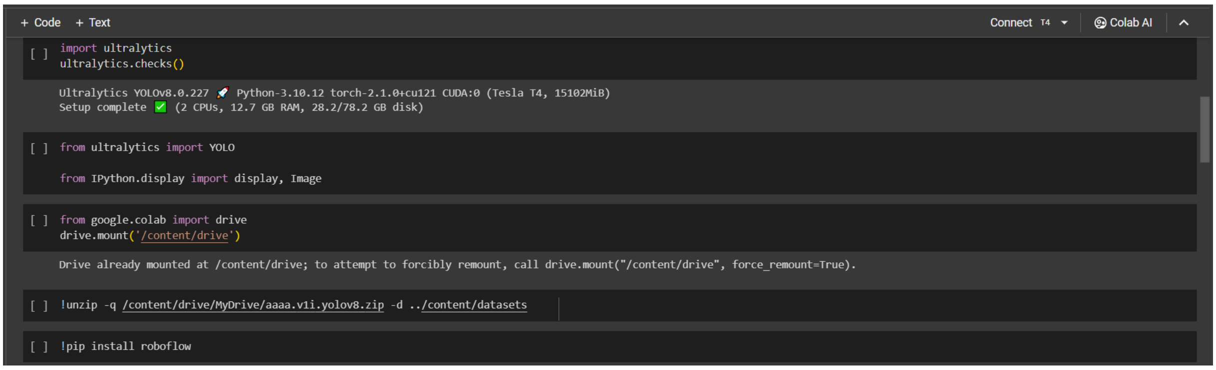
make a playback using cv2

```
    # 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')
```

6.4 Testing Phases and Results

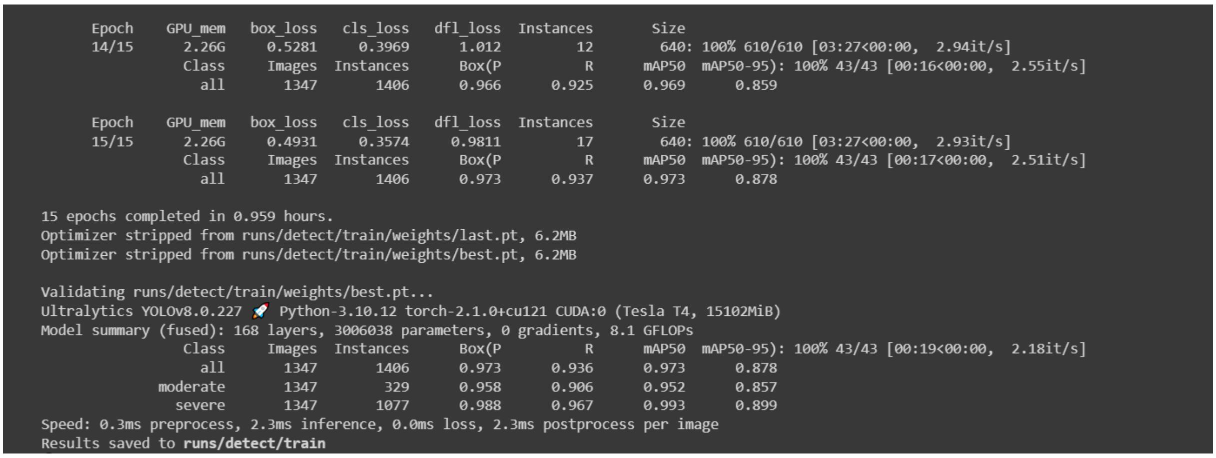
We used google colab to do our testing, as it’s one of the best and free software to run any machine learning code as it provides online clusters with a TGPU, CPU and RAM.

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%**



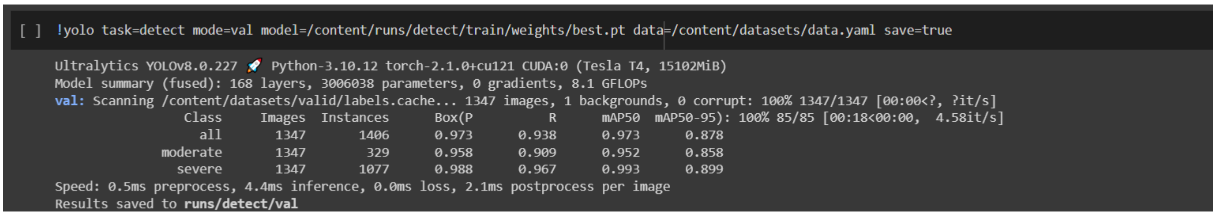
```
+ Code  + Text                                                  Connect T4   Colab AI
[ ] import ultralytics
    ultralytics.checks()

    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

6.4.1 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 (331/356) /content/drive/MyDrive/ColabNotebooks/testing.mp4: 384x640 (no detections), 7.7ms
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

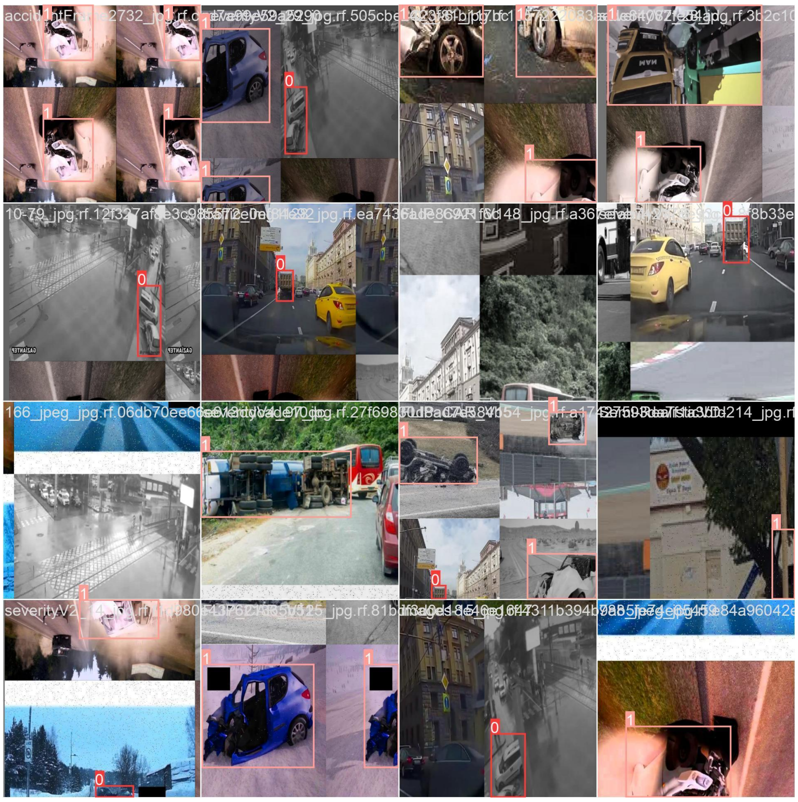


Figure 12

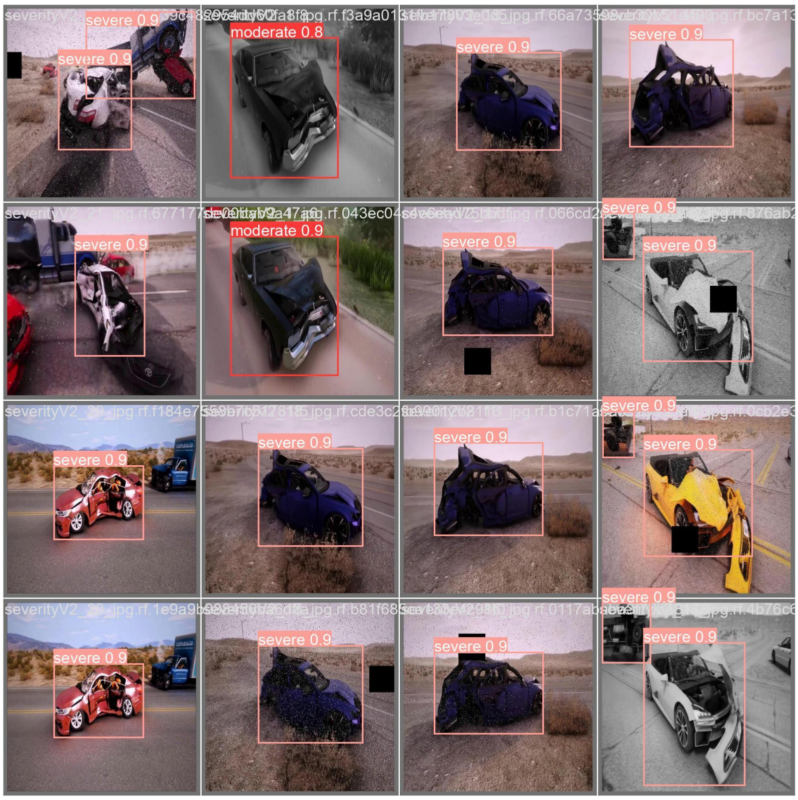


Figure 13

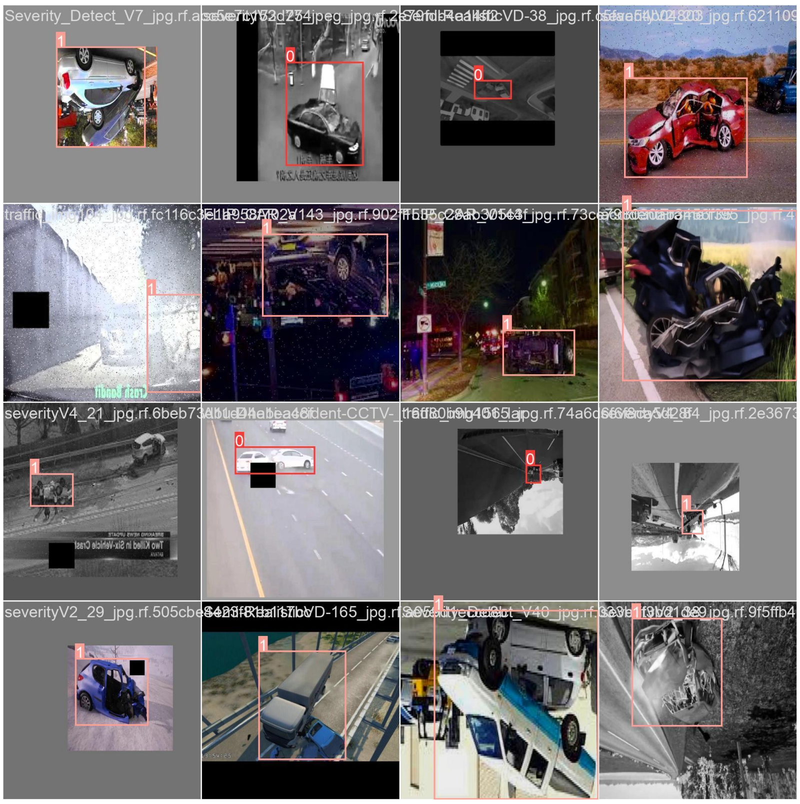


Figure 14

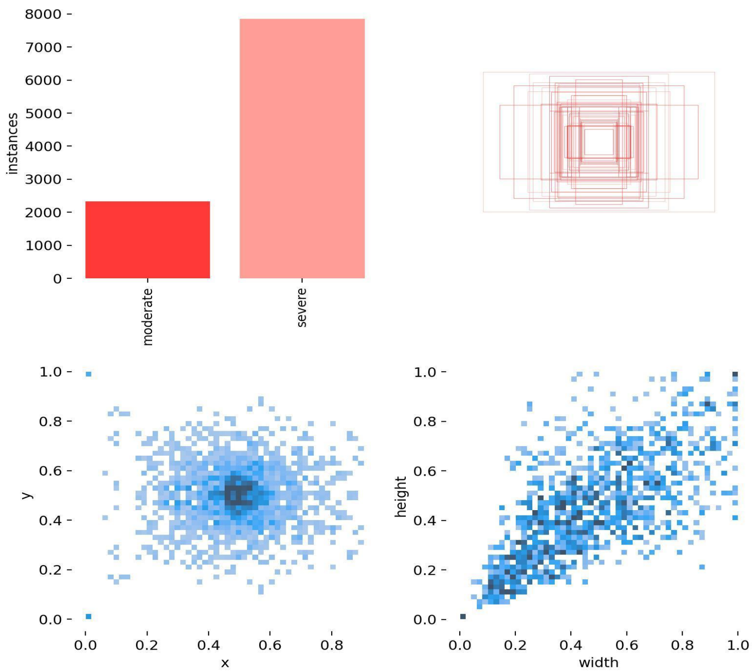
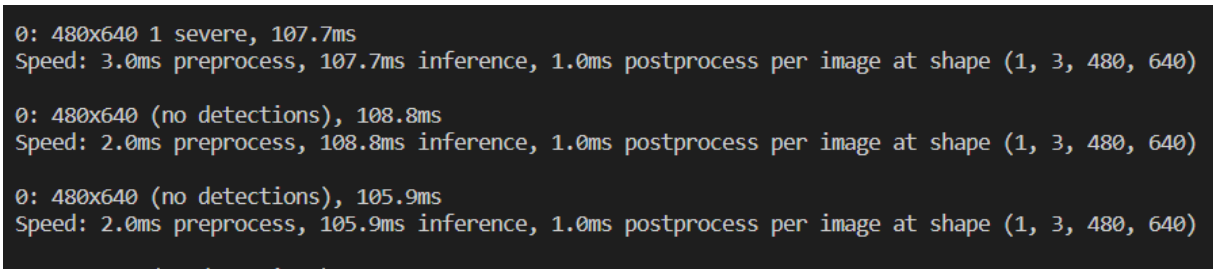

8000
7000
6000
5000
4000
3000
2000
1000
0
instances
moderate
severe
1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
x
height
1.0
width

Figure 15

6.4.2 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

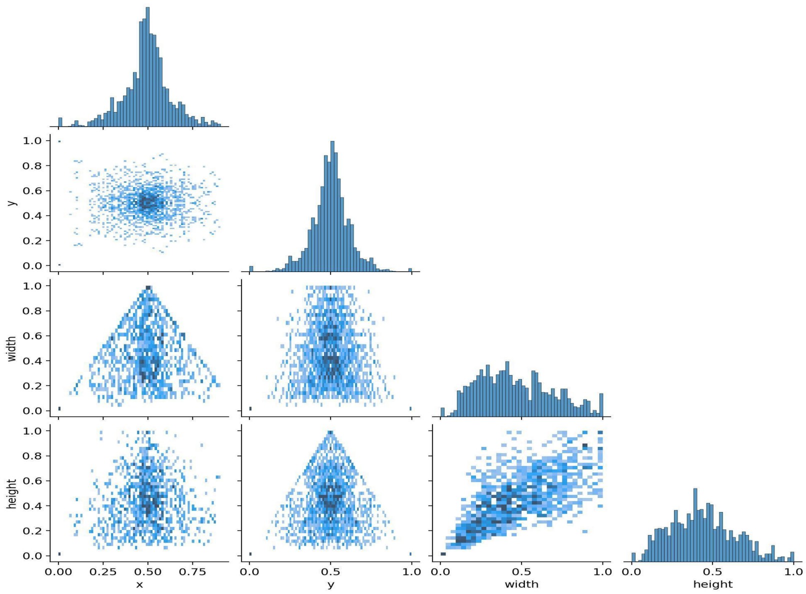


Figure 17

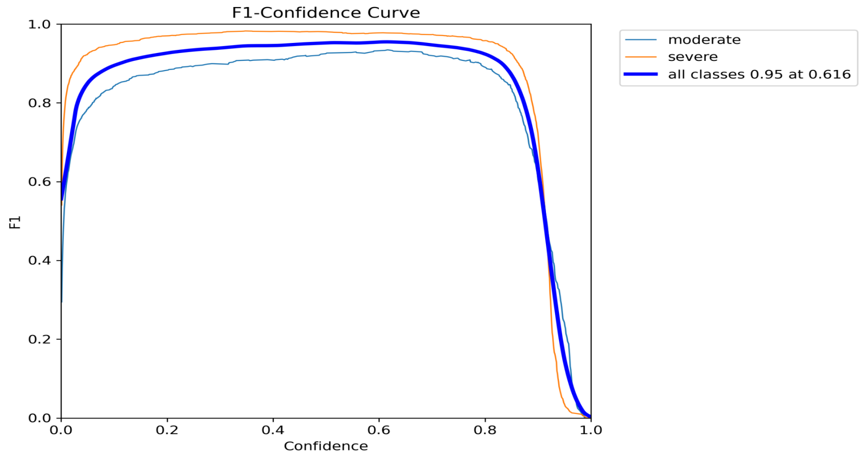


Figure 18

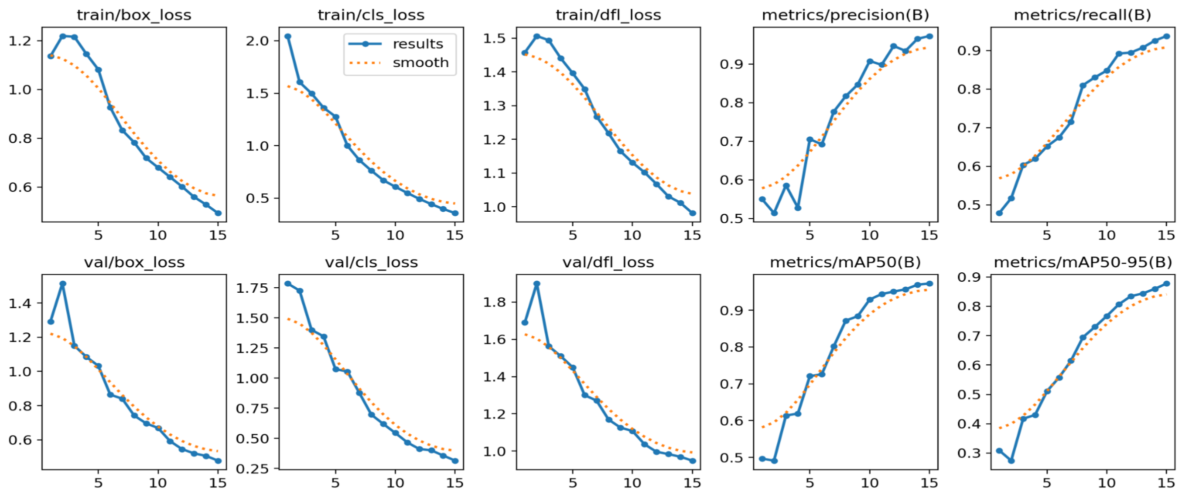


Figure 19

6.4.3 System Testing

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

6.4.4 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 numerous checking out, we reached a final format of the email to be dispatched, with a connected video clip of the accident in a .avi document with the time when it was dispatched.

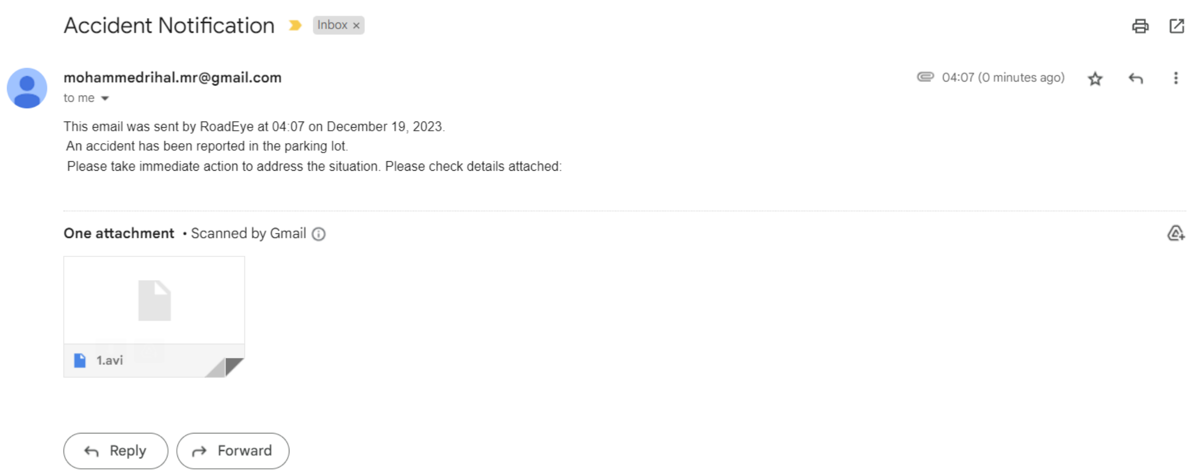


Figure 20: Email received by the higher authorities.

6.5 Discussion of Results and Comparison

After comparing the results of RoadEye with the other systems we found out that RoadEye is a capable competitor among them and has better readability score and it's easy to install. We do need to work on it more but that's explained in the future work, for now it's a prototype. With promising results, roadeye can be easily used in the University of Bahrain parking lot.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

In conclusion, the University of Bahrain's decision to use the Road Eye device is a significant step for improving our campus security and reducing incidents in the parking lots. Using YOLOv8n for object identification with the help of a python back-end and AWS S3 bucket led to the building of RoadEye.

7.1.1 Implementation

After we did the design, the implementation was smooth and we were ticking off the notelists. Vehicle detection was so

quick using the YOLOv8n model in our system. Real-world deployment was made easier by the Python back-end, and the AWS infrastructure deployment offered the flexibility and dependability needed for practical use.

7.1.2 Testing

Strict testing confirmed the efficacy and reliability of the Road Eye gadget. From the accuracy of the YOLOv8 model to the capability of the backend and the general device integration, complete testing eventualities had been employed. Simulated incident scenarios showed the gadget's responsiveness and its capability to make a contribution to campus protection.

7.1.3 Project Limitations

While acknowledging successes, it is important to understand the demanding situations that the project faced. Road Eye turned into stimulated with the aid of a number of factors, together with confined investment, challenging technological situations, and privateness issues. These demanding situations point out areas that must be evolved and advanced upon in destiny installments. In the stop, the Road Eye responsibility makes a full-size contribution to campus security. The gadget was placed to the test all through the design, deployment, and demonstration stages to make sure that it can have an effective, lasting effect on the security and welfare of the University of Bahrain network. With a commitment to constant innovation and improvement, the adventure embarked for this challenge serves as an anchor for ongoing improvements in campus protection.

7.1.4 Future Work

Future plans call for significant improvements to the Road Eye system in order to increase both its functionality and user satisfaction. The primary objective is to offer an intuitive online user interface that facilitates user interaction with event reports, real-time camera feeds, and related information. Simultaneously, a strong database integration can be positioned close by to efficiently handle and archive data, enabling efficient communication between front stop and backstop agents. Numerous cameras on the university premises will be distributed and turned on as the system advances to oversee these factors. The establishment of a dynamic digital camera management system and an extensive monitoring network are intended outcomes of this project.

References

1- 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.

2- 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.

3- 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.

4- 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].

5- Jocher, G. (2020). YOLOv8 Documentation. [online] docs.ultralytics.com. Available at: https://docs.ultralytics.com/.

6- 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].

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