Smart Under-Age Driver Detection System

Project Team

Muhammad Shaheer	20P-0480
Syed Ali Hasnain	20P-0460
Ramzan Ali	20P-0131

Session 2020-2024

Supervised by

Dr. Qasim Jan

Co-Supervised by

Dr. Hafeez-ur-Rehman

Department of Computer Science

National University of Computer and Emerging Sciences
Peshawar, Pakistan

March, 2024

Student's Declaration

We declare that this project titled "*Smart Under-Age Driver Detection System*", submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Muhammad Shaheer Signature: _____

Syed Ali Hasnain Signature: _____

Ramzan Ali Signature: _____

Verified by Plagiarism Cell Officer

Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Smart Under-Age Driver Detection System*, submitted by Muhammad Shaheer (20P-0480), Syed Ali Hasnain (20P-0460), and Ramzan Ali (20P-0131), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor

Dr. Qasim Jan Signature: _____

Co-Supervisor

Dr. Hafeez-ur-Rehman Signature: _____

_____

Mr. Haroon Zafar

FYP Coordinator

National University of Computer and Emerging Sciences, Peshawar

_____

Mr. Dr. Fazl-e-Basit

HoD of Department of Computer Science

National University of Computer and Emerging Sciences

Acknowledgements

We declare that this project titled “Smart Underage Driver Detection System”, submitted as a requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of the Department of Computer Science, National University of Computer and Emerging Sciences, has a zero-tolerance policy towards plagiarism. Therefore, we, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

Muhammad Shaheer

Syed Ali Hasnain

Ramzan Ali

Abstract

Road accidents are a persistent issue that require careful management of road safety measures, protocols, and responsible driving. However, ensuring safe driving practices can be challenging due to the need to balance various road safety precautions and regulations. This project aims to develop a smart system for detecting underage drivers, specifically designed to prevent them from driving and to issue fines if they violate the rules.

The "Smart Underage Driver Detection System" project uses machine learning to deliver accurate detection of drivers to enhance road safety. This report serves as a foundational document for project stakeholders, offering vital insights into the project's objectives, requirements, and design elements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the field of road safety management, addressing the issue of underage driving presents unique challenges that require innovative solutions. Recognizing the critical need for effective measures to prevent underage driving and enhance overall road safety, the development of a specialized Smart Underage Driver Detector system represents a significant advancement in this area. This innovative project leverages advanced technology to identify underage drivers in real-time, helping to enforce traffic laws and promote safer driving practices.

The primary objective of the Smart Underage Driver Detector is to create a comprehensive solution that detects underage drivers and prevents them from operating vehicles. By integrating state-of-the-art facial recognition and machine learning technology, the system can identify drivers below the legal driving age and issue automatic fines or warnings in cases of rule violations. This approach ensures adherence to road safety regulations and minimizes the risk associated with inexperienced or unauthorized drivers.

Beyond detection, the system offers a broader framework for road safety management. It facilitates the enforcement of legal driving age regulations while also contributing to data collection on road safety patterns, enabling more informed decision-making and policy development. By keeping detailed records of incidents involving underage drivers, the system fosters greater accountability and supports ongoing efforts to improve traffic safety.

Overall, the Smart Underage Driver Detector project represents a transformative step forward in the domain of road safety. By integrating advanced detection technology with a comprehensive approach to enforcement and data analysis, this system empowers authorities to take proactive measures against underage driving, promoting safer roads and reducing the risk of accidents.Our investigation aims to deal with the problem of underage driving on public roads by utilizing innovative technology and creating specialized systems while adhering to the previously stated success criteria. test.

- the purpose of the investigation
- the problem being investigated
- the background (context and importance) of the problem (citing previous work by others)
- your thesis and general approach
- the criteria for your study's success

1.1 Purpose of the Investigation

The purpose is to enhance the accuracy of detecting underage drivers using advanced machine learning techniques, including generative models and face recognition systems. By analyzing drivers' facial images to determine their age group, we address underage driving, a key factor in road accidents. This work seeks to improve road safety, promote responsible driving, and contribute to safer communities by proactively identifying and mitigating potential risks.

1.2 Problem Being Investigated

The core problem is the inaccuracy of driver image detection and the no availability of a suitable dataset for training an effective underage driver detection system. Current algorithms struggle with poor performance under varying conditions, such as low lighting,

obstructions, and diverse facial features, making it difficult to reliably distinguish between underage and non-underage drivers. Additionally, the absence of a well-labeled, diverse dataset tailored to this specific task is made more difficult the training of accurate machine learning models.

To address these issues, our research focuses on developing a robust data collection process and utilizing advanced algorithms like YOLOv3 and generative models to improve detection accuracy. By creating a comprehensive and diverse labeled dataset, we aim to overcome existing limitations and establish a reliable automated detection system, contributing to enhanced road safety.

1.3 Background (Context and Importance) of the Problem

The background and significance of this issue are illustrated by the rapidly changing nature of accidents on busy highways. Driving while hiding from police detection has changed rapidly with billions of drivers globally. Our aim is to provide an automated system that can recognize underage drivers from facial images by using previous studies and developments in generative machine learning, face recognition.

1.4 Thesis and General Approach

Our thesis is based on the idea that identifying underage drivers on public roads and toll plazas can be modernized by combining face recognition technology with generative machine learning. We propose a method that uses facial recognition to identify drivers who violate the law. Our strategy is to develop a platform that uses machine learning to identify drivers both during the day and at night with the highest level of accuracy.

1.5 Criteria for Your Study's Success

The success of our study will be evaluated based on the following criteria:

1. **Accurate Detection of Underage Drivers**: The system's ability to reliably identify underage drivers and generate actionable alerts for the control center.
2. **Impact on Road Safety**: Measured by the system's potential to reduce traffic accidents caused by underage and immature driving.
3. **System Performance and Efficiency**: Assessed through metrics such as detection accuracy, response time, and resource utilization.
4. **Adherence to Timeline and Budget**: Tracked through regular project milestones and efficient use of allocated resources.

In literature review, each paper is not reviewed individually like this. There must be some flow between the papers you are reviewing.

Chapter 2

Review of Literature

Recent advancements in computer vision and machine learning have driven progress in developing smart underage driver detection systems. Combining facial recognition, deep learning, and real-time picture analysis, these technologies play a vital role in enhancing traffic safety and supporting law enforcement. This literature review evaluates existing methods, highlights their effectiveness, and addresses key challenges, providing a foundation for further innovation in underage driver detection.

2.1 Related Works

2.1.1 Age And Gender Detection

Basic Idea: This project tackles the problem of unreliable internet content causing fear and harm. By identifying and categorizing the age and gender of the people sending these messages, the goal is to stop their spread and prevent negative outcomes.

Methodology: The methodology involves data input, tokenization, string-to-word vectorization, feature selection, classification using algorithms like random forest and naive Bayes, and result evaluation.

Result: Able to detect age

Limitation: Insufficient data, text tokenization challenges, biases in feature selection,

potential overfitting, and difficulty predicting age/gender accurately from text alone. Performance may vary based on data quality and diversity.

Citation Count:1

2.1.2 Development of Model for Estimation of Age Group of Drivers by using Soft Computing Tools

Basic Idea: The paper proposes a method to estimate drivers' age groups by analyzing 263 facial parameters extracted from traffic surveillance videos. These parameters aid in categorizing drivers by age, facilitating targeted interventions to improve traffic safety and reduce accidents.

Methodology: The methodology involves collecting traffic video from signal areas for clear facial images, followed by face detection using the Skin Colour method. Then, an Artificial Neural Network (ANN) is trained with calculated parameters and distance-based features to accurately estimate drivers' age groups.

Result: The model detects the age

Limitation: Model can give result only if it is straight towards the driver.

Sideway pictures are not handled

2.1.3 Underage Driving Detection - Age Recognition Using Face Detection

Basic Idea: The study faces challenges in predicting age accurately from facial features due to factors like genetics and lifestyle. It aims to detect and prevent underage drivers, with room for improvement in model accuracy through additional data and better network designs.

Methodology: The system faces challenges in preventing underage driving due to human limitations. Machine learning automates face detection and age prediction using a DNN Face Detector and Caffe model trained on the Adience dataset, enhancing accuracy for

proactive measures against underage driving.

Result: Able to detect age of the person.

Limitation: The camera should be in front of the driver's face only to get accurate results

2.1.4 Driver Distraction Detection Methods

Basic Idea: Reviews driver distraction detection methods and proposes a framework to enhance detection accuracy.

Methodology: Analyzed existing literature, identified gaps, and presented a conceptual framework for distraction detection using machine learning and computer vision.

Result: Proposed methods demonstrated potential for reducing accident risks, with enhanced detection speed and accuracy.

Limitation: The framework lacks implementation or real-world testing.

Citation Count:32

2.1.5 Vehicle License Plate Detection and Recognition Using Symbol Analysis

Basic Idea: Introduces a symbol-analysis-based method for recognizing vehicle license plates.

Methodology: Used symbol segmentation and classification to improve plate detection accuracy in various lighting conditions.

Result: Achieved 94accuracy in controlled environments.

Limitation: Struggles with poor lighting or occluded plates.

Citation Count:145

In the end, there can be a table that shows the citation

2.1.6 Driver Behavior Analysis for Safe Driving

Basic Idea: Focuses on understanding and modeling driver behaviors for safe driving.

Methodology: Employed data from in-car sensors and cameras to analyze driving patterns and risky behaviors.

Result: Enhanced prediction of high-risk driving situations, achieving 90simulations.

Limitation: Requires expensive equipment and large datasets.

Citation Count:67

2.1.7 A Lightweight Driver Distraction Detection Method Based on Three Level Attention Mechanisms

Basic Idea: Introduces a lightweight model to detect driver distractions with attention mechanisms.

Methodology: Combined computer vision and a novel three-level attention model to detect gaze shifts and distractions.

Result: Achieved real-time detection with 95accuracy

Limitation: Computational efficiency is high but limited to well-lit environments.

Citation Count: Not available yet (new paper)

2.1.8 Aerial Target Detection Based on the Improved YOLOv3 Algorithm

Basic Idea: Enhance YOLOv3 for detecting aerial targets.

Methodology: Modified backbone network, improved feature extraction, and optimized loss function.

Result: Improved detection accuracy and speed.

Limitation: Limited dataset size and potential overfitting.

Citation Count: 56

2.1.9 CRTSII Track Slab Crack Detection Based on Improved YOLOv3 Algorithm

Basic Idea: Enhance YOLOv3 for detecting cracks in railway tracks.

Methodology: Modified backbone network, improved feature extraction, and data augmentation.

Result: Improved detection accuracy and robustness.

Limitation: Limited dataset diversity and potential environmental impact.

Citation Count: 20

2.1.10 Fisheye Image Object Detection Based on an Improved YOLOv3 Algorithm

Basic Idea: Enhance YOLOv3 for object detection in fisheye images.

Methodology: Modified backbone network, improved feature extraction, and geometric transformation.

Result: Improved detection accuracy and robustness.

Limitation: Limited dataset size and potential computational cost.

Citation Count: 20

2.1.11 Edge Detection-Based Boundary Box Construction Algorithm for Improving the Precision of Object Detection in YOLOv3

Basic Idea: Improve YOLOv3's bounding box prediction using edge detection.

Methodology: Edge detection-based refinement of bounding boxes.

Result: Improved detection accuracy and precision.

Limitation: Potential computational overhead and sensitivity to edge detection quality.

Citation Count: 22

2.1.12 Underwater Object Detection Model Based on YOLOv3 Architecture Using Deep Neural Networks

Basic Idea: Enhance YOLOv3 for underwater object detection.

Methodology: Modified backbone network, improved feature extraction, and data augmentation.

Result: Improved detection accuracy and robustness.

Limitation: Limited dataset size and challenging underwater imaging conditions.

Citation Count: 8

2.1.13 YOLOv3-SPP for Vehicle Detection in Aerial Images

Basic Idea: Enhance YOLOv3 with a Spatial Pyramid Pooling (SPP) module for vehicle detection in aerial images.

Methodology: Incorporate the SPP module into the YOLOv3 architecture to capture multi-scale features.

Result: Improved detection accuracy and robustness, especially for small and distant vehicles.

Limitation: Potential computational overhead due to the additional SPP layer.

Citation Count: 57

2.1.14 Improved YOLOv3 for Real-Time Object Detection on Edge Devices

Basic Idea: Optimize YOLOv3 for deployment on resource-constrained edge devices.

Methodology: Employ techniques like model pruning, quantization, and knowledge distillation to reduce model size and computational cost.

Result: Improved inference speed and reduced model size, enabling real-time object detection on edge devices.

Limitation: Potential loss of accuracy due to aggressive model compression.

Citation Count: 10

Chapter 3

Project Vision

3.1 Problem Statement

Currently, underage driving remains a major concern due to its significant contribution to traffic accidents and fatalities. Despite strict legal regulations, underage drivers often go undetected, primarily because of inefficient and error-prone manual detection methods. This issue impacts road safety, as young drivers lack the necessary experience and understanding of traffic rules. Our FYP aims to address this problem by developing an automated system using advanced technologies like facial recognition and machine learning to accurately identify underage drivers in real-time, improving traffic enforcement and reducing road accidents.

3.2 Business Opportunity

- **Enhanced Road Safety:** The system offers significant improvements to road safety by accurately identifying underage drivers, reducing accidents, and saving lives. This can be integrated with existing traffic monitoring systems to enhance public safety.
- **Law Enforcement and Transportation Integration:** The technology can be marketed to law enforcement agencies and transportation authorities, offering them

real-time solutions for preventing underage driving and improving traffic enforcement.

- **Vehicle Manufacturer Collaboration:** Vehicle manufacturers can adopt this technology to enhance in-car safety features, providing added value to their product offerings while promoting responsible driving.

- **Standalone Solutions:** The system can be developed as a standalone product, offering a scalable solution for deployment in high-risk areas, ensuring widespread impact on road safety.

- **Data Analytics and Insights:** The system can collect valuable data on driving behavior, which can be used for traffic management, urban planning, and insurance purposes, potentially generating additional revenue through data services.

3.3 Objectives

The objectives of the project are as follows:

1. Develop a machine learning model capable of accurately detecting underage drivers based on facial recognition and other relevant features.

2. Implement digital image processing techniques to extract and analyze facial characteristics for age estimation.

3. Integrate the detection system with real-time monitoring mechanisms to alert authorities and stakeholders in case of underage driving incidents.

4. Conduct rigorous testing and validation to ensure the system's accuracy, reliability, and efficiency under various conditions.

5. Provide a user-friendly interface for stakeholders to access and manage the system effectively.

3.4 Project Scope

The scope of our project involves developing a machine learning-based system for detecting underage drivers using facial recognition and age estimation techniques. It will integrate advanced digital image processing for feature extraction and preprocessing. The system will be designed for real-time underage driver detection, either through integration with existing traffic monitoring infrastructure or as a standalone solution. The project includes testing and optimizing the system's performance, ensuring its reliability across various conditions. Additionally, the development process and functionalities will be documented for future reference and academic evaluation.

3.5 Constraints

The project faces certain constraints, including:

1. **Data Constraints**: The project faces challenges related to the availability and quality of high-quality training data, which is essential for machine learning model development.

2. **Computational Constraints**: Significant computational resources are required for training and testing complex algorithms efficiently.

3. **Regulatory Constraints**: The project must adhere to legal and regulatory considerations, particularly concerning privacy and data protection.

4. **Budget Constraints**: Financial limitations may affect the ability to invest in necessary hardware, software, and infrastructure.

5. **Time Constraints**: The project has strict time limits for completion and deployment, requiring efficient task management and timely execution.

3.6 Stakeholders Description

3.6.1 Stakeholders Summary

The stakeholders involved in the project include:

1. **Law Enforcement Agencies:** Responsible for enforcing regulations and ensuring road safety.
2. **Transportation Authorities:** Concerned with implementing measures to improve road safety and traffic management.
3. **Vehicle Manufacturers:** Involved in integrating safety technologies into vehicles and complying with regulatory standards.
4. **Parents and Guardians:** Concerned about the safety of underage drivers and seeking measures to prevent unauthorized vehicle usage.
5. **Insurance Companies:** Interested in risk assessment and offering policies based on underage driving detection data.
6. **Research Institutions:** Focused on advancing AI and machine learning technologies for traffic safety.
7. **General Public:** Beneficiaries of enhanced road safety measures and reduced risks associated with underage driving.
8. **Software Developers:** Responsible for the development, deployment, and maintenance of the detection system.

3.6.2 Key High-Level Goals and Problems of Stakeholders

- **Law Enforcement Agencies:** *Goal:* To reduce incidents of underage driving and enhance overall road safety. *Problem:* Difficulty in efficiently detecting underage drivers without relying on intrusive methods.

- **Transportation Authorities:** *Goal:* To implement effective measures to prevent underage driving and minimize associated risks. *Problem:* Limited availability of technological solutions for proactive detection and prevention.

- **Vehicle Manufacturers:** *Goal:* To integrate advanced safety features into vehicles while meeting regulatory standards. *Problem:* Challenges in incorporating underage driver detection systems into vehicle designs.

- **Parents and Guardians:** *Goal:* To ensure the safety of underage drivers and restrict unauthorized vehicle usage. *Problem:* Lack of reliable, non-intrusive monitoring methods for underage driving activities.

- **Educational Institutions:** *Goal:* To promote safe driving habits among students and discourage underage driving. *Problem:* Limited access to tools that help monitor and educate young drivers effectively.

- **Insurance Companies:** *Goal:* To minimize risk and ensure accurate driver profiling for underwriting policies. *Problem:* Difficulty in obtaining reliable data about driver age and associated risks.

- **Developers:** *Goal:* To create innovative solutions that address societal challenges like underage driving. *Problem:* Challenges in developing accurate, efficient, and scalable detection algorithms.

- **General Public:** *Goal:* To experience improved road safety and reduced risks from underage driving incidents. *Problem:* Concerns over the effectiveness and privacy implications of detection technologies.

Chapter 4

Software Requirements Specifications

4.1 List of Features

- Machine learning-based underage driver detection
- Facial recognition for driver identification
- Digital image processing for age estimation
- Real-time monitoring and alerting mechanisms
- User-friendly interface for stakeholders
- Integration with existing infrastructure
- Standalone deployment options
- Testing and validation of system performance
- Documentation of development process and functionalities
- High-quality training data acquisition
- Computational resource optimization
- Regulatory compliance considerations

- Budget management for hardware and software
- Timely project completion and deployment

4.2 Functional Requirements

1. User Registration:
    - Users can sign up using their email.
    - They'll get an email to confirm their account.
2. User Profile Management:
    - Users can upload and change their profile picture.
    - They can also update their name, location, and bio.
3. Age Detection:
    - The system should guess how old the driver is.
    - It should do this by looking at their face.
4. Add New Camera Location:
    - The system should add a new camera location.
    - The user can modify details
5. Real-Time Monitoring:
    - Keep an eye on the driver's face and guess their age continuously.
    - Show the result right away.
6. Alert System:
    - The data captured is sent to the screen to view as a pop-up if it's underage.
7. User Interface:

- Make a simple and easy-to-use screen for authorized users.
- They should be able to see what's happening in real time, manage alerts, and update profiles.

8. Privacy Protection:
   - Keep people's private information safe.
   - Make sure only authorized people can see the data.
   - The data should be stored securely so nobody can snoop on it.

4.3 Quality Attributes

- **Accuracy:** The system should accurately detect underage drivers to minimize false positives and negatives.
- **Reliability:** The system should operate reliably under various conditions, ensuring consistent performance.
- **Efficiency:** The system should process data efficiently, minimizing computational resources and response time.
- **Scalability:** The system should be scalable to accommodate increasing data volume and user demands.
- **Robustness:** The system should be robust against noise, variations in lighting conditions, and other environmental factors.
- **Security:** The system should ensure the security and privacy of sensitive data, complying with regulatory standards.
- **Usability:** The system should have a user-friendly interface, making it easy for stakeholders to access and manage.
- **Maintainability:** The system should be easy to maintain and update, facilitating long-term support and enhancements.

- **Interoperability:** The system should be interoperable with existing infrastructure and compatible with different platforms.
- **Adaptability:** The system should be adaptable to evolving technological advancements and changing requirements.

4.4 Non-Functional Requirements

- **Performance:** The system should respond to requests within a maximum latency of 500 milliseconds.
- **Availability:** The system should be available 99 percent of the time, allowing for scheduled maintenance windows.
- **Scalability:** The system should scale horizontally to accommodate up to 100,000 concurrent users.
- **Reliability:** The system should have a mean time between failures (MTBF) of at least 10,000 hours.
- **Security:** The system should encrypt all sensitive data in transit and at rest using AES-256 encryption.
- **Privacy:** The system should comply with GDPR regulations, ensuring the anonymity of user data.
- **Usability:** The system should provide clear error messages and intuitive user interfaces for all user interactions.
- **Maintainability:** The system should be modularized and well-documented to facilitate future updates and maintenance.
- **Compatibility:** The system should be compatible with the latest versions of popular web browsers (Chrome, Firefox, Safari).
- **Regulatory Compliance:** The system should adhere to relevant laws and regulations governing underage driver detection and data privacy.

4.5 Use Cases / Use Case Diagram

4.5.1 Use Case Diagram

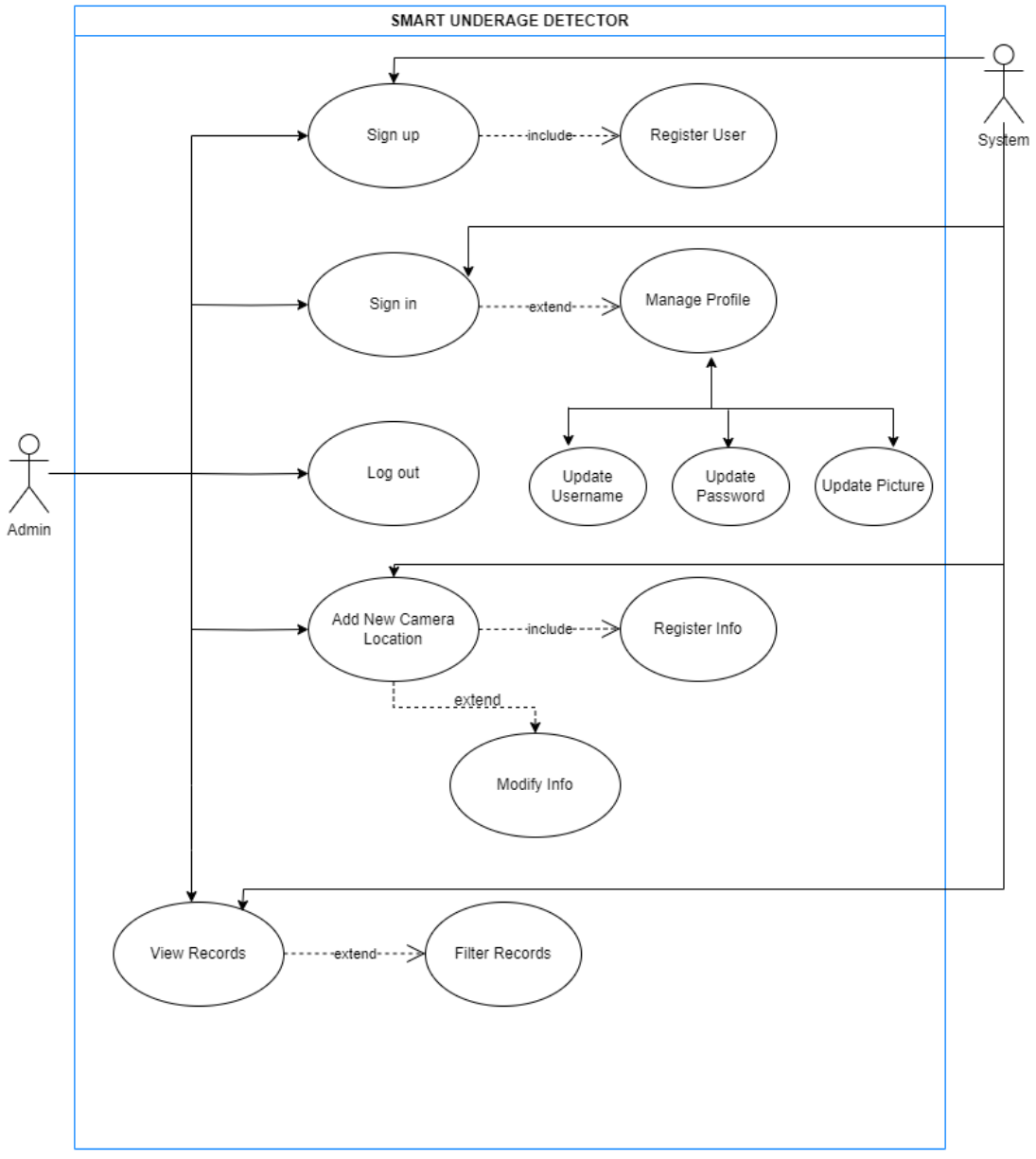


Figure 4.1: Use Case Diagram

4.5.1.1 Use Cases

Use Case ID	1
Use Case Name	Signup
Use Case Description	This use case allows new users to register and create an account within the system.
Actor	Admin
Precondition	Internet Access Should be Working
Main Flow	1. The user clicks on "Signup" to start the registration process. 2. A form appears asking for the user's details, such as username, email, and password. 3. The user fills in the required information. 4. After completing the form, the user submits it. 5. The system checks if the provided information is correct. 6. If everything checks out, the system creates a new user account. 7. The user is then logged in and redirected to their dashboard.
Alternate Flow	None
Post Condition	User has successfully registered and is logged into the system.
Exceptions	1. If there's a system error during registration: • The system displays an error message. • Provides options for the user to retry or contact support. 2. If the user provides incomplete or invalid registration information: • The system displays error messages. • Prompts the user to correct the information. 3. If the user's chosen username is already in use: • The system displays a message asking the user to select a different username. 4. If the email provided is already associated with an existing account: • The system notifies the user to use a different email address.

Table 4.1: Use Case: Signup

Use Case ID	2
Use Case Name	Sign in
Use Case Description	Allows registered users to sign in by entering their username and password. This use case allows users to manage their profile information, including updating profile details.
Actor	Admin
Precondition	The User must have a registered account. User has selected the "Manage Profile" option.
Main Flow	1. The user accesses the login page. 2. The system displays the login form. 3. The user enters their username and password. 4. The system validates the provided credentials. 5. If valid, the system logs in the user. 6. The user is redirected to their dashboard or the system's main page. 7. The user can update their username and password in the profile. 8. The user can update their profile picture. 9. The user selects the "Manage Profile" option from the user dashboard. 10. The system presents options for updating different profile attributes. 11. The user may choose to update the profile picture, password, interests, and username. 12. The system associates the "Manage Profile" use case with the relevant sub-use cases (e.g., "Update Password," "Update Interests," "Update Username") to handle the specific updates.
Alternate Flow	None
Post Condition	The user successfully signs in and gains access to their account and can manage the profile.
Exceptions	1. If incorrect credentials, the system displays an error message and offers a retry option. 2. For forgotten passwords, the system provides a password reset option via email verification. 3. In case of system errors, the system displays an error message with retry or support contact options.

Table 4.2: Use Case: Signin

Use Case ID	3
Use Case Name	Log out
Use Case Description	Allows users to log out from their accounts, ending their current session.
Actor	Admin
Precondition	The User must be logged in.
Main Flow	1. User selects the "Log Out" option, typically found in the user dashboard or settings. 2. The system confirms the user's decision. 3. The system terminates the user's session. 4. The user is logged out and returned to the login page or a logged-out state.
Alternate Flow	None
Post Condition	User's session is terminated, and they are logged out.
Exceptions	None

Table 4.3: Use Case: Logout

Use Case ID	4
Use Case Name	Add New Camera Location
Use Case Description	Allows users to add a location from where the camera will detect the results.
Actor	Admin
Precondition	The User must be logged in.
Main Flow	1. The user accesses the add location page. 2. The user views the added location data.
Alternate Flow	None
Post Condition	Success message shown by system.
Exceptions	If there is a system error, the system displays an error message.

Table 4.4: Use Case: Add Camera Location

Use Case ID	5
Use Case Name	View Records
Use Case Description	The user can check all underage drivers detected by the camera and then filter them.
Actor	Admin
Precondition	User is registered and logged in.
Main Flow	1. The user clicks on one of the entries. 2. The user is able to see the person's face image and the location where it was captured. 3. The system displays the info of that driver. 4. The system then filters the records.
Alternate Flow	None
Post Condition	The data is shown by the system to the user.
Exceptions	If there is a system error during the interaction, the system displays an error message and offers the user the option to retry or contact support.

Table 4.5: Use Case: View Records

4.6 Sequence Diagrams/System Sequence Diagram

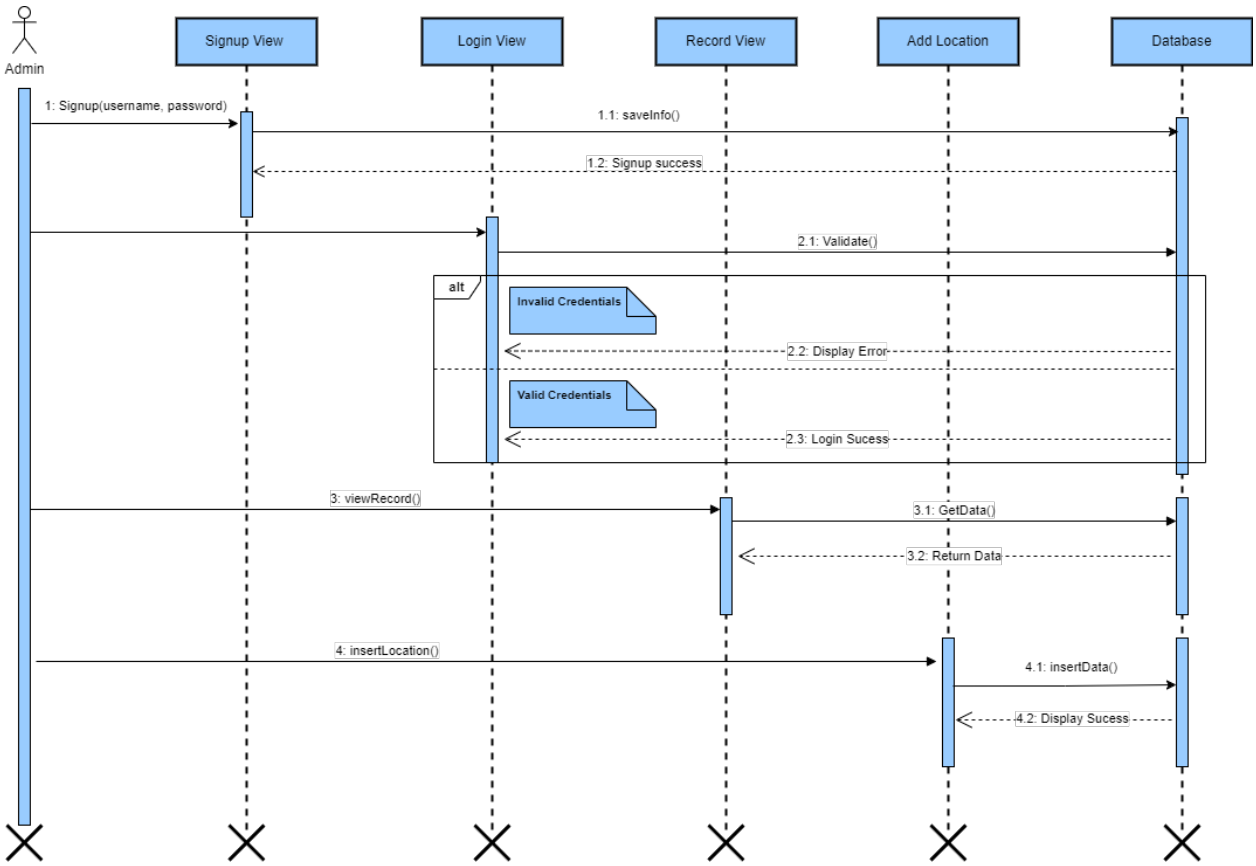


Figure 4.2: Sequence Diagram

4.7 Test Plan (Test Level, Testing Techniques)

Use Case: Register User

Description: The users can create an account, providing basic information such as username, email, and password.

Use Case: Register User

Description: A new user registers on the platform using valid credentials.

Use Case: Register User

Description: A new user registers on the platform using valid credentials.

Serial	Test Case	Test Case Description	Test Steps	Input Values	Expected Result
REG-TC001	Valid Registration	Verify successful registration of a new user with valid information.	• Access the registration page. • Enter valid user details (username, email, password, etc.). • Click the "Register" button.	- Username- Email- Password	User registration is successful, and the system confirms registration.
REG-TC002	Existing Email	Ensure the system handles registration with an email that already exists.	• Access the registration page. • Enter a valid user email that already exists in the system. • Fill in other required details. • Click the "Register" button.	- Email	The system detects the existing email and prompts the user to choose a different email for registration.
REG-TC003	Weak Password	Validate that the system enforces password strength requirements.	• Access the registration page. • Enter valid user details, including a weak password. • Click the "Register" button.	- Password	The system rejects the registration due to the weak password and provides guidance on password requirements.

Table 4.6: Test Cases: RegisterUser

Use Case: Login User

Description: The registered users log in using their credentials.

Serial	Test Case	Test Case Description	Test Steps	Input Values	Expected Result
LOGIN-TC001	Valid Login	Verify successful login with valid credentials.	• Access the login page. • Enter valid email and password. • Click the "Log In" button.	- Email- Password	User is logged in successfully, and the system directs the user to the dashboard.
LOGIN-TC002	Incorrect Email	Check the system's response to an incorrect email.	• Access the login page. • Enter an incorrect email and valid password. • Click the "Log In" button.	- Email	The system identifies the incorrect email and prompts the user to enter a valid email.
LOGIN-TC003	Incorrect Password	Ensure the system handles an incorrect password.	• Access the login page. • Enter a valid email and an incorrect password. • Click the "Log In" button.	- Password	The system detects the incorrect password and requests the user to enter the correct password.

Table 4.7: Test Cases: LoginUser

4.8 Software Development Plan

4.9 Introduction

This software development plan outlines the strategy for creating the **Smart Underage Driver Detection System**, a solution designed to identify and prevent underage individuals from driving vehicles, thereby improving road safety. The system uses advanced technologies like machine learning and face recognition to analyze drivers' facial features and accurately determine their age group. The development process includes gathering a suitable dataset, designing a robust detection mechanism, training machine learning models, and integrating the system with existing traffic monitoring infrastructure. The ultimate goal is to reduce accidents caused by inexperienced drivers and provide law enforcement

with an efficient tool to ensure compliance with traffic laws, contributing to safer roads and communities.

4.10 Project Overview

The **Smart Underage Driver Detection System** uses advanced machine learning and image processing to identify underage drivers based on facial recognition and other key features. It analyzes live images from traffic cameras and vehicle dashboards to detect underage individuals accurately, even in challenging conditions. Once identified, the system sends real-time alerts to authorities and stakeholders, enabling swift action to prevent accidents or violations. Designed for easy integration with existing traffic infrastructure, it offers a scalable and efficient solution to enhance road safety and promote responsible driving practices.

4.11 Development Methodology

The development methodology for this project will follow the **Agile approach**, which supports iterative development and continuous feedback from stakeholders. This methodology ensures that the project can evolve and adapt in response to changing requirements and feedback throughout the development lifecycle. By breaking the project into smaller, manageable tasks or sprints, Agile allows for regular evaluation and refinement of the system. This approach promotes flexibility, enabling the development team to quickly address challenges and make adjustments as needed. As the system evolves, constant communication with stakeholders ensures that the end product aligns with user needs and project goals, leading to a more effective and user-centered solution.

4.12 Project Organization

The project team will consist of software engineers, machine learning specialists, and domain experts. Roles and responsibilities will be assigned as follows:

- Project Manager: Oversees the overall project execution and ensures alignment with project objectives.
- Software Engineers: Responsible for software development, coding, and testing.
- Machine Learning Specialists: Develop and fine-tune machine learning algorithms for facial recognition and age estimation.
- Domain Experts: Provide insights into underage driving patterns and regulatory requirements.

4.13 Development Tools

The development tools and technologies to be used include:

- Programming Languages: Python for backend development, HTML/CSS/JavaScript/Flask for frontend development.
- Frameworks: TensorFlow and OpenCV for machine learning and image processing.
- Version Control: Git for source code management and collaboration.
- IDEs: Jupyter Notebook for Python development, Visual Studio Code for web development.

4.14 Development Environment

The development environment will require:

- High-performance computers with GPU acceleration for machine learning model training.
- Web servers for hosting the application and database servers for storing user data securely.

4.15 Project Schedule

The **project schedule** will outline all key milestones, tasks, and deadlines for the development of the **Smart Underage Driver Detection System**. It will cover all stages of the project, including requirement analysis, data collection, system design, model development, testing, and deployment. Each phase will have specific tasks with clear timelines to ensure progress is tracked and the project stays on schedule. The schedule will be continuously updated and shared with stakeholders, allowing for timely adjustments based on feedback and progress. Regular meetings will ensure all team members are aligned and that any issues are addressed promptly, helping the project stay on track for successful completion.

4.16 Risk Management

Potential risks and challenges associated with the development and deployment of the **Smart Underage Driver Detection System** will be thoroughly identified, assessed, and managed. Risks may arise in various areas, such as data privacy concerns, regulatory compliance, and technical complexities. For example, data privacy concerns may emerge due to the use of facial recognition technology, and steps will be taken to ensure compliance with data protection laws like GDPR. Additionally, the system's integration with existing infrastructure may pose technical challenges, such as compatibility with different camera systems or software platforms. A robust risk mitigation strategy will be implemented, which includes regular risk assessments, implementing security protocols, and creating contingency plans for addressing unforeseen issues during development and deployment.

4.17 Quality Assurance

A comprehensive **quality assurance** process will be put in place to ensure the Smart Underage Driver Detection System meets the highest standards of performance, reliability, and user satisfaction. This will involve several layers of testing, including **unit testing** to

verify the functionality of individual components, **integration testing** to check the interaction between system modules, and **user acceptance testing (UAT)** to ensure the system meets the needs and expectations of stakeholders. Clear and measurable quality metrics, such as system accuracy, processing time, and user experience ratings, will be defined to evaluate the success of each phase. A feedback loop will be established to address issues promptly and ensure continuous improvement in the system's overall quality.

4.18 Documentation

Comprehensive **documentation** will be created for all aspects of the Smart Underage Driver Detection System. This will include **user manuals** to guide end-users on how to interact with the system, **technical specifications** detailing the system architecture, components, and configuration requirements, and **developer documentation** to assist future developers in maintaining or expanding the system. Clear, concise, and well-organized documentation will facilitate ease of use for stakeholders and ensure that the system can be understood, managed, and maintained by both technical and non-technical users. Additionally, documentation will be updated regularly to reflect any system updates or enhancements.

4.19 Deployment Plan

The **deployment plan** will be developed to ensure a smooth transition from development to a live operational environment. This plan will include detailed steps for installing, configuring, and deploying the Smart Underage Driver Detection System, including server setup, database configuration, and system integration with existing traffic monitoring infrastructure. To ensure stakeholders can effectively utilize the system, user training sessions will be organized, covering how to interact with the system, interpret alerts, and perform basic troubleshooting. Support documentation and training materials will also be provided to ensure users are fully equipped to use the system efficiently.

4.20 Maintenance and Support

A comprehensive **maintenance and support plan** will be established to ensure the system remains reliable and effective post-deployment. This will include routine **bug fixes**, addressing issues that arise in the operational environment, and **system updates** to incorporate new features or enhancements. Additionally, regular performance evaluations will be conducted to ensure the system operates at peak efficiency. Any necessary updates to the system will be rolled out, including security patches, to address emerging threats and vulnerabilities. Support will also be available to end-users, with a helpdesk system for reporting issues and receiving assistance. This proactive approach will ensure the system continues to meet safety standards and adapt to new requirements.

4.21 Conclusion

The software development plan for the **Smart Underage Driver Detection System** outlines the comprehensive strategy and methodologies necessary to develop, deploy, and maintain an effective solution aimed at enhancing road safety. By following the outlined approach, which incorporates risk management, quality assurance, thorough documentation, and continuous support, we aim to deliver a robust, reliable, and user-friendly system. This system will not only reduce the risk of accidents caused by underage drivers but also provide a proactive tool for law enforcement and stakeholders, contributing to safer roads and communities.

4.22 Wireframes

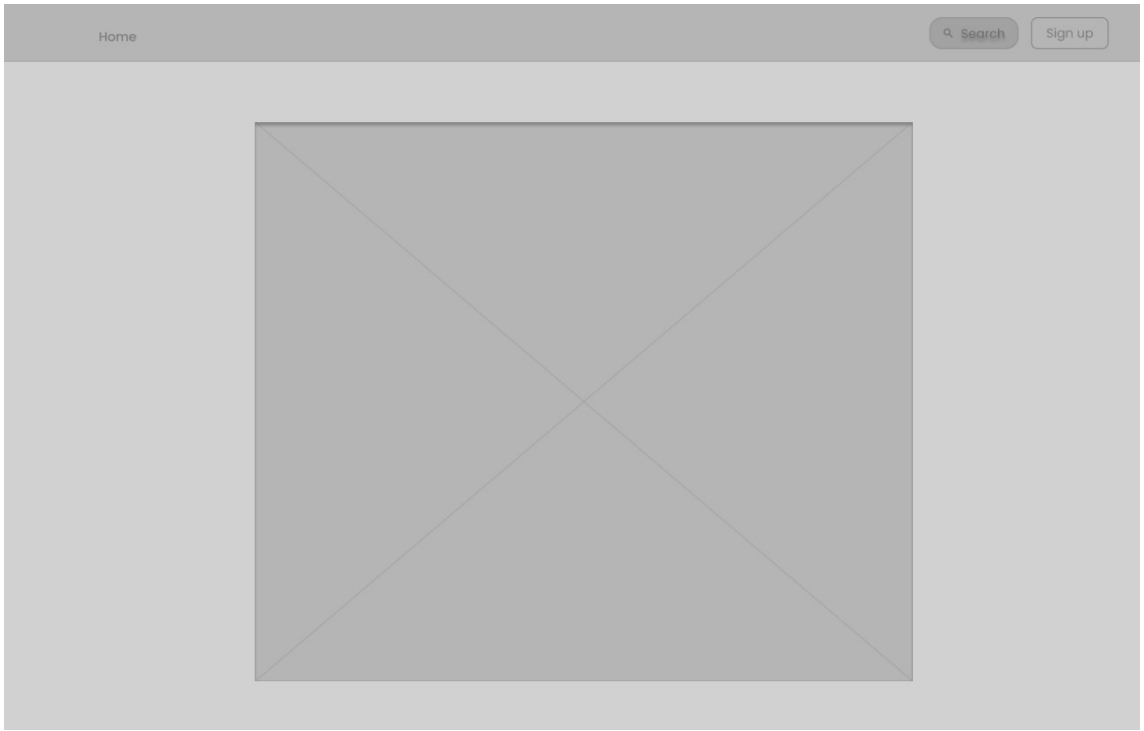


Figure 4.3: Wireframe Diagram

4.23 UI Screens

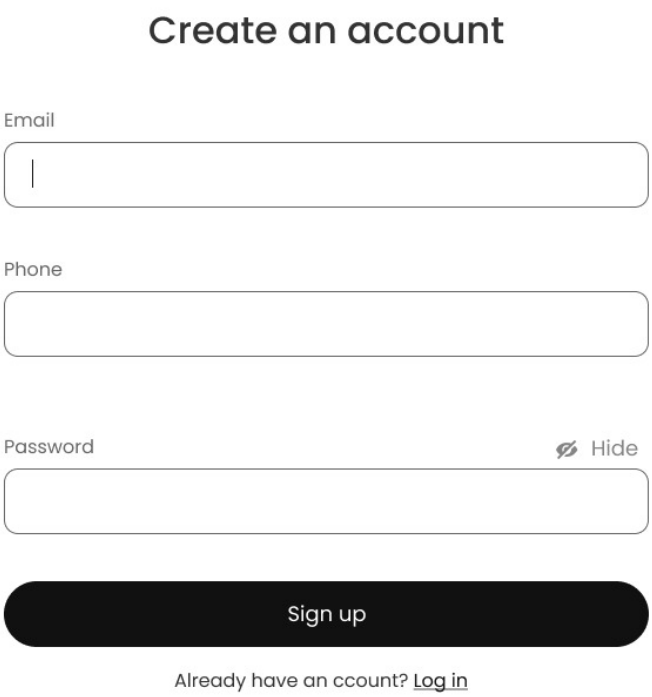


Figure 4.4: UI Screen 1

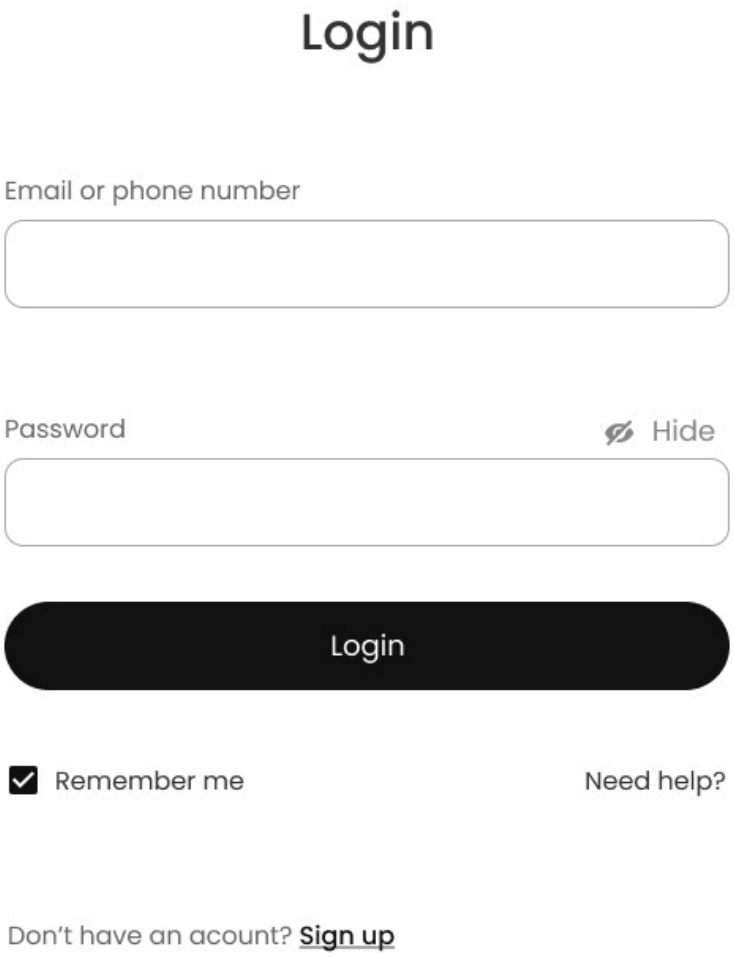


Figure 4.5: UI Screen 2

4.24 Updated And Finalized UI Screens

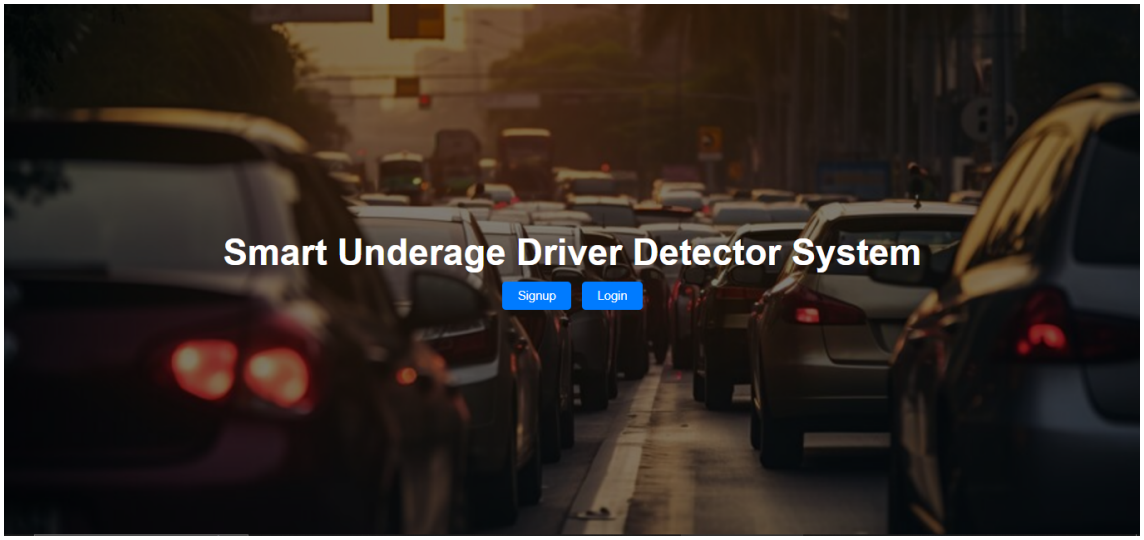


Figure 4.6: UI Screen 1

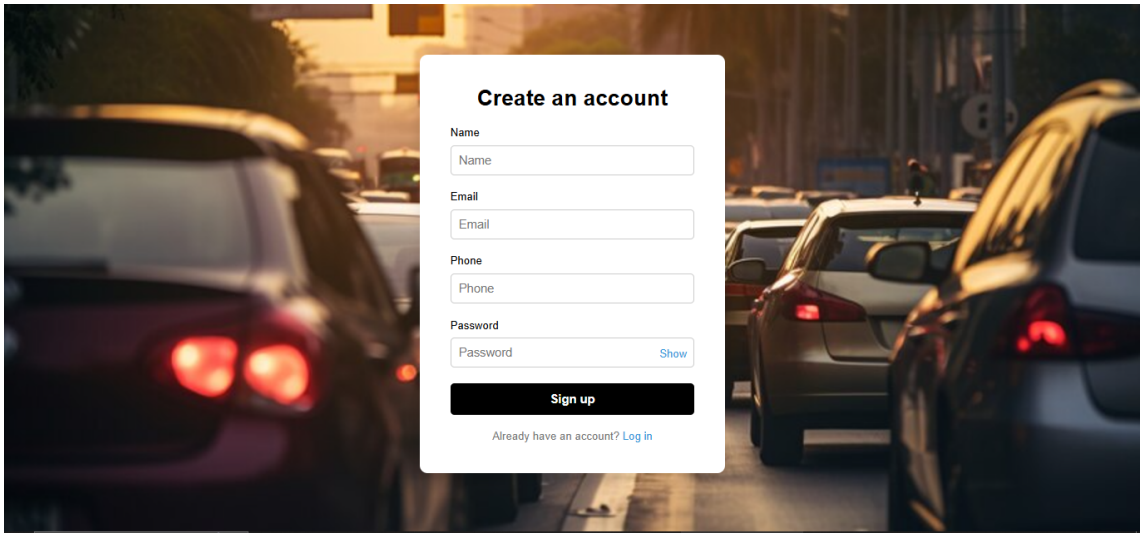


Figure 4.7: UI Screen 2

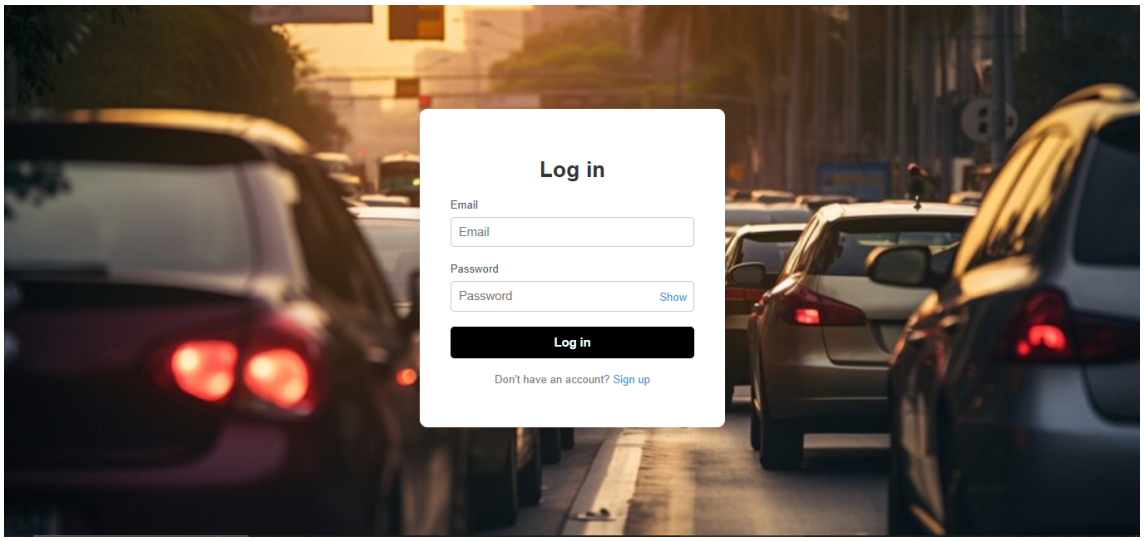


Figure 4.8: UI Screen 3

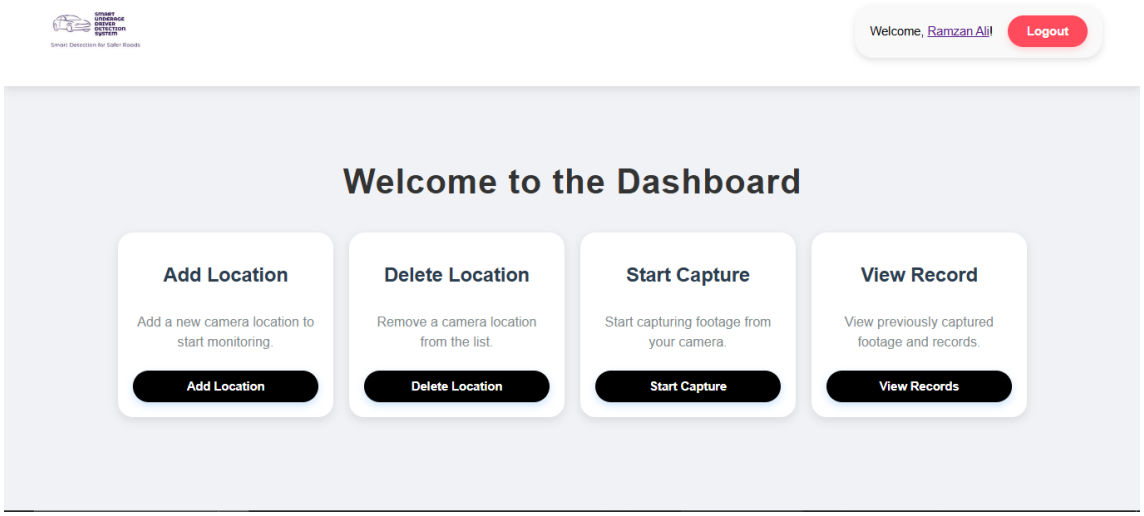


Figure 4.9: UI Screen 4

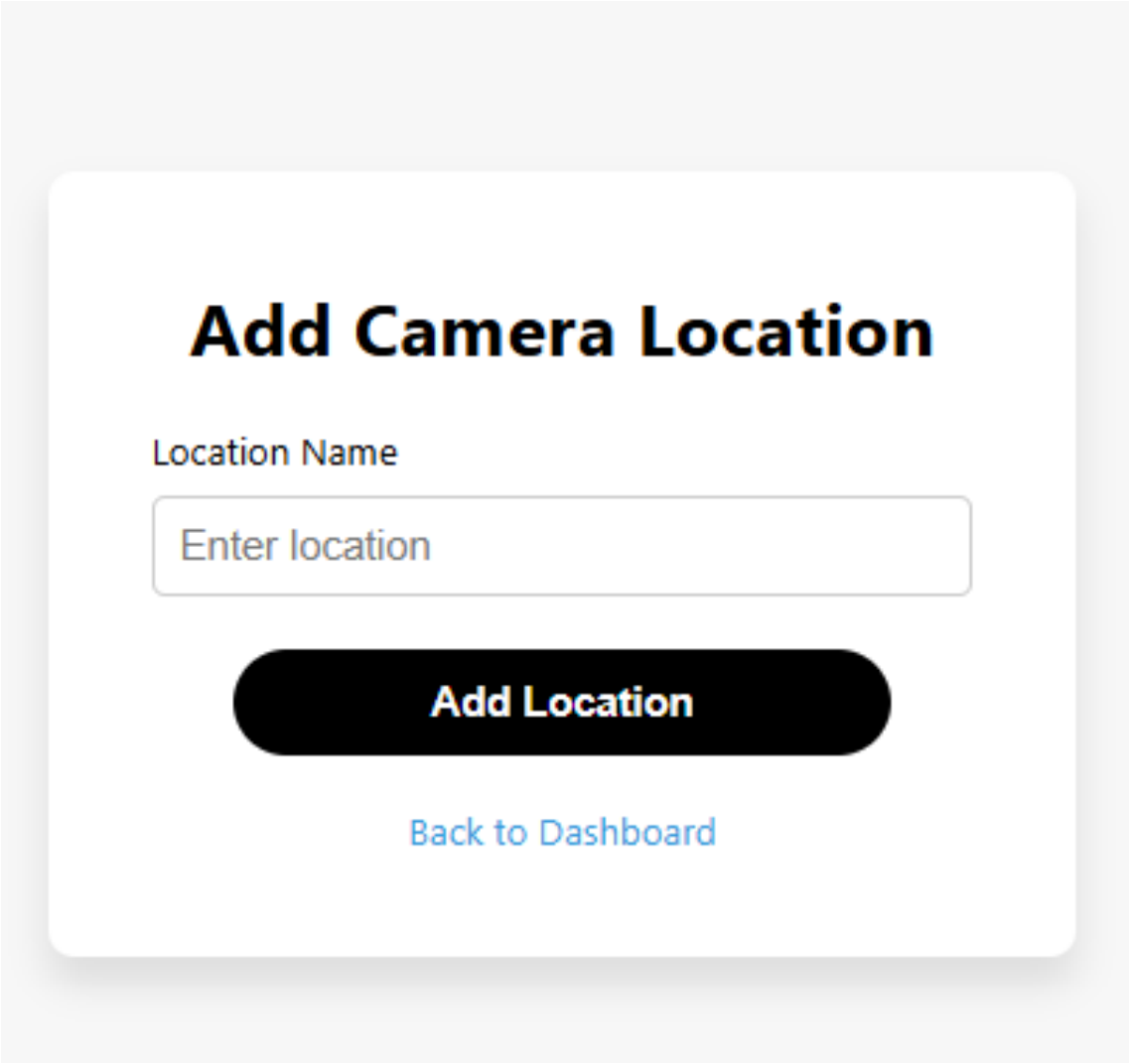


Figure 4.10: UI Screen 5

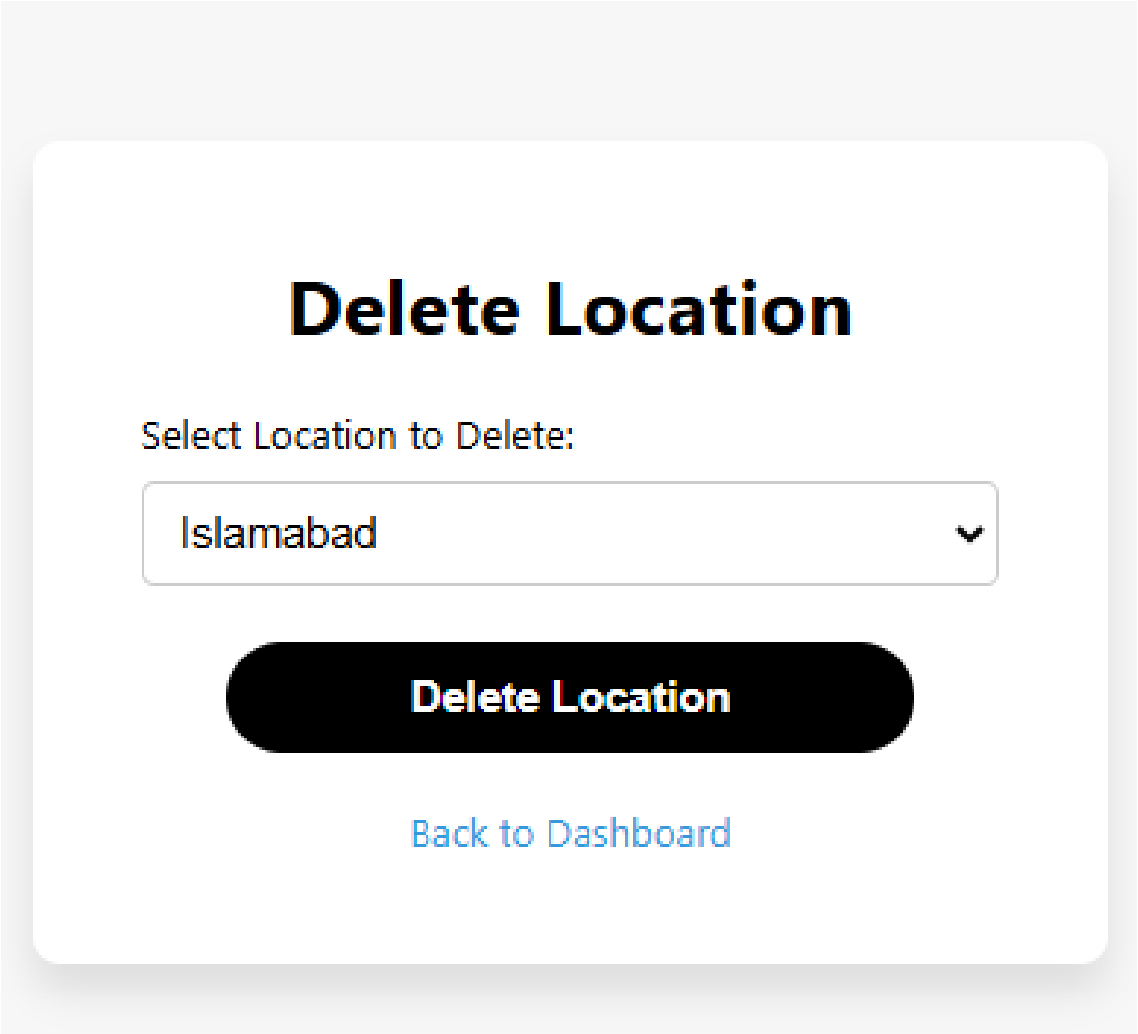


Figure 4.11: UI Screen 6

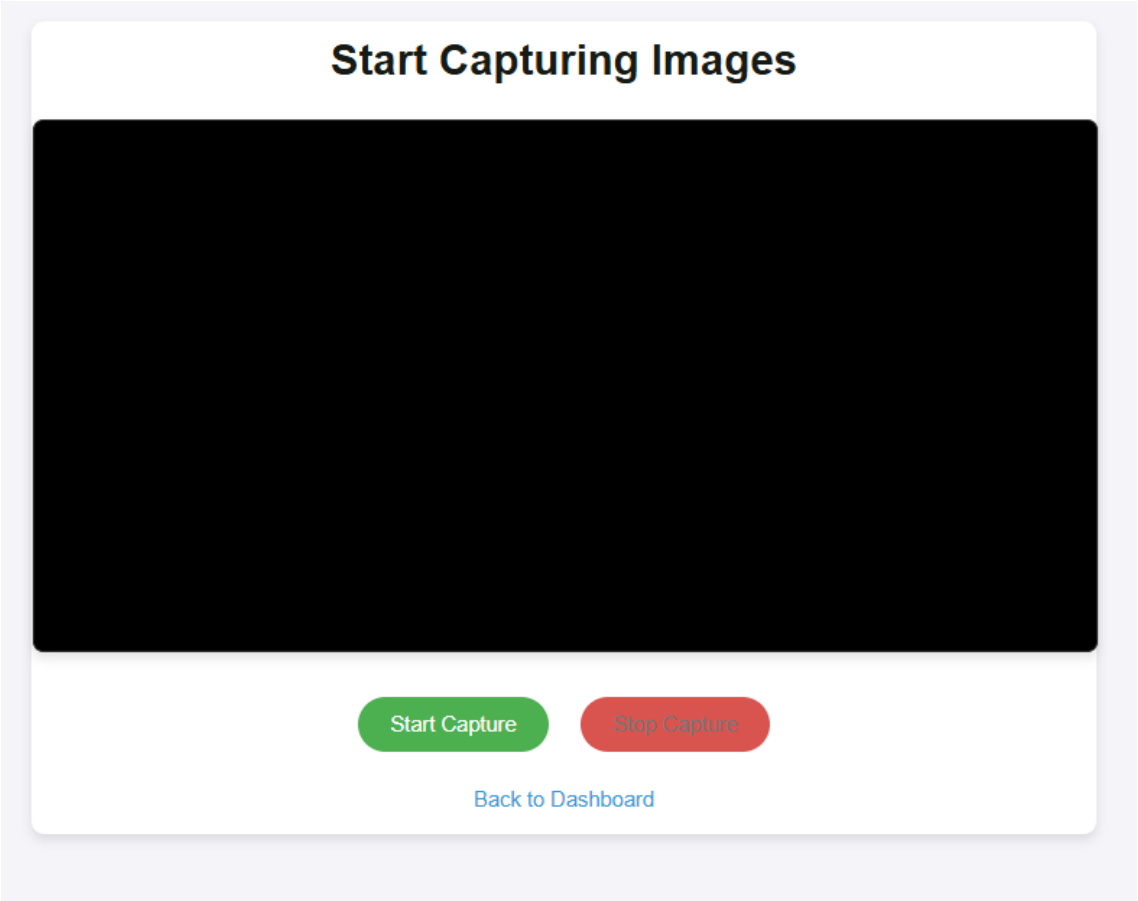


Figure 4.12: UI Screen 7

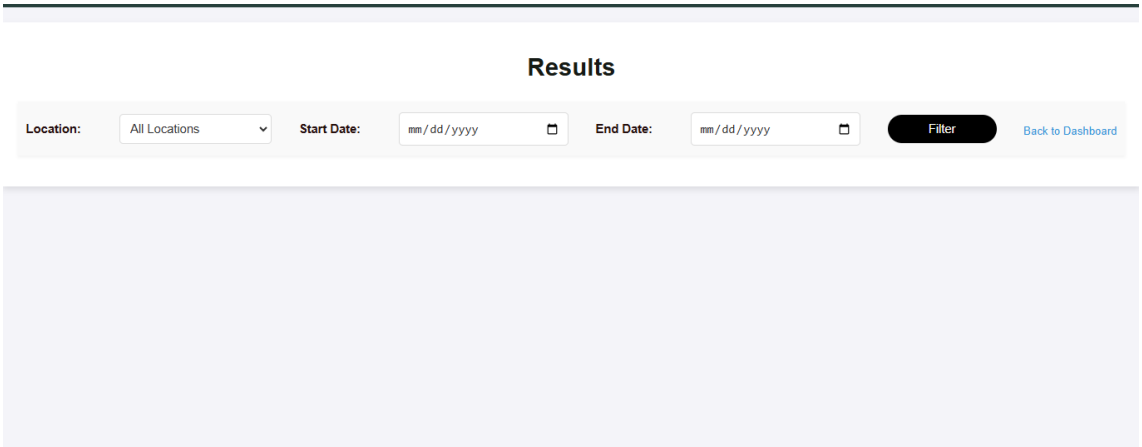


Figure 4.13: UI Screen 8

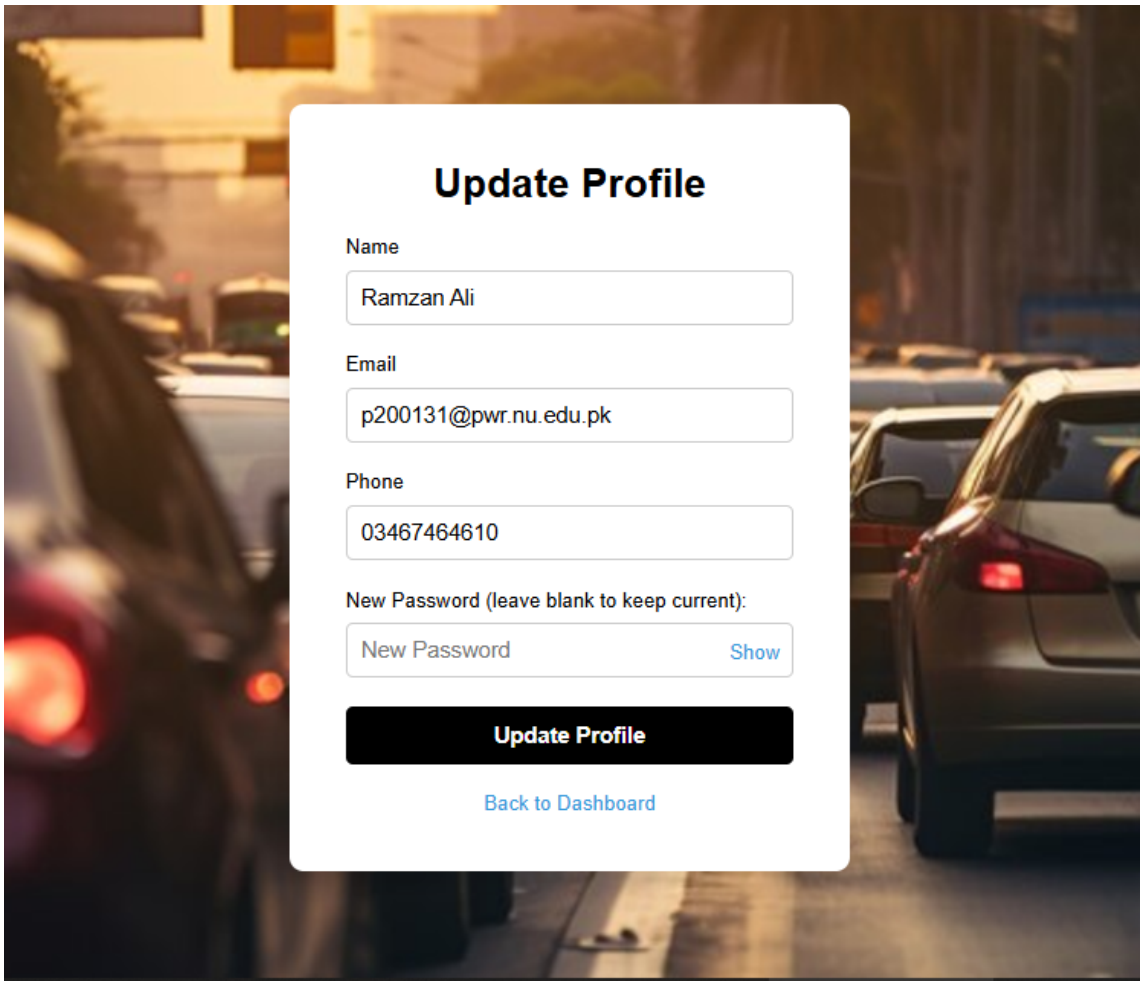


Figure 4.14: UI Screen 9

Chapter 5

Iteration Plan

This chapter outlines the project's iteration plan, describing how the project will progress to fulfill all its requirements. It provides a structured approach to guide the development of project modules. This chapter discusses the execution plan for the project, divided into phases, with a primary focus on the Midterm FYP 1 phase.

5.1 Midterm FYP 1

The Midterm FYP 1 phase marks the initial stage of project development. During this phase, the project team will focus on:

- Identifying project requirements and objectives.
- Defining the scope of the project.
- Setting up the development environment.
- Conducting preliminary research and feasibility studies.
- Creating an initial project plan, including timelines and milestones.
- Initiating the development of essential project modules.
- Conducting regular progress assessments and making necessary adjustments.

This phase serves as a foundation for the subsequent stages of the project and lays the groundwork for achieving the project's goals and deliverables. The following chapters will delve into the details of each phase as the project progresses.

5.2 Final FYP 1

5.2.1 5.2.1 Completed Activities

- **Module Development:** We successfully developed 40% of the project's core modules, which are critical functionalities driving the system. This represents substantial progress since the Midterm FYP 1, where the focus was on preliminary planning and development initiation.
- **User Interface Design and Development:** We have completed the design and development of a functional user interface (UI) for our project. This UI provides a user-friendly and intuitive interface for interacting with the system and accessing its functionalities.
- **Model Integration Research:** We conducted extensive research and evaluation of various models suitable for integrating into our project. This included exploring different model architectures, performance metrics, and compatibility with our project's requirements.

5.2.2 Challenges Faced and Solutions Implemented

- **Module Development Delays:** We encountered some unforeseen delays in the development of certain modules due to unforeseen technical complexities. To overcome these challenges, we implemented the following strategies:
  - Restructuring the development plan to prioritize critical modules.
  - Utilizing open-source libraries and frameworks to expedite development.
  - Collaborating with mentors and peers for guidance and technical assistance.

- **UI Design Optimization:** The initial UI design required improvements in terms of usability and aesthetics. To address this issue, we conducted user testing sessions and incorporated feedback to refine the UI for a better user experience.
- **Model Selection and Integration:** Selecting the best model for integration proved to be difficult because there were numerous options to consider. We implemented a systematic evaluation process involving performance benchmarking and compatibility assessments to identify the most suitable model for our project's specific needs.

5.2.3 Next Steps and Future Plans

Building upon the progress achieved in Final FYP 1, we are excited to move forward with the following next steps:

- Complete the development of all remaining modules: We will continue focusing on developing and refining the core modules of the project to ensure complete functionality and achieve our desired objectives.
- Integrate the chosen model: We will integrate the selected model into the system and conduct extensive testing to evaluate its performance and effectiveness.
- Refine and enhance the UI: We will continue to refine the UI based on user feedback and usability testing results to provide an optimal user experience.

5.2.4 Midterm FYP 2

In Midterm FYP 2, the project transitioned from the initial development phase to a phase focused on refinement, testing, and evaluation. The key activities undertaken during this phase include:

1. **Dataset Collection:** We focused on collecting a diverse set of labeled image data that includes both underage and non-underage drivers. This dataset consists of images from different angles, distances, ages, lighting conditions, and varying times

of day and night. **Preprocessing of Dataset:** The collected data was preprocessed to enhance model performance. This involved tasks such as face detection using YOLOv3, image normalization, resizing, and augmentation techniques like rotation and flipping to ensure that the model could generalize well across different situations.

2. **Fine-tuning the Model:** Building upon the foundational work from FYP-1, the team dedicated efforts to fine-tune the image recognition model. This involved optimizing model parameters, adjusting algorithms, and enhancing the model's accuracy to distinguish between underage and non-underage drivers effectively.

3. **Completion of the Website:** The essential project modules that were initiated in FYP-1 were completed during this phase. This included coding, debugging, and integrating the various system components to ensure that the application met the project's objectives and requirements.

4. **Testing the Website:** Rigorous testing procedures were conducted to evaluate the website's performance and reliability. This included unit testing to assess individual components, integration testing to validate interactions between modules, and system testing to verify the overall functionality.

5. **Consent Form Signed by Volunteers:** In compliance with ethical guidelines, consent forms were collected from all volunteers participating in the data collection phase. These forms ensured that volunteers were fully informed about the purpose of the project and gave their consent for their data to be used in the development of the smart underage driver detection system.

6. **Progress Assessment and Adjustment:** Throughout Midterm FYP 2, regular progress assessments were conducted to track milestones and ensure alignment with the project plan. Based on these assessments, necessary adjustments were made to address any deviations from the original timeline or project objectives.

Overall, Midterm FYP 2 served as a critical phase in the project's development, focusing on dataset preprocessing, model fine-tuning, testing, and optimization to prepare the system for its final implementation and deployment.

5.2.5 Final FYP 2

In the Final FYP 2 phase, our project reached its culmination with the completion and refinement of our product, which involved transitioning to a website built using Flask. Here's an overview of the key activities accomplished during this phase:

1. **Website Development:** Building upon the groundwork laid in the previous phases, the team successfully developed the website using Flask. This platform allowed for better accessibility and usability across different web browsers.

2. **Code Quality Improvement:** To ensure high-quality code throughout the development process, we implemented various Python tools, including Flake8, Pylint, and Black. Flake8 was used to enforce PEP 8 style guidelines and catch syntax errors, while Pylint provided comprehensive static code analysis to identify potential bugs, code smells, and areas for improvement. Additionally, Black was utilized to automatically format code, ensuring consistency and readability. These tools helped maintain a clean and efficient codebase, facilitating smoother collaboration and reducing technical debt.

3. **Optimization for Web Environment:** Significant efforts were made to optimize the website for different browsers and devices. This included refining user interface elements, ensuring responsive design, and optimizing performance for varying internet speeds.

4. **Deployment Preparation:** As the project neared completion, preparations were made for the deployment of the website. This involved finalizing documentation, setting up hosting infrastructure, and ensuring all necessary resources were in place for a smooth launch.

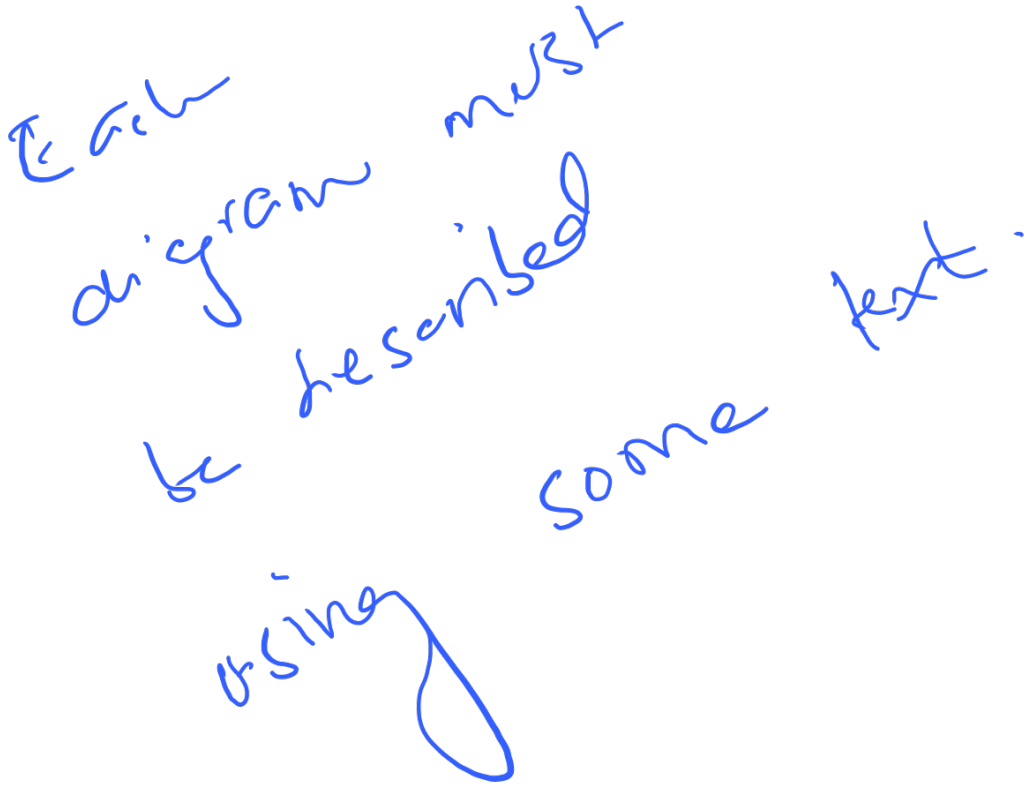
5. **Reflection and Documentation:** The team engaged in reflection sessions to evaluate the project's journey, lessons learned, and areas for future development. Comprehensive documentation was prepared to capture the project's evolution, technical specifications, and user guidelines for future reference.

The Final FYP 2 phase marked the culmination of months of dedicated effort, resulting in the successful development and refinement of our website. This phase showcased our technical abilities and demonstrated our adaptability in a dynamic project environment.

Chapter 6

Iteration 1

This iteration was focused on identifying project requirements and defining project scope. We conducted research, did feasibility studies, and also set up the development environment. This iteration was devising a plan, creating timelines, and defining the milestones.



6.1 ERD Diagram

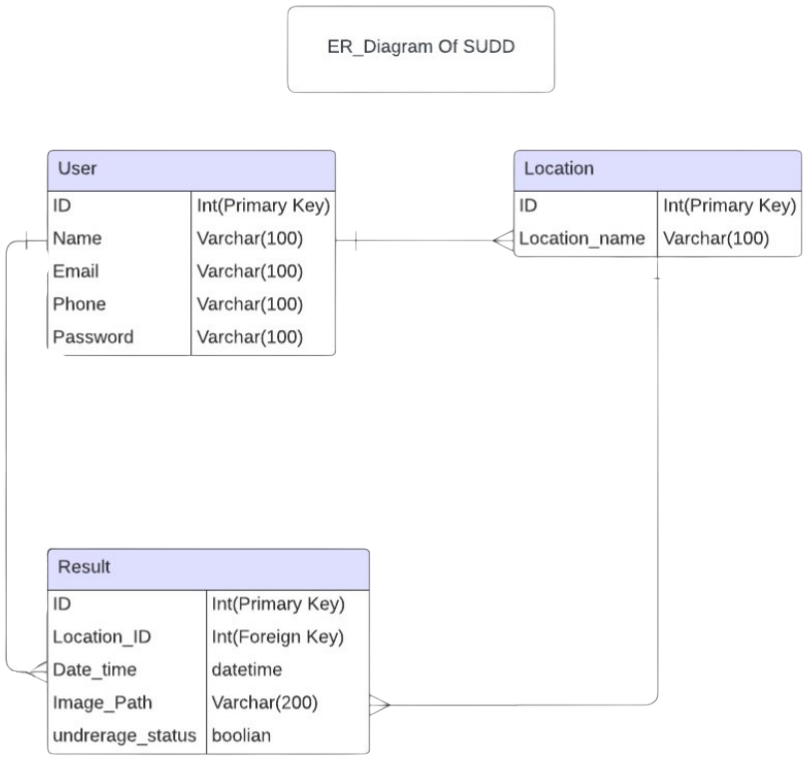


Figure 6.1: Entity-Relationship Diagram

6.2 Data Flow Diagram (DFD)

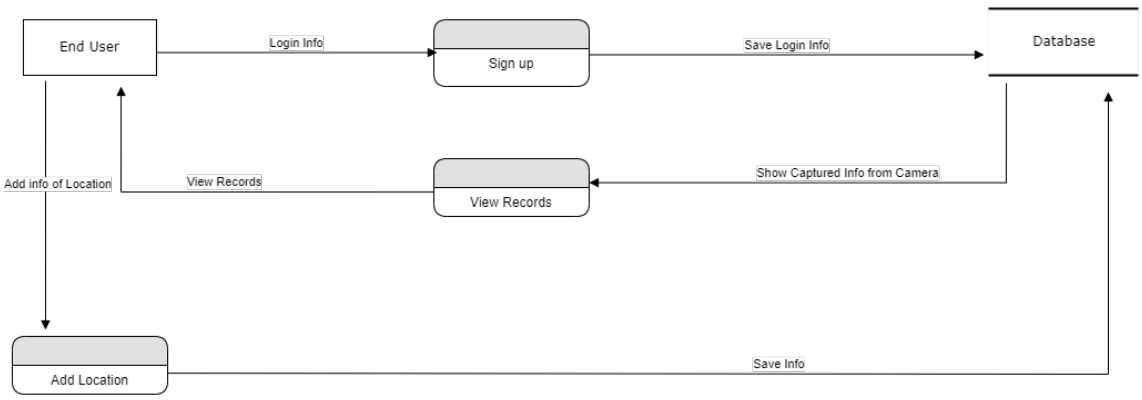


Figure 6.2: Data Flow Diagram (Level 1)

6.3 Architecture Diagram

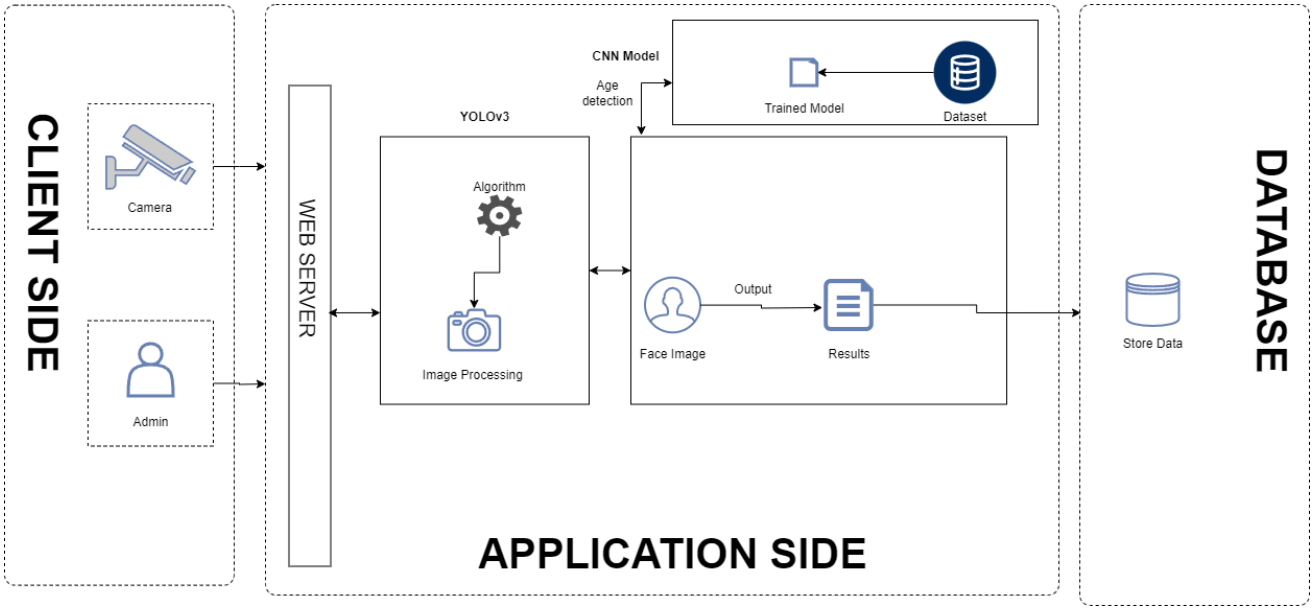


Figure 6.3: Architecture Diagram

6.4 Activity Diagram

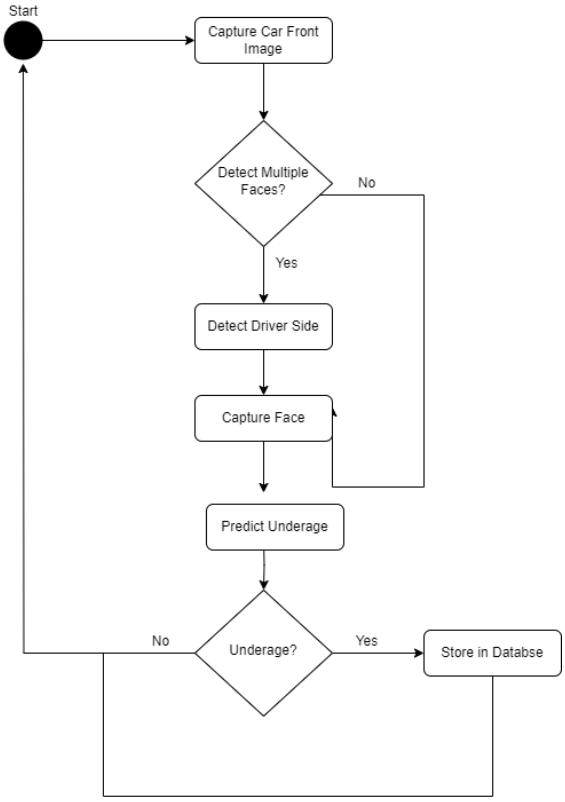


Figure 6.4: Activity Diagram

Chapter 7

Iteration 2 - Data set collection

7.1 Introduction

Concern about the increase in underage driving is spreading around the world, since it poses major threats to road safety and raises the possibility of collisions. Real-time detection of underage drivers is difficult, though, particularly in the absence of appropriate datasets that replicate the real-world circumstances needed for model training. By gathering a unique, real-time dataset, our project, Midterm FYP 2, seeks to close this gap and pave the way for the creation of machine learning-based underage driver identification systems. A systematic and exacting data gathering procedure has been put in place because datasets pertaining to the identification of underage drivers are not available online. Participants of various ages, including adults and minors, were chosen and photographed in a controlled setting to replicate real-world situations in which drivers are photographed via car windscreens under various structural design. The complete procedure, including imaging configurations, data gathering procedures, and analysis, is covered in depth in this study. In order to train models that can differentiate between underage and adult drivers based on facial traits, posture, and vehicle interactions, the main objective is to provide a realistic, high-quality dataset.

7.2 Problem Statement

Even though underage driving is a risky habit that can lead to deadly collisions, it is still a challenging problem to solve. There are currently no reliable methods for detecting underage drivers in traffic systems. Using machine learning algorithms that can determine the age groups of drivers based on their look, an efficient solution must be built on visual cues. The lack of a representative dataset is one of the primary obstacles to creating such models. It is difficult to train models with high accuracy if real-world data that reflects the range of circumstances in which underage drivers could be recognized is not available. By producing a dataset that mimics actual settings, our effort closes this gap and aids in the creation of detection algorithms that are more precise.

7.3 Objectives

This project's main goal is to create an extensive, real-time dataset that will subsequently be utilized to build machine learning models that can identify drivers who are underage. The particular goals are:

- To gather pictures of participants in a variety of age groups—adults and children—that mimic actual driving situations.
- To guarantee dataset variety by photographing the participants in various lighting situations, distances, and perspectives.
- To guarantee that the data is arranged and annotated in a way that is appropriate for machine learning applications.
- To provide the framework for a detection system that uses visual information to automatically identify drivers who are underage.

7.4 Data Collection Protocols

7.4.1 Participants And Criteria

The participants were split into two primary groups: adults (over 18) and minors (under 18). Participants were chosen based on the following criteria, which guaranteed a balanced distribution across age groups:

- Ages underage: 8, 10, 12, 14, and 16
- Adults: 20, 25, 30, 35, and 40.

To guarantee accuracy and traceability, each participant's name, father's name, birthdate, and photo were documented.

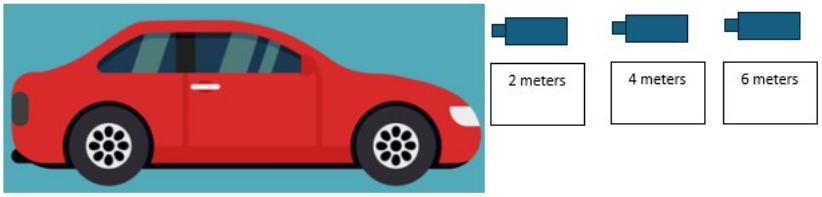


Figure 7.1: Car Distances

7.4.2 Imaging Conditions

In order to accurately simulate real-world conditions for detecting underage drivers, participants were positioned inside a vehicle, replicating a scenario in which a traffic camera would capture an image of the driver through the car's windshield. This setup was designed to mimic the typical imaging conditions that would occur in a traffic surveillance context, where cameras are often mounted outside vehicles or along roads to monitor traffic.

To simulate the environment as closely as possible, images were captured from outside the vehicle, with a focus on ensuring the windscreen was clearly visible in each shot. This provided a realistic representation of the type of images that might be captured by traffic

cameras, which often have to contend with factors like reflections, glare, and distortion caused by the car's glass. Care was taken to ensure that these conditions were properly represented to assess the system's ability to accurately detect facial features under varied lighting and environmental conditions.

In addition to these physical considerations, the images were taken under different lighting conditions, such as daylight, dusk, and nighttime, to assess how well the system performed in diverse visual environments. Special attention was also given to capturing images from varying angles and distances, as traffic cameras may not always capture drivers head-on. This variability was incorporated into the imaging process to better train the detection system and ensure its robustness in handling real-world scenarios.

By simulating these realistic imaging conditions, the system was tested for its ability to handle the challenges typically faced by traffic cameras, such as windscreen reflections, obstructions, and variable lighting, while maintaining accuracy in detecting underage drivers based on facial recognition and other relevant features.

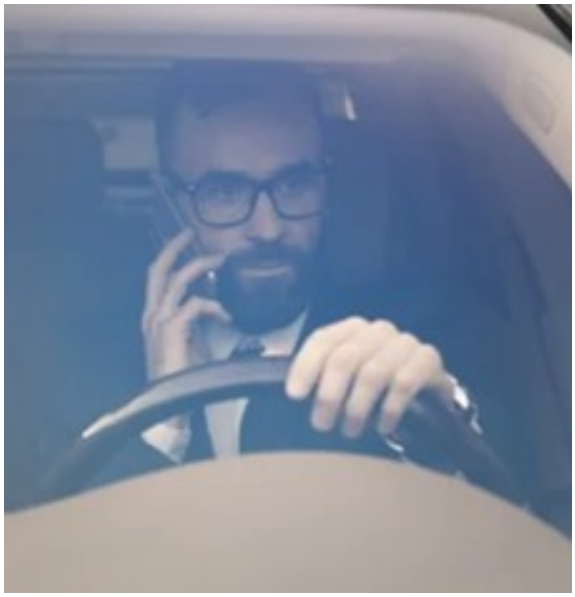


Figure 7.2: Man in Car

7.4.3 Camera Angles and Distances

Images were taken from three distinct angles: -30°, 0°, and 30° relative to the car's front. To capture a realistic variance in perspective, the images were also taken from three different distances: 2 meters, 4 meters, and 6 meters.

This extensive imaging protocol ensures a robust dataset that covers a wide range of conditions, essential for training machine learning models.

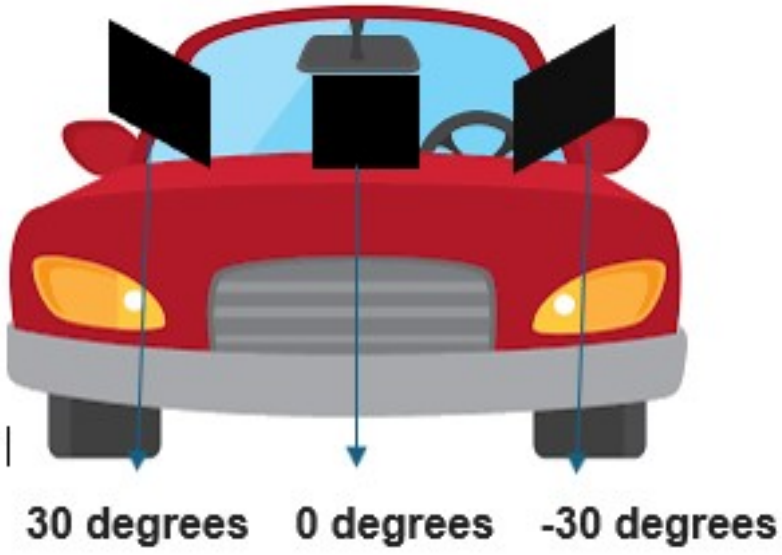


Figure 7.3: Front Angles

7.5 Methodology

Three primary steps comprised the data gathering process: imaging, data storage, and participant selection.

1. Participant Selection: An equitable proportion of adult and juvenile subjects was guaranteed by carefully choosing volunteers based on age groups.

2. Imaging: The automobile was parked outside, and the participants were sitting in the driver's seat. High-resolution cameras positioned at specified distances from the vehicle (2, 4 and 6 meters) were used to take the pictures. For every angle (-30°, 0°, and 30°) and

lighting condition (morning and evening), the procedure was repeated.

3. Data Storage: Every picture was named by participant age, angle, distance, and time of day and saved in the Portable Network Graphics (PNG) format. After that, the photos were tagged for use in machine learning tasks in the future.

Participant	**Morning**			**Evening**			**Total**
	-30	0	30	-30	0	30	
8 years	5	5	5	5	5	5	30
10 years	5	5	5	5	5	5	30
12 years	5	5	5	5	5	5	30
14 years	5	5	5	5	5	5	30
16 years	5	5	5	5	5	5	30
20 years	5	5	5	5	5	5	30
25 years	5	5	5	5	5	5	30
30 years	5	5	5	5	5	5	30
35 years	5	5	5	5	5	5	30
40 years	5	5	5	5	5	5	30

Table 7.1: DataSet Matrix

7.6 Results

The dataset consists of 1000 photos that accurately reflect real driving conditions. This comprehensive and diverse dataset captures a wide range of scenarios, including variations in lighting, camera angles, and distances. The variety ensures robust coverage of different environmental conditions that could be encountered in real-world traffic monitoring. Preliminary data analysis reveals that underage drivers and adults exhibit distinguishable features based on face shape, posture, and overall appearance, as observed through a car windshield.

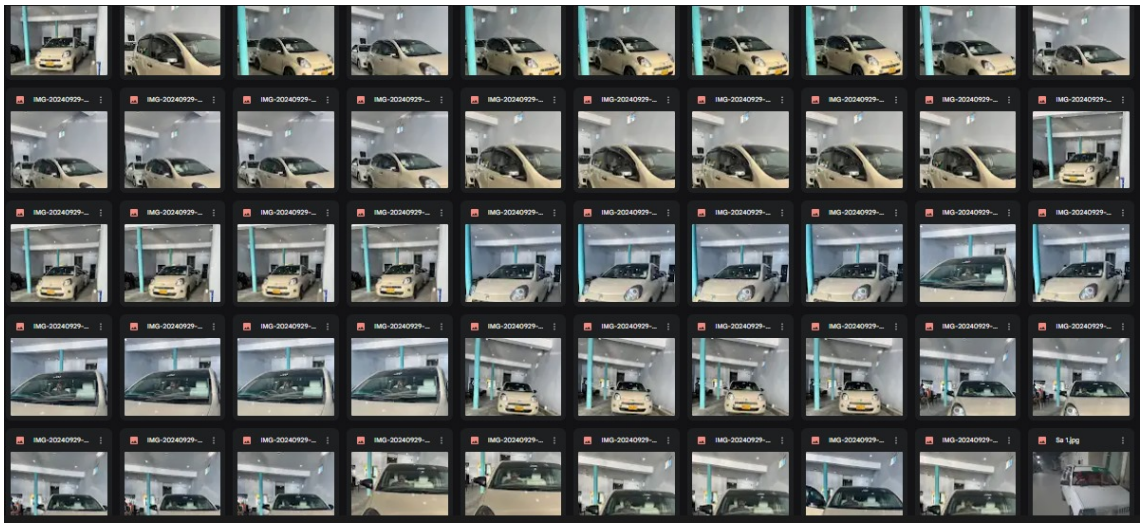


Figure 7.4: Illustration of the dataset collection process during different times of day. Images were captured under varying lighting conditions to reflect daytime scenarios, ensuring diverse data for further analysis.

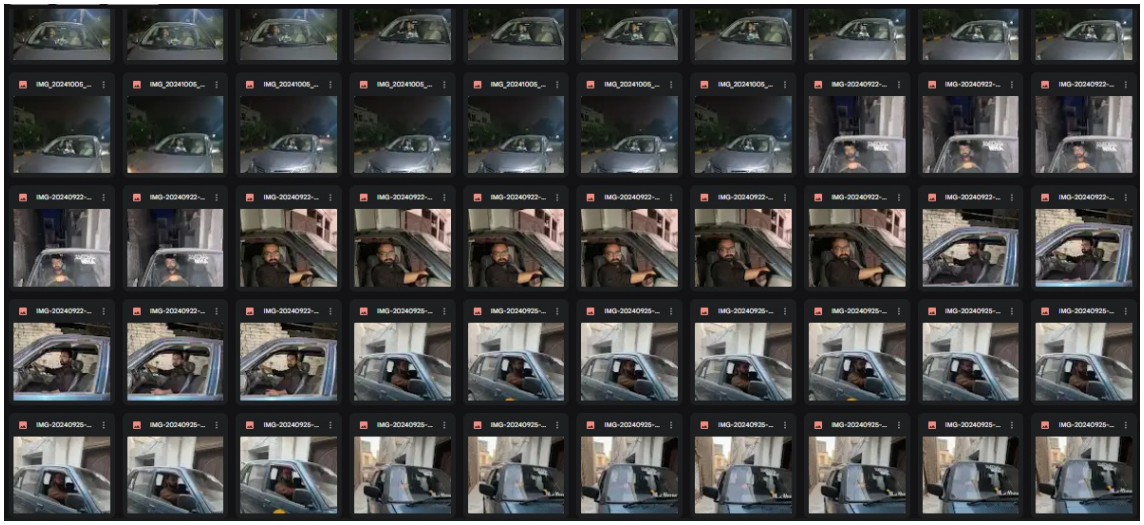


Figure 7.5: Example of evening time dataset collection. This image demonstrates the conditions under which data was captured at night.

7.7 Challenges

The method of gathering data presented a number of difficulties:

- **Lighting Variability:** It was challenging to take consistent photos in various lighting scenarios (morning vs. evening), particularly when it came to preserving con-

stant exposure settings.

- **Cooperation from Participants:** Some participants, especially younger ones, had trouble staying still for several picture takes, which resulted in minor differences in image quality.
- **Environmental Factors:** Weather-related factors, including sun glare or reflections on a car's windshield, occasionally impacted the image quality.

7.8 Conclusion And Future Work

The Midterm FYP 2 project effectively filled the data gap for this crucial issue by creating a bespoke dataset for underage driving detection. In the future, machine learning algorithms that attempt to automatically identify drivers who are underage can be trained using this dataset. Future research will concentrate on improving the procedure for gathering data and incorporating this information into a more comprehensive detection system.

Chapter 8

Iteration 3 - Driver Side Detection

Our project initially goal is to detect underage drivers; however, accurately identifying the driver became challenging in scenarios where multiple passengers were present in the vehicle. Further analysis revealed that YOLOv3 faced limitations in effectively distinguishing between multiple faces within a single image, contributing to the complexity of accurate driver detection.

8.1 Data Labeling

Labeling is the first step in preparing the dataset for supervised learning. It involves annotating images with critical details like driver positions and relevant features, allowing the model to learn patterns and recognize drivers in vehicles effectively.

To improve performance, the dataset includes a variety of images depicting drivers in different scenarios, poses, and lighting conditions. This diversity helps the model generalize better and perform reliably under varying camera angles and illumination.

8.2 Driver Detection Using YOLOv3

To enhance system precision, we used the YOLOv3 object detection model for detecting drivers. YOLOv3 is well-suited for real-time applications due to its balance of speed and

accuracy, making it ideal for detecting drivers efficiently.

8.2.1 Why Driver Detection?

- **YOLOv3 Face Detection**: YOLOv3 operates by detecting faces within an image and providing the corresponding x and y coordinates based on the image size. This allows the model to identify faces with precise location information.
- **Choosing the Leftmost Face**: When multiple faces are detected, such as in the case of passengers, the model focuses on the leftmost face based on the x-coordinate. This approach assumes that the driver is typically seated in the leftmost position, making it a reliable method for identifying the driver.

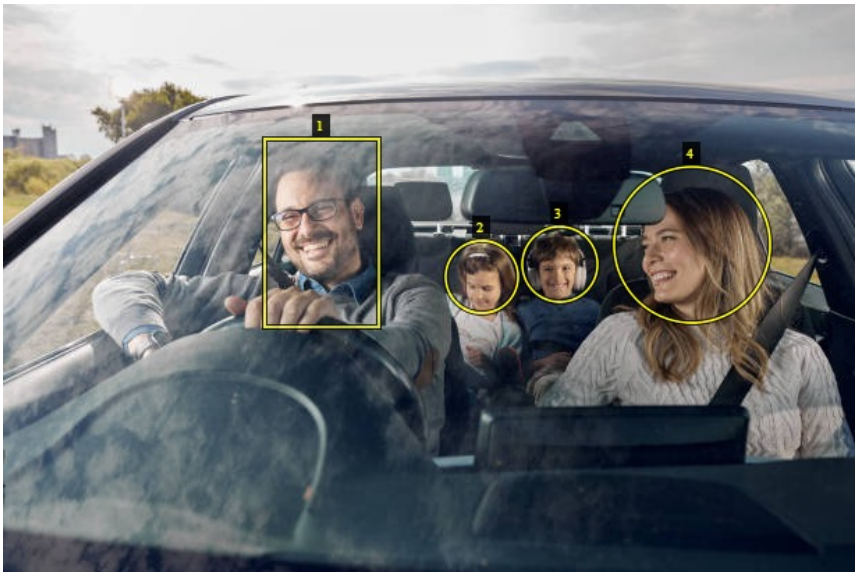


Figure 8.1: Multiple face detections are shown, with the leftmost face selected as the driver.

Chapter 9

Iteration 4 - Model Training

In the next phase of our Smart Underage Driver Detection system We will improve the model's performance by concentrating on binary classification, namely differentiating between drivers who are and are not underage. The efficiency of our system depends on this classification activity because it has a direct impact on identifying drivers who might not be authorized to drive a vehicle.

9.1 Model Architecture

To achieve this classification, we will employ a combination of Convolutional Neural Networks (CNNs) and the YOLO (You Only Look Once) architecture. The YOLO model will initially be used for face detection, allowing us to focus on the driver's face for further analysis. The extracted face images will then be processed through a CNN to classify them into two categories: underage and non-underage.

9.1.1 Image For Binary Classification

The collected dataset was divided into two categories: "underage" and "not underage." Our project focuses specifically on detecting underage individuals, so we are not concerned with predicting other age groups. Additionally, our model uses a sigmoid activation function, which is well-suited for binary classification tasks like this one.

9.1.2 YOLOv3 for Face Extraction

YOLO model will be fine-tuned to detect faces in images captured during various driving scenarios. This model operates in real-time, making it ideal for our application, where quick detection is essential for immediate responses.

We used YOLOv3 for face extraction due to resource limitations during model training. Since we couldn't use large images directly, face extraction became necessary to ensure that the model could operate efficiently with the available resources.

9.2 Data Preparation

To train our model effectively, the following data preparation steps will be undertaken:

9.2.1 Splitting the Dataset

The dataset will be divided into training, validation, and test sets. A common split would be 70% for training, 15% for validation, and 15% for testing to ensure the model is evaluated on unseen data.

9.2.2 Training the Model

The model will be trained over multiple epochs, monitoring performance on the validation set to prevent overfitting. Early stopping callbacks may be employed to halt training when validation performance stops improving.

9.2.3 Evaluating Model Performance

After training, the model's performance will be evaluated using the test set, measuring metrics such as accuracy, precision, recall, and F1 score. Confusion matrices will be generated to visualize performance in distinguishing between underage and non-underage

classifications.

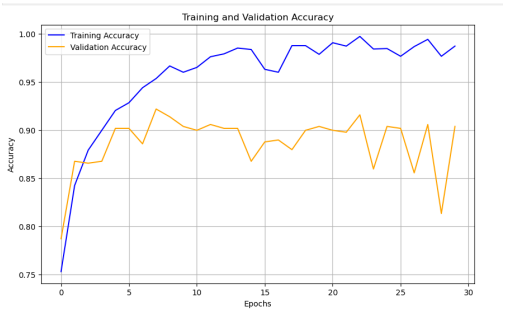


Figure 9.1: Model accuracy

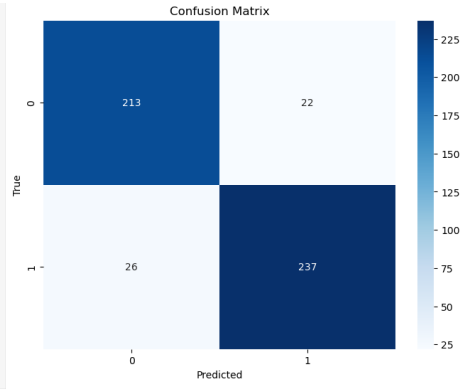


Figure 9.2: F1 Score of the model

Chapter 10

Evaluations And Results

10.1 Iteration 5

This was the final iteration. In this phase, we transitioned from a basic web interface built with flask to a fully functional web application for real-time monitoring and enforcement. Key activities included optimizing the system for better performance, improving detection accuracy, conducting extensive testing, and refining the system's documentation to reflect the new features and changes implemented during this phase.

10.1.1 Code Comments

Comments are used to explain complex sections of the code to ensure readability and facilitate future maintenance. A comprehensive and consistent approach was applied to commenting throughout the system's code.

Guideline	Description
Use Clear and Concise Comments	Comments should be brief and focused, explaining the intent or purpose of the code block without stating the obvious.
Comment the Why, Not the What	Comments should explain why something is done a certain way or why a particular approach was chosen, not just restate what the code does.
Follow PEP 257 Docstring Conventions	Use docstrings to describe modules, functions, classes, and methods. Docstrings should be enclosed in triple quotes and follow the PEP 257 conventions.
Comment Formatting	Use proper formatting and indentation for comments to improve readability. Avoid long lines of comments that can be difficult to read.
Update and Maintain Comments	Keep comments up to date. When you modify code, ensure that associated comments are modified to reflect the changes.
Avoid Redundant Comments	Don't write comments that duplicate the code. Comments should provide insights that aren't immediately obvious from the code itself.
Use Inline Comments Sparingly	Inline comments should be used sparingly and explain complex or non-intuitive code segments. Overusing inline comments can clutter the code.
Avoid Excessive Comments	Well-written code should be self-explanatory, reducing the need for excessive comments.

Table 10.1: Guidelines for Effective Code Comments

10.1.2 Naming Conventions

Industry-standard naming conventions, descriptive and meaningful names for variables, functions, and classes were used for ensuring readability and better understanding of components.

Element	Convention
Variables, Functions, Methods	snake_case
Constants	Uppercase_snake_case
Classes	snake_case
Testing	snake_case

Table 10.2: Flask Naming Conventions

10.2 Code Quality and Testing

10.2.1 Static Analysis of Code

Static analysis tools, such as Flask's built-in debugging options and pytest's linting features, were used to analyze the codebase for potential issues. These tools helped identify code smells, bugs, inefficiencies, and ensured adherence to best practices, contributing to improved code reliability.

10.2.2 Code Refactoring

Periodic code refactoring was performed to improve code quality, eliminate redundancy, enhance performance, and simplify complex code. Flask's modular structure and pytest's ease of testing helped maintain clean, maintainable code.

10.2.3 Unit Testing and Quality Assurance

Unit tests were created using pytest to verify the functionality of individual components. The quality assurance procedures ensured that the software met requirements and worked as intended.

10.2.4 Version Control and Collaboration

Git was used for version control and collaboration, with proper branching strategies and versioning practices followed to ensure smooth teamwork and code consistency.

Chapter 11

User Manual

11.1 Introduction

This user manual provides a comprehensive guide to the key features and functionalities of the system, ensuring that users can effectively utilize its capabilities for real-time monitoring and enforcement. By following this guide, users will be equipped to operate the system, understand its processes, and contribute to minimizing road accidents and ensuring compliance with driving regulations.

11.2 Authentication

11.2.1 10.1.1 Signup Page:

Description: This section shows the signup process.

Action Steps:

1. Enter a unique username in the field.
2. Enter your email address in the field.
3. Enter a strong password and ensure you can remember it.

4. Click the signup button.

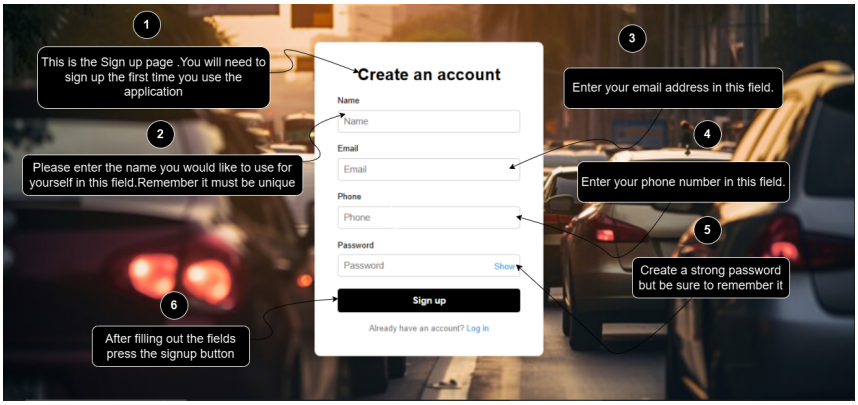


Figure 11.1: Sign Up Page

11.2.2 Login Page:

Description: This section shows the login process.

Action Steps:

1. Enter your username in the first field.

2. Input your password in the second field.

3. Click the login button to access your account.

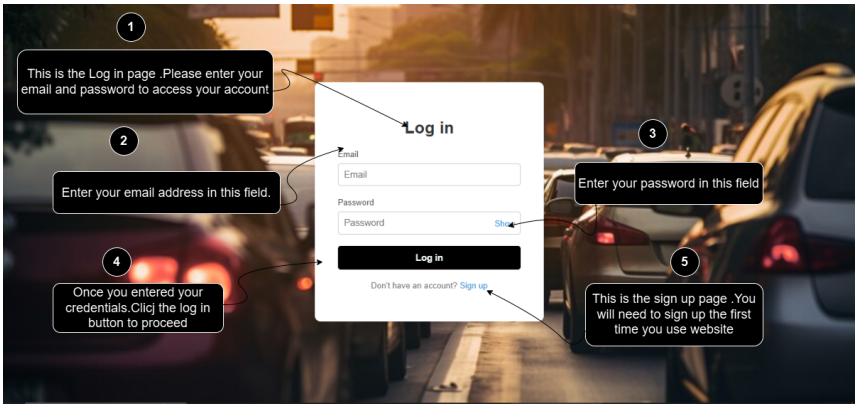


Figure 11.2: Login page

Website Dashboard

Welcome to the Dashboard

The dashboard allows you to manage monitoring locations, capture footage, and view records.

Features

1. Add Location

Description: Add a new camera location to start monitoring.
Steps:

1. Click the **"Add Location"** button.
2. Enter the details of the camera location.
3. Press "Add Location" button to start monitoring.

2. Delete Location

Description: Remove an existing camera location.
Steps:

1. Click the **"Delete Location"** button.
2. Select the camera location to be deleted from the list.
3. Press Delete Location button.

3. Start Capture

Description: Begin capturing footage from your camera.
Steps:

1. Click the **"Start Capture"** button.
2. Ensure the camera is connected and operational.
3. Monitor the live feed as the system captures footage.

4. View Records

Description: Access previously captured footage and records.

Steps:

1. Click the **"View Records"** button.
2. Browse through the list of saved records.
3. Enter specific date
4. Select a record to view detailed footage.

Logout

To securely exit the system, click the **"Logout"** button in the top-right corner.

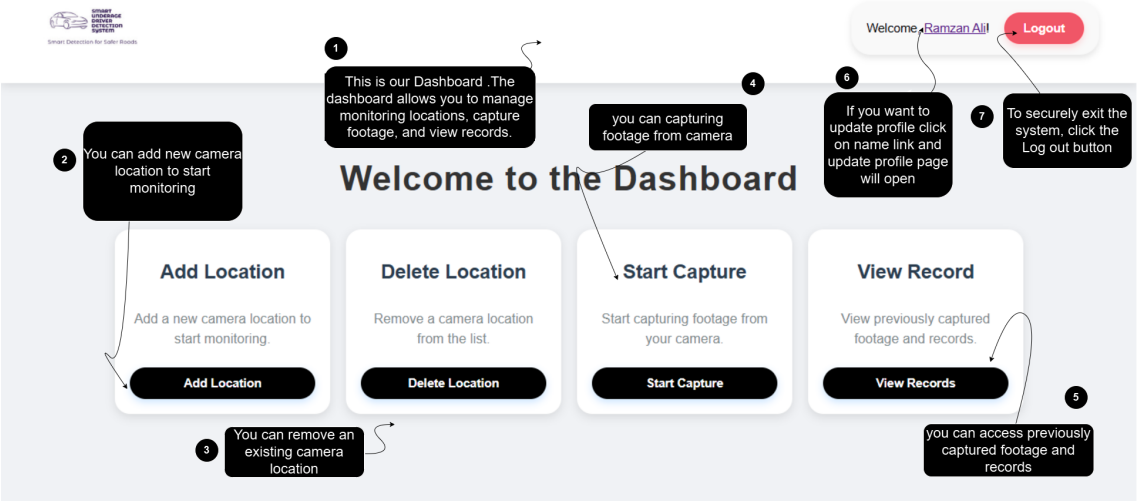


Figure 11.3: Dashboard Page

Update Profile Feature

Purpose

This page allows the user to update their personal details, such as name, email, phone number, and password.

Action Steps

1. **Name:** Edit the "Name" field with your new name.

2. **Email:** Update your email address in the "Email" field.

3. **Phone:** Enter your new phone number in the "Phone" field.

4. **Password:**

    - To change your password, type the new password in the "New Password" field.
    - Leave it blank to keep the current password.

5. **Submit:** Click the **Update Profile** button to save changes.

6. **Navigation:** To return to the main dashboard, click **Back to Dashboard**.

Notes

- Ensure all required fields are correctly filled before submitting.

- Password updates are optional.

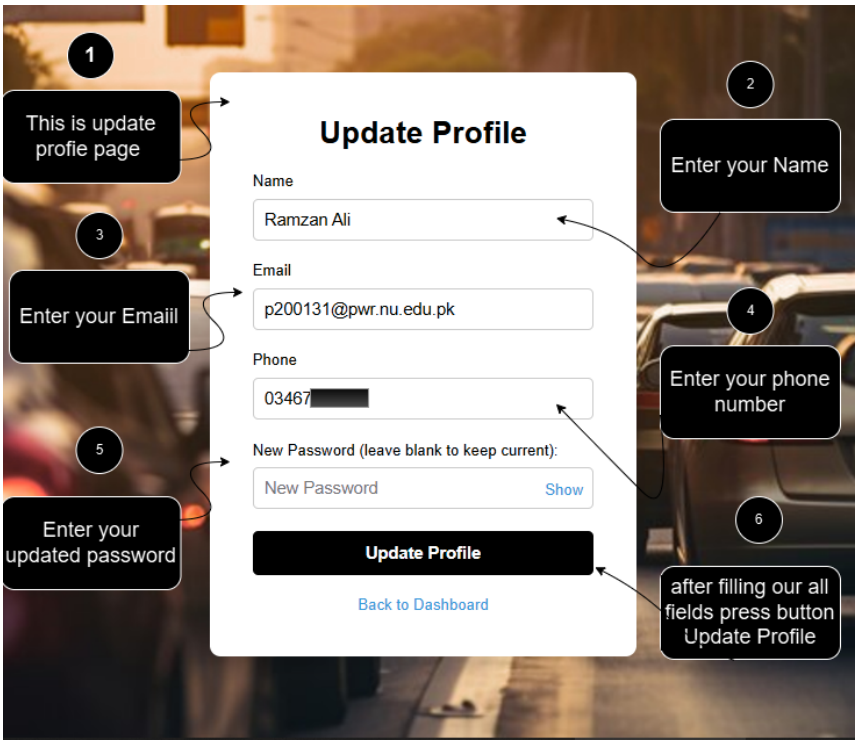


Figure 11.4: Update Profile Page

Chapter 12

Conclusions and Future Work

12.1 Conclusions

The Smart Underage Driver Detection System demonstrates an innovative approach to enhancing road safety by accurately identifying underage drivers in real-time. Utilizing advanced deep learning models, including convolutional neural networks (CNNs) and YOLO (You Only Look Once) for object detection, our system reliably processes images to analyze driver facial characteristics, accurately classifying individuals as either underage or legally eligible drivers. Through careful pre-processing, data labeling, and model training, the system achieves a high level of accuracy and efficiency, even under variable conditions, providing a robust solution for automated age verification.

Our project achieves a key milestone in binary classification (underage vs. non-underage) and presents significant potential for integration into live surveillance systems, facilitating proactive detection and immediate notification to authorities upon identifying an underage driver. The success of this system marks a promising advancement in the field of intelligent transportation systems, where automated driver verification can play a critical role in compliance with legal driving age requirements.

12.2 Future Work

We have identified key areas for future work to improve the **Smart Underage Driver Detection System** and enhance its effectiveness:

1. **Dataset Expansion:** Broadening the dataset to include more age groups, ethnicities, and diverse environmental conditions to enhance model generalizability.
2. **Advanced Classification Layers:** Implementing additional layers for more precise age detection, especially for regions with varying legal driving ages.
3. **Multimodal Data Integration:** Combining facial analysis with other technologies like voice recognition or motion detection for better age estimation.
4. **Edge Device Optimization:** Enabling real-time monitoring via portable or embedded vehicle systems for wider accessibility.
5. **Continuous Enhancement:** Ongoing model improvements and updates to maintain its relevance in real-world applications, contributing to road safety and regulatory enforcement.

Bibliography

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer, 2010.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[5] S. Chai, X. Zhang, X. Li, and Z. Yu, "Driver distraction detection methods: A literature review and framework," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1181–1186.

[6] D. Bai and K. Sidiropoulos, "Vehicle license plate detection and recognition using symbol analysis," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 1217–1220.

[7] L. Sun, J. Yeo, and D. Lee, "Driver behavior analysis for safe driving," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 755–760.

[8] X. Zhang and X. Xu, "A lightweight driver distraction detection method based on three-level attention mechanisms," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[9] Y. Li and X. Liu, "An adaptive technique for computer vision based vehicles license plate detection system," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2780–2784.

[10] Z. Zhang and Y. Wang, "Driver fatigue detection based intelligent vehicle control," in *2005 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, 2005, pp. 68–73.

[11] R. Ramakrishnan and X. Chen, "Modeling driver lane selection decisions in a toll plaza," in *2014 IEEE Intelligent Vehicles Symposium*. IEEE, 2014, pp. 430–435.

[12] H. Wang, C. Zhao, and J. Wang, "Aerial target detection based on the improved yolov3 algorithm," in *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*. Chongqing, China: IEEE, 2019, pp. 1–5.

[13] X. Zhang, S. Li, and C. Ma, "Crtsii track slab crack detection based on improved yolov3 algorithm," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. Chongqing, China: IEEE, 2021, pp. 2272–2276.

[14] Z. Wu, Z. Xie, J. Zhang, X. Li, and X. Cheng, "Fisheye image object detection based on an improved yolov3 algorithm," in *2020 Chinese Automation Congress (CAC)*. Shanghai, China: IEEE, 2020, pp. 3314–3318.

[15] J. Liu, Z. Zhang, W. Han, and H. Zhou, "Edge detection-based boundary box construction algorithm for improving the precision of object detection in yolov3," in *2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. Guilin, China: IEEE, 2019, pp. 1–5.

[16] A. S. A. Kumar and S. Mathew, "Underwater object detection model based on yolov3 architecture using deep neural networks," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. Coimbatore, India: IEEE, 2021, pp. 243–248.