SafeCommuteAI:
Image based Real-time vehicle unsafe passenger behavior detection using Deep Learning

Project Team

Mustafa	22P-9221
Hamza	22P-9222
M Abdullah Aamir	22P-9235

Session 2022-2026

Supervised by

Mr. Fazlebasit
Dr. Hafeez Anwar

Department of Computer Science
National University of Computer and Emerging Sciences
Peshawar, Pakistan

October, 2025

Student's Declaration

We declare that this project titled "SafeCommuteAI: Image based Real-time vehicle unsafe passenger behavior detection using Deep Learning", 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.

Mustafa (22P-9221) Signature: _____

Hamza (22P-9222) Signature: _____

M Abdullah Aamir (22P-9235) 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 "SafeCommuteAI: Image based Real-time vehicle unsafe passenger behavior detection using Deep Learning", submitted by Mustafa (22P-9221), Hamza (22P-9222), and M Abdullah Aamir (22P-9235), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor
Mr. Fazlebasit
Signature: _____

Co-Supervisor
Dr. Hafeez Anwar
Signature: _____

Mr. Riaz Nawab
FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

Dr. Qasim Jan
HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

This project would not have been possible without the support of many people. Many thanks to our advisors, Sir. Fazlebasit and Dr. Hafeez Anwar, who read the numerous revisions we presented to them and gave appreciable and actionable feedback. Also a token of appreciation for our coordinator, Mr. Riaz Nawab, and every other faculty member from the Computer Science department for offering their guidance, support, and most importantly, their valuable time.

Thanks as well to the National University of Computer and Emerging Sciences for giving us the opportunity to complete this project.

Mustafa (22P-9221)
Hamza (22P-9222)
M Abdullah Aamir (22P-9235)

Abstract

Unsafe public transport is a major issue in South Asia and Africa, where buses, vans, and rickshaws (chingchis) are often dangerously overloaded. Passengers riding on rooftops or hanging from vehicles lead to frequent accidents and fatalities. Manual enforcement by traffic authorities is inconsistent, inefficient, and difficult to scale.

This project, "SafeCommuteAI," presents an AI-powered real-time system designed to automatically detect unsafe passenger behaviors and traffic violations using existing traffic and CCTV camera footage. The system integrates multiple deep learning modules to provide a comprehensive solution for data-driven enforcement.

The core functionalities include:

1. **Vehicle Type Classification:** Identifying specific vehicle types (Suzuki, Bus, Chingchi) to apply relevant safety rules.
2. **Passenger Counting:** A computer vision-based module to detect and count passengers, identifying overcrowding conditions (e.g., roof riding, hanging).
3. **License Plate Recognition (LPR):** Automatically identifying the license plates of violating vehicles to log evidence.
4. **Analytics Dashboard:** A web-based dashboard for authorities to monitor real-time violations, view captured evidence (images/clips), and analyze violation hotspots.

To achieve this, a custom dataset of over 2000+ images per class is being developed, capturing local transport conditions in Pakistan. By automating violation detection, SafeCommuteAI empowers traffic authorities with scalable, data-driven tools to reduce accidents, save lives, and improve urban transport safety.

Contents

Acknowledgements **3**

Abstract **4**

1 Introduction **1**
1.1 Purpose . . . 1
1.2 Problem Statement . . . 1
1.3 Background and Significance . . . 1
1.4 Approach . . . 1
1.5 Project Objectives . . . 2
1.6 Scope of Work . . . 2

2 Review of Literature **3**
2.1 Introduction . . . 3
2.2 Summary of Findings . . . 4
2.3 Research Gap . . . 4
2.4 Summary of Findings . . . 4
2.5 Research Gap . . . 4

3 Project Vision and Scope **6**
3.1 Problem Statement . . . 6
3.2 Project Impact . . . 6
3.3 Objectives . . . 6
3.4 Project Scope (Final Deliverables) . . . 6
3.5 Future Extensions . . . 7
3.6 Stakeholders . . . 7

4 Software Requirements Specifications (SRS) **8**
4.1 List of Features . . . 8
4.2 Functional Requirements . . . 8
4.3 Non-Functional Requirements . . . 10
4.4 Use Case Diagram . . . 11
4.5 System Architecture . . . 12

5 System Design and Methodology **13**
5.1 Development Methodology . . . 13
5.2 Tools and Technologies . . . 13
5.3 AI Model Design . . . 13

6 Implementation and Testing **15**
6.1 Project Milestones 15
6.2 Dataset Collection and Annotation 15
6.3 Model Training 15
6.4 Test Plan 16
6.5 Sample Test Cases 16

Bibliography **18**

Chapter 1

Introduction

1.1 Purpose

The purpose of this project is to design, develop and implement an AI-powered system that automatically detects unsafe overcrowding and other violations on passenger vehicles in real time. The system aims to provide traffic enforcement authorities with an efficient, scalable, and data-driven tool to improve road safety.

1.2 Problem Statement

Public transport in South Asian and African regions, particularly in Pakistan, is often and dangerously overloaded. Passengers are often seen sitting on the rooftops of buses or hanging dangerously from the sides of vehicles like Suzuki pickups and chingchis. This practice leads to a high number of road accidents, injuries, and fatalities.

Current enforcement methods rely on manual monitoring by traffic police, which is inefficient, inconsistent, and cannot provide 24/7 coverage. There is no automated system in place to detect these specific unsafe behaviors, log evidence, and alert authorities in real-time.

1.3 Background and Significance

The lack of automated enforcement leads to unchecked violations, contributing to traffic chaos and preventable loss of life. By building an intelligent system, we can empower traffic authorities with data-driven enforcement capabilities. An automated system reduces the need for constant physical monitoring, minimizes opportunities for corruption, and creates a scalable solution that can be deployed across a city. This project aims to make urban transport smarter and safer by reducing accidents and saving lives.

1.4 Approach

The project will be implemented in the following phases:

- **Dataset Collection & Annotation:** A custom dataset of 5,000-6,000+ images will be collected and annotated. This dataset is crucial as it will feature vehicle types (Suzuki, Bus, Chingchi) and violation scenarios (roof riding, hanging) specific to Pakistan.
- **Model Development:** Deep learning object detection models (such as YOLOv8) will be trained on the custom dataset to perform vehicle classification, passenger detection, and license plate recognition.

- **System Integration:** A backend system (using Flask/Django) will be developed to process video streams from cameras. This system will orchestrate the AI models to detect violations in real-time.
- **Dashboard Development:** A web-based analytics dashboard (using React/Flutter) will be built for authorities. This dashboard will display real-time alerts, violation logs with evidence (images, vehicle number), and statistics.

1.5 Project Objectives

- Collect and annotate a custom dataset of public transport vehicles with unsafe behavior.
- Train deep learning models (e.g., YOLOv8) to detect and classify vehicles, passengers (on roof, hanging), and license plates.
- Develop a backend system for processing video streams and logging violations.
- Build a dashboard for traffic authorities to visualize violations and hotspot areas.
- Enable real-time alerts for enforcement authorities.

1.6 Scope of Work

The final deliverables for this project are:

1. **Vehicle Type Classification:** Suzuki, Bus, and Chingchi.
2. **Custom Dataset Development:** 2000+ images per class.
3. **License Plate Recognition (LPR):** Automatic violator identification.
4. **Passenger Counting:** Computer vision-based overcrowding detection.
5. **Analytics Dashboard:** Real-time violation monitoring for authorities.

Chapter 2

Review of Literature

2.1 Introduction

This chapter reviews existing research and systems related to automated traffic monitoring, vehicle classification, object detection, passenger counting via density estimation, and license plate recognition. It summarizes key findings from foundational and recent survey papers, highlighting methods, performance benchmarks, and limitations relevant to the SafeCommuteAI project.

Table 2.1: Selected Studies Relevant to SafeCommuteAI

Sr	Author(s) & Year	Methodology	Key Findings	Gaps / Limitations
1	Redmon et al. (2016)	Introduced YOLO (You Only Look Once), treating object detection as a single regression problem on a grid system for real-time, single-pass detection.	Achieved significantly faster real-time performance than prior two-stage detectors while maintaining competitive accuracy. Unified architecture.	Struggled initially with small objects, dense object clusters, and precise localization compared to two-stage methods.
2	Ren et al. (2015)	Proposed Faster R-CNN, integrating a Region Proposal Network (RPN) directly into the CNN to generate object proposals efficiently within a two-stage detection framework.	RPN dramatically increased the speed of region proposal compared to methods like Selective Search, enabling near real-time high-accuracy detection. Set accuracy benchmarks.	Remained slower than single-stage detectors like YOLO due to its two-stage nature (proposal then classification).

Continued on next page

Table 2.1: (Continued) Selected Studies Relevant to SafeCommuteAI

Sr	Author(s) & Year	Methodology	Key Findings	Gaps / Limitations
3	Gao et al. (2020)	Surveyed CNN-based crowd counting methods, focusing on density map estimation techniques which learn to predict spatial density rather than detecting individuals.	Density estimation methods show strong performance for dense crowds and handle occlusion better than detection-based counting. Various CNN architectures explored.	Accuracy can still be affected by extreme scale variations, complex backgrounds, and severe occlusion. Requires specialized training data.
4	Usama, Hafeez Anwar & Saeed Anwar (2025)	Proposed a toll collection framework using YOLO (v2–v11, incl. Nano) for vehicle type recognition, LPR localization & reading. Created novel 10k image dataset addressing local challenges (decorations, occlusions).	Large YOLOv5/v8/v11 achieved ∼99% mAP@0.5. Nano variants achieved high mAP (97–98%) suitable for edge devices (Raspberry Pi tested). Addressed specific Pakistani vehicle/plate variations.	Focused on toll context; dataset size may limit generalization. Primarily tested specific YOLO versions.

2.2 Summary of Findings

(Placeholder: Summarize the key takeaways from these papers...)

2.3 Research Gap

(Placeholder: Describe the specific gap your project addresses based on these findings...)

2.4 Summary of Findings

(Placeholder: Summarize the key takeaways from the papers you reviewed. e.g., "The literature shows a clear trend towards deep learning, particularly CNNs like YOLO and Faster R-CNN, for object detection in traffic scenarios due to their superior accuracy and increasing speed. While LPR is a mature field, deep learning offers robustness improvements. Crowd counting via density estimation appears promising for handling occluded passengers...")

2.5 Research Gap

(Placeholder: Describe what is missing from existing research, building on the table. For example: "Despite advances, there is a lack of publicly available datasets and models specifically addressing the unique vehicle types (Chingchi, local Suzuki variants) and dangerous overcrowding behaviors (roof riding, extreme side-hanging) common in Pakistan and similar regions. Existing traffic

datasets primarily focus on standard cars, trucks, and pedestrians in less chaotic environments. Furthermore, integrating vehicle classification, LPR, and density-based passenger counting into a single real-time system for enforcement remains an open challenge.")

Chapter 3

Project Vision and Scope

3.1 Problem Statement

South Asian and African public transport is characterized by dangerously overloaded vehicles, and authorities lack the real-time, evidence-based tools for effective enforcement.

3.2 Project Impact

- **Improved Safety:** Aims to reduce accidents and save lives.
- **Smarter Enforcement:** Provides authorities with data-driven tools.
- **Automation & Efficiency:** Reduces reliance on manual, inconsistent monitoring.
- **Scalability:** The system can be expanded to cover multiple cities.

3.3 Objectives

- To improve road safety via AI automation.
- To provide traffic police with scalable monitoring tools.
- To build the first annotated dataset of unsafe transport in Pakistan.
- To deliver a working prototype with a real-time dashboard.

3.4 Project Scope (Final Deliverables)

The system will deliver the following core features:

1. **Vehicle Type Classification:** The system will identify and classify vehicles into three distinct categories: **Suzuki**, **Bus**, and **Chingchi**.
2. **Custom Dataset Development:** A comprehensive dataset of **2000+ annotated images per class** will be developed to train the AI models.
3. **License Plate Recognition (LPR):** A module to automatically detect and read the license plates of vehicles that are flagged for violations, enabling automatic violator identification.
4. **Passenger Counting:** A computer vision-based component to detect and count passengers, specifically identifying overcrowding violations such as people riding on the roof or hanging from the sides.

5. **Analytics Dashboard:** A web-based, real-time violation monitoring dashboard for authorities to view live alerts, violation history, and visual evidence.

3.5 Future Extensions

If time allows, the following features may be explored:

- **Pose Estimation:** For skeleton-based passenger counting (inside view).
- **Behavioral Safety Analysis:** Detection of helmet, seatbelt, or footboard hanging.
- **Explainable AI (XAI):** For transparent AI decisions.
- **Edge AI / Federated Learning:** For on-camera processing.
- **Predictive Modeling:** Heatmaps of accident-prone zones & violations.

3.6 Stakeholders

- **Traffic Police / Authorities:** Primary users of the dashboard for enforcement.
- **Public Transport Users:** Indirect beneficiaries from improved safety.
- **Vehicle Owners:** Subject to automated violation logging.
- **City Administration:** Interested in smart city initiatives and public safety metrics.

Chapter 4

Software Requirements Specifications (SRS)

4.1 List of Features

- F1: Real-Time Camera Feed Processing
- F2: Vehicle Type Detection
- F3: Unsafe Condition Detection (Overcrowding)
- F4: Person Detection and Counting (External)
- F5: Number Plate Detection and Recognition (LPR)
- F6: Violation Data Capture
- F7: Violation Data Transmission
- F8: Violation Data Storage
- F9: Admin Dashboard - Violation Viewing
- F10: Admin Dashboard - Search Functionality
- F11: Admin Dashboard - Filter Functionality
- F12: Admin Dashboard - View Violation Details
- F13: Admin Dashboard - Delete Violation Records
- F14: User Authentication (Admin)

4.2 Functional Requirements

This section details the specific functions the SafeCommuteAI system must perform.

- **FR1: Real-Time Camera Feed Acquisition**
  The system **shall** be able to process real-time video feeds streamed from fixed surveillance cameras for continuous monitoring.
- **FR2: Vehicle Type Detection**
  For each detected vehicle in the video feed, the system **shall** determine and classify its type as 'Bus', 'Suzuki pickup', 'Chingchi', or 'Other'.

- **FR3: Unsafe Condition Detection**
  The system **shall** determine whether a detected vehicle constitutes an unsafe condition. An unsafe condition is defined as having one or more passengers standing outside the designated passenger cabin or sitting/standing on the vehicle's rooftop.

- **FR4: Person Detection and Counting**
  The system **shall** detect individual persons located outside the normal vehicle cabin (e.g., standing on side steps, sitting/standing on the roof). It **shall** provide an estimated count of such persons for each vehicle instance. (Note: This count is an estimate due to potential occlusions and camera limitations).

- **FR5: Number Plate Detection and Recognition**
  Upon detecting an unsafe condition (FR3), the system **shall** attempt to:

  a. Detect the region corresponding to the vehicle's number plate.
  b. Extract the alphanumeric registration number from the detected plate region using Optical Character Recognition (OCR).
  c. If the number plate cannot be read (e.g., due to occlusion, blur, or distance), the system shall still log the violation but indicate that the LPR failed or the plate was unreadable.

- **FR6: Violation Data Capture**
  For each instance where an unsafe condition (FR3) is detected, the system **shall** capture and associate the following data:

  – A snapshot (image frame) clearly showing the vehicle and the unsafe condition.
  – The precise timestamp of the detection.
  – The location identifier of the camera feed.
  – The estimated count of external persons (from FR4).
  – The detected vehicle type (from FR2).
  – The recognized vehicle number plate (from FR5), if successfully read.

- **FR7: Violation Data Transmission**
  The system **shall** securely transmit the captured violation data (from FR6) to the central backend server for storage.

- **FR8: Violation Data Storage**
  The system **shall** store all transmitted violation data persistently in a structured database. Each record must include all associated metadata captured in FR6.

- **FR9: Admin Dashboard - Violation Viewing**
  The system **shall** provide a web-based admin dashboard that allows authorized personnel to view a list or log of all recorded violations. Display should include key information like snapshot thumbnail, timestamp, person count, vehicle type, and location.

- **FR10: Search Functionality**
  The admin dashboard **shall** provide functionality allowing the admin to search for specific violations based on the recognized vehicle number plate.

- **FR11: Filter Functionality**
  The admin dashboard **shall** provide functionality allowing the admin to filter the displayed violations based on:

  – Camera location.

  - Date and time range.
  - Detected vehicle type.

- **FR12: View Violation Details**
  From the violation list (FR9), the admin **shall** be able to select a specific violation to view its detailed information, including the full-resolution snapshot and all associated metadata (FR6).

- **FR13: Delete Violation Records**
  The admin dashboard **shall** provide functionality allowing an authorized admin to delete selected violation records from the database.

- **FR14: User Authentication**
  Access to the admin dashboard **shall** be restricted. The system **shall** require users to authenticate using secure login credentials before granting access.

4.3 Non-Functional Requirements

- **NFR1 (Performance):** The system must be able to process video streams in near-real-time (target < 500ms per frame end-to-end latency from frame input to violation detection).

- **NFR2 (Accuracy):**
  - Vehicle classification and unsafe passenger detection models should achieve a Mean Average Precision (mAP) of $> 85\%$ on the held-out test dataset.
  - LPR accuracy should be > 90

- **NFR3 (Usability):** The analytics dashboard must be web-based, accessible via standard browsers, secure, and intuitive for use by traffic monitoring personnel with minimal training.

- **NFR4 (Reliability):** The core violation detection and logging system should be designed for continuous operation (24/7) and include robust error handling and logging mechanisms.

- **NFR5 (Scalability):** The system architecture (particularly the backend processing and database) should be designed to potentially support processing feeds from multiple cameras concurrently in the future.

- **NFR6 (Security):** Data transmission (FR7) must use encrypted protocols (e.g., HTTPS/TLS). Access control (FR14) must prevent unauthorized access. Sensitive data in storage (FR8) should be protected.

4.4 Use Case Diagram

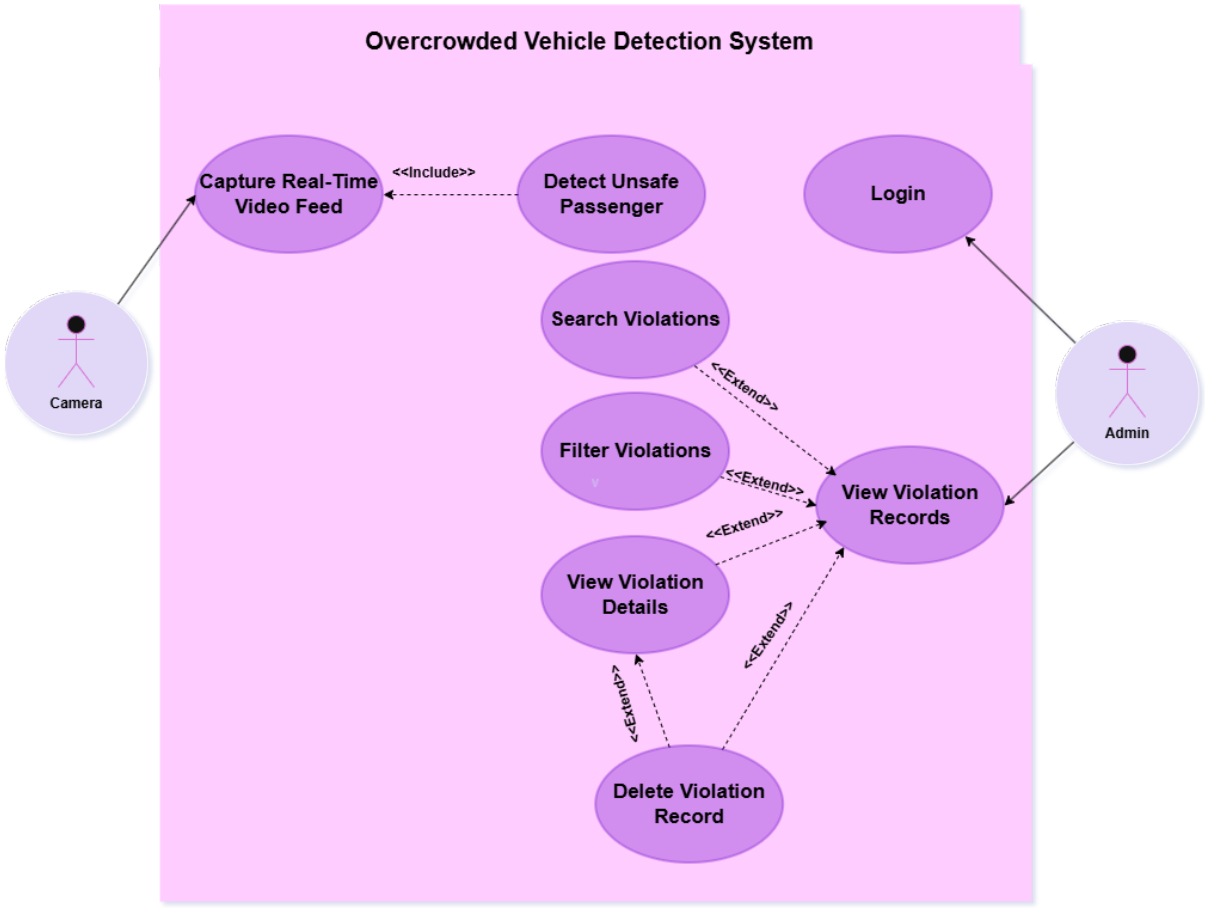

Figure 4.1: System Use Case Diagram.

4.5 System Architecture

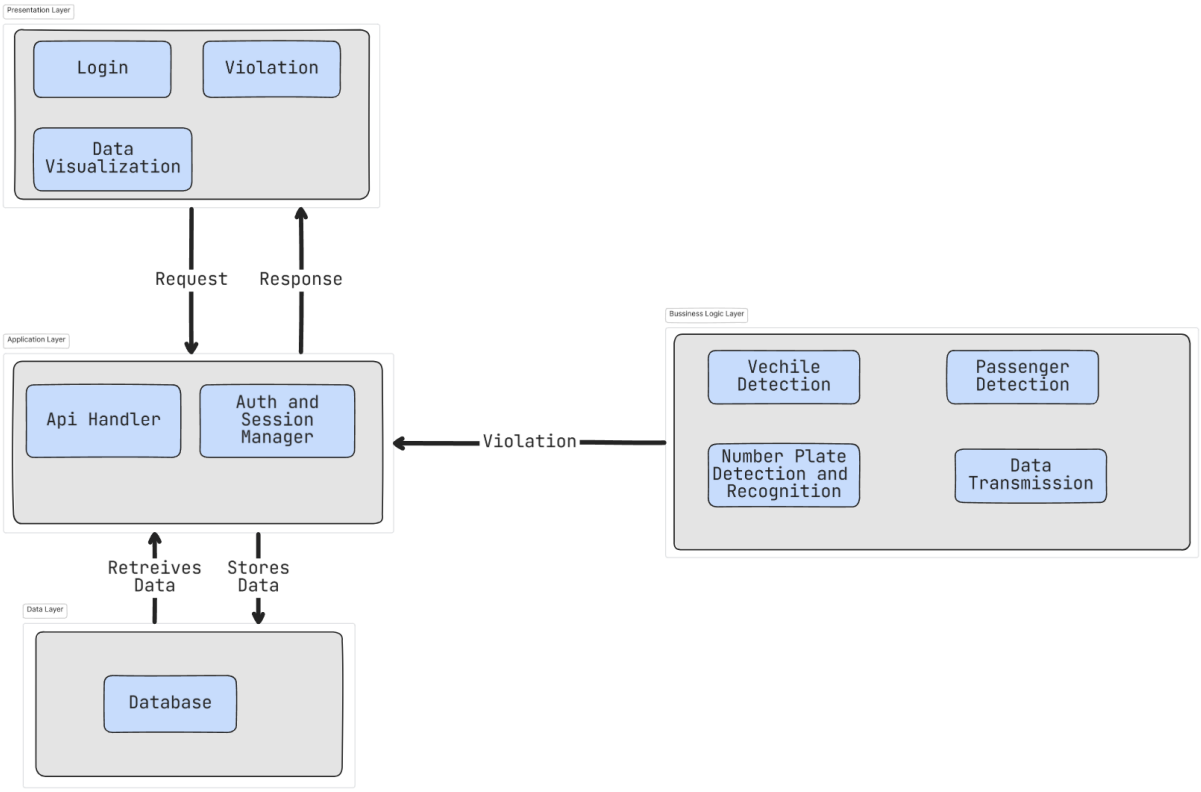

Figure 4.2: High-Level System Architecture.

Chapter 5

System Design and Methodology

5.1 Development Methodology

The project follows an iterative development methodology, adapting principles from Agile approaches. Development is planned in phases corresponding roughly to the FYP semester milestones, allowing for focused work on specific components like dataset creation, model development, and system integration. A key aspect of this approach is the iterative nature of the AI model development itself; models will be initially trained, evaluated, and then potentially retrained or refined as more data is collected or annotated, ensuring continuous improvement throughout the project lifecycle. This allows for flexibility and incremental testing of the dataset, models, and dashboard components.

5.2 Tools and Technologies

- **Programming Language:** Python
- **AI/Deep Learning:** PyTorch/TensorFlow, YOLOv8 (or chosen version), OpenCV
- **LPR Specifics:** May involve additional libraries like Tesseract OCR or specialized LPR models.
- **Backend:** Flask or Django (Web server, API for dashboard and data ingestion)
- **Frontend (Dashboard):** Flutter or React (Web-based UI)
- **Database:** MongoDB (To store violation logs)
- **Annotation:** CVAT
- **Version Control:** Git / GitHub

5.3 AI Model Design

The core of the SafeCommuteAI system is based on deep learning models for visual perception. The primary models and processing pipeline are designed as follows:

- **Model 1: Vehicle Detection.** The initial stage uses a primary object detection model (e.g., YOLOv8) trained to identify vehicles within the input frame. This model detects bounding boxes for relevant classes: 'Bus', 'Suzuki', 'Chingchi'. Processing only proceeds if a vehicle is detected.

- **Model 2: Unsafe Passenger Behavior Detection.** For each detected vehicle bounding box, the corresponding image region is cropped. A secondary detection model (potentially another specialized YOLOv8 or a similar architecture) is then applied *only* to this cropped region. This model is specifically trained to detect the class 'Passenger_Unsafe', identifying individuals sitting/standing on the roof or hanging from the sides. The count of these detections triggers the violation condition.
- **Model 3: License Plate Detection.** If an unsafe passenger condition is detected (passenger count > 0) on a vehicle crop, a dedicated license plate detection model (e.g., another YOLO variant trained specifically for plates) is run on the *same* vehicle crop to locate the license plate region.
- **Model 4: Optical Character Recognition (OCR).** If a license plate region is successfully detected, this region is cropped. An OCR model (such as Tesseract, EasyOCR, or a custom sequence model) is then applied to the plate crop to extract the raw text. Post-processing is applied to clean the OCR result and format the license plate text. If no plate is detected or OCR fails, a placeholder value (e.g., "none" or "unreadable") is used.

This cascaded approach, where subsequent models operate only on cropped regions identified by previous stages, aims to optimize processing speed. Upon confirmation of unsafe behavior and successful LPR (if possible), a violation record containing vehicle type, passenger count, plate text, timestamp, camera ID, and a saved image frame reference is generated and transmitted to the backend system.

Chapter 6

Implementation and Testing

6.1 Project Milestones

- **FYP-I (Semester 1 / Sep '25 - Jan '26):**
  - Proposal Submission & Defense (Sep '25)
  - Dataset Collection (Sep-Oct '25)
  - Dataset Annotation (Oct '25)
  - Model Selection & Initial Training (Vehicle/Passenger Model) (Nov '25)
  - Basic Detection Prototype (Vehicle/Passenger Detection) (Dec '25)
- **FYP-II (Semester 2 / Jan '26 - May '26):**
  - LPR Model Training Integration (Jan-Feb '26)
  - Real-time Detection Pipeline Setup (Jan-Feb '26)
  - Backend Development (Violation Logging, API) (Feb-Mar '26)
  - Dashboard Development (Feb-Mar '26)
  - Alerts & Evidence Logging Integration (Mar-Apr '26)
  - Final System Testing & Evaluation (Apr-May '26)
  - Final Documentation & Defense (May '26)

6.2 Dataset Collection and Annotation

Data was collected using mobile cameras at various locations in Peshawar and elsewhere, also scrapped from different internet sources known for public transport routes during peak and off-peak hours. Annotation was performed using CVAT, creating bounding boxes for 'Bus', 'Suzuki', 'Chingchi' classes.

6.3 Model Training

The YOLOv8m model was chosen for the primary detection task due to its balance of speed and accuracy. It was trained for 150 epochs on our custom dataset using transfer learning from COCO pre-trained weights. An Adam optimizer with a learning rate schedule was used. The dataset was split 70% training, 15% validation, 15% testing. For LPR, Tesseract OCR was initially evaluated.

6.4 Test Plan

- **Unit Tests:** Test individual Python functions (e.g., video frame reading, database insertion logic, LPR character filtering). Use PyTest.
- **Model Evaluation:** Assess model performance (mAP, precision, recall) on the held-out test set for both detection and LPR.
- **Integration Tests:** Verify data flow, e.g., ensure that a detected 'Passenger_Unsafe' triggers the LPR attempt and the result (or failure) is correctly logged in the database. Test API endpoints for the dashboard.
- **System Tests:** End-to-end testing using recorded video files and potentially a live camera feed. Verify that violations appear on the dashboard with correct data and snapshots. Test search and filter functionality.
- **Performance Tests:** Measure the average frames per second (FPS) the system can process. Measure the latency between an event occurring in the video and it appearing on the dashboard.

6.5 Sample Test Cases

Table 6.1: Sample Test Cases

Test ID	Description	Test Steps	Expected Result
TC-01	**Violation: Bus Overcrowding**	1. Input a video segment showing a bus with 3 people detected as 'Passenger_Unsafe'.	1. System classifies vehicle as 'Bus'. 2. System identifies 'Unsafe Condition'. 3. System logs violation with person count >= 3.
TC-02	**LPR Success**	1. Use video from TC-01 where the bus plate 'ABC-123' is clearly visible.	1. Violation logged (as per TC-01). 2. LPR module processes the frame. 3. Violation record in DB includes plate_number 'ABC-123'.
TC-03	**LPR Failure (Obscured)**	1. Input video of Chingchi with unsafe passengers, but plate is muddy/unreadable.	1. System classifies as 'Chingchi'. 2. System identifies 'Unsafe Condition'. 3. Violation logged with person count > 0. 4. Violation record in DB indicates plate_number as 'unreadable' or NULL.
TC-04	**No Violation: Normal Suzuki**	1. Input video of a Suzuki pickup with passengers only inside the cabin.	1. System classifies vehicle as 'Suzuki'. 2. No 'Passenger_Unsafe' detected. 3. No 'Unsafe Condition' flagged. 4. No violation record created.
TC-05	**Dashboard Search**	1. Ensure violations from TC-01/TC-02 exist. 2. Access dashboard, log in. 3. Search for plate 'ABC-123'.	1. Login successful. 2. Search results display the violation record(s) corresponding to TC-01/TC-02.
TC-06	**Dashboard Filter**	1. Ensure violations for Bus (TC-01) and Chingchi (TC-03) exist. 2. Access dashboard, log in. 3. Filter by Vehicle Type: 'Bus'.	1. Login successful. 2. Filtered list shows TC-01 violation but not TC-03 violation.

Bibliography

[1] The Express Tribune, "Action Sought Against Overloaded Public Transport," March 15, 2023. Available: https://tribune.com.pk/story/2406116/action-sought-against-overloaded-public-transport

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788. https://arxiv.org/pdf/1506.02640

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in neural information processing systems*, 28. https://arxiv.org/pdf/1506.01497

[4] Gao, G., et al. (2020). CNN-based Density Estimation and Crowd Counting: A Survey. https://arxiv.org/pdf/2003.12783

[5] Usama, M., Anwar, H., & Anwar, S. (2025). Vehicle and license plate recognition with novel dataset for toll collection. *Pattern Analysis and Applications*, 28, 57. https://link.springer.com/article/10.1007/s10044-025-01443-8