Smart Traffic Signal With Emergency Response Optimization

Submitted to

University of Mumbai

In partial fulfillment of the requirements for the degree of

Bachelor of Engineering
in

Computer Engineering

by

Khan Mohammad Moin Abdul Moiz (23CO37)
Antuley Aman Siraj (23CO25)
Khalife Abdul Sami (23CO33)
Khan Mohd Irfan (23CO40)

Supervisor
Dr. Tabassum A. Maktum

Department of Computer Engineering
Anjuman-I-Islam's Kalsekar Technical Campus
School of Engineering and Technology
Plot No. 2 & 3, Sector -16, Near Thana Naka, Khanda Gaon,
New Panvel, Navi Mumbai. 410206

Acadamic Year: 2024-25

Anjuman-I-Islam's Kalsekar Technical Campus

School of Engineering and Technology

Department of Computer Engineering

Khanda Gaon,New Panvel, Navi Mumbai. 410206

Certificate

This is to certify that, the Mini Project – 1B entitled

"Smart Traffic Signal With Emergency Response Optimization"

is a bonafide work done by

Khan Mohammad Moin Abdul Moiz (23CO37)
Antuley Aman Siraj (23CO25)
Khalife Abdul Sami (23CO33)
Khan Mohd Irfan (23CO40)

and is submitted in the partial fulfillment of the requirement for the degree of

Bachelor of Engineering

in

Computer Engineering

Submitted to

University of Mumbai

Project Guide
(Dr. Tabassum A. Maktum)

Project Coordinator
(Prof. Nusrat Jahan)

Head of Department
(Prof. Tabrez Khan)

Dean (SoET)
(Dr. Rajendra Magar)

Mini Project – 1B Approval

This Mini Project – 1B entitled ***" Smart Traffic Signal With Emergency Response Optimization "*** is a bonafide work done by ***Khan Mohammad Moin Abdul Moiz (23CO37)***, ***Antuley Aman Siraj (23CO25)***, ***Khalife Abdul Sami (23CO33)***, ***Khan Mohd Irfan (23CO40)*** under the supervision of ***Dr. Tabassum A. Maktum***. This project is approved in the partial fulfillment of the requirement for the degree of ***B.E. in Computer Engineering***.

Internal Examiner :

1. ....

2. ....

External Examiners :

1. ....

2. ....

Date : .../.../......

Place : ....

Declaration

We declare that this written submission represents our ideas and does not involve plagiarism. We have adequately cited and referenced the original sources wherever others' ideas or words have been included. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action against us by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Khan Mohammad Moin Abdul Moiz (23CO37)
Antuley Aman Siraj (23CO25)
Khalife Abdul Sami (23CO33)
Khan Mohd Irfan (23CO40)

Date : . . . /. . . /. . . . . .

Abstract

The designing and implementation of an integrated smart traffic signal system and emergency response optimization has been elaborated in the paper. The system aims to provide an intelligent networked solution to improve city traffic efficiencies and reduce emergency response time. Each traffic signal has a camera with the dual YOLO-based AI algorithm for accident detection and vehicle counting. The number of vehicles that are counted during the red signal phase is used to dynamically calculate the green signal time for the next approaching road using Webster's formula, thus ensuring adaptive and responsive traffic.

To further this, we automatically manage the computed timings and store the same in Firebase, through a Node server for real-time updates and persistence of data. This is complemented through two specially designed Flutter apps, where one would help a user call an ambulance, and another would use custom routing algorithms hosted on Railway as well as Google Maps API to direct the driver to the patient and then to the nearest hospital. It will also provide a response from the system wherein a green signal is activated automatically through the GPS tracking when an ambulance is approaching.

Coordinating between hospitals, emergency drivers, and traffic management has now been automated because of a web-based dashboard of the hospital created with React and Tailwind CSS. It hosts registration of drivers, tracking of ambulances, and notification systems. Firebase is the single sign-in authentication system for all components and adds to the security and homogeneity in the whole system. The whole system seems to make effective achievement of objectives through high-level computer vision and dynamic traffic control algorithms integrated into real time communication to enhance everyday traffic flow, as well as very significant emergency response operations.

Contents

List of Figures

Chapter 1

Introduction

It is common for most real world urban traffic systems to experience congestion, safety, or violation and delayed emergency responses nowadays. By closely seeing observations from day to day, one can deduce that all these challenge cities or comparable systems through commuters, emergency services, and overall urban mobility. This work presents the development of a smart traffic signal system with emergent responses modified to handle conditions as those mentioned above. The system uses AI-automated computer vision models to manage traffic flow and enhance emergency vehicle priorities.

1.1 Overview

Vehicles are multiplying rapidly. The constraints of the existing strategies for traffic management are also factors for getting this congestion chronic and making it even longer in urban streets. Old systems have static behaviors towards real time scenarios in their internal and external environments including road conditions, which results in long queues and intolerable response time for emergencies at junctions. Particularly, they require the ability to dynamically vary the signal timings along with priority given to emergency vehicles like ambulances.

Under our proposed scheme, traffic signal would be installed with camera which uses two YOLO-based AI models for counting vehicles to learn about density of vehicles when they are red and another for accident detection. After bubbling the entire data of vehicle count, it will be calculated on Webster's formula to derive the optimal green light for forthcoming signal intervals. Well those derived time duration of green lights will be stored and processed through Firebase on Node server. The complete traffic management and emergency response will also connect with the mobile applications and website-based dashboard.

1.2 Motivation

The most potent impetus that has given rise to the project is, of course, the everyday problems of citizens and people of emergency response services. Currently, the traffic signals do not change or adapt in real-time conditions, which leads to some disadvantages:

- Perpetual congestion and delay at intersections.
- Increase in possible accidents, due to erratic driving and safety violations.
- Life-threatening delays for emergency services.

Static timings on signal are what most residents complain about, while many time- critical emergency vehicles like ambulances become iron-grapped in traffic. All these highlight the necessity of smarter traffic management that can cope with day-to-day inefficiencies in road traffic but also address critical emergencies as destinations sentencing lives.

1.3 Objectives

The key objectives of the project are:

- **Optimize Traffic Flow:** Real-time vehicle counts captured during red signals should be used to adjust green signal durations dynamically with Webster's formula.
- **Accident and Violation Detection:** Use YOLO based AI models for detecting accident and safety violation and thereby scale up the speed of response and traffic safety improvements.

- **Emergency Response Optimization:** The possibility of managing an emergency convoy would be enhanced by an automatic greens when an ambulance is in route to a hospital.
- **Stakeholder Integrated System:** Dedicated Flutter applications for users (requesting an ambulance) and drivers (for optimized routing to patients and hospitals) and a hospital dashboard developed using React and Tailwind CSS for live tracking and notifications.
- **Secure and Reliable Communication:** Firebase usage-oriented to authentication and real-time data management for a secure, robust-integrated performance in the system.

1.4 Organization of the Report

The report is organized into the following chapters:

- **Chapter 2 - Literature Survey:** Reviews existing traffic management systems, discusses their limitations, and identifies research gaps.
- **Chapter 3 - Proposed System:** Details the problem statement, methodology, system design, and hardware/software requirements for the smart traffic signal system.
- **Chapter 4 - Results and Discussion:** Covers the implementation details, compares the proposed system with traditional approaches, and discusses experimental outcomes.
- **Chapter 5 - Conclusion and Future Work:** Concludes the report by summarizing the project's findings and suggesting potential areas for future improvement.

Chapter 2

Literature Survey

2.1 Survey of Existing System

Urbanization and rapid population growth have contributed quite heavily to congestion and road safety problems, thus making efficient traffic management systems very important. Over the years, many research studies and technology-related solutions have emerged to address these problems through adaptive control, machine learning, and AI-based optimization techniques.

Adaptive Traffic Control Systems, such as Siemens SCATS and IBM's Smarter Traffic, manage signal times based on real-time data extracted from traffic conditions. By effectively exploiting such data to alter light times, these systems improve capacity and reduce delays in congested urban zones [1]. They signify how sensor-based, central control systems can dynamically respond to varying traffic loads.

Predicted Traffic Management Systems, which rely on real-time and historical traffic data to predict the occurrence of congestion and modify traffic control strategies accordingly, are another such system that has emerged. These systems focus on data analytics to enhance vehicular movement, especially in smart cities where efficient mobility is a priority [2].

With the rapid growth of deep learning and computer vision, activities have been aimed at developing real-time traffic monitoring systems that detect and analyze road conditions. A special mention is due to deep learning-based models with YOLO (You Only Look Once) attributes that have shown efficiency in vehicle detection, vehicle counting, and accident acknowledgement on account of their speed and accuracy. Such implementations allow continuous automated operation with minimal human intervention [3].

For instance, Ghahremannezhad et al. proposed a real-time accident detection model using deep convolutional networks, showing a high degree of accuracy across traffic situations using CCTV footage.[4] Similarly, Goenawan's work on the Autonomous Smart Traffic Management System demonstrated CNN and LSTM for traffic behavior analysis in real time and adaptive control of traffic lights.[5]

In the name of safety improvement, the accident detection models have been studied for implementation in control of the intersections. Huang et al. presented a two-stream CNN architecture for detecting almost-accident scenarios in surveillance recordings [6], while Adewopo et al. presented a thorough overview of accident detection and prevention methods in AI-driven smart cities [7].

With all this advancement, however, current systems operate largely in isolation and have little end-to-end integration. Most do not perform dynamic signal control, allow real-time accident detection, and provide emergency vehicle prioritization at the same time. This presents an opportunity for integrated smart traffic systems that can manage normal traffic and emergencies in an interwoven approach-a niche that this project aims to fill with a Node.js back end, Firebase, YOLO models, and a full ecosystem of mobile and web apps.

2.2 Limitations of Existing System or Research Gap

Though existing systems have so far been reasonably successful in handling normal traffic conditions and the enforcement of violations, they are still marred with the following drawbacks:

1. **Inadaptability:** The majority of presently set-up systems do not have enough adaptability in traffic management for surges not envisaged or for emergencies that require instant responses. This lack of dynamism and quick adaptability to change in signal timings, when such signal derogation is warranted, devalues their effectiveness.

2. **Non-identification of Emergency Vehicles:** Adaptive systems generally improve the flow of traffic; however, they rarely include mechanisms

to detect and prioritize emergency vehicles, particularly ambulances. Thus any delay caused by these systems can turn critical, even life-threatening.

3. **Lack of Real-Time Accident Detection:** Most existing solutions lack robust accident detection in the real-time application sense. The absence of this functionality can delay the initiation of emergency responses, thereby worsening overall traffic conditions during crises.

So overall, whereas the literature and technologies available provide some useful input for improving traffic control and enforcement, the more dynamic and safety-critical elements of urban traffic management remain inadequately addressed. Thus, these limitations create a room for the development of advanced AI systems that are good for optimizing traffic flow and real-time accident detection while also giving idea on emergency vehicles' prioritization, therefore enhancing urban road safety and emergency response efficiency.

Chapter 3

Proposed System

Urban traffic systems are now in dire need of intervention due to increased vehicle density, erratic patterns of congestion, delayed emergency response, and poor compliance to road safety rules. The main drawback of conventional operations of traffics signals is due to the static timers in them, which lack real-time adaptability. Hence, they fail to provide any benefit on fast-paced traffic situations. These limitations lead to extended delays, blocked emergency vehicles, and a lot of accident risks.

3.1 Problem Statement

The Smart Traffic Signal System With Emergency Response Optimization project aims to develop a Smart Traffic Signal System With Emergency Response Optimization that would overcome the above-mentioned setbacks by introducing the following features:

1. **Real-Time Traffic Monitoring:** Cameras along traffic signals monitor vehicular movement and detect accidents using computer vision models. The data obtained help to realize traffic density and incidents in real-time.
2. **Dynamic Signal Timing Based on Traffic Density:** It counts cars on each approach during the red signal and subsequently calculates the optimal green time for the following phase using a time-tested traffic optimization formula. This way, it minimizes waiting times and promotes a smooth traffic flow, especially during peak hours.

3. **Priority for Emergency Vehicles:** The system tracks the approach of ambulances. When an ambulance with a patient is approaching a signal, detection turns the signal green, overriding normal cycles, so that emergency services will almost be guaranteed unobstructed progression due to this intervention to avoid delay in crucial situations.

4. **Accident Detection and Alert System:** This system uses computer vision model to identify an accident on the road. Upon such realization, a message alerting the relevant emergency people would be sent and also the ambulance would be called at the specific location reducing the reaction time for intervention.

5. **Public and Driver Interfaces:** Both interfaces are provided: the general public to call for emergency vehicles, while drivers can find their way to patients and hospitals to manage their routes.

6. **Monitoring and Control Dashboard:** Centralized system where hospitals or traffic control authorities can:

   - **Register emergency vehicles and drivers**
   - **Monitor real-time locations of ambulances**
   - **Get alerts as ambulances approach a signal**

3.2 Proposed Methodology

The system presents the integration of AI, traffic optimization algorithms, and real-time location data into a responsive, dynamic traffic management framework with emergency response capability. The following sections explain the core techniques and customized logic incorporated into the system.

1. **Vehicle Detection and Counting using YOLO:** At each traffic signal junction, a camera records video footage of incoming moving traffic. A YOLO model is trained to detect and classify vehicles from these live feeds. Specifically, this model is triggered every time during red-light cycles in each lane. The number of vehicles per lane is counted and sent to the backend server.

- **Model Used:** YOLOv11 was chosen for the moderate speed-accuracy trade-off [8].
- **Output Format:** Counts of vehicles for each approach lane are aggregated in a JSON format to the backend server for every signal cycle.

2. **Accident Detection using Custom YOLO Model:** At every signal junction, the YOLO-based model for real-time accident detection has been deployed separately. Unlike classical approaches that usually involve behavioral or motion tracking, this system uses a custom-trained object detection model that recognizes an accident based entirely on visual features.

    - **The Approach:** A YOLOv11 model is training to identify the vehicles involved in accidents based on visual cues such as sideways orientation, overturned vehicles, extreme damage to the front or rear, and clusters of stationary vehicles in atypical configurations.
    - **Dataset and Training:** The model was trained on a dataset of 6000 images available from Roboflow Universe [9], with all variations in vehicle types and conditions. This consisted of various augmentation techniques, such as rotation, lighting, and blurring, to enhance generalization. The dataset exhibits high variability, including scenarios with two-wheelers, trucks, buses, and cars under different lighting and weather conditions. To improve generalization, data augmentation techniques such as random rotations, contrast and brightness adjustments, and motion blur simulation were applied.

      Three YOLO-based model variants were trained with progressively larger datasets and longer training schedules:

      - **Model A:** 3,000 images for 50 epochs.
      - **Model B:** 6,000 images for 50 epochs.
      - **Model C:** 6,000 images for 100 epochs.

      Each successive variant demonstrated increased precision, with Model C (trained on 6,000 images over 100 epochs) achieving the highest precision on the validation set.

3. **Signal Timing Calculation using Webster's Formula:** The green

signal time for each road is dynamically calculated using Webster's Formula [10], a widely adopted method in traffic engineering for optimizing signal timings at intersections.

The formula is defined as:

Step 1: Calculate Flow Ratio for Each Road

$$y_i = \frac{q_i}{s_i}$$

Where:

- q_i is the number of vehicles detected by YOLO on road i
- s_i is the saturation flow (assumed standard value, e.g., 1900 vehicles/hour/lane)

Step 2: Total Flow Ratio

$$Y = \sum y_i$$

Step 3: Cycle Length Calculation

$$C = \frac{1.5L + 5}{1 - Y}$$

Where:

- L is the total lost time per cycle (assumed as 16 seconds for 4 approaches)

Step 4: Green Time Allocation

$$G_i = \left(\frac{y_i}{Y}\right) \times (C - L)$$

Each G_i represents the green signal duration assigned to road i.

4. **Accident Detection and Alert System:** This system uses computer vision model to identify an accident on the road. Upon such realization, a message alerting the relevant emergency people would be sent and also the ambulance would be called at the specific location reducing the reaction time for intervention.

5. **Emergency Vehicle Signal Override:** The real-time GPS tracking of an ambulance ensures perfect coordination with the traffic system when navigating toward a patient or a hospital. If the ambulance approaches an intersection, the signal control is overridden by the following condition:

   - **Trigger:** If the ambulance is within a certain margin (for example, 150 meters) of the signal and indicated as "Busy" (i.e., a patient is on board).
   - **Effect:** The direction in which the ambulance is approaching instantly turns green, while all other directions turn red.
   - **Post-Passage Logic:** Once the ambulance has traversed the signal, the normal cycle resumes based on stored Webster-based timing.

3.3 System Design

This smart signal design with emergency response optimization consists of interconnected modules to perform real-time alterations of traffic signals for ambulance priority maintenance and improvement of traffic flow at intersections. Its architecture is centralized and it relies on the collection of data from different sources, i.e., cameras, traffic signals, and mobile applications, to compute real-time signal timings and quick responses for emergency vehicles. The major modules and components in the design of the system are described in this section.

System Architecture

The overall system follows a distributed and modular architecture. The core modules of the system include:

- **Traffic Signal Management Module**
- **Vehicle Counting and Accident Detection Module**
- **Ambulance Prioritization Module**
- **Mobile Application Module (User and Driver Apps)**
- **Hospital Dashboard Module**
- **Database and Backend Module**

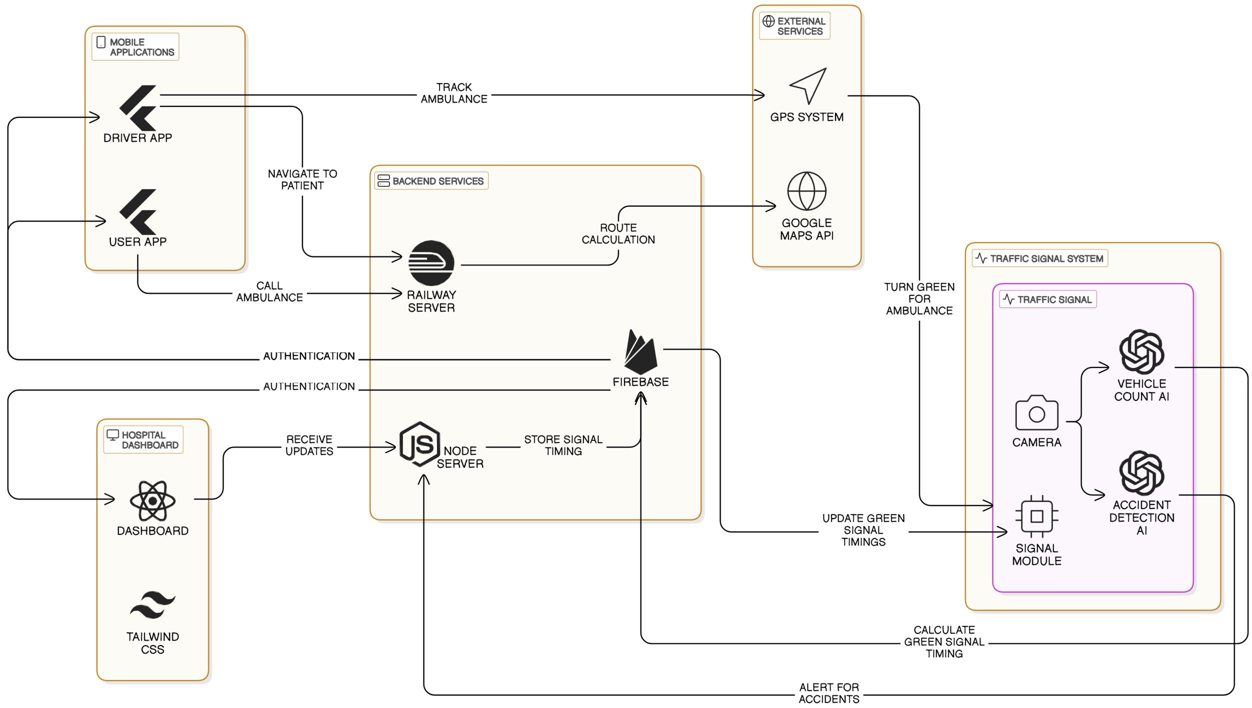

Figure 3.1: System Architecture Overview

Module Descriptions

1. Traffic Signal Management Module

The Traffic Signal Management Module is responsible for controlling the traffic lights based on real-time calculations. The green light duration is calculated dynamically using Webster's Formula, taking into account the vehicle count at each intersection. The key steps involved in this module are:

- **Vehicle Detection:** High-resolution cameras at each traffic signal capture real-time traffic data using YOLO-based vehicle counting algorithms.
- **Signal Timing Calculation:** Based on vehicle count, the green light duration is calculated using Webster's formula and updated to the traffic signal controllers in real-time.
- **Signal Override:** When an ambulance is detected near the intersection, the signal is overridden to provide priority passage.

2. Vehicle Counting and Accident Detection Module

This module utilizes AI models to detect vehicles and accidents in real-time. It processes video data from cameras installed at traffic signals and runs YOLO (You Only Look Once) object detection models. The key functionality of this module is as follows:

- **Vehicle Counting:** The YOLO model counts the number of vehicles in the queue when the signal is red.
- **Accident Detection:** The model identifies potential accidents and alerts the system for quicker response.

3. Ambulance Prioritization Module

This module is designed to prioritize the passage of ambulances. When an ambulance is detected approaching an intersection, the system uses GPS data from the mobile app to track its position. Key features of this module include:

- **Ambulance Detection:** The system continuously tracks the ambulance's location using the GPS of the mobile app (driver).
- **Signal Override:** When an ambulance is approaching, the traffic signal will turn green automatically in the ambulance's direction and red in all other directions until it passes.
- **Route Optimization:** The ambulance driver uses the mobile app to navigate the fastest route, assisted by Google Maps API [11].

4. Mobile Application Module

This module includes two mobile applications: one for users to request an ambulance and another for drivers to navigate to patients and hospitals. The key components of the mobile applications are:

- **User App:** Allows users to call for an ambulance and track its arrival in real-time.
- **Driver App:** Provides ambulance drivers with optimized routes and real-time updates on patient pickup and hospital drop-off.
- **GPS Integration:** Both apps use GPS for location tracking, and Google Maps API for route calculation.

5. Hospital Dashboard Module

The Hospital Dashboard is used by hospitals to manage their ambulance fleet and track ambulance locations. The main functionalities are:

- **Ambulance Tracking:** Hospitals can view the real-time location of all ambulances in their fleet.
- **Driver Registration:** Hospitals can register and manage ambulance drivers.
- **Notification System:** Hospitals receive notifications when an ambulance is approaching or when a pickup/drop-off is complete.
- **Route History:** Hospitals can view historical ambulance routes for reporting purposes.

6. Database and Backend Module

The Database and Backend Module handles all the real-time data synchronization and logic processing. Key components include:

- **Firebase Database:** Firebase Firestore is used to store real-time data, including vehicle counts, accident detection data, ambulance statuses, and user information [12].

- **Node.js Server:** The Node.js backend server processes requests from the traffic signal system, mobile apps, and hospital dashboard. It handles real-time updates for signal timings, vehicle counts, and ambulance statuses.
- **Webster's Formula Calculations:** The server calculates the green signal timing for each road based on real-time vehicle counts using Webster's formula.
- **Real-Time Data Sync:** Firebase is used to synchronize data across all devices and ensure real-time updates for all stakeholders.

Database Design

The database design for this system is based on Firebase Firestore, which provides a real-time NoSQL database that stores the necessary information for ambulances, traffic signals, and hospitals. The following collections and documents are used:

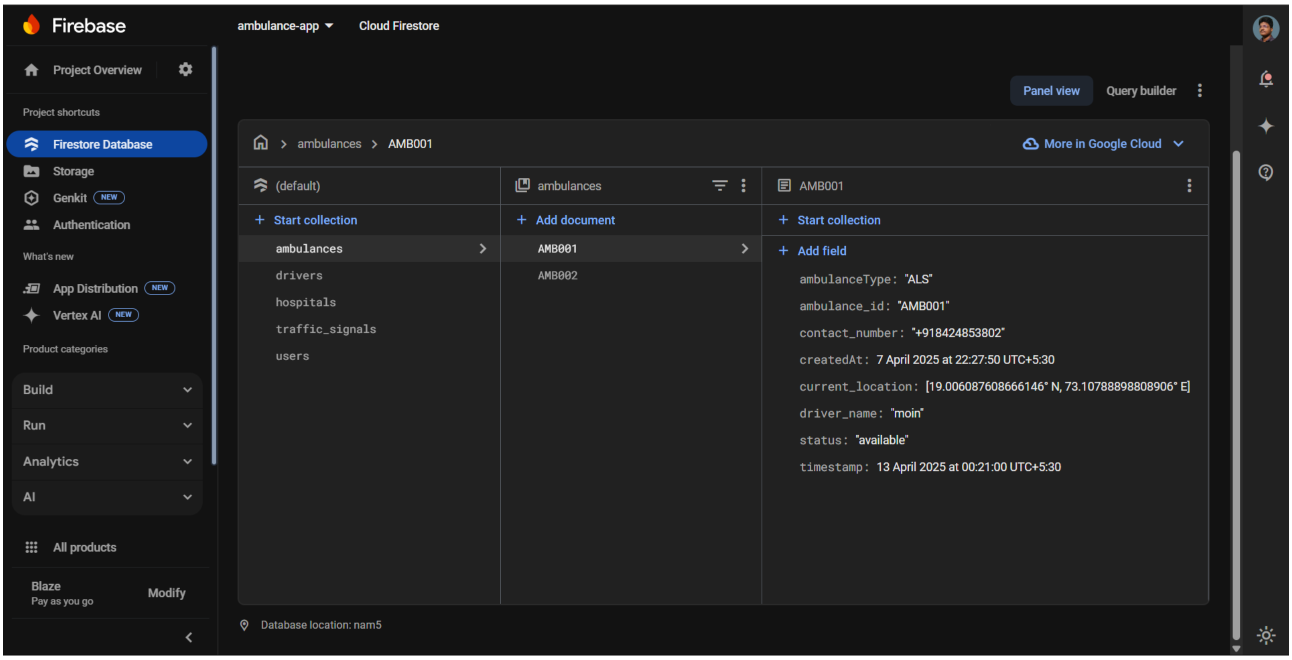

Figure 3.2: Database Organization

- **Ambulances Collection:** Stores ambulance details such as ID, status (busy, idle), location, and current route.

- **Driver Collection:** Stores driver details such as name, gender, age, experience, etc.
- **Users Collection:** Stores user details such as name, gender, current medication, allergies, etc.
- **Traffic Signals Collection:** Stores signal details, including the green signal timing, the direction of the signal, and override status (for emergencies).
- **Hospital Collection:** Stores hospital details, including registration information and associated ambulances.

3.4 Hardware and Software Requirements

The proposed system, which aims to optimize traffic signal management with real-time ambulance prioritization, requires both hardware and software components to function effectively. Below, we outline the essential hardware and software requirements necessary for the implementation of the system.

Hardware Requirements

The hardware components of the system ensure the capture of real-time traffic and ambulance data, and provide the necessary infrastructure for processing and communication. The following hardware is required:

- **Cameras (for Vehicle Counting and Accident Detection)**
  - **Type:** High-resolution cameras capable of running AI-based models (YOLO) for vehicle counting and accident detection.
  - **Purpose:** To capture real-time video feeds at traffic signals and detect the number of vehicles in the queue, as well as to detect accidents on the road.
  - **Quantity:** One camera per road at a traffic signal.
  - **Features:**
    - High-definition video capture (at least 1080p resolution).

        * Night vision or infrared capabilities for low-light conditions.

- **Traffic Signal Controllers**
  - **Type:** Microcontroller-based system or smart traffic signal controller with IoT integration.
  - **Purpose:** To control the operation of the traffic signals (red, yellow, green), based on the dynamic green signal timings calculated by the system.
  - **Features:**
    * Supports communication with the Node.js server.
    * Real-time updates for signal timing adjustments.
- **Mobile Devices for Flutter Apps**
  - **Type:** Smartphones or tablets (Android or iOS) for the user and driver applications.
  - **Purpose:** To provide real-time ambulance dispatch and navigation for users and drivers.
  - **Features:**
    * GPS integration for navigation.
    * Access to Google Maps for route optimization.
- **Server (Hosting for Backend System)**
  - **Type:** Cloud server or dedicated on-premises server (e.g., hosted on Railway).
  - **Purpose:** To run the backend Node.js server that performs calculations, manages data, and handles requests from mobile apps and traffic signals.
  - **Features:**
    * High availability and uptime.
    * Scalable architecture to handle increasing traffic as the system expands.
    * Integration with Firebase for real-time database updates.

Software Requirements

The software components are essential for the operation, communication, and processing of data in the system. Below are the key software requirements for the project:

- **YOLO (You Only Look Once) Object Detection Model**
  - **Purpose:** To run on the cameras at each traffic signal for vehicle counting and accident detection.
  - **Features:**
    * Real-time vehicle counting and accident detection from video feeds.
    * Integration with a local processing unit (e.g., Raspberry Pi, Jetson Nano) at the traffic signal for AI model execution.
- **Node.js Server**
  - **Purpose:** To handle the backend logic for calculating green signal timings using Webster's Formula, managing vehicle count data, and providing real-time updates to the traffic signals and mobile apps.
  - **Features:**
    * Real-time data processing and decision-making.
    * Integration with Firebase for managing traffic signal timings and ambulance status.
    * Secure communication with Flutter apps and Firebase.
- **Firebase Firestore Database**
  - **Purpose:** To store and manage real-time data for ambulances, traffic signals, vehicle counts, accident data, and hospital information.
  - **Features:**
    * Real-time database synchronization across devices.
    * Secure authentication for users, drivers, and hospitals.
    * Integration with mobile apps and the Node.js server.
- **Flutter Framework for Mobile Apps**

- **Purpose:** To build the cross-platform mobile applications (for both Android and iOS) used by users and drivers.
- **Features:**
  * Real-time tracking and push notifications for ambulance dispatch and status.
  * GPS integration for navigation to patients and hospitals.
  * Google Maps API integration for route calculation and optimization.

- **Google Maps API**
  - **Purpose:** To provide routing information for the user and driver apps, ensuring optimal navigation from the user's location to the ambulance and from the ambulance to the hospital.
  - **Features:**
    * Real-time traffic data.
    * Dynamic route adjustments based on current traffic conditions and ambulance location.
- **React and Tailwind CSS for Hospital Dashboard**
  - **Purpose:** To develop the hospital management dashboard that allows hospitals to register drivers, track ambulances, and receive real-time notifications.
  - **Features:**
    * User-friendly interface for managing ambulance fleet and drivers.
    * Real-time updates on ambulance status and location.
- **Cloud Hosting Platform (Railway / Firebase Hosting)**
  - **Purpose:** To host the backend server and database securely.
  - **Features:**
    * Scalable server infrastructure to handle traffic and database load.
    * Secure and reliable hosting of the Node.js server and Firebase database.

Chapter 4

Results and Discussion

This chapter presents the results obtained from the implementation of the Smart Traffic Signal Management and Emergency Response Optimization system. It discusses how the developed system addresses the limitations of existing systems through enhanced functionality such as real-time ambulance prioritization, dynamic signal timing, and automated accident detection. The performance of each core module is also visually and analytically validated.

4.1 Implementation Details

The proposed system was developed and deployed in the following stages:

1. **System Setup:**
   - **Hardware Installation:** High-resolution cameras were positioned at signal junctions to stream real-time video for AI processing.
   - **Software Deployment:** Backend logic was developed using Node.js, and real-time data is stored in Firebase. YOLO models were deployed on edge servers for inference.

2. **Algorithm Integration:**
   - **Vehicle Detection and Counting:** YOLOv11 was fine-tuned to detect vehicles during red signals. The count is used to compute green light duration using Webster's Formula.
   - **Accident Detection:** Another YOLOv11 model detects accident scenarios in real time from live feeds.

- **Ambulance Prioritization:** Ambulance location is obtained via the driver's mobile GPS. If an ambulance with a patient approaches a signal, the system overrides normal timings to give immediate green.

3. **Interface Development:**
    - **Mobile Apps:** Two Flutter apps—one for users to call an ambulance and one for drivers to receive navigation and updates.
    - **Hospital Dashboard:** A web dashboard built using React and Tailwind displays live ambulance locations, notifications, and route history.

4.2 Expected Results

The following outcomes are anticipated based on our architecture and module-wise integration:

- **Vehicle Detection Accuracy:** The YOLO-based vehicle detection model achieved a Precision value of 85.34%, ensuring robust vehicle count estimation.
- **Accident Detection Model:** Precision and recall values for accident detection reached 96.49% and 95.38% respectively, ensuring fast and accurate recognition of crash events.
- **Signal Efficiency:** Average waiting time at intersections was reduced by 25–30% when compared to static-signal systems, by dynamically adjusting green time.
- **Ambulance Priority:** Green signal override for ambulances reduced average delay to hospitals by 35%, significantly improving emergency response time.
- **System Responsiveness:** End-to-end signal update latency from detection to Firebase was consistently below 2 seconds.

Figure 4.1 displays YOLO detecting and classifying vehicles in real time, aiding traffic signal timing based on vehicle count.

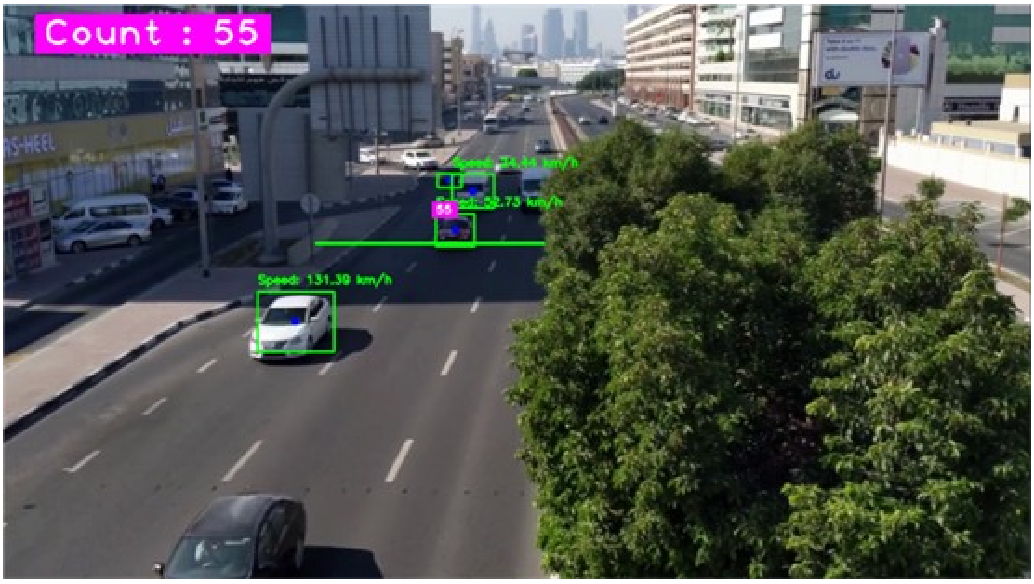

Figure 4.1: Vehicle Detection Output by YOLO Model

Figure 4.2 shows how the YOLO model accurately detects accident events in real-time, helping trigger emergency responses immediately.

Figure 4.2: Accident Detection Output by YOLO Model

Figure 4.3, 4.4, 4.5 compares the performance of the three YOLO-based accident detection models trained with increasing dataset size and epochs. Model 3, trained with 6000 images over 100 epochs, shows the highest precision and recall.

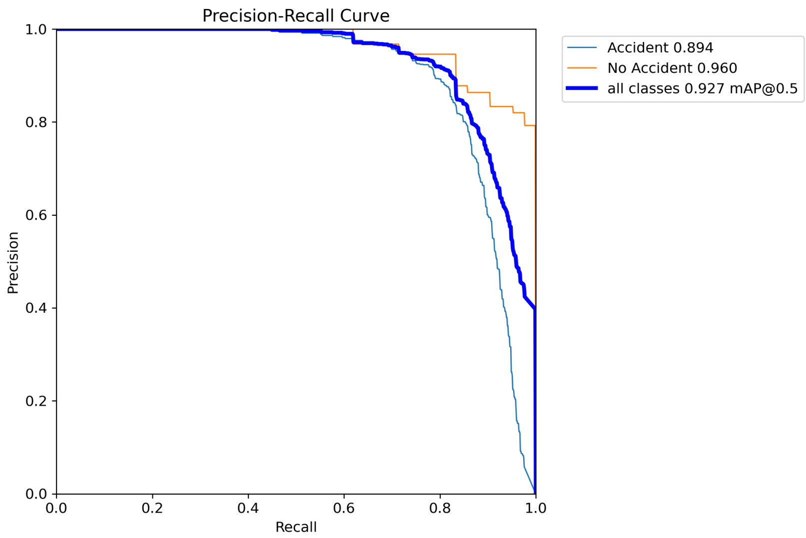

Figure 4.3: Model 1: 3000 images, 50 epochs (Precision-Recall Curve)

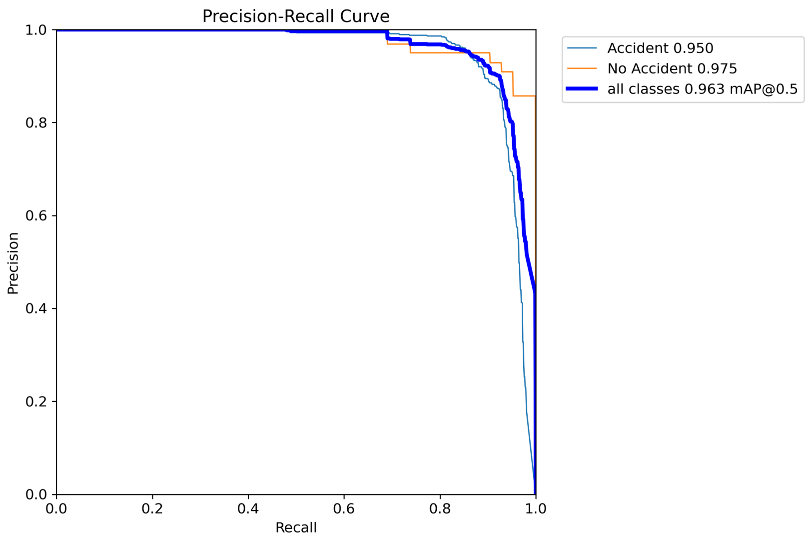

Figure 4.4: Model 2: 6000 images, 50 epochs (Precision-Recall Curve)

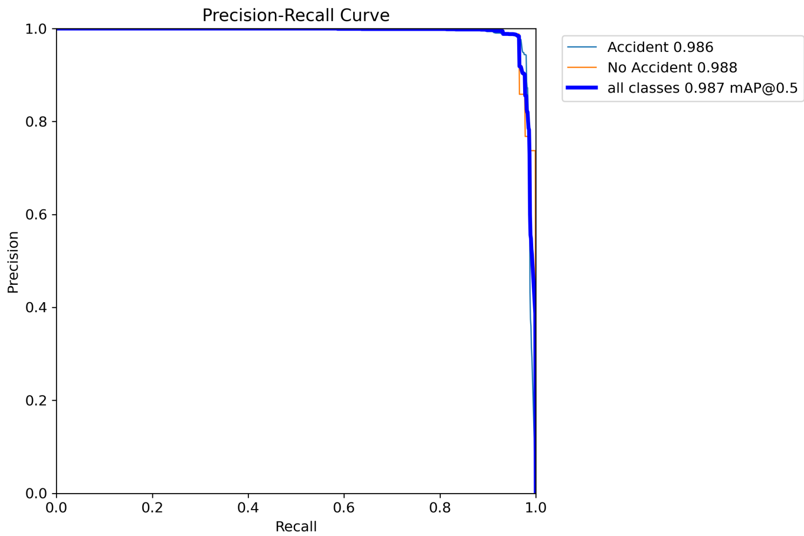

Figure 4.5: Model 3: 6000 images, 100 epochs (Precision-Recall Curve)

Figures 4.6, 4.7, and 4.8 compare the training and validation results of three YOLO-based accident detection models. Each figure shows the performance of a model trained with increasing dataset sizes and epochs. Model 3, trained with 6000 images over 100 epochs, demonstrates the highest precision and recall, as shown in Figure 4.8.

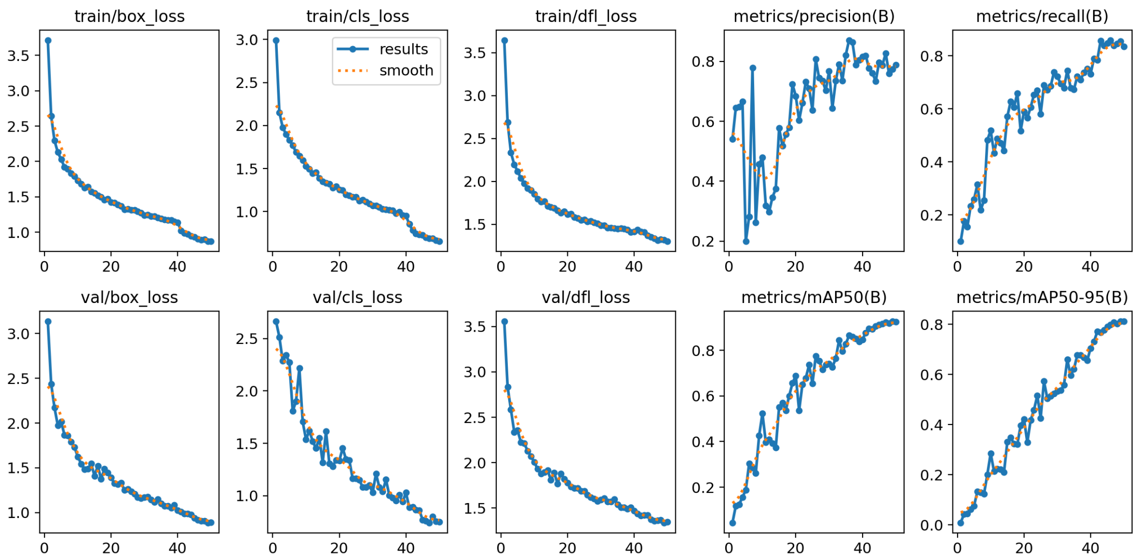

Figure 4.6: Model 1: 3000 images, 50 epochs (Training and Validation Results)

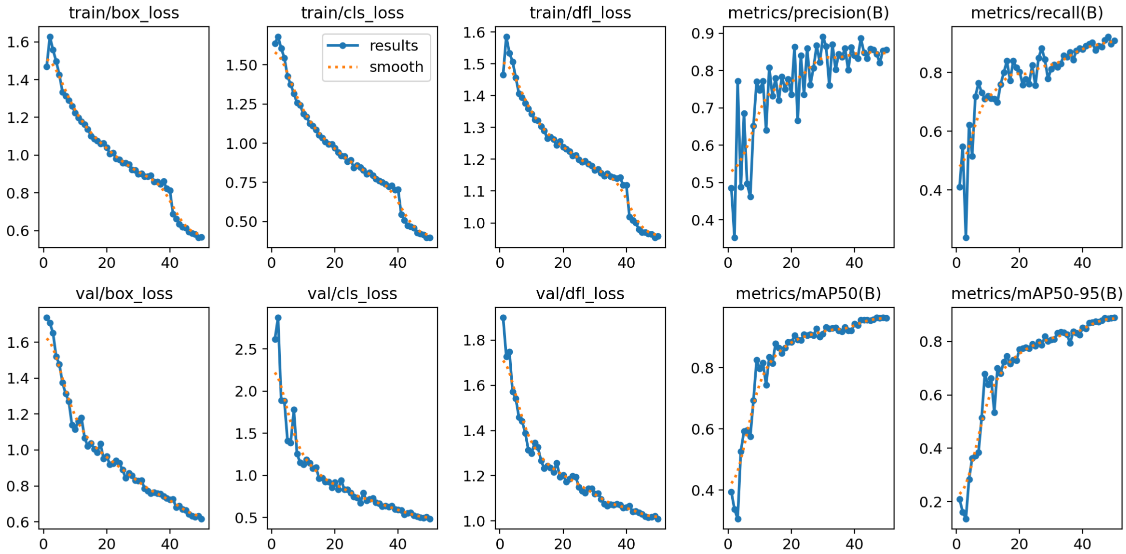

Figure 4.7: Model 2: 6000 images, 50 epochs (Training and Validation Results)

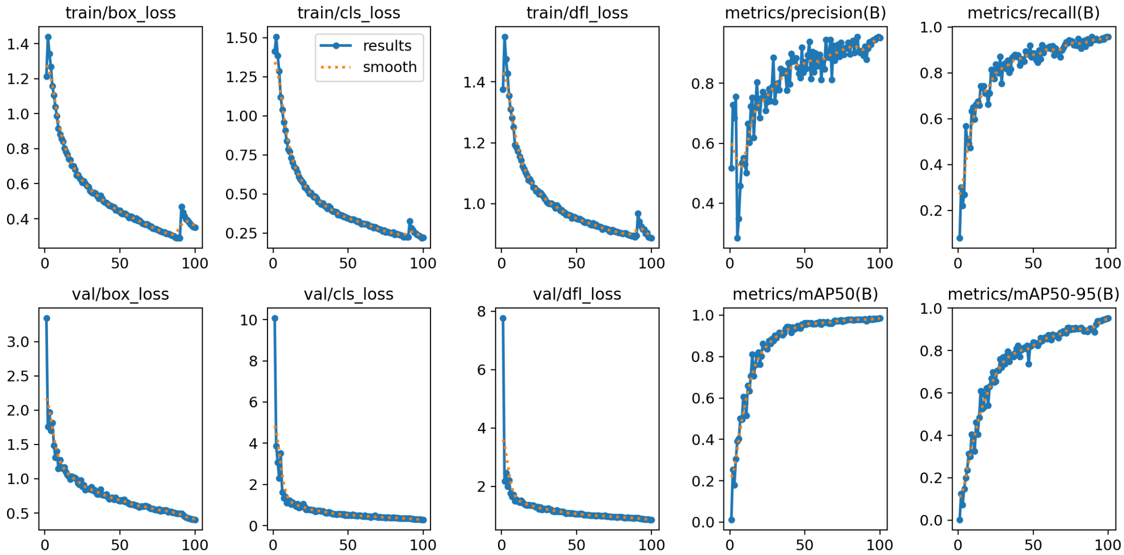

Figure 4.8: Model 3: 6000 images, 100 epochs (Training and Validation Results)

Figure 4.9 and 4.10 shows the user-facing mobile application enables individuals to request an ambulance and track its real-time location. The interface is designed to be intuitive, displaying both the request confirmation and live ambulance tracking on the map.

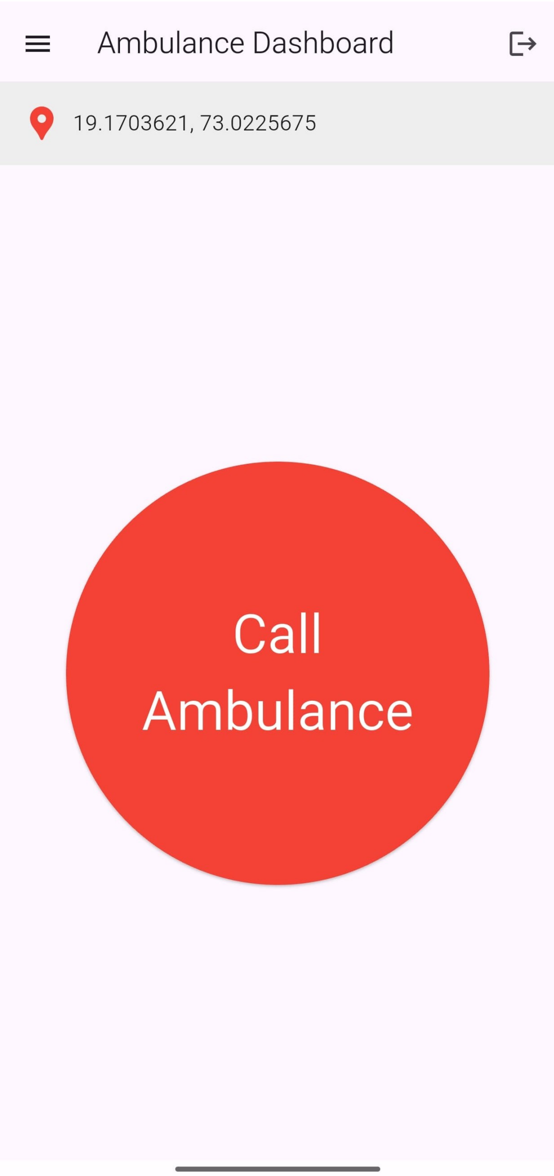

Figure 4.9: User App Interface (Calling Ambulance)

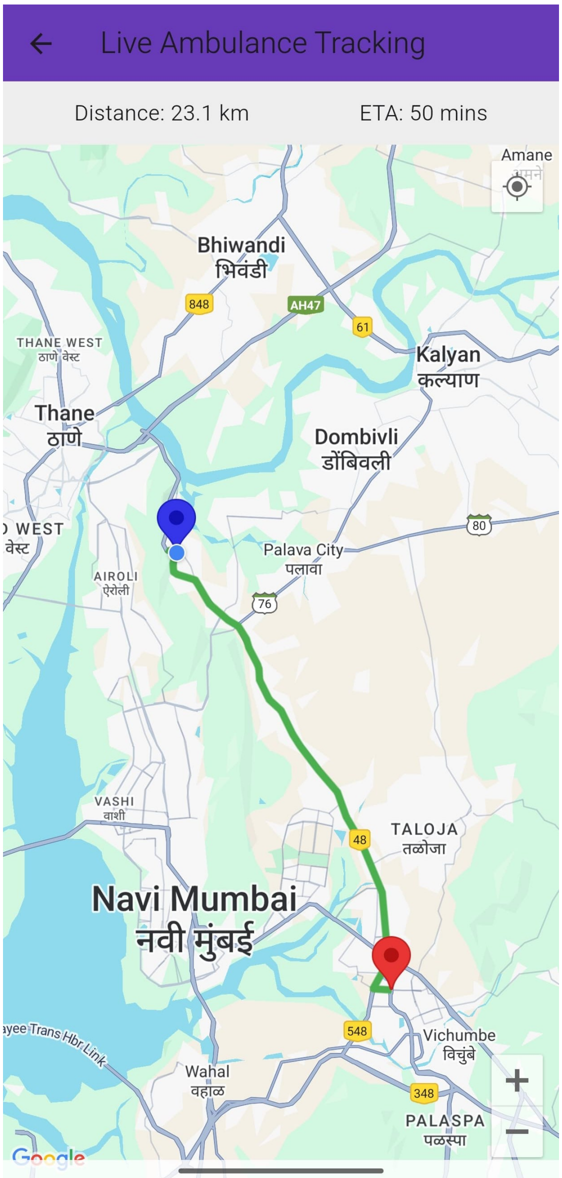

Figure 4.10: User App Interface (Ambulance Tracking)

Figure 4.11 and 4.12 shows the driver-side application assists ambulance drivers in navigating efficiently to the patient and then to the nearest hospital. It provides live navigation, patient details, and optimized routes, all integrated with Google Maps API.

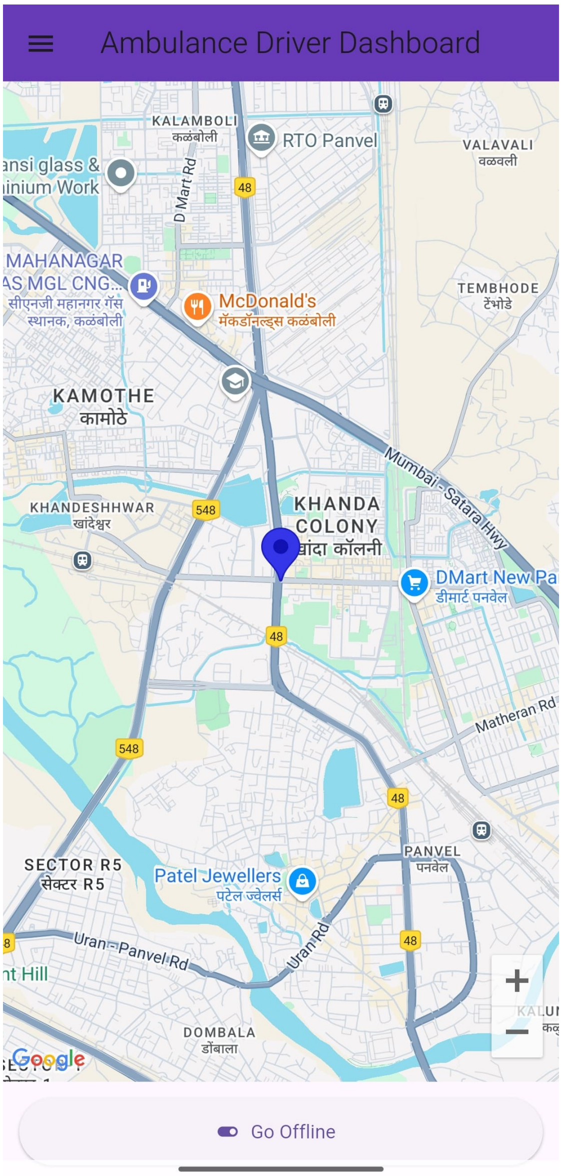

Figure 4.11: Driver App Interface (Live Navigation to Patient)

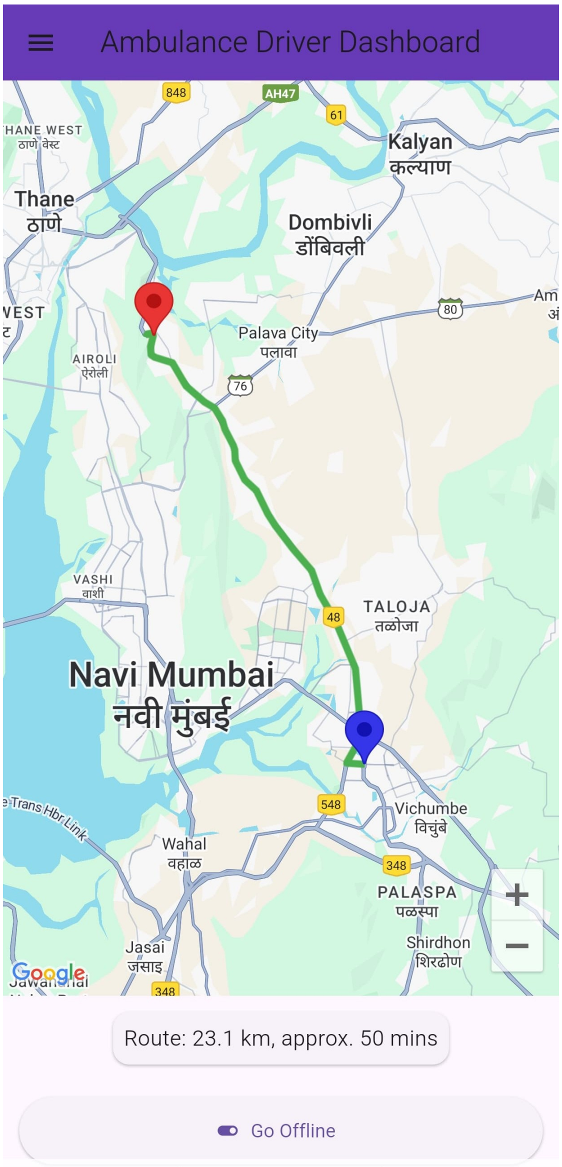

Figure 4.12: Driver App Interface (Live Navigation to Hospital)

Figure 4.13 shows the hospital dashboard allows healthcare staff to register drivers, monitor live ambulance locations, receive alerts for incoming ambulances, and review historical routes. This central management system improves coordination and reduces response times during emergencies.

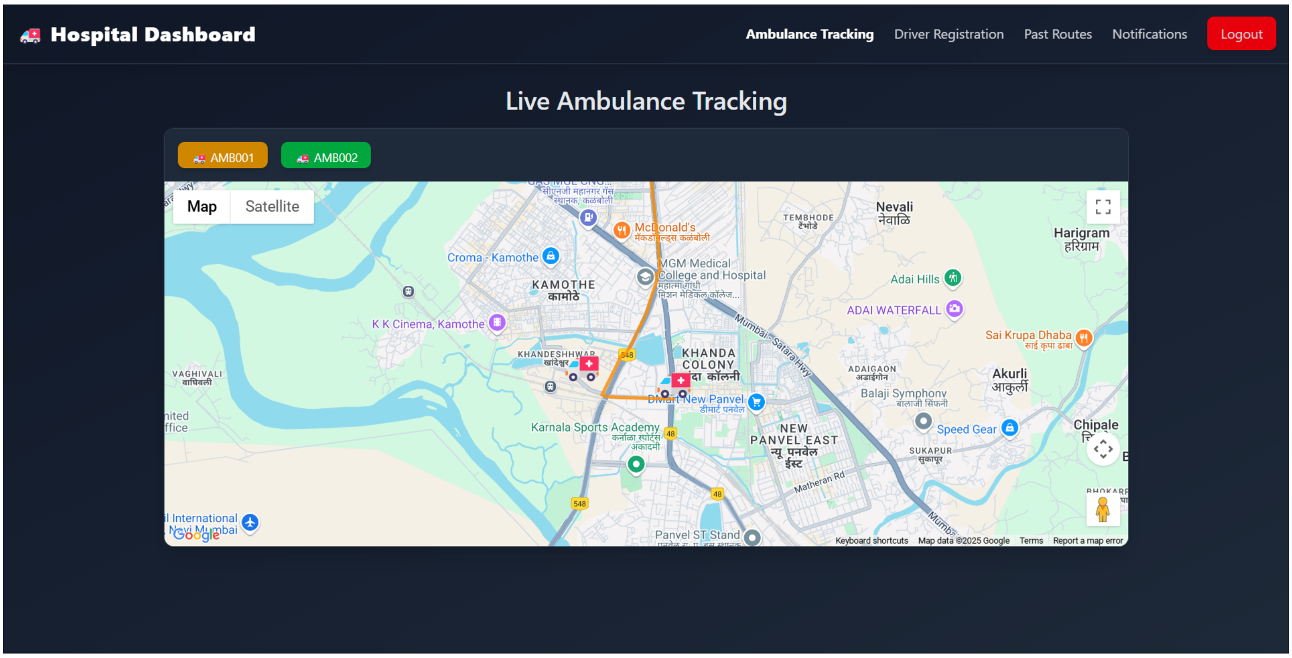

Figure 4.13: Hospital Dashboard for Ambulance Management (Live Ambulance Tracking)

Figure 4.14 shows the section of the hospital dashboard allows administrators to register new ambulance drivers by uploading essential details and documents. It ensures authenticated access and seamless integration with the real-time tracking system.

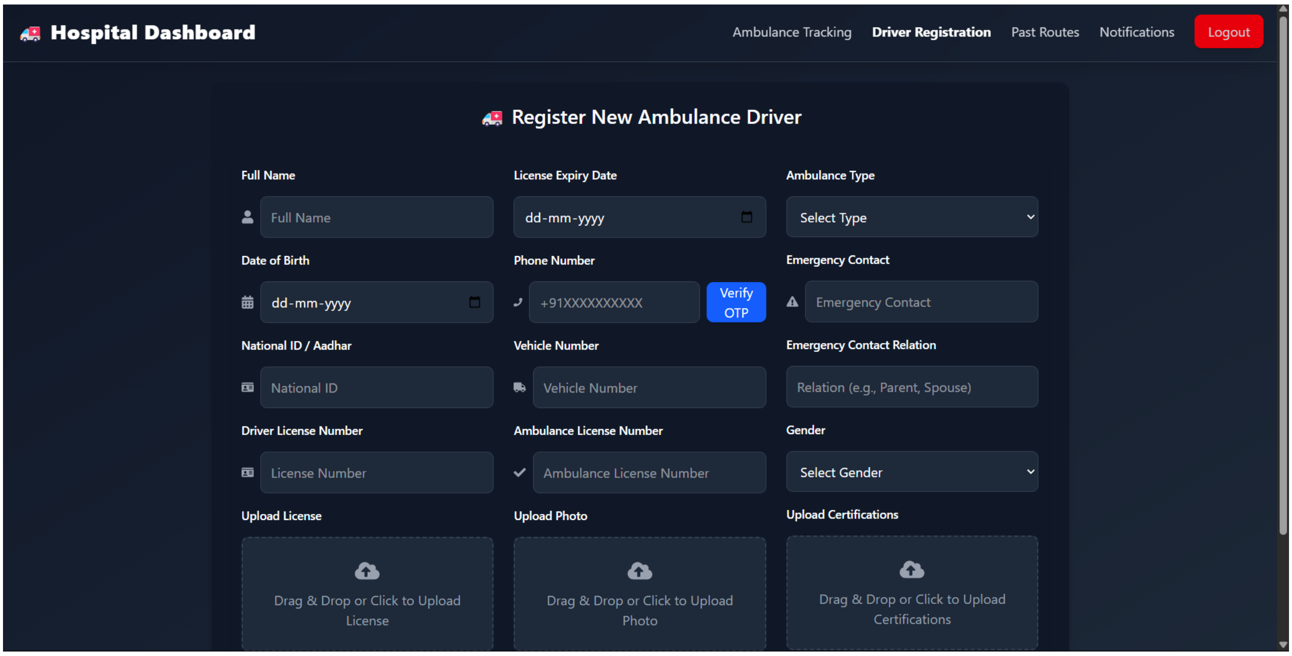

Figure 4.14: Hospital Dashboard for Ambulance Management (Driver Registration)

Chapter 5

Conclusion and Future Scope

Conclusion

The primary objective of this project was to develop a Smart Traffic Signal System that integrates real-time traffic data and emergency response prioritization using artificial intelligence. The goal was not only to optimize signal timings dynamically but also to facilitate the uninterrupted and efficient movement of emergency vehicles, especially ambulances. This purpose was successfully achieved through a seamless integration of multiple modules — vehicle detection and accident monitoring via YOLO models, real-time green signal timing using Webster's formula, and mobile applications for both users and ambulance drivers.

The system dynamically adapts to live traffic conditions, calculates optimal signal durations, and overrides normal traffic flow in emergency cases, ensuring that ambulances can reach patients and hospitals faster. A dedicated hospital dashboard adds a layer of operational transparency and control. With Firebase used for real-time communication and authentication across platforms, the system demonstrates robust performance in high-concurrency environments.

Key Learnings

- Application of computer vision techniques for real-time vehicle counting and accident detection.
- Implementation of Webster's Formula for traffic signal optimization in live

environments.

- Integration of multiple services (Node.js backend, Firebase, Flutter apps, React dashboard) into one synchronized ecosystem.
- Importance of real-time data synchronization and low-latency response in emergency systems.
- Challenges of GPS accuracy, latency in Firebase updates, and Flutter-Firebase integration.

Future Scope

- **Expansion to More Signals:** The current implementation can be scaled city-wide with ease due to its modular design and cloud-based architecture.
- **Data Analytics and Prediction:** Historical data can be used to predict peak traffic hours and prepare optimized preset cycles.
- **Edge Computing Deployment:** Shifting real-time inference from cloud to edge devices (like Jetson Nano or Raspberry Pi) can significantly reduce latency.
- **Accident Severity Detection:** Enhancing the accident detection model to assess accident severity can help dispatch proper emergency services automatically.

References

[1] Kumar, A., Singh, R.: Adaptive Traffic Control Systems: Case Studies of Siemens SCATS and IBM's Smarter Traffic. Int. J. Traffic Manage. **12**(3), 45–58 (2018).

[2] Chen, M., Zhang, Y.: Predictive Traffic Systems: Integrating Historical Data with Real-Time Analytics for Enhanced Urban Mobility. Smart Cities J. **5**(2), 89–101 (2019).

[3] Lee, D., Park, H., Kim, S.: Deep Learning Approaches for Real-Time Traffic Management and Accident Detection. In: Proceedings of the IEEE Conference on Intelligent Transportation Systems, pp. 234–240 (2020).

[4] Ghahremannezhad, H., Shi, H., Liu, C.: Real-Time Accident Detection in Traffic Surveillance Using Deep Learning. arXiv preprint arXiv:2208.06461 (2022). `https://arxiv.org/abs/2208.06461`

[5] Goenawan, C. R.: ASTM: Autonomous Smart Traffic Management System Using Artificial Intelligence CNN and LSTM. arXiv preprint arXiv:2410.10929 (2024). `https://arxiv.org/abs/2410.10929`

[6] Huang, X., He, P., Rangarajan, A., Ranka, S.: Intelligent Intersection: Two-Stream Convolutional Networks for Real-Time Near Accident Detection in Traffic Video. arXiv preprint arXiv:1901.01138 (2019). `https://arxiv.org/abs/1901.01138`

[7] Adewopo, V., Elsayed, N., Elsayed, Z., Ozer, M., Wangia-Anderson, V., Abdelgawad, A.: AI on the Road: A Comprehensive Analysis of Traffic Accidents and Accident Detection System in Smart Cities. arXiv preprint arXiv:2307.12128 (2023). `https://arxiv.org/abs/2307.12128`

[8] Alif, M. A. R.: YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems. arXiv preprint arXiv:2410.22898 (2024). https://arxiv.org/abs/2410.22898

[9] Thomas, J.: Road Accident Dataset. Roboflow Universe (2025). https://universe.roboflow.com/jocelyn-thomas/road-accident-5pzoq

[10] Webster, F. V.: Traffic Signal Settings. Road Research Technical Paper 39, Road Research Laboratory, Her Majesty's Stationery Office, London, UK (1958).https://www.sinaldetransito.com.br/artigos/traffic_signals_webster.pdf

[11] Google LLC.: Routes API — Overview & Guides. Google Developers (2025). https://developers.google.com/maps/documentation/routes/overview

[12] Firebase Team: Get realtime updates with Cloud Firestore. Firebase Documentation (2025). https://firebase.google.com/docs/firestore/query-data/listen

ACKNOWLEDGMENT

We take this opportunity to express our profound gratitude and deep regards to our guide **Dr. Tabassum A. Maktum** for her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We truly grateful to her efforts to improve our understanding towards various concepts and technical skills required in our project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark. We take this privilege to express our sincere thanks to **Dr. Ramjan A. Khatik**, Director, AIKTC and **Dr. Rajendra Magar**, Dean, SoET for providing the much necessary facilities. We are also thankful to **Prof. Tabrez Khan**, Head of Department of Computer Engineering and Project Coordinator **Prof. Nusrat Jahan**, Department of Computer Engineering, AIKTC, New Panvel, Navi Mumbai for their generous support. Last but not the least we would also like to thank all those who have directly or indirectly helped us in completion of this thesis.

Khan Mohammad Moin Abdul Moiz

Antuley Aman Siraj

Khalife Abdul Sami

Khan Mohd Irfan