Adaptive Traffic Signal Management

(Artificial Intelligence Project)

Session: 2022 – 2026

Submitted by:

Mian Muhammad Ashhad	2022-CS-114
Muhammad Abdul Barr	2022-CS-131

Supervised by:

Dr. Samyan Qayyum Wahla

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Table of Contents

Table of Figures

1. Introduction:

1.1. Background and Motivation:

The rapid increase in urbanization has led to an exponential increase in the number of vehicles on roads, resulting in difficulty in managing traffic on roads. Smart traffic solutions are becoming a necessity to manage congestion, reduce delays, and ensure smoother movement of vehicles, particularly emergency services.

The primary motivation behind the project is to improve traffic management at intersections. Traffic congestion often occurs due to the assignment of inappropriate signal timings where traffic is less, however, insufficient timing is given to the signals experiencing high vehicle density. This results in significant delays, increased fuel consumption, and unnecessary idling at intersections, which affects both the economy and the environment. By developing an accurate and efficient vehicle detection system, this project aims to contribute to smart city initiatives and provide a scalable solution to minimize wait times at critical junctions.

1.2. Problem Statement:

Current traffic management systems lack the ability to manage traffic flow at intersections. Signal timings are set by default, or overridden manually if a problem arises due to traffic overload for a specific way at an intersection. This leads to long waiting times at signals and traffic congestions, leading to economic, social, and environmental problems.

Therefore, a robust, real-time solution capable of processing variable traffic density at intersections and setting appropriate green signal timings to avoid starvation for other ends is needed. Also, the solution must be scalable, to be easily adaptable to real-world changing road infrastructure and traffic management requirements.

1.3. Objectives:

1) Minimize congestion and improve traffic flow by dynamically managing signal timings using real-time vehicle density data.
2) Decrease fuel consumption and vehicle emissions by reducing idle time at traffic intersections.
3) Provide a future-ready system capable of integrating with smart city infrastructures and evolving traffic management technologies.

1.4. Scope of Report:

This report outlines the design and development of a vehicle detection and classification system. It covers data collection, preprocessing, model training, and evaluation. The report also discusses system architecture, implementation challenges, and future improvements.

2. Literature Review:

Traffic congestion has been a problem no matter the place or country. Manual override of signal timing is a hassle that would require the availability of a human resource at all times 24/7 at each intersection. This is not a feasible solution which was recognized by many individuals and organizations.

A wide-scale implementation of the smart traffic system is implemented and working by “Rexgen” in Korea called ITS (Intelligent Traffic System) [1] . It’s a wide-scale project for surveillance of traffic. It is also integrated with AI to detect traffic density and set signal timings. However, it works on special sensors that are too expensive to be implemented in a developing country such as Pakistan.

Another system was implemented in Ohau, Hawaii, US [2]. Due to long wait times and traffic congestion, the system was implemented at various traffic intersections to allow smooth traffic flow. However, the system lacked the scalability option. This means the system was too strict to adapt to dynamic infrastructure and road development.

In Pakistan, a system was proposed by a startup called “InLights”, by students of NUST [3]. They worked on a similar principle of using camera feeds at intersections to gather data and set appropriate green signal timings. However, the solution provided was not easily scalable to adapt to changing city roads and infrastructure development. Manual requests to InLights to change the settings.

Our proposed system provides adaptive AI-based traffic solutions real-world maps data and A CRUD system that allows traffic authorities to easily adapt to respective needs. Moreover, our system focuses more on decentralized management rather than centralized, making sure that decisions are adopted as soon as possible and with ease.

3. Methodology:

3.1. Model Designing:

Since our system required to recognizing different types of vehicles on the road, a AI-model was required to process the traffic at intersections.

3.1.1. Data Gathering:

Data was gathered from various sources. A basic vehicle dataset, including cars, trucks, motorbikes, and buses, was sourced from the COCO dataset. This collection features over 2,000 labeled instances of these vehicles, offering a diverse range of examples. Additionally, three-wheeler data, consisting of 400 labeled instances, was gathered from Kaggle to enhance the dataset's variety.

For emergency vehicles, data was sourced from Roboflow, contributing over 1,300 labeled examples. This dataset is rich in different types of emergency vehicles, ensuring it is well-suited for specialized tasks like detection and classification.

Real-world road data was included using Dataninja's Bangladesh Dataset, which provided more than 800 labelled instances. This dataset focuses on authentic road environments, offering valuable insights for applications such as traffic analysis and navigation systems.

3.1.2. Organizing and Cleaning:

Since the data was too diverse and collected from different sources. It had to be organized and cleaned. First of all, all the labels were redefined from different formats (JSON, txt, etc.) to a single scale of txt with the same pattern.

Instances that were not required were dropped and removed. More over blurry and too complex images that might produce noise in training were removed.

3.1.3. Data Imbalance:

Due to Data being gathered from different resources, the dataset combined was imbalanced, containing cars, trucks, and bikes with about 2000+ instances making them in the majority while emergency vehicles and three-wheelers with 1300+ and 400+ instances respectively making them in the minority.

3.1.4. Handling Class Imbalance using SMOTE:

The imbalanced classes can become hurdle in training of model. They can cause vehicles to be unrecognised and result in in accurate results.

This imbalancement was handled using SMOTE (Synthetic Minority Over-sampling Technique). The Minority class of three-wheeler was increased by further images and majority class was reduced to match the minority class. A mean value of about 800 instances in total was chosen for each class.

3.1.5. Choosing and Training the Model:

Ultralytics' YOLOv8 model was chosen for the our system's AI-model. YOLO (You Look Only Once) is used because of it's speed in real time scenarios.

The model was trained based on hyperparameters of 50 epochs, a batch size 4, and an image size of 614. The initial learning rate was set to 0.01, with momentum at 0.937 and weight decay at 0.0005. Early stopping was configured with a patience of 100, and deterministic settings with a seed of 0 ensured reproducibility. Pretrained weights were utilized, and validation was performed on the split defined in the dataset.

3.2. Live Feed Monitoring:

Live Feed is transferred directly from the intersection to the Application. This enables the user of system to monitor the signal activities. For the current system, due to the limitation of prototyping, IP address based camera feed was used. However, for the final product this can be upgraded to inline-based camera feeds to ensure stability and speed.

3.3. Vehicle Detection and Setting Timings:

The Feed from each signal is transferred to the model which detects the number of vehicles inside the frame and returns the count of each vehicle present at that moment.

The system then calculates the count of each vehicle and based on the vehicle count category wise. Based on average 5 seconds for each car, bike, three-wheeler, van and suv and 7 seconds for each truck. However, this may vary based on the traffic density at other signals lanes as well by reducing time ensuring none of the lane is starved.

3.4. User Interaction:

3.4.1. Selective Intersection:

The administrator station creates the intersection on the live real-world map and assigns it to the respective stations along with editing (if needed).

The respective station selects the required intersection from its assigned intersections right from the real-world map. He can then modify the signal preferences according to needs, as per required by the conditions.

3.4.2. Configuring Area for Detection:

The station configures the area for each signal available at the respective intersection. This allows the model to more effectively and accurately determine the number of vehicles.

3.4.3. Verify The Configured Area:

The station user verifies that if the coordinates are set correctly or not. If not set correctly, he/she can reconfigure it.

3.4.4. Monitoring The Signal:

The station user can now monitor the respective intersection and keep track of how much each signal was set for with records.

3.5. Fault Protections and Override:

If due to any issue camera feed stops working or an error is detected in the system, the system reverts to default mode setting each signal at 20 sec timer and alerting the station that an error has been detected at the respective intersection.

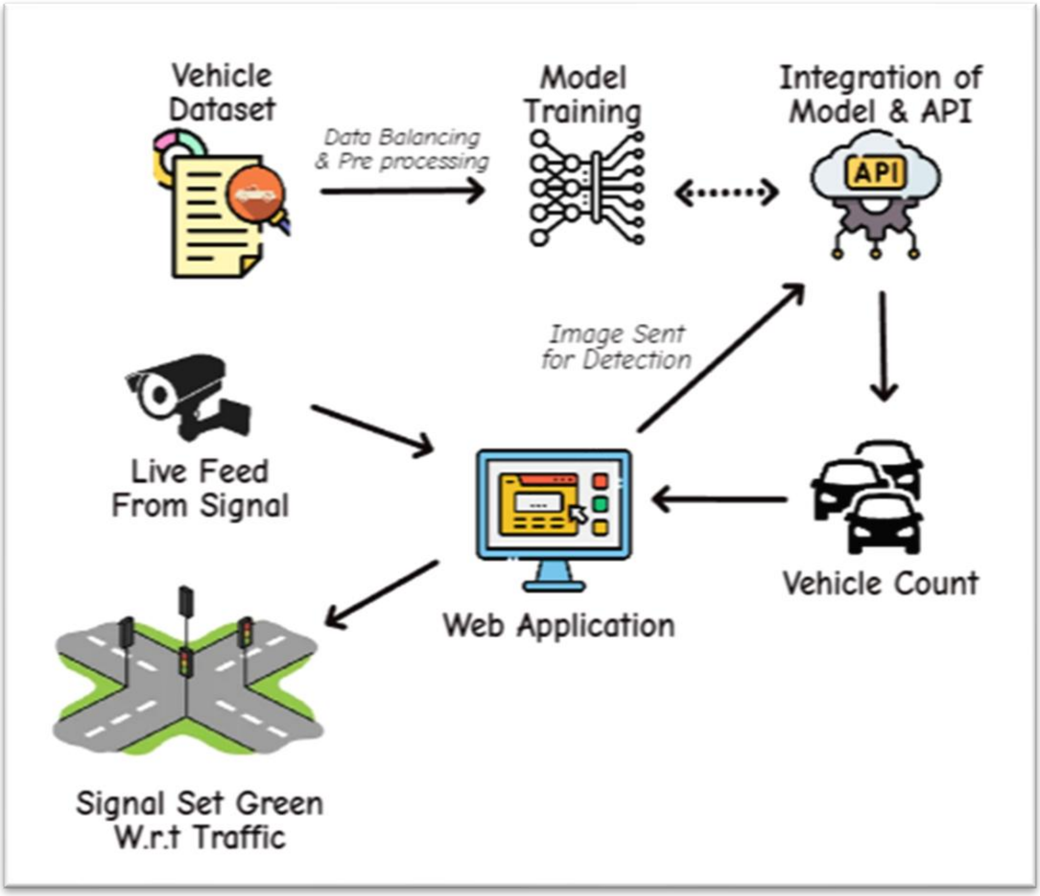
Moreover, due to any special reasons or conditions customs timings are needed to be set. The station can override the smart signal system to manual override, by setting custom timings for each signal or choosing from the preset.

4. System Design and Architecture:

4.1. System Overview:

The AI-based Adaptive Traffic Signal Management system, by integrating real-time video feeds from cameras positioned at intersections, can detect and count vehicles, classify them by type (e.g., cars, buses, motorcycles) using YOLO object detection algorithm, and adjust traffic signals based on the observed density and priority rules. The use of OpenStreetMaps allows for precise real-world mapping of signals and road layouts, ensuring that the system aligns with actual traffic infrastructure.

A web-application based system is designed so that it's easily accessible and helpful in achieving the decentralized approach of our system.



4.2. System Architecture:

The system is divided into two components, the front end of the application that displays the user interface of the system to the admin or station user meanwhile the backend that handles the database and the API-integrated vehicle detection AI model.

4.2.1. Frontend:

The frontend is developed using React and TailwindCSS. React makes it easier to create dynamic web applications because it requires less coding and offers more functionality. React uses Virtual DOM, thereby creating the web-based application faster.

4.2.2. Backend:

Backend of application is made using Node and Express.js framework. This allows our system to be scalable, easy to integrate new features and improve time efficiency.

4.2.3. Database:

MongoDB was used for the database of our traffic system. MongoDB provides the flexibility, scalability, and performance required especially in our rapidly changing environment. Currently, the images are stored locally, due to the limitations of prototype system, however, this image handling can easily be integrated with firebase storage in the final product for deployment.

4.2.4. AI-Model Integration:

Our YOLO, the AI-based Model, is integrated with the system using FastAPI. FastAPI allows up to integrate our python environment to our adaptive traffic management system.

4.2.5. Post Processing After Vehicle Detect:

After processing the feed, and determining vehicle count, OpenCV is used to map the labels provided by the model onto images, this allows us to keep record of the actions made and check achieves if needed.

4.3.Database design:

The Database is created using MongoDB however, a relational schema below can be interpreted to understand the database structure of the system.

4.4. User Characteristics:

The System caters two users, i.e. Admin and Station.

4.4.1. Administrator:

The Administrator is a singular role within the system, assigned to a single individual situated at the city's head office. The Administrator oversees the overall configuration and operational management of all user stations and traffic signals across the city. Key responsibilities for the Administrator include:

- CRUD Operations for Local Stations and for signal intersections in city
- Application Log Monitoring
- Signal responsibility assignment to stations.

4.4.2. Station:

Station Operators are responsible for the daily management and configuration of traffic signals within their designated areas. Multiple Station Operators may be present across the city, each with access to the signals in their specific area. Their responsibilities include:

- Signal Configuration of its respective area's intersections
- Traffic Signal logs of its respective area's intersections

4.5. Scalability using OpenStreetMaps:

OpenStreet maps are integrated into the system. This allows our system to modify the intersections and signals with respect to changing road and city infrastructure. Also, the intersection data is handled in a private environment of the system in the database. This ensures that open contributions from the public do not affect our system.

4.6. Use Cases:

4.6.1. Administrator Level:

1) Intersection Management:
    - Add New Intersection (UC1)
    - Delete Existing Intersection (UC2)
    - Edit Existing Intersection (UC3)
2) Station Management:
    - Add New Station (UC4)
    - Remove Existing Station (UC5)
    - Edit Existing Station (UC6)
    - Assign Intersection to Station. (UC7)
3) Application Management:
    - View Changes Made to the Application. (UC8)

4.6.2. Station Level:

1) Signal Management:
    - Override The Signal (UC9)
    - Configure Signal detection zone (UC10)
2) Monitoring Intersection:
    - Monitor live Feeds from the Intersection (UC11)
    - View Logs for Changed Signal Timings. (UC12)
    - View Snapshots for detected vehicles (UC13)

5. System Role in Sustainable Development Goals of Pakistan:

The project helps in achieving:

5.1. Industry, Innovation and Infrastructure

The project aims to use AI technology to improve urban infrastructure which, in this case, is traffic management. The system reduces traffic congestion which is a major infrastructure-related issue faced urban areas.

5.2. Sustainable Cities and Communities

The project aims to make cities more sustainable and improve quality of life by improving the flow of traffic. The project helps authorities to timely respond to emergencies and reducing emissions by reducing traffic congestion.

5.3. Responsible Consumption and Production

The main problem solved by this system, traffic congestion, is a big contributor to fuel wastage and environmental degradation. Vehicles stuck in traffic jam not only consume more fuel but also release more greenhouse gases. By reducing traffic blockage, the system helps resolve these issues.

6. Machine Learning Approach:

6.1. Mapping ML to Business Level:

6.1.1. Baseline:

Already existing solutions exist such as [1,2,3] but they are either expensive to implement or not easily scalable with changing road and city infrastructure.

Our system provides a simple solution that can be easily integrated with already existing signal infrastructure by utilizing the already installed cameras on each intersections.

6.1.2. Usefulness Threshold:

Our System don’t require high-level human scale strict performance. The failback system in system helps us counter unseen circumstances and reverts to default timings, and alerting authorities.

6.1.3. False Negatives and False Positives:

Our System is designed in such a way that false positives and false negatives don’t impact much on the environment and it’s objects. In case of no vehicle/ false negative, the signal still follows the default 15 sec timer, this ensures the system keeps running reliably. In case of false positive, only a few seconds of timer is extra added.

6.1.4. Interpretability:

The system is designed to be simple and easily interpretable to general audience, Even for a simple person, the System provides hints and popups to help interpret the functionality.

6.1.5. Confidence Measurement:

The system doesn't require high level of confidence because the signal timings are based on the density of vehicles, not by their category. Therefore, Detection of objection is required but false classification of vehicle doesn't affect.

Moreover, the set detection area provides a boost in confidence for detection of vehicle density at the desired signal of intersection, and not detecting side objects away from the signals but in the image.

Each log is added into system, that can be reviewed and appropriate fine tuning to the model can be requested to be made.

6.1.6. Feasibility of System:

The System will utilize the already installed camera systems at intersections so implementation requires little cost. The system requires high speed internet to operate properly. However, if feed is delayed or any error in the work flow happens, the system will revert the intersection to default signal timings. Thus, providing a feasible solution.

The System will prove to reduce the human cost, proving to be the middle man of traffic operations and will be working 24/7 whatever the situation or weather conditions. Hence, proving to be a useful and effective solution.

Also, the adaptive timings to ensure traffic congestion do not occurs and waiting time is reduced, the overall mood of public will improve, effect to fuel consumption savings and reduce accidents due to signal breaks.

6.2. Data Specification:

The Data consisted of over 1300 images with 800 plus instances of car, three-wheeler, rickshaw, bus, motorbike, truck, minivan and suv. The data was previously pre-processed and cleaned. The data also consisted of real world images to fine tune model according to road traffic.

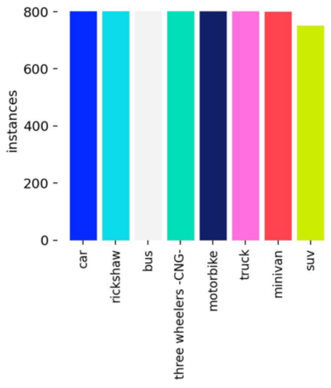


Figure 1- Classes Instances

6.3. Evaluation Metrics:

The model was evaluated using precision, recall and F1-score. These evaluation metrics provided a detailed overview of the model's performance. A confusion matrix was also used to understand the confidence in classification correction.

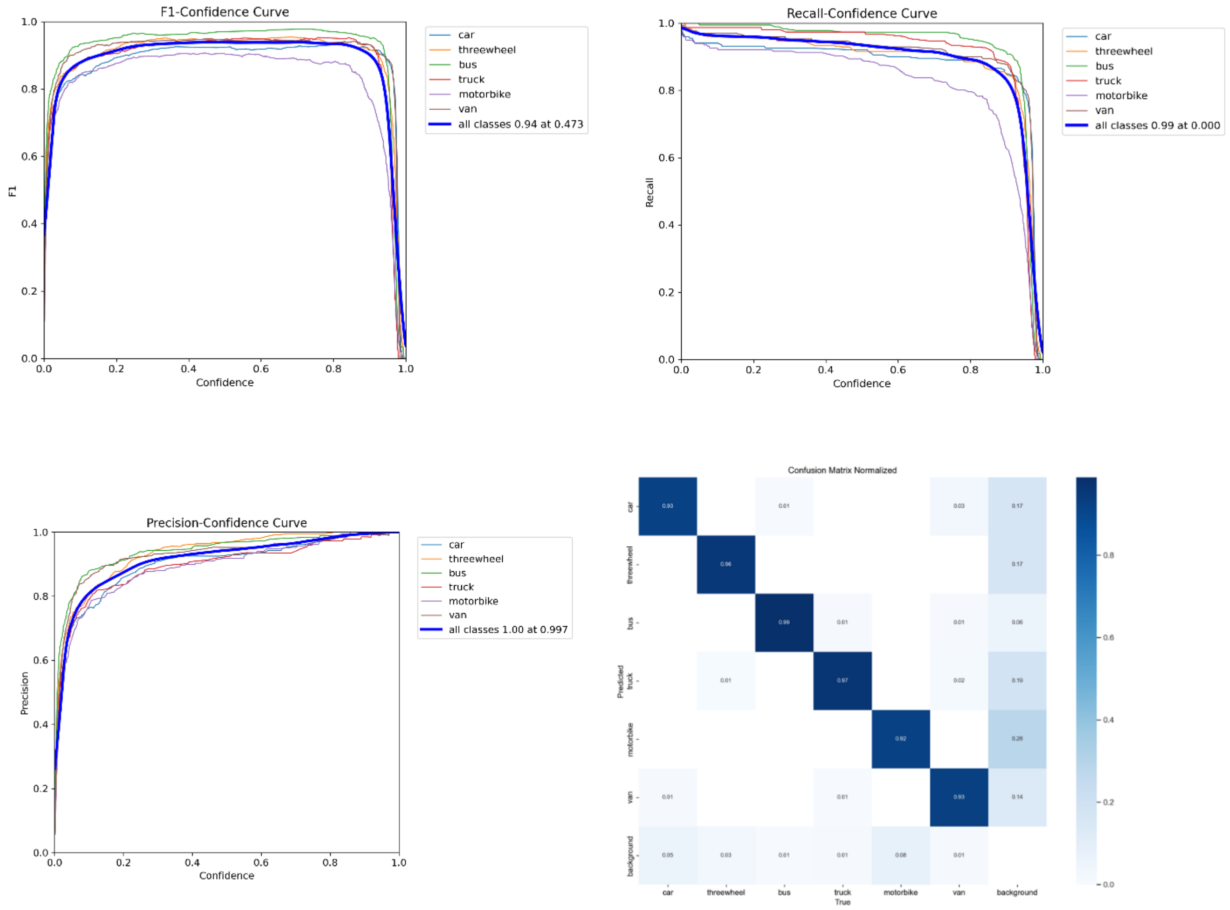


Figure 2- Evaluation of Model using Precision, Recall and F1 Score along with Confusion matrix

Th system was able to give 82% accuracy in detection, which is good considering the data provided. However, the model can be further fine tuned to provide even better results.

8. Implementation:

8.1. Component Connections:

All the different components are connected together to form a unified system. MongoDB handles the database of the system, storing signal info, credentials, intersection info and logs. Express handles and connect this to the frontend which is made in React and Tailwind CSS.

The model is integrated into Flask and Fast API is used to integrate the model with the web application.

Live feed from camera is transferred using IP address (for the current prototype). For future purposes, we can adapt to wired camera connection.

8.2. Challenges Faced:

The integration of camera feed proved to be a difficult task and making it responsive and fast to make the feed smooth. This was tackled using private connections based on Java protocol to handle rather than html2canvas protocol.

The configuration of signal for model to detect vehicles in specific area was a challenging task. This was necessary to improve confidence and accuracy of the system and make sure it was feasible to adopt. This was overcame by power of python and OpenCV.

Moreover, smooth and collision free transfer of data between different layers of system was a great hurdle in development. This was catered using various techniques, firebase, camera endpoints, timers and async function handling, etc.

8.3. Wireframes:

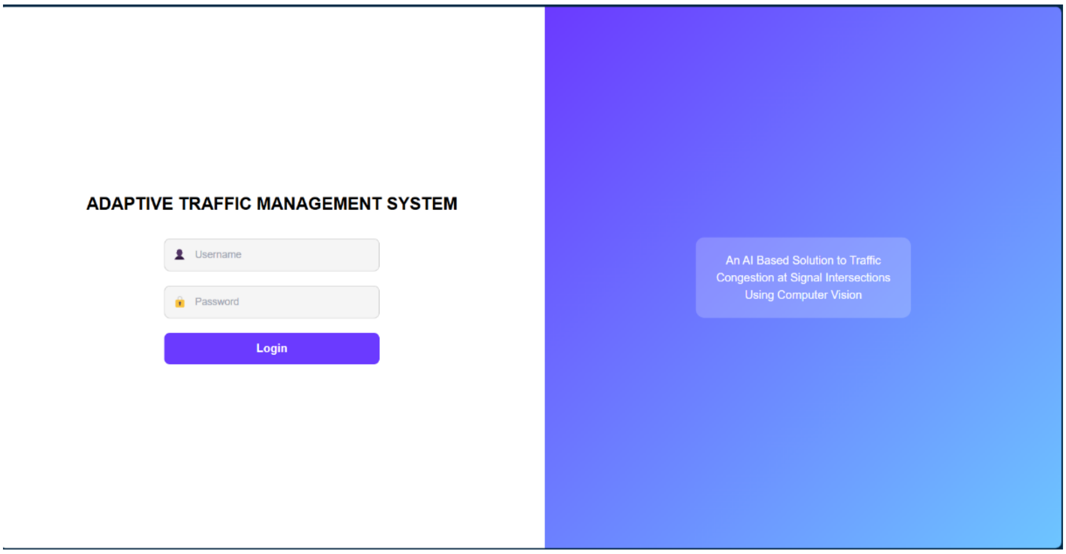


Figure 3- Login Page

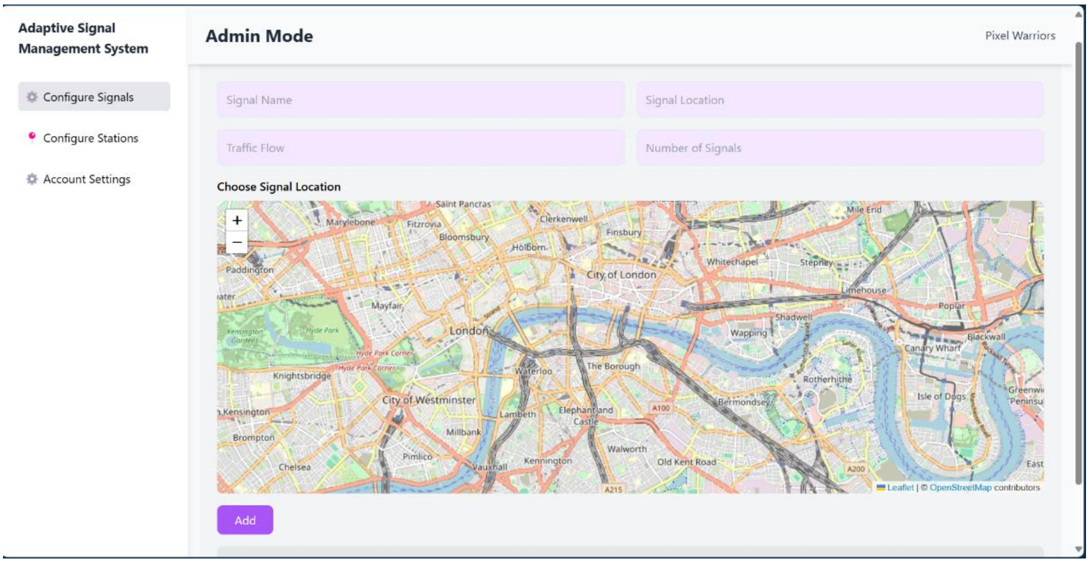


Figure 4- Adding New Intersection

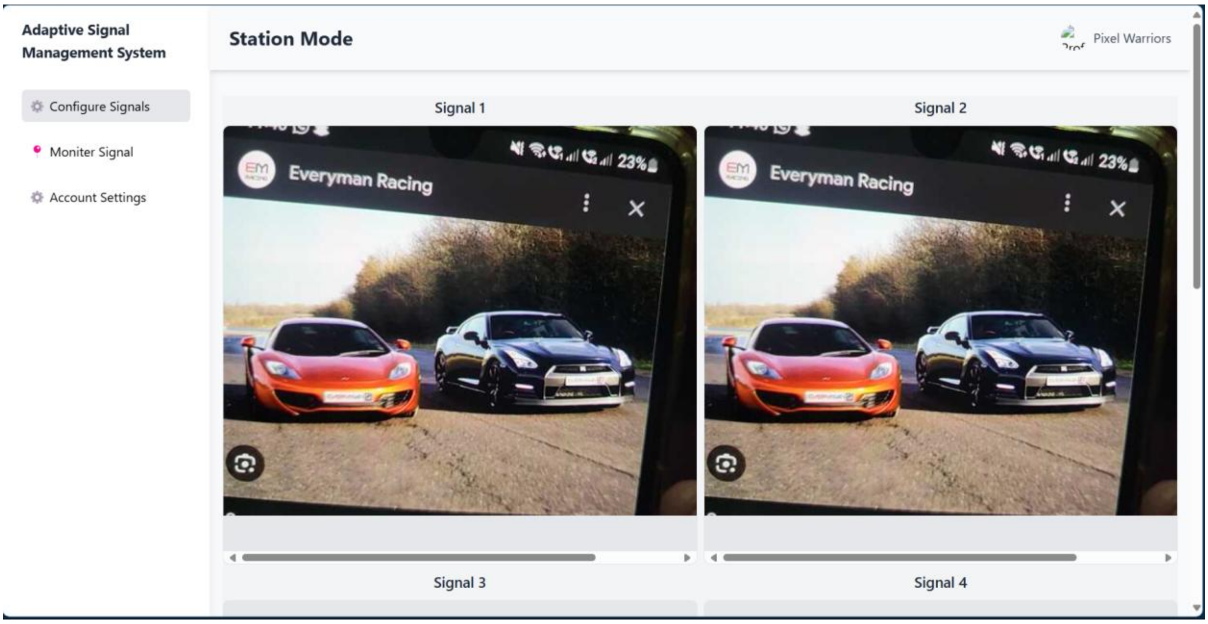


Figure 5- Live Feeds from the Intersection

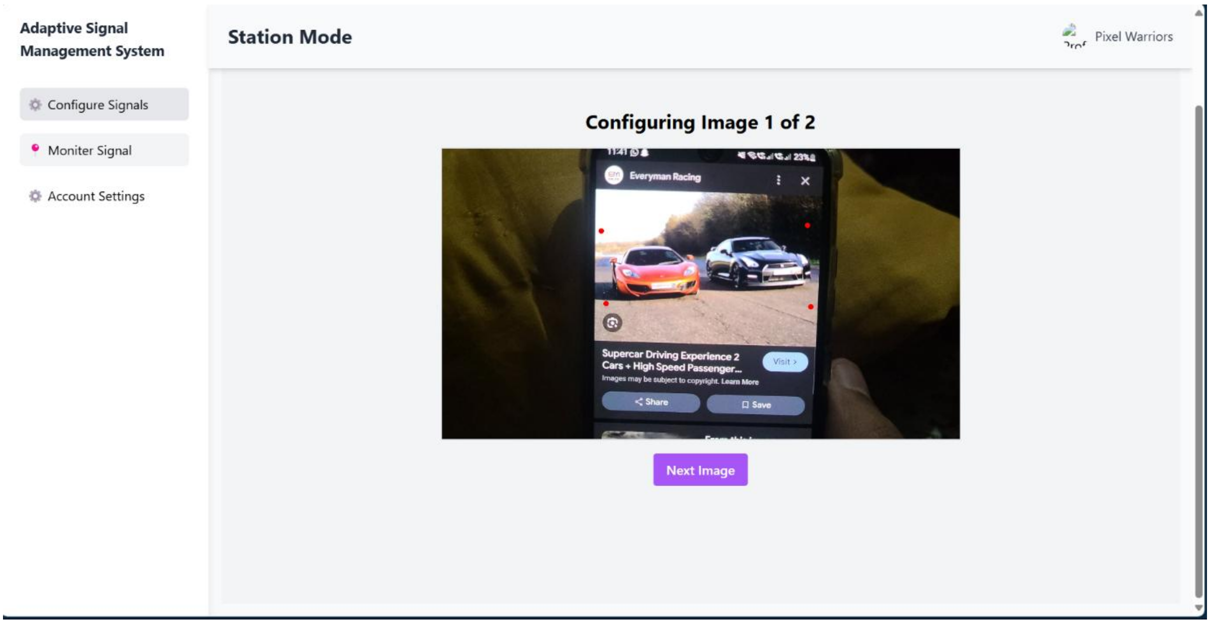


Figure 6- Configuration of Detection Area

9. Conclusion:

9.1. Summary:

Adaptive traffic management system provides a robust & scalable system, that not only feasible to implement but also provides a great platform for decentralized management of traffic at intersections in Pakistan. The system proves to be reasonable when mapped according to business logic and implementation. Real-time feeds and integration of AI-Based model are expected to improve public mood, reduce average wait time at intersections, avoid starvation of signal lane, improve traffic flow and reduce the risk of accidents due to breaking of traffic lights.

9.2. Future Insights:

Since this system was a prototype of the original real-scale implementation. A few shortcoming were faced which could be improved.

The system currently uses local system to store images which can be upgraded to firebase or cloud solution. This could result in fast delivery between different layers of system.

The current live feed system is based on IP web cam. This can be later upgraded to fiber optic based wired connection to improve reliability and speedy transfers of live feeds.

10. References:

1) Intelligent Transport Systems (ITS) Korea, "Overview of ITS Korea," *ITS Korea*. [Online]. Available: https://www.its.go.kr/Eng/overview/overview01.html.
2) KHON2 News, "Smart traffic signals: part of the state's plan to modernize roads," *YouTube*. [Online]. Available: https://www.youtube.com/watch?v=frzlvECik5I.
3) Inlights Pakistan, "Islamabad green signals: AI-based traffic management solution by Inlights Pakistan," *NBS Digital*, Mar. 17, 2023. [Online]. Available: https://nbsdtps.wordpress.com/2023/03/17/islamabad-green-signals-ai-based-traffic-management-solution-by-inlights-pakistan.