Smart Traffic Control

Smarter signals, smoother streets.

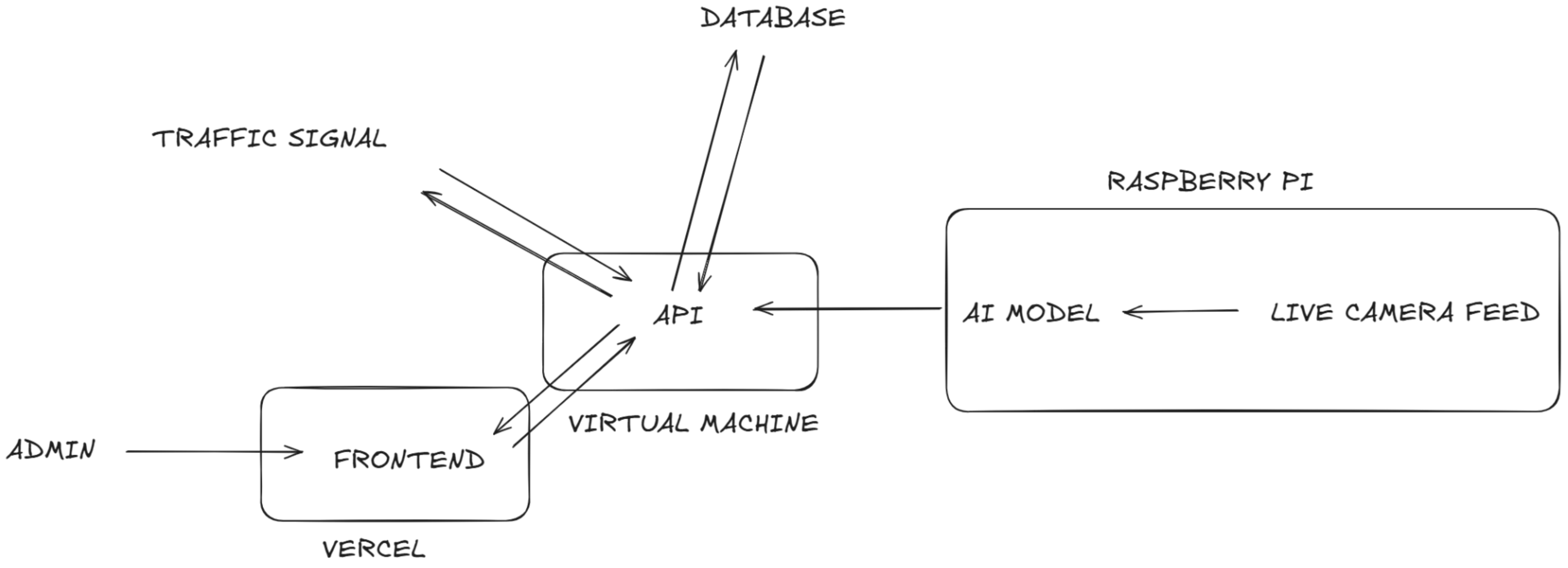
Introduction

Problem Statement: Advanced Traffic Control System for Urban Road Network

Problem: Urban traffic management needs adaptive solutions to optimize flow and reduce environmental impact.

Solution: The Advanced Traffic Control System is a smart solution for urban traffic management. It integrates real-time traffic monitoring via cameras, AI-powered predictive modeling, and a web interface to dynamically adjust traffic signal timings, optimizing flow and minimizing delays. Object detection models (YOLOv11) and predictive models (PyTorch) analyze data, anticipate congestion, and detect accidents. The dashboard provides traffic authorities with insights, including real-time notifications, heatmaps, and manual override. The system prioritizes emergency vehicles and public transport. Built for scalability and sustainability, it incorporates modular intersections and renewable energy. The goal is to reduce wait times, enhance safety, minimize environmental impact, and create a resilient system. Utilizes React, Next.js, Node.js, and Vercel, with DevOps practices via Docker.

Workflow Diagram



Features

1. Traffic Monitoring and Data Collection
2. AI-Powered Signal Control
3. Interactive Admin Dashboard
4. Traffic Prioritization
5. Scalable and Sustainable Design

Expected Outcome

- **Improved Traffic Flow**: Reduced wait times and congestion, leading to faster commutes.
- **Enhanced Safety**: Early accident detection and prompt notifications.
- **Environmental Benefits**: Reduced emissions due to optimized signal timings.
- **Scalability**: A modular design that can be implemented across various urban settings.

Challenges Expected

1. **Electronics**: Ensuring accurate calibration and durability of sensors and cameras in varying conditions.

2. **Integration**: Seamlessly combining hardware and software components while debugging microcontroller setups.

3. **AI Model Accuracy**: Training models to handle diverse weather, lighting, and traffic scenarios.

4. **Scalability**: Ensuring the system performs efficiently as new intersections are added.

5. **Cybersecurity**: Protecting the system from unauthorized access and data breaches.

Technologies Used:

1. Hardware:

• Microcontrollers: Raspberry Pi, ESP-32

• Electronics: Resistors, capacitors, LEDs, transistors, breadboards, wires

• Sensors: Cameras

Technologies Used:

2. Software and Frameworks:

- Frontend: React, Next.js
- Backend: Node.js
- Cloud Hosting: Vercel
- DevOps: Docker

Technologies Used:

3. AI and Machine Learning:

- Object Detection Models: YOLOv11, OpenCV
- Predictive Models: PyTorch
- Data Handling: Pandas, NumPy
- Visualization: Matplotlib, Plotly

Artificial Intelligence and Machine Learning

- **Intelligent Vehicle Detection:** Real-time identification and classification of cars, ambulances, and school buses using YOLO-based object detection.
- **Dynamic Traffic Signal Optimization:** AI algorithm optimizes traffic light timing based on real-time congestion data and vehicle type, prioritizing emergency vehicles and school buses during peak hours.
- **Accident Detection & Alert System:** Automated accident detection with immediate alerts, enabling rapid response and incident management.
- **Data-Driven Congestion Mitigation:** Adaptive traffic signal adjustments informed by continuous data analysis, minimizing overall congestion and improving traffic flow.

Application Programming Interface (API)

- **Interfacing AI model with frontend and hardware:** The API continuously receives the state of all traffic signals from the AI model which is stored in a database. The data is frequently updated if a new version is posted by the AI model and both the frontend application and hardware can request the data whenever needed.

- **Admin UI:** The frontend is used by admins to access the database and in addition to viewing the traffic signals state they can view accident reports, manually modify signal state, etc which is all stored and served by the API.

Frontend

Utilizes a dynamic frontend - Next.js and React

1. Real-time user-readable traffic data from API

2. Interactive visualizations of traffic flow, congestion levels, and incident reports.

3. Promotes informed decisions; smoother journeys

Electronics

An AI-powered advanced traffic management system integrates several electronic components for efficient traffic flow. Cameras monitor traffic conditions, sending data to an intelligent AI box that uses deep learning models to optimize traffic management. IoT sensors, embedded in roads, collect data on vehicle counts, speed, and traffic density. Microcontrollers, such as Raspberry Pi, process data and control traffic signals. Additional components include resistors, capacitors, LEDs, transistors, breadboards, and wires. The AI box communicates optimized plans to neighboring intersections via the cloud.

Conclusion

Thus, our AI-powered traffic management system utilizes a robust network of electronic components – from IoT sensors and cameras to microcontrollers like Raspberry Pi – for comprehensive data collection. Paired with a dynamic Next.js/React frontend, the system processes and visualizes real-time traffic data, offering actionable insights. By intelligently analyzing traffic patterns and providing users with clear, informative displays, the system optimizes traffic flow, reduces congestion, and improves the overall commuting experience for everyone.