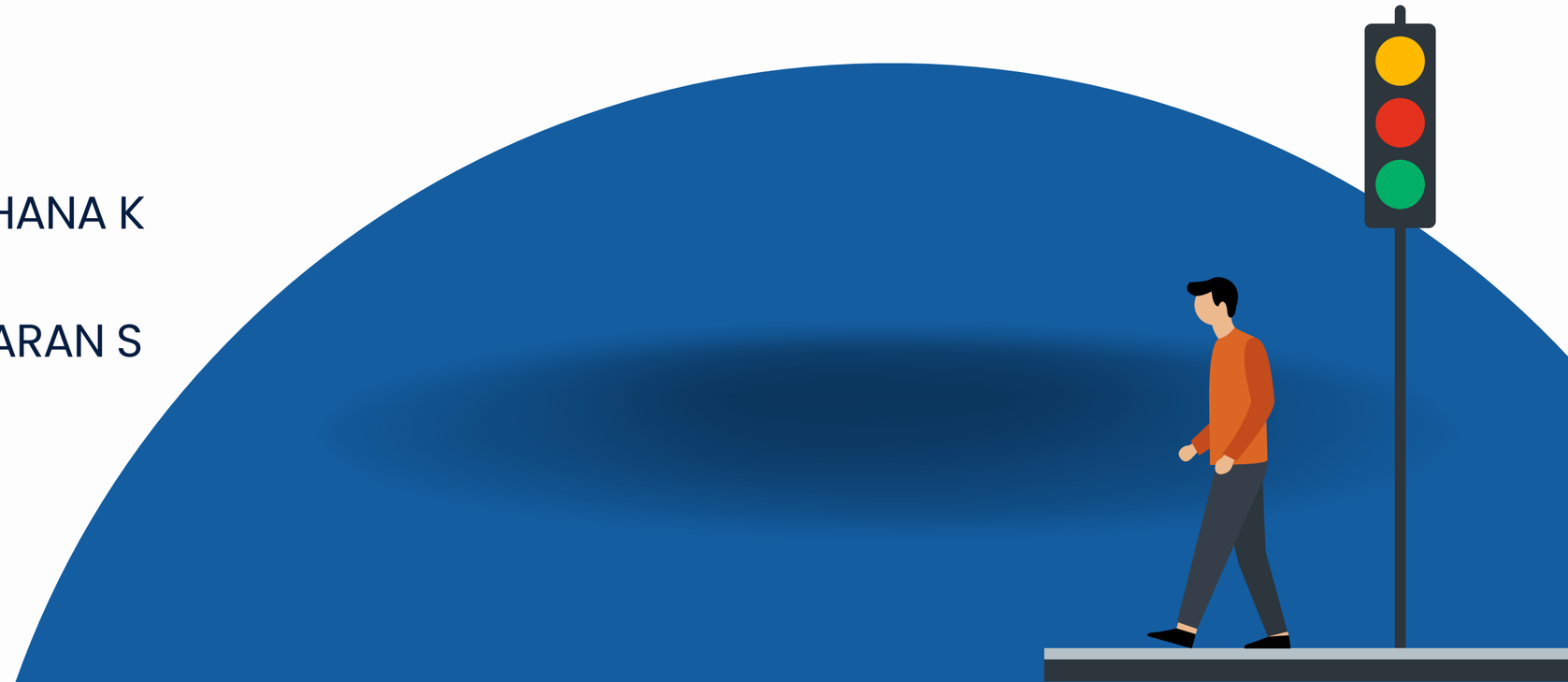
Smart Traffic Management System

19EEE381 – Open Lab

Team Members:

CB.EN.U4ELC22017 - SUBA KEERTHANA K
CB.EN.U4ELC22025 - LAKHSYA P
CB.EN.U4ELC22028 - MITHILESHWARAN S
CB.EN.U4ELC22037 - NIVETHA G K



Introduction

- Traffic congestion is a major urban issue causing delays, fuel wastage, and pollution.
- Traditional traffic signals operate on fixed time intervals, leading to inefficiencies.
- This project aims to create an adaptive traffic light system using AI and real-time data.

Problem Statement

Problem 01

- Inefficient Traffic Flow: Fixed signal timings increase congestion.

Problem 02

- Delayed Emergency Response: Emergency vehicles get stuck in traffic.

Problem 03

- Unsafe Pedestrian Crossings: Short signal times increase accident risks.

Methodology Overview

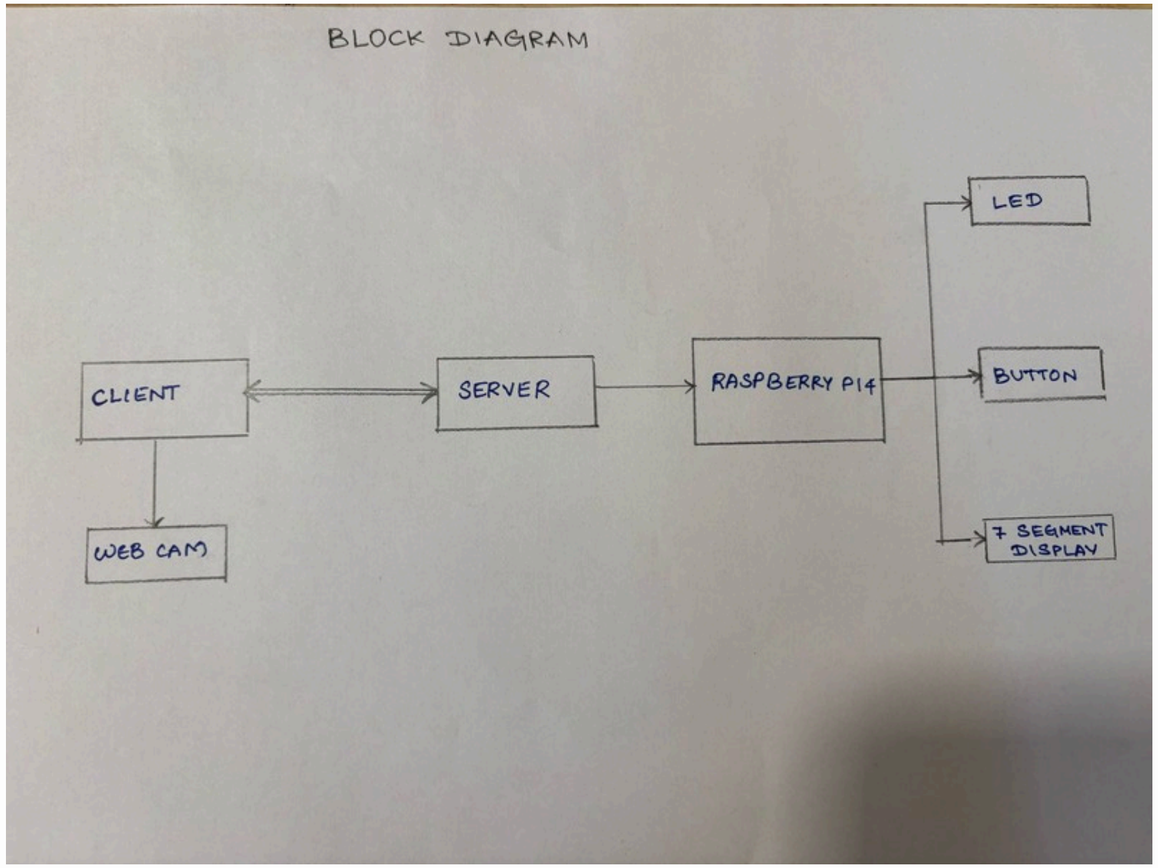
01 Traffic Density-Based Control: Adjusting signals based on vehicle count.

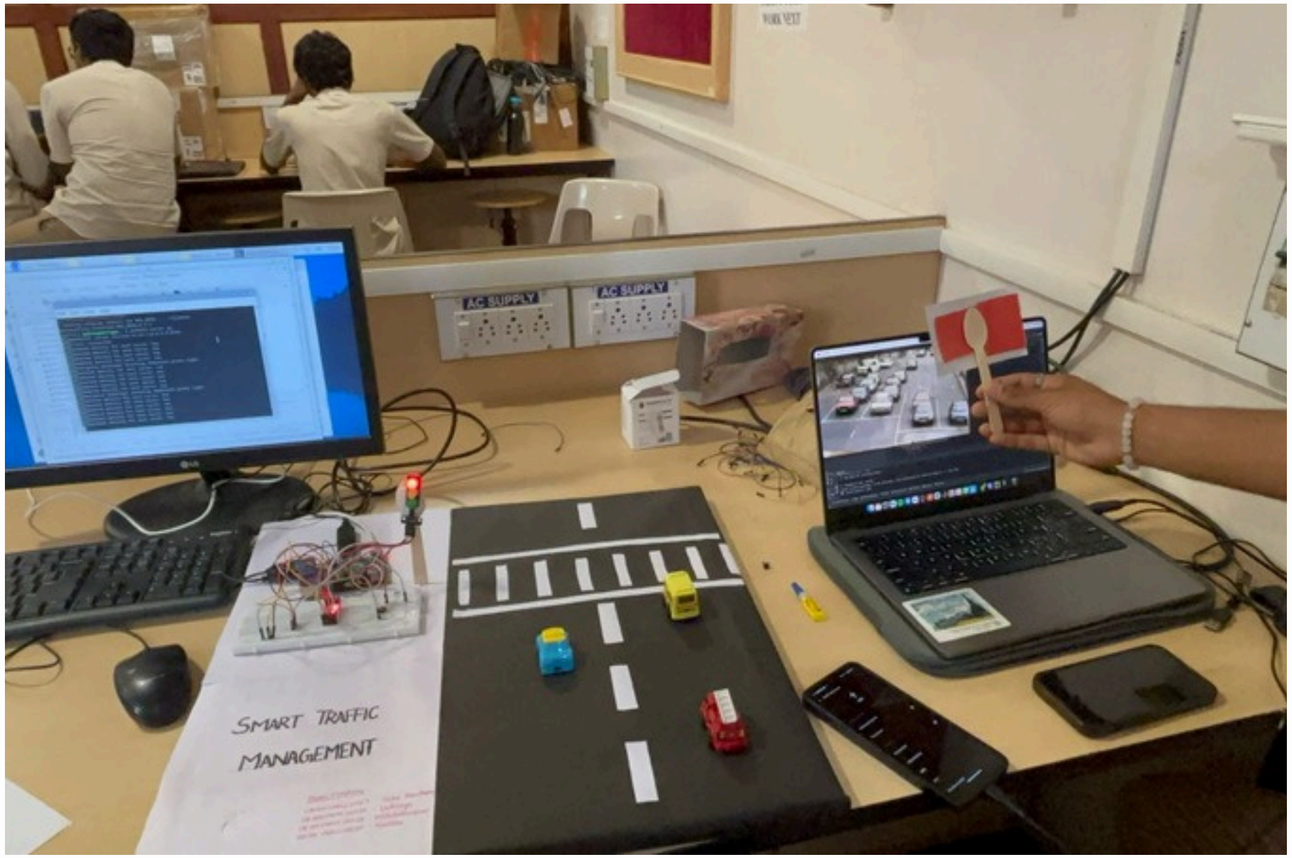
02 Emergency Vehicle Detection: Granting priority to emergency vehicles.

03 Pedestrian Button Control: Extending red light duration for safe crossing.

04 Client-Server Communication: Efficient real-time processing via Raspberry Pi and laptop.

Block diagram





Traffic Density-Based Control

- Webcam captures real-time footage.
- YOLOv8 detects and counts vehicles.
- Signal duration is dynamically adjusted based on density:
  - Low (0-4 vehicles): Shorter green time.
  - Medium (5-14 vehicles): Moderate green time.
  - High (15+ vehicles): Longer green time.

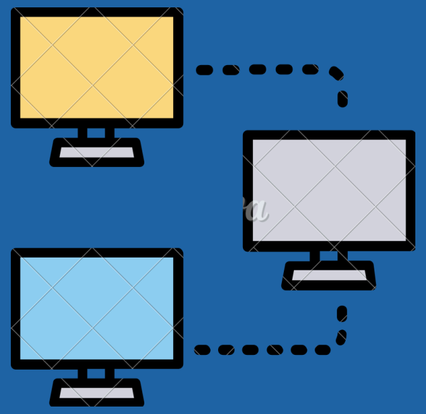
Emergency Vehicle Detection

- Uses image processing to detect flashing red and blue lights.
- HSV color filtering to identify emergency signals.
- Automatically grants immediate green signal priority.

Pedestrian Button Control

- Push-button installed at intersections.
- Extends red light for safer crossings without disrupting traffic flow.
- Real-world applications show significant accident reduction.

Client-Server Communication



- Raspberry Pi acts as a WebSocket server controlling signals.
- Laptop runs YOLOv8 for vehicle detection.
- Enables real-time decision-making and efficient traffic control.

Why WebSocket for Communication?

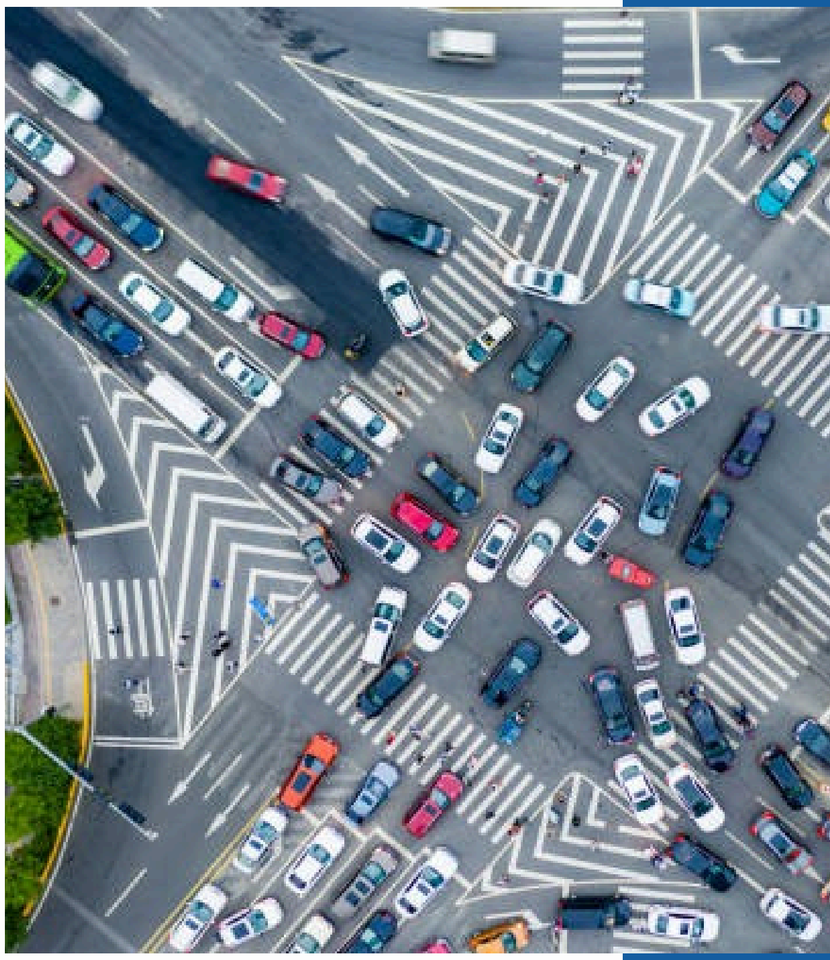
- Direct implementation of ML on Raspberry Pi resulted in only 2 FPS, causing delayed signal adjustments.
- To enhance real-time responsiveness, ML processing is offloaded to a laptop.
- WebSocket ensures fast, low-latency communication between Raspberry Pi and the laptop for seamless traffic control.

Why Raspberry Pi over ESP32?

- Linux-based OS → Supports YOLOv8 and Python-based AI models.
- Higher computational power → Unlike ESP32, which lacks real-time image processing capability.
- Seamless integration with OpenCV, deep learning models, and GPIO-controlled traffic lights.

Conclusion

- Smart Traffic System = AI + Real-Time Decision Making
- Outcome: Reduced congestion, better emergency response, and sustainable urban mobility.
- Impact: Improves safety, efficiency, and environmental sustainability.



THANK YOU!