CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
Approved by AICTE & Affiliated to JNTU Hyderabad, Medchal Road, Hyderabad.

Department of Computer Science & Engineering

A Mini Project Presentation
on

MACHINE LEARNING DEPLOYMENT IN AWS SAGEMAKER

Batch No: D11

Internal Guide
Ms. Sai Manasa,
Assistant Professor,
Dept. of CSE.

B. Ravi Kumar (228R1A05L1)
K. Kavitha (228R1A05N0)
K. Sai Saanvi (228R1A05N4)
N. Eswari (228R1A05P9)
K. Akhil Yadav (238R5A0528)

CONTENTS

ABSTRACT

- We present a smart traffic management system to detect and count vehicles from real-time images using the YOLOv8 object detection model.
- The primary goal is to deploy this machine learning model on Amazon SageMaker to ensure scalable, efficient, and cloud-based inference capabilities.
- Our system accepts traffic images as input, detects and classifies vehicles into categories such as cars, motorcycles, buses, and trucks, and calculates an adaptive green signal time based on traffic density.
- The trained YOLOv8 model is packaged and deployed using SageMaker's custom model deployment features.

INTRODUCTION

- Traffic congestion is a major issue in urban areas, leading to delays, fuel wastage, and increased pollution.
- Traditional traffic lights follow fixed timers, which do not adapt to real-time traffic conditions.
- This project proposes a smart traffic system that detects vehicles using YOLOv8 (You Only Look Once) and adjusts green light duration dynamically.
- The system leverages Amazon SageMaker to deploy the ML model for scalable and real-time inference.

LITERATURE SURVEY

S.No	Paper Title	Author(s)	Year	Algorithms Used	Conclusion
1	End-to-End ML Deployment Using Flask and SageMaker	M. Iyer, V. Kapoor	2025	YOLOv8, Flask, AWS	Full-stack ML deployment with Flask and SageMaker offers seamless scalability and integration.
2	Smart Traffic Light Control System Using Deep Learning	B. Thomas, L. Javed	2024	YOLOv5, RNN	Smart systems using DL can reduce traffic congestion significantly.
3	Cloud-Based Deployment of ML Models Using AWS SageMaker	S. Kumar, P. Reddy	2024	XGBoost, Linear Models	SageMaker provides scalable and efficient deployment for ML applications in production.
4	Flask vs FastAPI for ML Model Deployment: A Comparative Study	T. Raj, M. Roy	2024	REST API frameworks	Flask is easy to integrate but FastAPI offers better performance in high-concurrency scenarios.

EXISTING SYSTEMS

- **Fixed Time Signal Systems -**

  Operate traffic lights based on pre-set timers.

  Cannot adapt to real-time traffic flow, causing unnecessary delays and congestion.

- **Manual Traffic Control** -

  Traffic police manage signals based on human judgment.

  Prone to human error, inconsistent, and not scalable for large cities.

- **Traditional Image Processing Technique -**

  Use basic motion detection and background subtraction.

  Struggle with accuracy in poor lighting, occlusions, or complex traffic scenarios.

LIMITATIONS

- Inefficient during peak and off-peak hours.
- Requires constant manpower and lacks scalability.
- Not scalable for city-wide implementation.
- Requires constant manpower and lacks scalability.
- Cannot adapt to real-time traffic changes or emergencies.

PROPOSED SYSTEM

- **Real-Time Vehicle Detection**

  Uses YOLOv8 to detect and classify vehicles from traffic images quickly and accurately

- **Smart Signal Timing**

  Calculates dynamic green light duration based on the number of detected vehicles.

- **AWS SageMaker Integration**

  ML model is deployed using SageMaker for scalable and reliable inference.

- **Support for Multiple Vehicle Types**

  Recognizes cars, buses, trucks, and motorcycles using COCO class IDs.

ADVANTAGES

- Dynamic Traffic Management: Adjusts signal timings in real-time based on actual traffic conditions, reducing congestion.
- Improved Traffic Flow: Optimizes green light duration to minimize waiting times and prevent traffic jams.
- Cost-Effective: Eliminates the need for expensive physical sensors or infrastructure upgrades.
- Scalable and Flexible: Cloud deployment on AWS SageMaker allows easy scaling and updates without hardware changes.

SOFTWARE REQUIREMENTS

- **Operating System** Windows / Linux / macOS
- **Python Version** Python 3.8 or higher
- **Libraries** ultralytics, opencv-python, flask, etc.
- **Cloud Platform** AWS Free Tier (SageMaker, EC2)
- **Browser** Chrome / Firefox / Edge (for AWS UI)

HARDWARE REQUIREMENTS

- **Processor** Dual Core / i3 / i5 / i7
- **RAM** 4 GB (Minimum)
- **Hard Disk** 100 GB or above
- **Internet** Stable connection (for accessing AWS services)

TOOLS AND TECHNOLOGIES USED

Programming Language

- Python : Used for backend logic, ML model integration, and image processing.

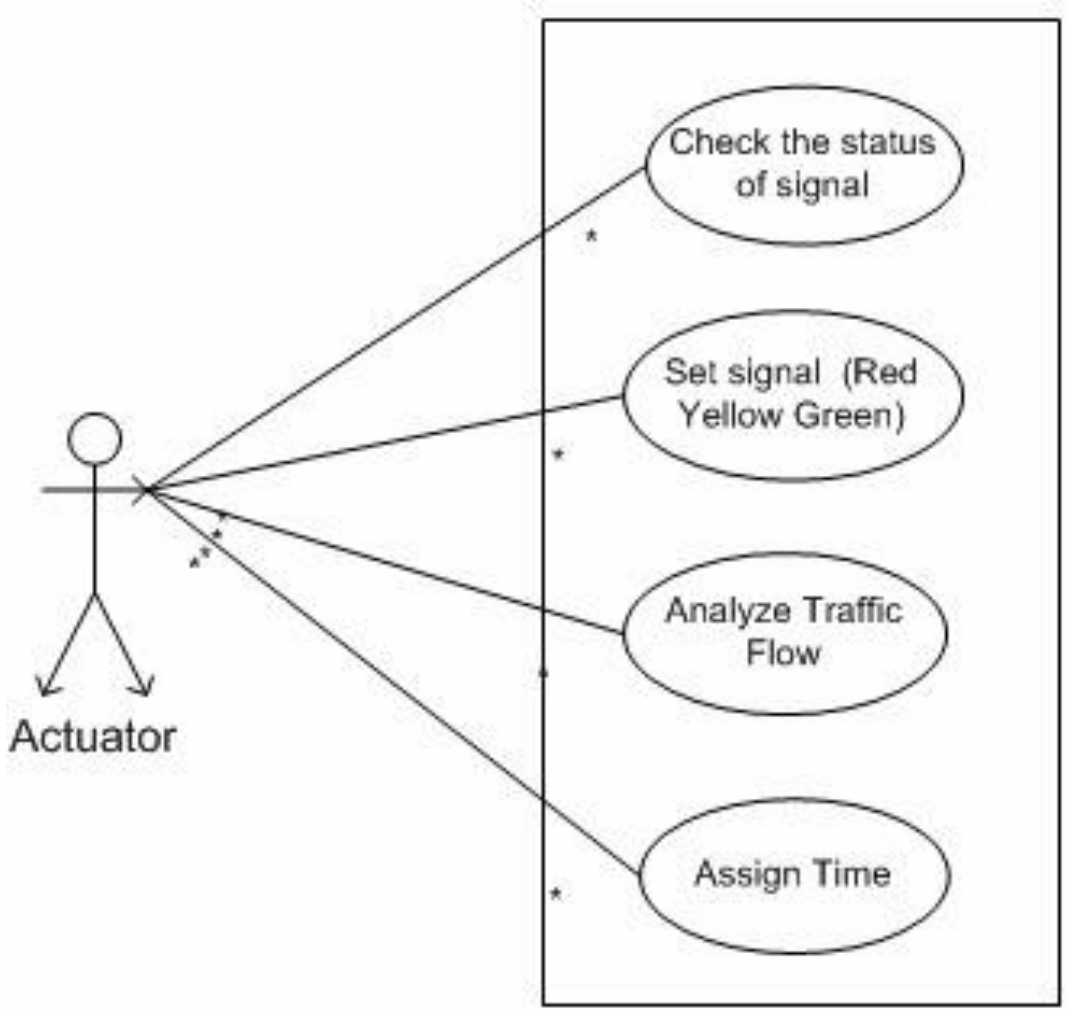
Machine Learning

- YOLOv8 (You Only Look Once) : Pre-trained object detection model for detecting cars, trucks, buses, etc.

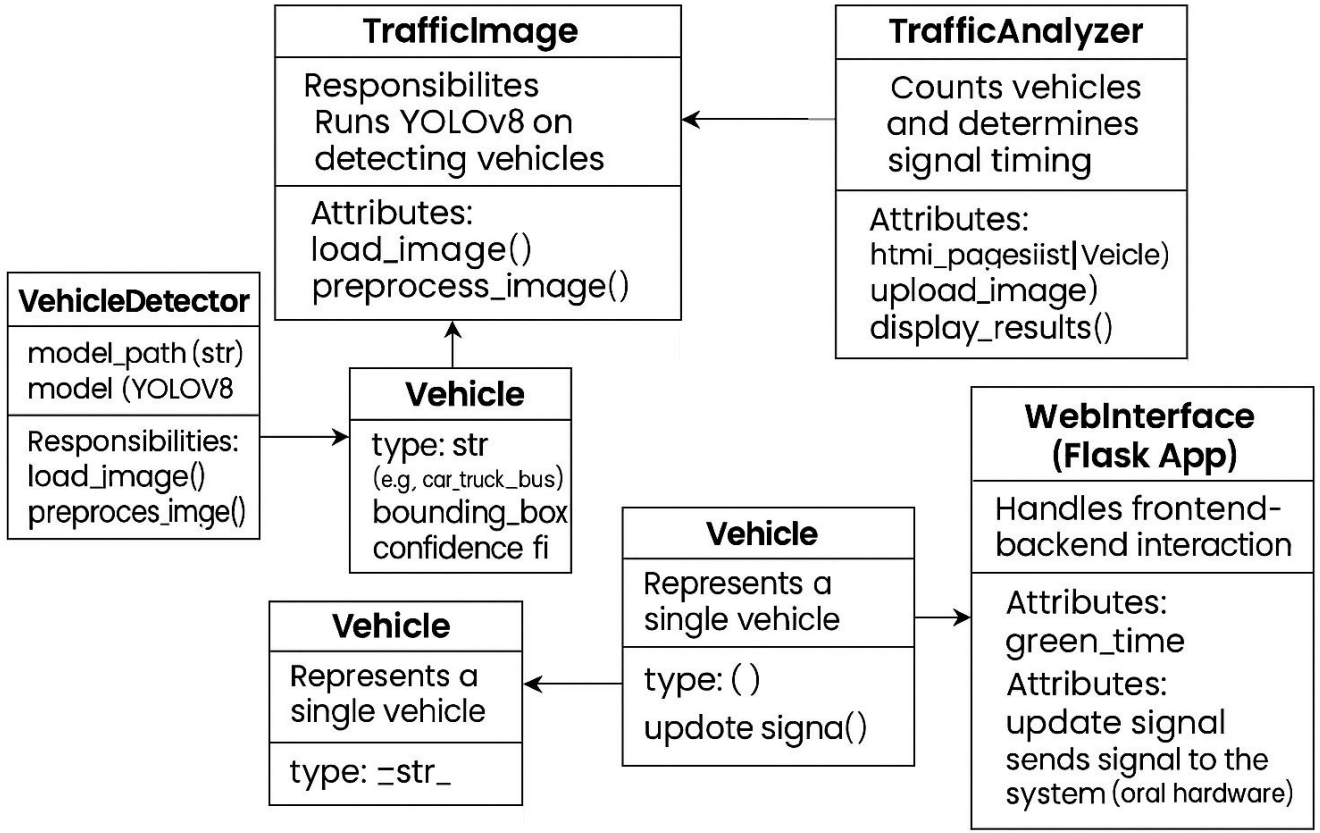
Cloud Services

- Amazon SageMaker : For ML model deployment and inference
- Amazon EC2 *:* For full-stack deployment (Flask + frontend)

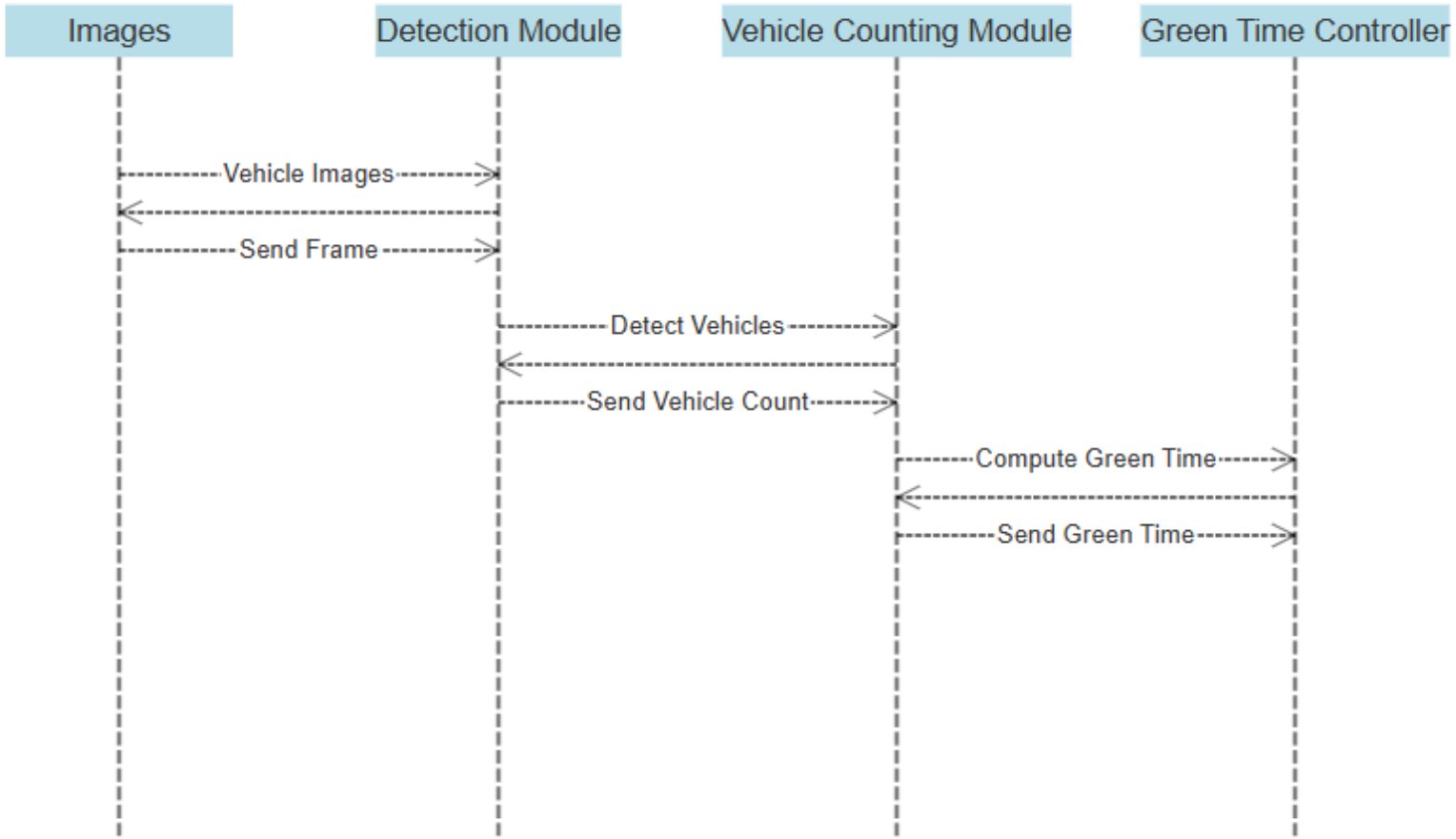
USE CASE DIAGRAM



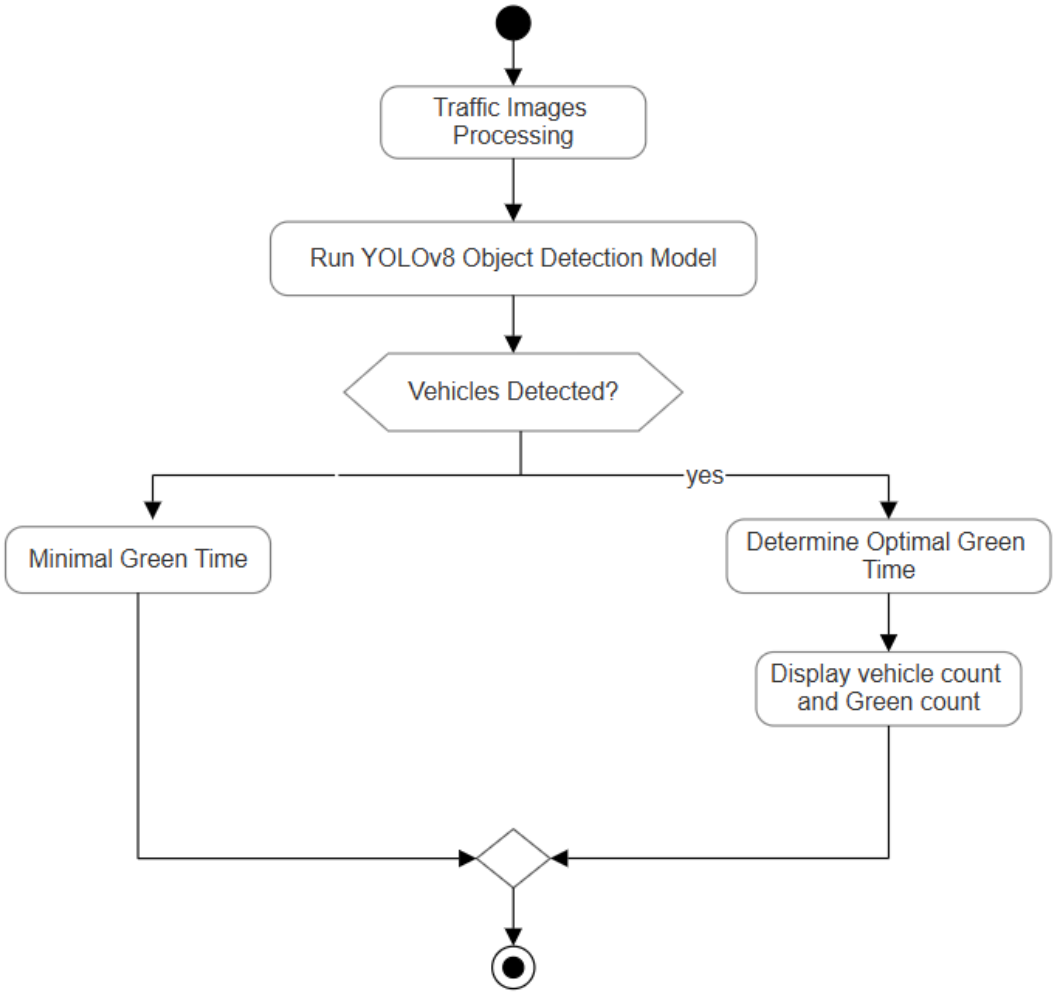
CLASS DIAGRAM



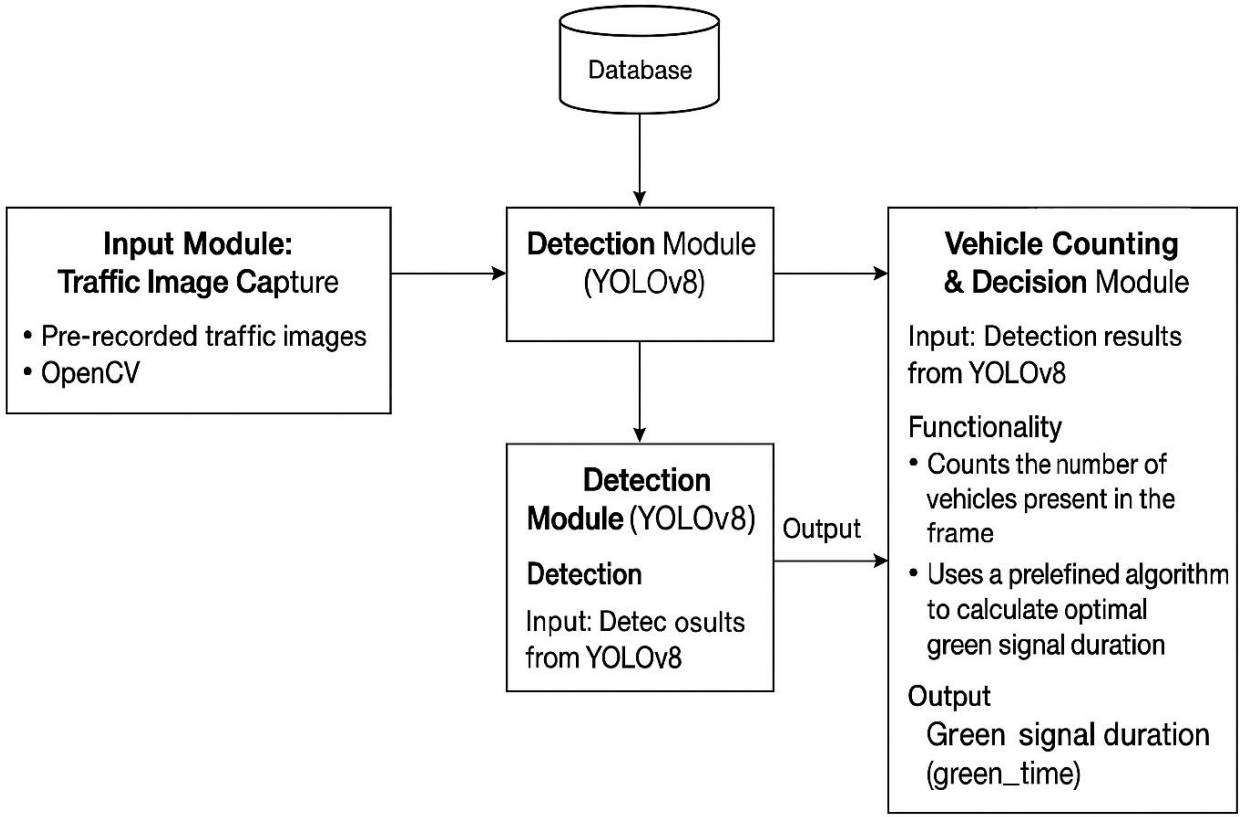
SEQUENCE DIAGRAM



ACTIVITY DIAGRAM



SYSTEM ARCHITECTURE



SYSTEM TESTING

- **Functional Testing:** Verify real-time detection of vehicles and ambulances.
- **Integration Testing:** Confirm compatibility between Flask backend and SageMaker deployment.
- **Performance Testing:** Stress test the system with high traffic volume to observe stability.
- **Security Testing:** Verify secure API endpoints and protection from unauthorized access.
- **Usability Testing:** Assess frontend interface for clarity and ease of use.

OUTPUT SCREENS

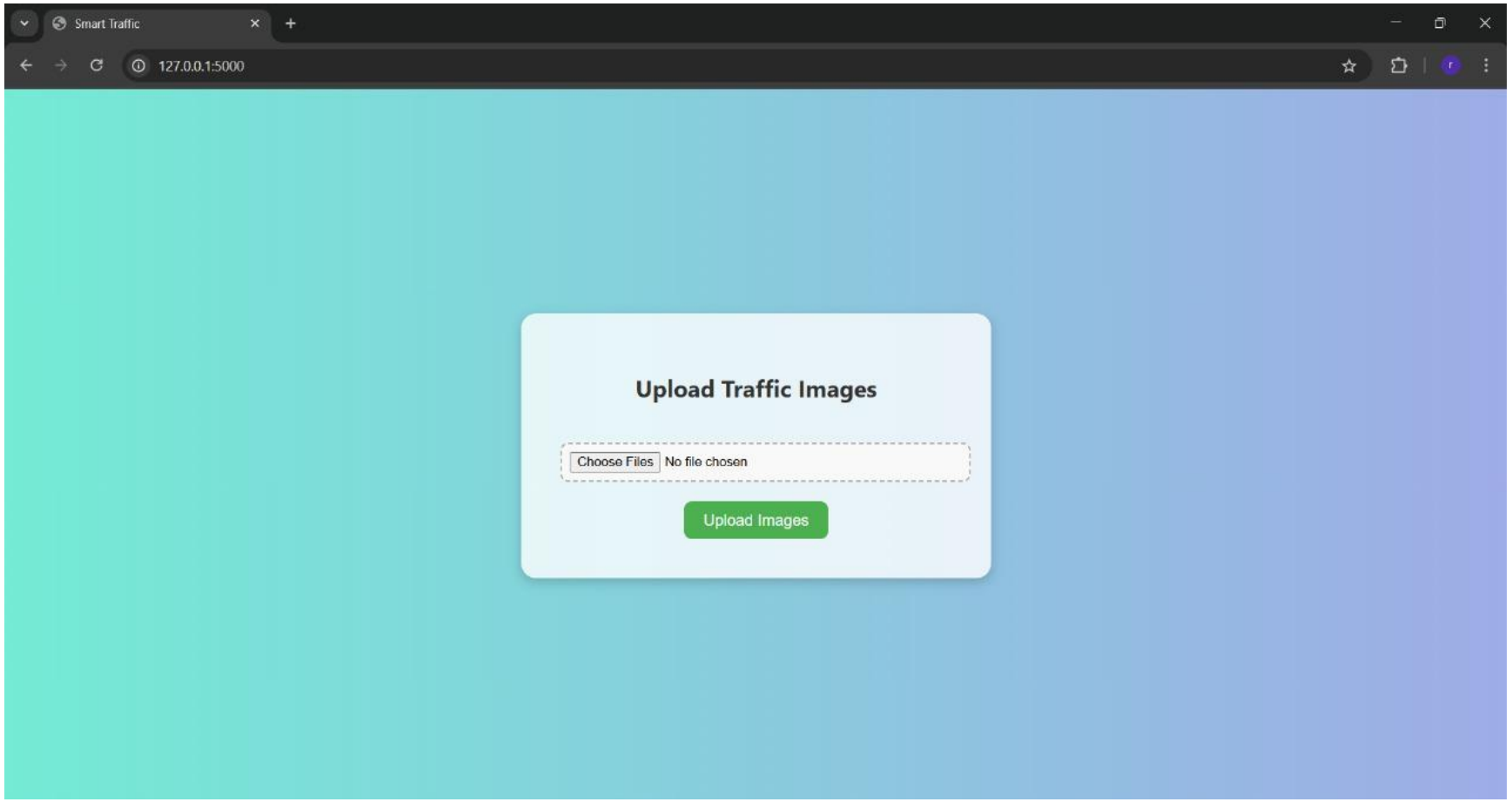


Fig (a) : Main Page

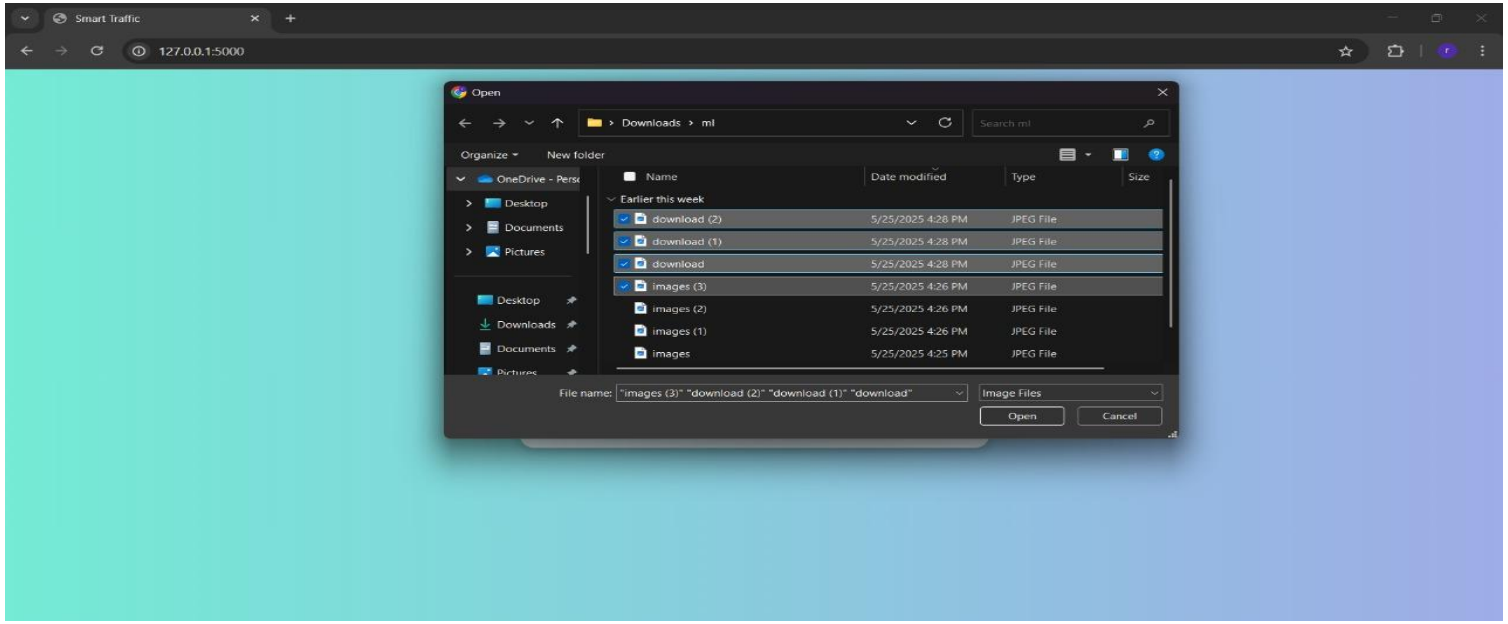
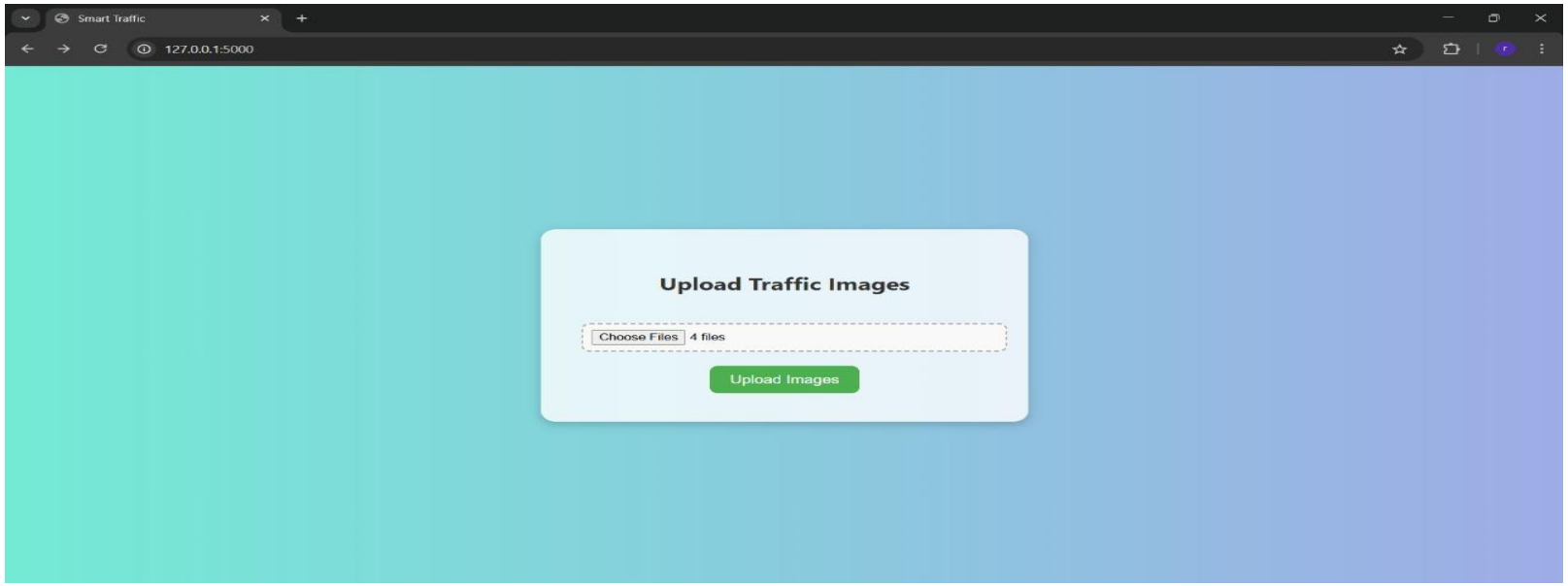


Fig (b) : Selection of Images



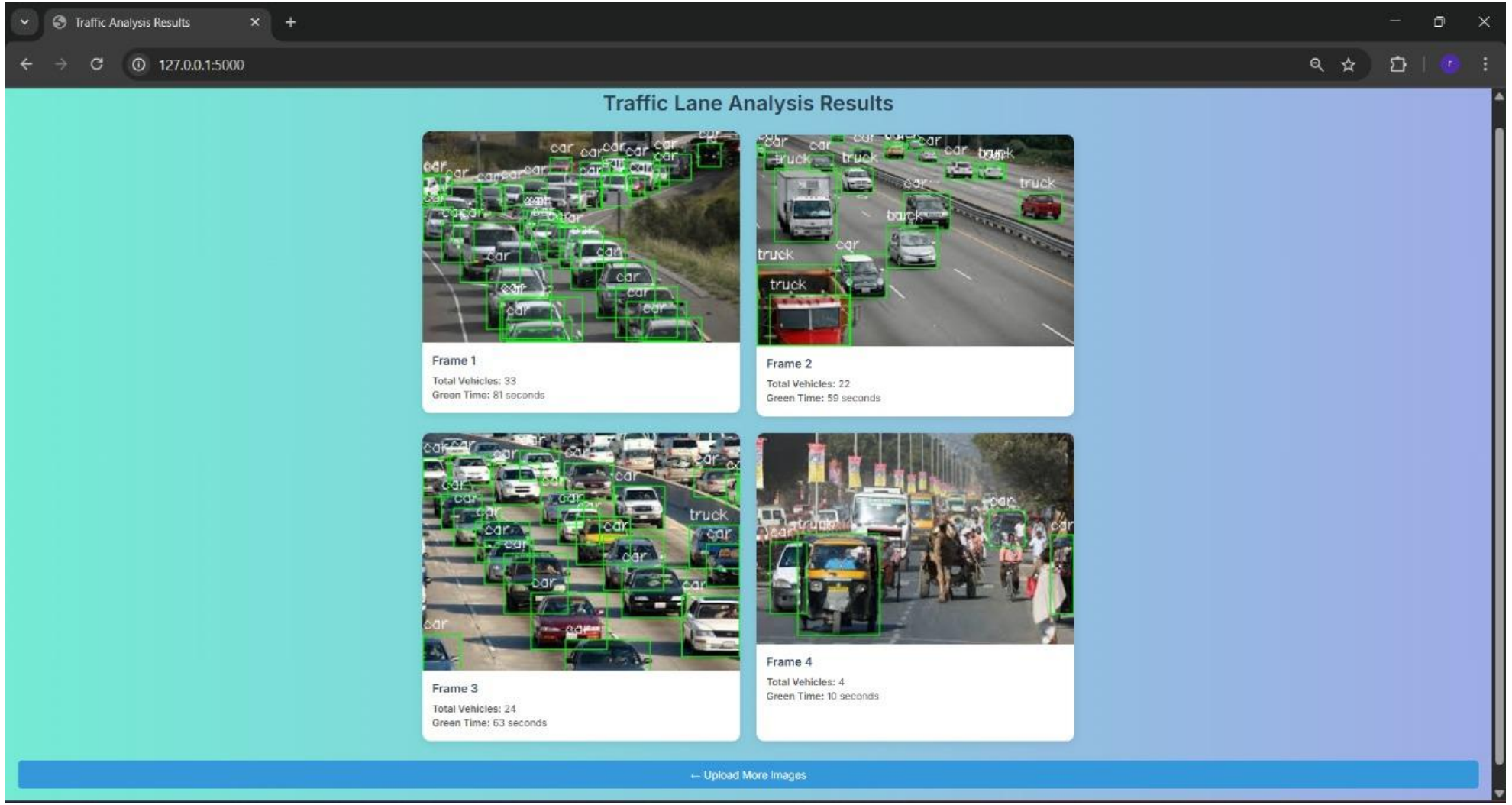


Fig (d) : Result page

CONCLUSION

- The Smart Traffic System developed in this project provides a practical and efficient solution for real-time traffic management using computer vision and machine learning.
- The system leverages the YOLOv8 object detection model to identify and count various types of vehicles in traffic footage.
- By analysing the density of vehicles, it can make informed decisions to optimize traffic signal timings dynamically, thus contributing to smoother traffic flow and reduced congestion.
- Deployed using AWS SageMaker, the system benefits from cloud scalability, fast inference, and seamless integration with other AWS services.

FUTURE ENHANCEMENTS

- **Ambulance Detection and Priority Handling:** The system can be enhanced to identify emergency vehicles like ambulances in real-time.
- **Accident Detection System:** accident detection capabilities can be included to identify collisions based on motion anomalies or object displacement, triggering immediate emergency response protocols.
- **Weather and Lighting Adaptation:** To ensure accuracy during night time, rain, etc the system can be upgraded to use infrared cameras alongside standard video feeds.
- **Multi-Camera Integration:** To scale the solution beyond a single intersection, the system can be extended to support multiple camera feeds from various junctions across a city.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2016. *https://doi.org/10.1109/CVPR.2016.91*

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," .2020. *https://arxiv.org/abs/2004.10934*

[3] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016. *https://arxiv.org/abs/1603.04467*

[4] Himanshu Singh, "Practical Machine Learning with AWS: Process, Build, Deploy, and Productionize Your Models Using AWS". 2021. *https://www.amazon.com/Practical-Machine-Learning-AWS-Productionize/dp/1484262212*

THANK YOU