



UrbanBot – AI Powered Smart City Analytics & Decision Support System

Final Capstone Project

Developed by:

Vishvashwarran V B

1 ABSTRACT

UrbanBot is an end-to-end AI-powered Smart City Analytics and Decision Support System designed to monitor, analyze, and assist urban infrastructure management in real time. The platform integrates Computer Vision, Machine Learning, Natural Language Processing, Large Language Models, and a MySQL database into a unified Streamlit-based dashboard.

The system enables automated traffic congestion detection, road accident severity analysis, crowd density monitoring, pothole detection, air quality analytics, infrastructure issue tracking, and citizen complaint sentiment analysis. In addition, an LLM-powered chatbot with Retrieval-Augmented Generation (RAG) provides natural language access to city data and operational insights.

The application is deployed on AWS EC2 for real-time cloud access, making it a scalable and production-style Smart City intelligence solution.

2 INTRODUCTION

With rapid urbanization, cities require intelligent and automated systems to monitor infrastructure, ensure public safety, and support data-driven governance. Traditional monitoring methods are manual, reactive, and lack real-time analytical capabilities.

UrbanBot addresses this challenge by combining multiple AI technologies into a single unified platform that delivers real-time urban intelligence through an interactive dashboard and an AI-powered assistant.

3 PROBLEM STATEMENT

Existing city monitoring systems:

- Depend heavily on manual observation
- Do not provide real-time insights
- Lack predictive and analytical capabilities
- Operate as isolated subsystems

There is a need for an integrated AI-based Smart City platform that can:

- Monitor multiple urban domains simultaneously
- Generate intelligent alerts
- Provide decision-support insights

- Enable natural language interaction with city data
-

OBJECTIVES

The main objectives of this project are:

- To design a unified Smart City analytics dashboard
 - To detect and classify road accidents based on severity
 - To analyze traffic congestion levels using AI
 - To monitor AQI and environmental conditions
 - To detect potholes and streetlight infrastructure issues
 - To perform sentiment analysis on citizen complaints
 - To develop an LLM-powered Smart City chatbot using RAG
 - To deploy the complete system on AWS cloud
-

SYSTEM ARCHITECTURE

The system follows a modular AI-driven architecture:

User Interface (Streamlit)



AI Processing Modules



MySQL Database



RAG Knowledge Index



Groq LLM (LLaMA 3)



AWS S3 (Image Storage)



AWS EC2 Cloud Deployment

This architecture enables real-time inference, intelligent alerting, and natural language querying of Smart City data.

METHODOLOGY / MODULES

6.1 Smart City Unified Dashboard

Displays real-time KPIs:

- High traffic zones

- Latest AQI
- Accidents today
- Crowd hotspots
- Potholes detected
- Streetlight issues
- Negative citizen complaints

Includes interactive visualizations and heatmaps.

6.2 Traffic Analysis

- AI-based congestion detection
 - Vehicle count estimation
 - High traffic zone identification
-

6.3 Air Quality Monitoring

- AQI trend analysis
 - Pollution category classification
 - Location-based environmental insights
-

6.4 Pothole Detection

- Road damage detection using Computer Vision
 - Automatic infrastructure issue logging
-

6.5 Road Accident Detection

- Image/video-based accident detection
 - Severity classification (Low / Medium / High)
 - Emergency alert generation
 - Response time estimation
-

6.6 Crowd Density Monitoring

- High-density area detection
- Public safety risk identification

6.7 Infrastructure Streetlight Monitoring

- Streetlight fault detection
 - City-wise issue tracking
-

6.8 Citizen Complaint Intelligence

- Complaint registration
 - Sentiment analysis
 - Priority-based routing
 - Negative complaint analytics
-

6.9 UrbanBot AI Chat Assistant

LLM-powered assistant that supports:

- Natural language database queries
 - Smart city operational reports
 - AI advisory generation
 - RAG-based knowledge retrieval
 - Monitoring image retrieval from AWS S3
-

7 TECHNOLOGY STACK

Frontend: Streamlit

Backend: Python

Artificial Intelligence:

- YOLO (Ultralytics) – Computer Vision
- Scikit-learn – Machine Learning
- NLTK – NLP Sentiment Analysis

LLM:

- Groq API – LLaMA 3 (8B)

Database:

- MySQL

Cloud:

- AWS EC2 – Application Hosting
- AWS S3 – Image Storage

Other Tools:

- OpenCV
 - Plotly
 - Boto3
-

8 IMPLEMENTATION

The system is implemented using a modular agent-based architecture:

- Intent detection agent
- Database query agent
- RAG knowledge agent
- Email automation agent
- S3 image retrieval agent
- AI advisory agent

Each module operates independently and is orchestrated through the UrbanBot AI interface.

The RAG pipeline converts structured Smart City data into embeddings for semantic retrieval and context-grounded LLM responses.

9 RESULTS AND OUTPUTS

The system successfully provides:

- Real-time Smart City KPI monitoring
- AI-based accident severity detection
- Automated email alerts for critical incidents
- Complaint sentiment-based prioritization
- Natural language querying of city data
- Cloud-hosted intelligent dashboard

Smart City Analytics Dashboard

Real-time Urban Intelligence for Data-Driven City Governance

Command Center Overview

This dashboard provides real-time monitoring of:

- Traffic congestion
- Air quality levels
- Road accidents
- Crowd density
- Infrastructure issues
- Citizen complaints

Enabling faster, data-driven urban governance.

Key Urban Risk Indicators

Category	Value
High Traffic Zones	34
Latest AQI	114
Accidents Today	6
High Crowd Locations	5
Potholes Detected	80
Streetlight Issues	9
Negative Complaints	10

Monitoring Station: Perungudi - Chennai

Latitude: 13.082700

Pollutant Values

PM2.5	44.86	- +	NO2	21.57	- +
PM10	184.12	- +	SO2	11.58	- +
CO	0.88	- +	O3	26.61	- +

Run AQI Analysis

AQI: 114 (Poor) | Stored in RDS

- [Dashboard](#)
- [Project Description](#)
- [Traffic Analysis](#)
- [Air Quality](#)
- [Pothole Detection](#)
- [Road Accident](#)
- [Overcrowding](#)
- [Infrastructure Streetlight](#)
- [Citizen Sentiment](#)
- [Urbanbot AI Intelligence](#)
- [About](#)

[Clear Chat](#)

@ UrbanBot AI Intelligence

Hi
Hello! I am your Smart City AI assistant. How can I help you today?

Give me recent crowd detections full report

Crowd Detections Report

- Pudukkottai + Anna Salai: 406 estimated count with a density level of "extreme" (234) and "high" (75, 97)

Operational Insights:

- The extreme crowd density at Anna Salai in Pudukkottai suggests that the area may require additional security measures to ensure public safety.
- The high crowd density at other locations in Pudukkottai and Chennai may indicate popular tourist attractions or events, warranting increased foot traffic management and emergency response planning.

Ask about traffic, AQI, accidents, complaints...

🚦 Traffic Analysis

📍 Enter Location Details

City

Trichy

Area / Junction

Railway colony

Latitude

105.000000

Longitude

95.854000

Lane Count

5

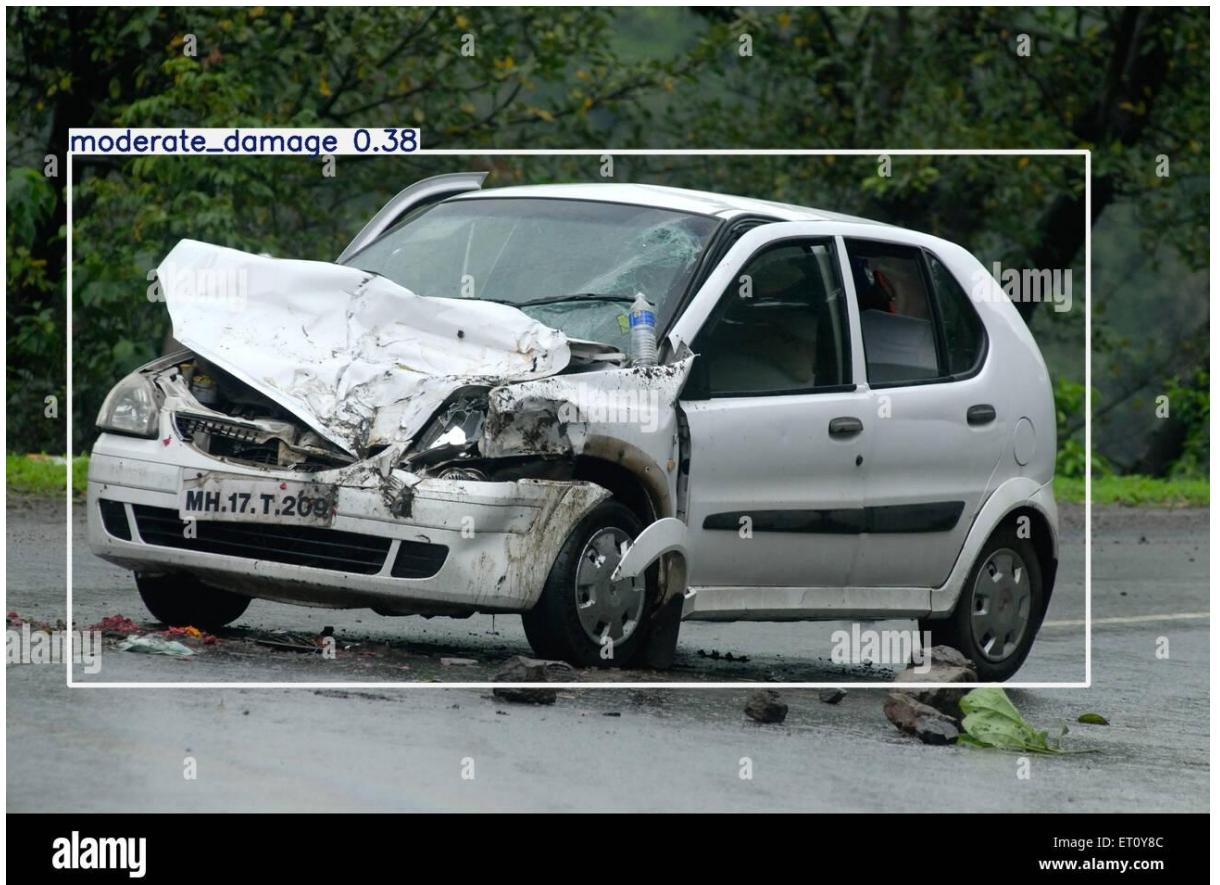
Weather

Clear

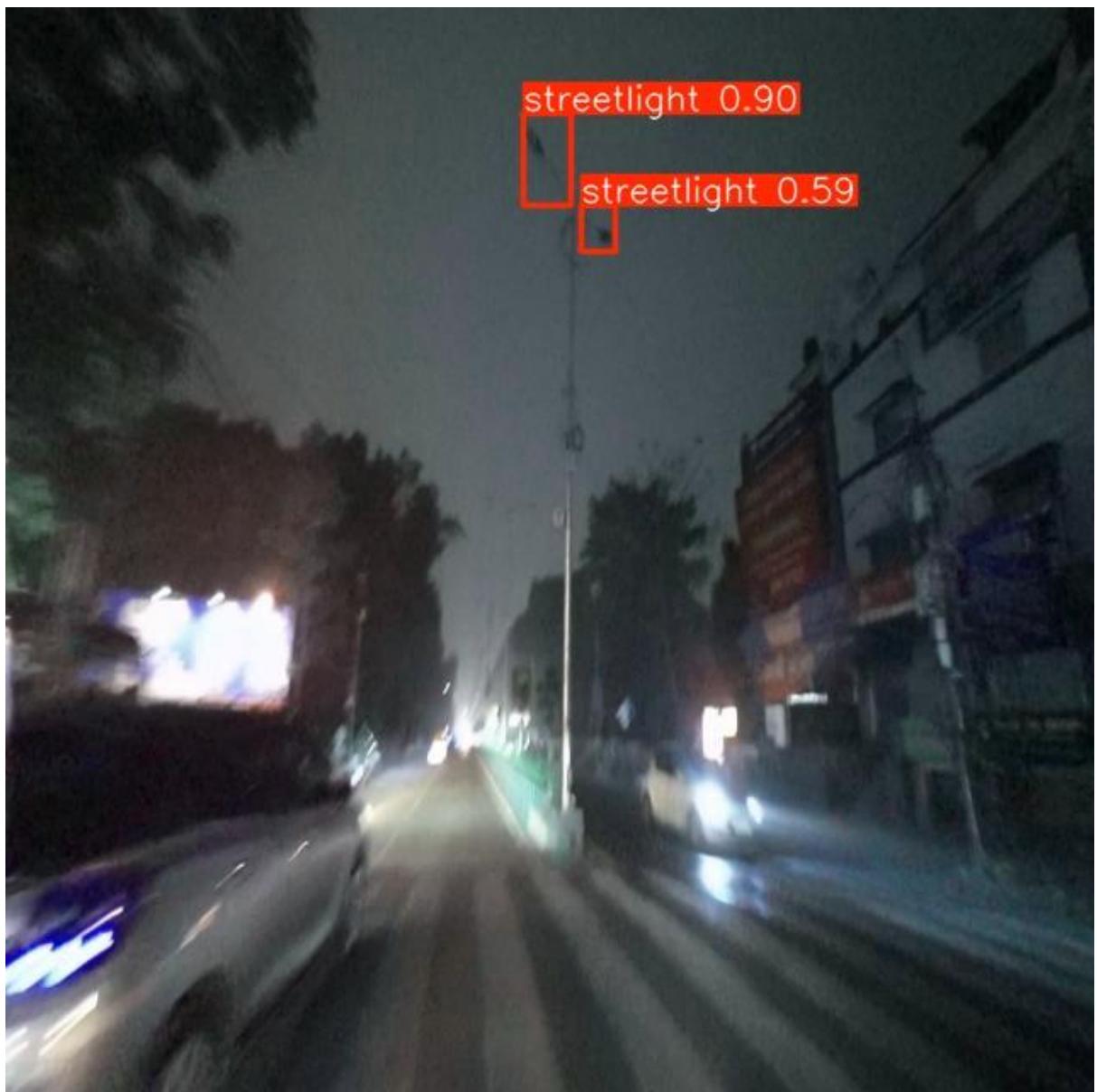
[Upload Traffic Image / Video](#)

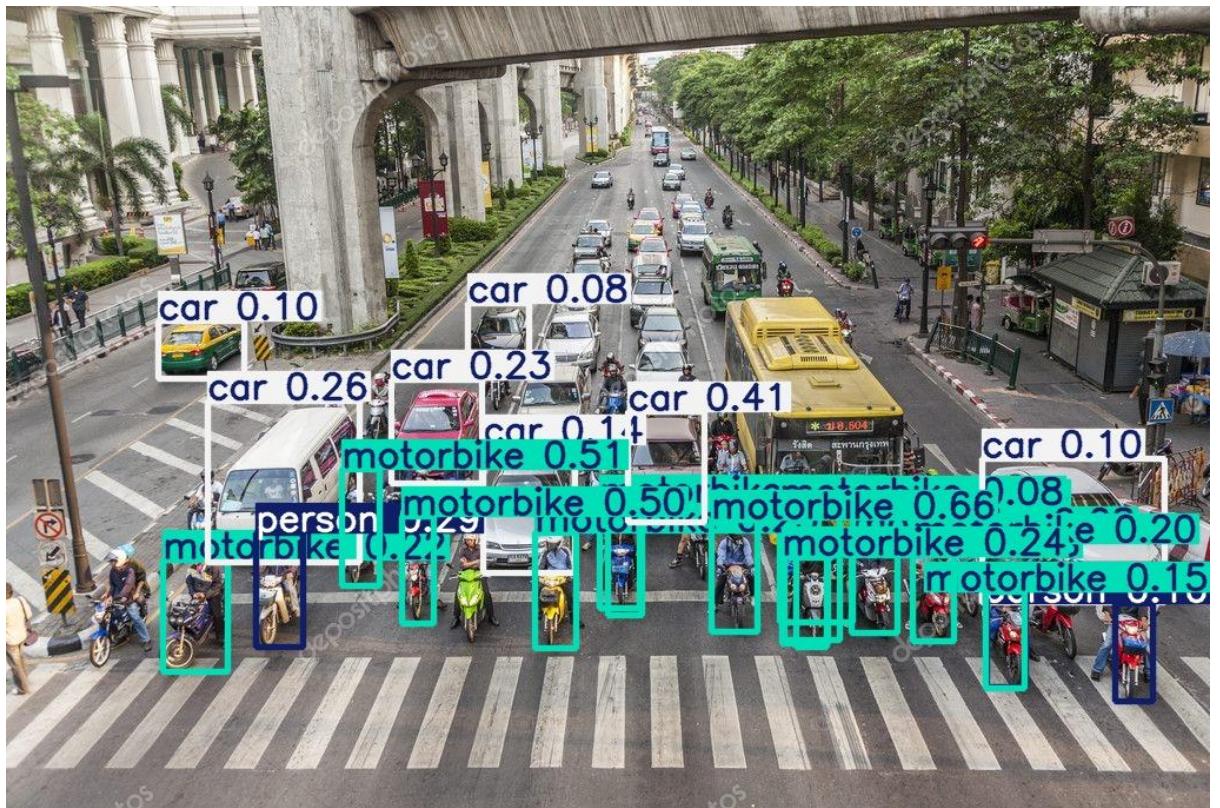


⚠️ Vehicles: 13 | 📈 Next Hour: 239 (high) | 📁 Stored in RDS | 🚨 Alert email sent









10 DEPLOYMENT

The application is deployed on AWS EC2 and is accessible through a public IP.

Live Application:

<http://40.192.16.177:8501>

Key deployment features:

- Cloud-based AI inference
 - Remote accessibility
 - Persistent database integration
 - Scalable architecture
-

1 1 CONCLUSION

UrbanBot demonstrates how multiple AI technologies can be integrated into a unified, real-time Smart City intelligence platform. The system transforms raw urban data into actionable insights, enabling efficient infrastructure monitoring, faster emergency response, and data-driven governance.

1 2 FUTURE ENHANCEMENTS

- Live CCTV stream integration
 - Predictive traffic forecasting
 - Mobile application interface
 - Role-based administrative dashboard
 - GPU-based cloud deployment for faster inference
-