1. INTRODUCTION

1.1 Project Overview

"Traffictelligence" is a machine learning-based system designed to estimate traffic volume intelligently using computer vision and deep learning techniques. It processes video feeds or camera footage to detect and count vehicles, classify them, and analyze traffic flow patterns to support smart city infrastructure and traffic management systems.

1.2 Purpose

The purpose of this project is to provide real-time traffic data analytics for city planning, congestion control, emergency response, and optimized signal control using deep learning models trained on traffic video datasets.

2. IDEATION PHASE

2.1 Problem Statement

Urban areas are experiencing a rapid rise in traffic congestion. Manual traffic volume surveys are inefficient, costly, and error-prone. There is a need for an automated, scalable system that can estimate traffic volume accurately and in real-time.

2.2 Empathy Map Canvas

User: Traffic authorities, commuters, urban planners.

Says: “We need a system that gives real-time traffic data.”

Thinks: “Manual methods are outdated and inefficient.”

Does: Uses CCTV footage, road sensors.

Feels: Frustrated by traffic jams and lack of data.

2.3 Brainstorming

Use YOLOv5 for vehicle detection.

Integrate with OpenCV to process live video.

Count and classify vehicles.

Display volume statistics via dashboard.

Apply model optimization for real-time use.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

City traffic department installs cameras.

Footage is fed into the system.

ML model detects and counts vehicles.

Output is visualized and stored.

3.2 Solution Requirement

Input: Real-time video feeds or pre-recorded footage.

Output: Vehicle count per class (cars, bikes, trucks), traffic density.

Performance: Real-time or near-real-time processing.

3.3 Data Flow Diagram

Camera Feed → Pre-processing → Object Detection → Vehicle Classification → Count Estimation → Dashboard/Storage

3.4 Technology Stack

Python

OpenCV

YOLOv5

PyTorch

Flask/Django (for dashboard)

SQLite/PostgreSQL

4. PROJECT DESIGN

4.1 Problem Solution Fit

The solution automates traffic volume analysis using real-time visual data, significantly reducing manpower and increasing efficiency and accuracy.

4.2 Proposed Solution

Use a YOLOv5 deep learning model trained on traffic footage to detect and classify vehicles, estimate counts, and generate statistics accessible via a web interface.

4.3 Solution Architecture

Frontend: Web dashboard (HTML/CSS/JS)

Backend: Python (Flask/Django)

Model: YOLOv5

Storage: SQLite/PostgreSQL

Deployment: Local server or cloud (AWS/GCP)

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week 1: Problem analysis, dataset collection

Week 2: Model selection, initial training

Week 3: Video processing integration

Week 4: UI development, testing

Week 5: Final integration and demo preparation

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

FPS: Achieved 20 FPS processing on live feed

Accuracy: 91% average detection accuracy on test videos

Latency: Under 2 seconds for dashboard update

7. RESULTS

7.1 Output Screenshots

Screenshot of vehicle detection

Live traffic counting dashboard

Plots of traffic volume over time

8. ADVANTAGES & DISADVANTAGES

Advantages

Real-time analysis

Low-cost compared to traditional sensors

Scalable and flexible

Disadvantages

Accuracy may reduce in poor lighting

Depends on video quality and angle

9. CONCLUSION

This project demonstrates a viable and efficient approach to traffic volume estimation using deep learning. It replaces traditional methods with an intelligent, automated system that can be deployed in urban settings for traffic analysis and planning.

10. FUTURE SCOPE

Integrate traffic prediction algorithms

Deploy on edge devices (Jetson Nano, Raspberry Pi)

Include license plate recognition

Enable accident and anomaly detection

GitHub & Project Demo Link: [https://github.com/Guru-prasad2004/TrafficTelligence-Advanced-Traffic-Volume-Estimation-with-Machine-Learning]