ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Project Documentation format

# Introduction

* + **Project Title:** Traffictelligence: Advanced traffic volume estimation with machine learning
  + **Team Members:**  Palagiri Swetha

# Project Overview

* + **Purpose:** The purpose of TraffiCTelligence is to develop an intelligent traffic monitoring and analysis system powered by computer vision and deep learning. It aims to automate the detection, classification, and tracking of vehicles, analyze traffic flow patterns, and support real-time traffic management. This project is designed to assist urban planners, transportation departments, and smart city infrastructures in making data-driven decisions, improving traffic efficiency, and enhancing road safety.
  + **Features:** a. Automated Vehicle Detection

b. Traffic Flow Analysis

c. Violation Detection

d. Real-Time Dashboard

# Architecture

* + **Frontend:** HTML,CSS
  + **Backend:** Python , OpenCV , CNN
  + **Database:** SQLite/MYSQL

# Setup Instructions

* + **Prerequisites:** Python , TensorFlow , Keras , Flask , Numpy , Pandas ,Scikit-learn.
  + **Installation:**

i.Clone the Repository :

git clone : <https://github.com/palagiriswetha/traffictelligence>

ii. Create a Virtual Environment :

python -m venv venv

Venv\Scripts\activate

iii. Install Dependencies :

pip install -r requirements.txt

iv. Setup the Environment Variables :

touch .env

v. Run the Application :

Flask run

# Folder Structure

* + **Client:** It contains the user interface (index.html , style ) in the frontend.
  + **Server:** It loads the deep learning model , process the uploaded image and returns the predicted class and communictes with the frontend using Flask routes(app.py) at the backend.

# Running the Application

Install dependencies , setup the environment and run the application using flask.

# Authentication

# Authentication : Verifies who the user is (identity)

# User Registration

# User Login

# Session Management

# 

# Authorization : Determines what the user can do (permissions)

# Use Flask-Login’s @login\_required decorator to protect sensitive routes.

# User Interface

# Traffic Volume Estimation displays estimated traffic volume in real-time. Provide insights into traffic patterns and trends. Allow users to customize dashboards with relevant metrics.

# Testing

# Model Testing(Deep Learning Pipeline)

# Tools Used :

# Python

# TensorFlow

# Scikit-learn

# Matplotlib/Seaborn

# Backend Testing(Flask API)

# Tools Used :

# Flask-Testing

# Frontend Testing(HTML- based Client)

# Tools Used :

# Manual Testing

# 

# Screenshots

# A screenshot of a computer AI-generated content may be incorrect.

# 

# Known Issues

# Limited Dataset Diversity

# Model Overfitting on Small Datasets

# Prediction Latency

# Error Handling for Invalid Inputs

# Authentication Token Expiry(if implemented)

# No Offline Functionality

# Frontend Compatability

# Lack of Real-Time Training Feedback

# Future Enhancements

1. Dataset Expansion
2. Mobile App Integration
3. Offline Prediction Mode
4. Pattern Detection and Segmentation
5. Pattern Similarity Search
6. User Feedback Loop
7. Multi-language Support
8. Voice-Based Interface