VEHICLE DETECTION AND COUNTING USING OPENCV(Open Source Computer Vision Library)

INTRODUCTION :

In cities managing traffic efficiently is crucial for safe and sustainable transportation. Counting vehicles is a key part of monitoring traffic, providing insights for planning roads, reducing congestion, and ensuring public safety. This study introduces a reliable and affordable method for vehicle counting using OpenCV, a free computer vision tool.

OBJECTIVE :

The aim of this research is to provide a versatile traffic management solution that doesn't require expensive equipment. By using easily available cameras and OpenCV, this system can be seamlessly integrated into existing urban setups, offering a cost-effective alternative to traditional sensor-based counting systems.

TOOLS / PLATFORM, HARDWARE AND SOFTWARE REQUIREMENT:

Hardware Requirements

- Processor: Intel core 13 or above
- Hard Disk: 1tb
- RAM: Minimum 4GB(recommended 8GB)

Software Requirements

- Operating System: Windows, Linux,Macos
- Language: Python
- Library: openCV(Open Source Computer Vision)
- IDE: Visual Studio Code
- Web Browser: Browser: Firefox,Google Chrome,Safari,Internet Explorer

PROBLEM DEFINITION :

Create a reliable real-time vehicle counting system in urban areas using OpenCV, which is crucial for efficient traffic management, safe transportation, and urban planning. This system should perform accurately in diverse environmental conditions, such as changing light and weather.

BASIC FUNCTIONALITIES:

1. Image/Video Input: Capture frames from a video feed or load images
2. Preprocessing: Convert frames to grayscale. Apply any necessary filters or adjustments to enhance features.
3. Vehicle Detection: Use a pre-trained model like Haar cascades or a deep learning model for object detection to identify vehicles in each frame.
4. Bounding Box Drawing: Draw bounding boxes around detected vehicles.
5. Counting Logic: Implement a counting mechanism to keep track of vehicles entering and leaving the frame.
6. Display Results: Show the processed frames with bounding boxes and count information.

INITIAL REQUIREMENTS:

- Real-Time Video Source
- Testing and Validation