

**SUBMITTED TO**

**DR.SYED MUHAMMAD ADNAN**

**SUBMITTED BY**

**AYESHA ATHAR(21-CS-22)**

**EMAN SHAHID (21-CS-13)**

**MAHEEN KHALID(21-CS-07)**

Vehicle Counting System

**Overview**

This project implements a Vehicle Counting System using Python and OpenCV. The system processes a video stream, detects vehicles, and counts the number of vehicles that pass a predefined line in the frame. The methodology involves background subtraction, contour analysis, and centroid tracking to achieve accurate vehicle counting.

**Methodology**

1. Video Capture and Background Subtraction:
   * The script captures frames from a video file and initializes a background subtractor using the MOG algorithm.
2. Image Processing:
   * The captured frame undergoes grayscale conversion, Gaussian blur, and morphological operations to enhance vehicle detection.
3. Contour Detection:
   * Contours of moving objects are detected using the **cv2.findContours** function.
4. Vehicle Tracking and Counting:
   * Detected contours are filtered based on specified width and height criteria, and vehicles are tracked using centroid calculation. The system counts vehicles that cross a predefined counting line.
5. Display:
   * The original video frame is displayed with bounding boxes around detected vehicles. The real-time vehicle count is updated on the frame.

**Usage**

1. Video Input:
   * Replace **'video.mp4'** with the path to your video file in the **cap = cv2.VideoCapture('video.mp4')** line.
2. Adjust Parameters:
   * Fine-tune parameters such as minimum width, minimum height, offset, and counting line position for optimal performance.
3. Run the Code:
   * Execute the script, and the system will display the processed video with vehicle counting.
4. Exit:
   * Press the **Esc** key to exit the program.

**Dependencies**

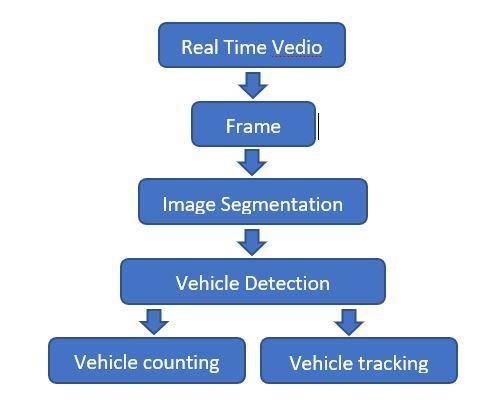
Ensure that you have the following dependencies installed:

* Python 3.x
* OpenCV
* NumPy

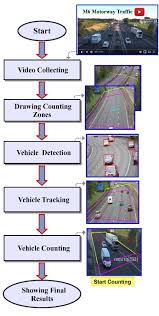
Install the required packages using:

**pip install opencv-python numpy**

**Flow chart of working :**

****

**Flow In real time video :**

****

**Conclusion:**

In conclusion, the Vehicle Counting System demonstrates an effective implementation of computer vision techniques for real-time vehicle tracking and counting. Leveraging OpenCV, the system employs background subtraction, contour analysis, and centroid tracking to accurately identify and monitor vehicles crossing a specified line in a video stream. The project's modular design allows for easy parameter customization, adapting to diverse video datasets. With its ability to provide real-time vehicle counts, this system holds potential applications in traffic management, surveillance, and urban planning. The successful execution of this project highlights the practicality and versatility of computer vision in addressing real-world challenges.

Top of Form