**Intelligent Signal Processing**

**End of Term Assignment**

**Exercise 1**

|  |  |  |
| --- | --- | --- |
|  | **Total number of cars** | **Cars per minute** |
| Traffic\_Laramie\_1.mp4 |  |  |
| Traffic\_Laramie\_2.mp4 |  |  |

**Analysis of Application**

This Python script utilizes the OpenCV library to analyze two traffic videos ("Traffic\_Laramie\_1.mp4" and "Traffic\_Laramie\_2.mp4"). It employs background subtraction, morphology operations, and contour detection to identify and track moving objects, assumed to be cars, within a defined region of interest representing a main street. The script counts the number of cars passing a specific checkpoint and prints the count at one-minute intervals, providing insights into traffic flow over time. The application combines computer vision techniques to process video frames, detect cars, and monitor their movement, offering a basic traffic analysis tool for the specified scenarios.

**Frame differencing and background subtraction**

My code utilizes frame differencing and background subtraction techniques for vehicle detection in traffic surveillance videos. Frame differencing is employed to highlight changes between consecutive frames, emphasizing moving objects. Initially, the video frames are captured and processed in a region of interest representing the main street. The frames are converted to grayscale and then blurred using GaussianBlur to reduce noise. Subsequently, background subtraction is performed using the createBackgroundSubtractorMOG2 method, which models the background of the scene and identifies foreground objects. The resulting foreground mask highlights moving vehicles and eliminates static background elements. To further enhance object segmentation, morphological operations are applied to the foreground, refining the shapes and filling gaps in the detected contours.

Contours are then extracted from the processed foreground, and the code iterates through each contour to identify potential vehicles. A minimum contour area threshold is set to filter out small irrelevant objects like people bicycles. Detected vehicles are outlined with bounding boxes, and the centroid of each vehicle is tracked using circles. The specific region of interest in the main street is monitored, and when a vehicle crosses a predefined line (checkpoint), its presence is detected. A car counting mechanism is implemented, focusing on a specific region within the frame of main street. If a vehicle is detected within this region, its centroid is added to a list, and a subsequent loop removes the current centroid in the list, indicating the counting of a single vehicle.