**Vehicle Detection**

**Library used:**

**OpenCV-Python:** OpenCV is an excellent tool for image processing and computer vision tasks. It is an open-source library that performs functions like face detection, objection tracking, landmark detection, and much more.

**NumPy:** NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices.

**Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy and hard things possible, [interactive figures](https://mybinder.org/v2/gh/matplotlib/mpl-brochure-binder/main?labpath=MatplotlibExample.ipynb) that can zoom, pan, update, visual style and layout.

**Time:** provides a function for getting local time from the number of seconds.

**Gspread:** It enables you to easily pull data from Google spreadsheets into Data Frames and push data into spreadsheets from Data Frames. Gspread is a Python API for Google Sheets. Features: Google Sheets API v4. Open a spreadsheet by title, key or URL. Read, write, and format cell ranges.

**Cvlib**: A simple, high-level, easy-to-use open-source Computer Vision library for Python.

Developed with a focus on enabling easy and fast experimentation. Keras(deep learning library) inspired it.

**Execution Steps:**

1. We import our libraries, cv2, NumPy, time, matplotlib etc.
2. We declare some variables like width\_min(minimum width), height\_min(minimum\_height), offset(minimum error), count(initially, we advise it 0, and then it is increased) and set the position of a line.
3. We make a list named detec; if anything is detected, it gets appended.
4. We make a function find\_center of a rectangle made on a vehicle after detecting and returning the centre points of a rectangle in the x and y direction.
5. We extract or open video from our system using videocapture in OpenCV.
6. Then we use one algorithm in cv2, known as a subtractor, to subtract the background of our object in the video.
7. We run a loop to read video (as we know, video is a set of collections of frames) and for performing functions.
8. We use cvtColor to convert colour to greyscale to increase accuracy, and then we store that frame in a variable named grey and blur that image by using GaussianBlur to reduce size.
9. Then we use one function dilate for broadening an image by using a specific structure element that also determines the shape of the pixel.
10. We use getStructuringElement and MORPH\_ELLIPSE to process images based on image shape; each pixel of the image pixel is adjusted based on the value of other pixels in its neighbourhood.
11. We use the morphologyEx method that gives black white phase in the backend shown in the video and then by findContours used to count the number of objects.
12. Then we make a function named enumerate for making rectangles on vehicles after detecting and putting text on the rectangle(VEHICLE COUNT: ) and putting a red circle in the centre of the rectangle.
13. We also print the count down on our terminal as Output.
14. In the final step, we release all frames, and by using destroyAllWindows, we remove all windows after completing all tasks.

Cv2.cvtColor

(Convert Color into Grey Scale

Run while loop and divide video into frames

(As we know video is set of collection of frames)

Cv2.bgsem. createBackgroundSubtractorMOG ()

(To remove background of our object in Video)

VideoCapture

(Input the Video)

Gaussian Blur

(Blurring an image and reducing the size of image)

Cv2.dilate

(Dilate the source image using the specific structuring element)

Cv2.getStructuringelement

(It is an elliptical structuring element, that is a filled ellipse inscribed into the rectangle)

morphologyEx

(Black and white phase of video in backend)

findContours

(Count the number of objects after the image form getting to subtractor)

Cv2.line

(Making a line)

Run for loop

(For putting the rectangle on every vehicle and find centre of it)

Run for loop

(For printing the output and count the number of vehicles on window)

putText

(For putting the text on video or image frames)

Imshow

(For showing or display that frames and videos)

destroyAllWindows

(After completing all task just to release all window)