YOLOv3 Darknet for Car Detection and Counting from COCO Dataset

# What is the Use Case?

The use case involves detecting and counting cars in a video titled "Driving Downtown - New York City 4K - USA" using a pre-trained object detection model. The objective is to accurately identify instances of cars in the video and assign unique identifiers (e.g., Car1, Car2) to each detected vehicle. This task is valuable for traffic monitoring, automated vehicle counting, and analyzing vehicle flow in urban environments.

# Which Model is Used?

The model used for this task is YOLOv3 (You Only Look Once, version 3) implemented in the Darknet framework. YOLOv3 is selected due to its efficiency and effectiveness in real-time object detection.

## Explain the Model in Short

YOLOv3 is a cutting-edge real-time object detection system that divides an image into a grid and predicts bounding boxes and class probabilities for each grid cell. It consists of 106 layers, which contributes to its improved accuracy over its predecessors. YOLOv3 employs the Darknet-53 backbone, a convolutional neural network with 53 layers trained on ImageNet, allowing it to extract features robustly. The model predicts bounding boxes at three different scales, which enhances its capability to detect objects of various sizes.

## What is the COCO Dataset?

The COCO (Common Objects in Context) dataset is a comprehensive large-scale dataset used for object detection, segmentation, and captioning. It includes over 200,000 labeled images covering more than 80 object categories. The dataset features a wide variety of everyday objects in diverse settings, providing a realistic and challenging environment for training and evaluating deep learning models.

## Which Class is Detecting?

In this use case, we are focusing on detecting the "car" class from the COCO dataset. Among the 80 different classes available in COCO, our objective is to identify and count instances of cars within the video "Driving Downtown - New York City 4K - USA." Each detected car is assigned a unique identifier for counting purposes (e.g., Car1, Car2).

## Frames Per Second (FPS)

The Frames Per Second (FPS) is calculated by dividing the number of rendered frames by the total seconds passed. In this instance, 200 frames were processed over a duration of 168.98 seconds.

*FPS=Number of Frames/Seconds Passed*

*=200/168.98*

*=1.19*

Therefore, the FPS for this detection and counting task is approximately 1.19.