YOLOv7 PyTorch for Object Detection from COCO Dataset

# What is the Use Case?

The use case involves detecting and identifying specific objects in a video titled "Driving Downtown - New York City 4K - USA" using a pre-trained object detection model. This is particularly useful for applications such as traffic analysis, urban planning, autonomous vehicle navigation, and public safety. The goal is to accurately identify instances of multiple object classes from the COCO dataset in the video frames.

# Which Model is Used?

The model used for this task is YOLOv7 (You Only Look Once, version 7) implemented in PyTorch. YOLOv7 is known for its superior accuracy and efficiency in object detection tasks.

# Explain the Model in Short

YOLOv7 is an advanced real-time object detection system that uses a single convolutional neural network (CNN) to predict multiple bounding boxes and class probabilities for each image. YOLOv7 improves upon previous versions with enhanced network architecture, training techniques, and optimization strategies, making it more accurate and faster. The model processes images in a single pass, which allows it to achieve high speed while maintaining high accuracy, making it suitable for real-time applications.

# What is the COCO Dataset?

The COCO (Common Objects in Context) dataset is a large-scale object detection, segmentation, and captioning dataset. It contains over 200,000 labeled images with more than 80 object categories and 91 classes in total. The dataset is widely used in the research community for benchmarking and training deep learning models for

various computer vision tasks. The images in COCO are diverse, capturing objects in their natural context, making it a challenging and realistic dataset for model training and evaluation.

# Which Classes are Detecting?

In this specific use case, we are focusing on detecting the following classes from the COCO dataset:

* + Backpack
  + Baseball glove
  + Bench
  + Bicycle
  + Bird
  + Bottle
  + Bus
  + Car
  + Cell phone
  + Clock
  + Door
  + Eye glasses
  + Fire hydrant
  + Handbag
  + Hat
  + Horse
  + Motorcycle
  + Parking meter
  + Person
  + Potted plant
  + Shoe
  + Stop sign
  + Street sign
  + Suitcase
  + Teddy bear
  + Traffic light
  + Train
  + Truck
  + TV
  + Umbrella
  + Window

These classes were selected based on their occurrence in the video "Driving Downtown - New York City 4K - USA".

# Frames Per Second (FPS)

The Frames Per Second (FPS) is calculated by taking the number of rendered frames and dividing it by the seconds passed. In this use case, 150 frames were processed in a total of 227.877 seconds.

FPS=Seconds Passed / Number of Frames FPS=227.877150

FPS=0.66

Therefore, the FPS for this detection task is approximately 0.66.