Smart Vision: Real-Time Object Detection with YOLOv3 on COCO Dataset Using OpenCV and Python

Objective:

The project aims to implement a real-time object detection system using the YOLOv3 deep learning model, OpenCV, and the COCO dataset. The primary objectives are:

1. **Real-Time Object Detection** – Detect and identify multiple objects in live video streams with high accuracy and low latency.
2. **Enhance Visual Intelligence** – Improve surveillance, automation, and monitoring capabilities with AI-powered object detection.
3. **Resource Optimization** – Utilize lightweight frameworks and optimized models for performance on edge devices.
4. **Wide Object Coverage** – Enable detection across 80 different object categories from the COCO dataset.
5. **Scalable Integration** – Make the system compatible with diverse applications like security, retail analytics, and traffic monitoring.

Methodology:

The system is developed in a structured manner to ensure speed, precision, and adaptability. The core methodology includes the following components:

1. **Dataset and Label Preparation**
    - The COCO (Common Objects in Context) dataset is used, which includes over 80 object classes such as people, vehicles, animals, and household items.
    - Object labels are parsed from a label file to ensure consistent class naming.
2. **Model Selection and Configuration**
    - YOLOv3 is selected due to its balance between speed and accuracy, making it suitable for real-time applications.
    - The system uses the official configuration file and pre-trained weights trained on the COCO dataset.
    - OpenCV's DNN module is utilized for efficient loading and real-time processing of the model.
3. **Detection Pipeline and Object Recognition**
    - Input frames are captured from CCTV footage, webcams, or video files.
    - Each frame is processed to predict bounding boxes and object classes.
    - Non-Maximum Suppression (NMS) is applied to refine results and reduce redundancy.

- Detected objects are labeled by matching the predictions with COCO dataset class names.

4. **Performance Optimization and Visualization**
    - OpenCV's optimization features are used to achieve high-speed inference.
    - Real-time feedback is provided through visual overlays, such as bounding boxes and class labels.
    - Frame performance is monitored to ensure smooth detection and tracking.
5. **Data Output and Scalability**
    - Detected objects and timestamps can be logged for future analysis.
    - The system can be extended to support data dashboards or cloud integration for remote monitoring.
    - Designed to be scalable across platforms such as embedded systems, servers, or edge devices.

Reference:

1. **Redmon, J., & Farhadi, A. (2018)** – YOLOv3: An Incremental Improvement
2. **Joseph Redmon et al.** – You Only Look Once: Unified, Real-Time Object Detection (arXiv:1506.02640)
3. **COCO Dataset** – Common Objects in Context (cocodataset.org)

Team:

1. **MRIDUL SHARMA** – E22CSEU0149
2. **AARYANAN SHARMA** – E22CSEU0147