OBJECT DETECTION AND COUNTING

-Neha Sara Sam

**OBJECTIVE:**

This project aims to detect and classify objects from a video file using the YOLOv10s model from the ultralytics library. The system processes each video frame, detects and classifies them according to their classes. These detected class names along with their timestamps are logged into a CSV file. It also keeps track of how many times each object appears.

**LIBRARIES USED:**

1. ultralytics
2. YOLOv10
3. OpenCV
4. datetime
5. collections.defaultdict
6. random
7. psycopg2

**1.ultralytics:**

It is a package which is used to access and run the YOLO (You Only Look Once) family of object detection models, including YOLOv10. It provides a simple python interface to run detection on images, videos and webcam input. It loads the “yolov10s.pt” pre-trained model and allows detection with a simple call of “model(frame)”.

**2.YOLOv10 (You Only Look Once, v10):**

It is an object detection model introduced in 2024. It detects objects in images and videos with bounding boxes and class labels. It is known for being fast, lightweight and accurate. The “yolov10s.pt” model used is a small version that runs faster and is suited for real-time or video processing tasks.

**3.OpenCV (Open Source Computer Vision):**

It is an open-source library used for computer vision and image processing. It supports video processing operations and integrates with many deep learning models. It captures video frames, draws bounding boxes and labels on detected objects according to their classes. It displays the video output in a window using the “cv2.imshow()” function and releases video & destroys windows using “cap.release()” and “cv2.destroyAllWindows()” functions.

**4.datetime:**

It is a module which is used to handle dates and timestamps in python. It retrieves the current date and time for each detection. Formats timestamps into human-readable strings for saving to a CSV file.

**5.collections.defaultdict:**

defaultdict is a subclass of Python’s dictionary that provides a default value for missing keys. It automatically initializes missing keys with a default value. It simplifies counting objects without manual checks.

**6.random:**

This module is used to generate random numbers. It assigns random RGB colours to each object class for visualization. It helps in seeding with “hash(class\_name)”, which ensures that the same colour is used consistently for each class.

**7.psycopg2:**

It is a powerful and popular PostgreSQL database adapter which is used in Python programming language. It connects python programs to a PostgreSQL database and execute SQL commands such as inserting, updating, deleting and retrieving data. It is known for its fast and efficient database operations with proper error handling.

**DATABASE DESCRIPTION:**

The code uses a **PostgreSQL** relational database to store detection results. For every object detected in the video, the details recorded are:

* Object: The name or class label of the detected object.
* Timestamp: The exact date and time when the object appeared in the video.

Each detection is inserted into a table called “OBJECT DETECTION” using the **psycopg2** library. This database helps in storing large volumes of detection records in an organized manner and allows further analysis using SQL queries.

**OUTPUT:**

* A video with bounding boxes of different colours and class labels for detected objects is obtained.
* All the detected object class names with timestamps are logged into the CSV file (detections.csv).
* A PostgreSQL database table (OBJECT DETECTION) with all detections is stored.
* Total number of counts for each object class detected is shown in the terminal.

**CONCLUSION:**

In conclusion, this project provided a practical understanding of object detection and counting using the YOLOv10 model. By integrating OpenCV for video processing and psycopg2 for PostgreSQL database interaction, a system that detects, counts and stores object data with timestamps was developed. This provides an efficient way to automate object tracking and data storage for future analysis or monitoring applications.