Human & Object Detection using YOLOv8

1. Objective

The aim of this assignment is to:

- Detect **humans and common objects** in images or videos
- Draw **bounding boxes** using OpenCV
- Perform **basic human posture classification** (optional)

A **pre-trained YOLOv8 model** is used to achieve fast, accurate detection on the CPU.

---

2. Methodology

Object Detection

- A **YOLOv8 Nano (yolov8n)** pre-trained model is used
- A video clip is passed to the model
- The model outputs bounding boxes, class labels, and confidence scores

Posture Classification

- Posture is inferred using a heuristic approach of **bounding box aspect ratio**
- For detected persons:
    - Taller boxes → **Standing**
    - Medium ratio → **Sitting**
    - Wider boxes → **Bending**

---

3. Environment Setup

Requirements

- Python 3.10 (Anaconda recommended)
- CPU-only system

Install Dependencies (Run Once)

```
conda install -c conda-forge numpy opencv -y

pip install ultralytics
```

Restart the Jupyter kernel before running the notebook.

---

4. Input and Output

Input

- A short video clip is used in this assignment

Output

- **Image Output**:
    - Bounding boxes with class labels and confidence scores
    - Posture label for detected humans (not used a relevant video for this task)
- **Video Output**:
    - Processed video (output_detected.mp4)
    - Bounding boxes and posture labels on each frame
    - Output saved in the notebook directory

---

5. Results

- YOLOv8 successfully detects humans and objects
- Bounding boxes are accurately drawn in real time
- The system runs efficiently on the CPU

---

6. Limitations

- Heuristic posture classification may fail for extreme angles or occlusions
- No person tracking across frames

---

7. Future Improvements

1. **YOLOv8-Pose** can be integrated for accurate posture detection
2. Add **person tracking** for smoother video analysis