**Name :** Chandrakant Dattatrey Thakare  
**Roll No.:**382014 **PRN:**22310303

**Class :** TY CSE AI **Batch :** B1

**Subject :** DL Lab Assignments

**Assignment 07**

**Object Detection using YOLO and Pretrained Model**

**Problem Statement**

The task of this assignment is to implement an object detection system using the **YOLO (You Only Look Once)** deep learning algorithm. The aim is to apply a pretrained YOLOv8 model on real-world images or videos and evaluate its ability to detect and classify multiple objects in a single frame. Unlike simple classification models, object detection not only identifies the type of object but also predicts its location with bounding boxes.

**Objective**

* To explore YOLO as a real-time object detection framework.
* To apply a pretrained YOLOv8 model on test images and videos.
* To visualize the predictions with bounding boxes and confidence scores.
* To understand how pretrained models can be used without additional training.

**Tools and Resources**

* **Software Used:** Google Colab / Jupyter Notebook
* **Libraries:** Ultralytics (YOLOv8), OpenCV, Pillow
* **Pretrained Weights:** yolov8n.pt (Nano version for faster inference, trained on COCO dataset)
* **Dataset:** Sample images (bus, persons, cars, etc.) provided from Ultralytics or custom images uploaded.

**Methodology**

1. **Install Dependencies**
   * Installed the Ultralytics YOLOv8 library.
   * Imported required packages (YOLO, PIL.Image, cv2).
2. **Load Pretrained Model**
   * Initialized the pretrained YOLOv8 model using:
   * model = YOLO("yolov8n.pt")
3. **Run Object Detection**
   * Downloaded a test image (bus.jpg).
   * Ran inference using:
   * results = model.predict("bus.jpg", save=True, conf=0.25)
   * This generated an annotated image with bounding boxes, labels, and confidence scores.
4. **Visualize Results**
   * Displayed the annotated image directly inside the notebook.
   * Saved outputs were stored automatically in runs/detect/predict.
5. **(Optional Extensions)**
   * The same approach can be extended to detect objects in a folder of images or video files.
   * Real-time detection using webcam was also tested (on local system).

**Results**

* The pretrained YOLOv8 model successfully detected multiple objects such as **bus** and **persons** in the input image.
* Bounding boxes were drawn with labels and confidence values.
* The model was efficient, producing results in real-time.
* Since YOLOv8 was trained on the COCO dataset (80 object classes), it could detect a wide variety of objects without additional training.

**Conclusion**

In this assignment, we implemented object detection using the **YOLOv8 pretrained model**. The experiment showed how YOLO can accurately detect and classify multiple objects within a single frame. The pretrained approach was efficient, requiring no dataset preparation or model training while still achieving high accuracy. This assignment highlighted the practical use of pretrained deep learning models for object detection in real-world applications such as traffic monitoring, security systems, and autonomous driving.