**Assignment No: - 7**

**Name:** Rushiraj Sunil Suwarnkar  
**PRN:** 22310315  
**Roll No:** 381018  
**Title:** Object Detection using YOLO and Pretrained Model

**Problem Statement**

Implementing real-time object detection using the YOLO (You Only Look Once) algorithm with a pretrained YOLOv8 model for detecting and classifying objects from webcam video feed.

**Objective**

* To understand the YOLO object detection algorithm.
* To use a pretrained YOLOv8 model for object detection.
* To capture real-time video feed and process frames using OpenCV.
* To visualize detected objects with bounding boxes and confidence scores.
* To evaluate the effectiveness of pretrained models in real-time applications.

**S/W Packages and H/W apparatus used**

* **Operating System:** Windows/Linux/MacOS
* **Kernel:** Python 3.x
* **Tools:** Google Colab, Jupyter Notebook
* **Hardware:** Webcam, CPU/GPU with minimum 8GB RAM recommended
* **Libraries:** Ultralytics YOLOv8, OpenCV, NumPy, TensorFlow, PIL, IPython

**Theory**

**YOLO (You Only Look Once)** is a real-time object detection algorithm that predicts bounding boxes and class probabilities directly from input images in a single forward pass of the network.

* **Key Features of YOLO:**
  + Processes the entire image at once, making it extremely fast.
  + Uses bounding box regression and classification simultaneously.
  + Generalizes well to unseen images due to global context analysis.
* **YOLOv8 Pretrained Model:**
  + Trained on the COCO dataset with 80 object classes.
  + Provides pretrained weights (yolov8s.pt) for fast deployment.
  + Outputs bounding box coordinates, class labels, and confidence scores.

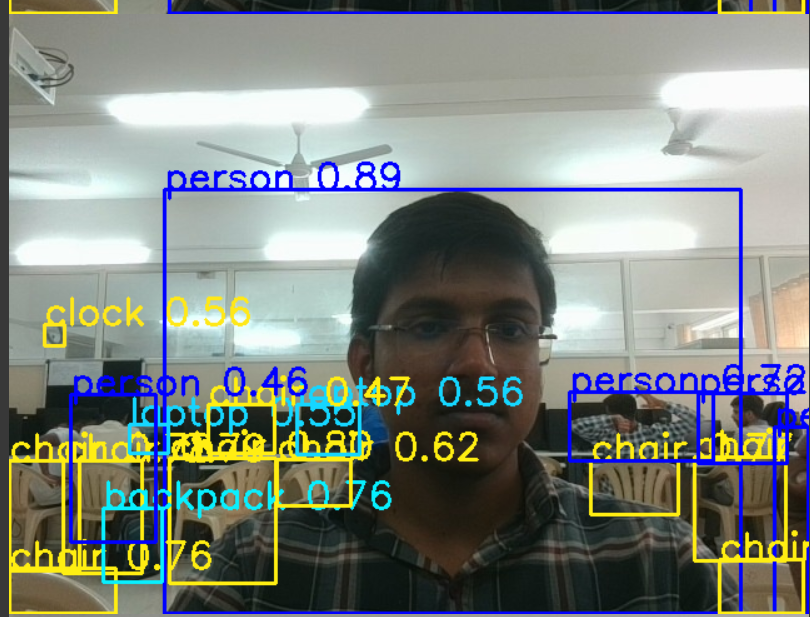
**Methodology**

1. **Installation & Setup:** Installed Ultralytics YOLO library and imported OpenCV, NumPy, and PIL for image handling.
2. **Camera Capture:** Used JavaScript + Google Colab integration to capture webcam frames.
3. **Preprocessing:** Converted captured frames to OpenCV format for YOLO processing.
4. **Model Loading:** Loaded pretrained YOLOv8s model (yolov8s.pt).
5. **Object Detection:**
   * For each frame, YOLO predicted bounding boxes, classes, and confidence scores.
   * Objects with confidence > 40% were retained.
   * Bounding boxes and labels were drawn on frames.
6. **Display:** Displayed real-time video stream with detected objects and confidence values.
7. **Exit Condition:** Detection loop stopped when user pressed q.

**Results**

* Successfully detected multiple objects in real-time from webcam feed.
* Objects were labeled with class names (e.g., *person, car, bottle*) and corresponding confidence scores.
* Bounding boxes were drawn in different colors for each class.

**Sample Output :**



Detected: person 0.87

Detected: cell phone 0.75

Detected: bottle 0.62

**Advantages**

* High speed and real-time detection capability.
* Pretrained YOLOv8 provides state-of-the-art accuracy.
* Can detect multiple objects in a single frame.

**Limitations**

* Performance depends on hardware (GPU recommended for smooth real-time processing).
* Accuracy may drop in poor lighting or occlusion conditions.
* Pretrained model limited to classes from the COCO dataset.

**Applications**

* Real-time surveillance and security systems.
* Autonomous vehicles (object detection for navigation).
* Robotics and industrial automation.
* Smart retail (product recognition, inventory monitoring).

**Working / Algorithm**

1. Import required libraries (Ultralytics, OpenCV, NumPy, PIL).
2. Load pretrained YOLOv8 model (yolov8s.pt).
3. Capture video frames using webcam (via JavaScript in Colab).
4. Preprocess and send frames to YOLO model.
5. Detect objects → extract bounding boxes, labels, and confidence.
6. Draw bounding boxes and display annotated video stream.
7. Stop detection when exit key is pressed.

**Conclusion**

The YOLOv8-based object detection system successfully detected and classified objects in real-time using a webcam feed. The pretrained model provided accurate results with minimal setup effort. This demonstrates the efficiency and versatility of YOLO for real-world applications in surveillance, automation, and computer vision tasks.