EX:No.4 DATE: 15/4/25	Object Detection with Realtime Object Detection

AIM:

To build and train a model for object detection with real time example.

ALGORITHM:

- **Import Libraries**

- TensorFlow/Keras, OpenCV, NumPy, etc.

- **Load & Preprocess Data**

- Load images and labels.
- Convert to grayscale or normalize color.
- Resize images to a fixed size.
- Encode labels (e.g., one-hot).
- Split into training/testing sets.

- Load YOLOv5 model (can change to yolov5s, yolov5m, yolov5l, yolov5x)

- Evaluate Model

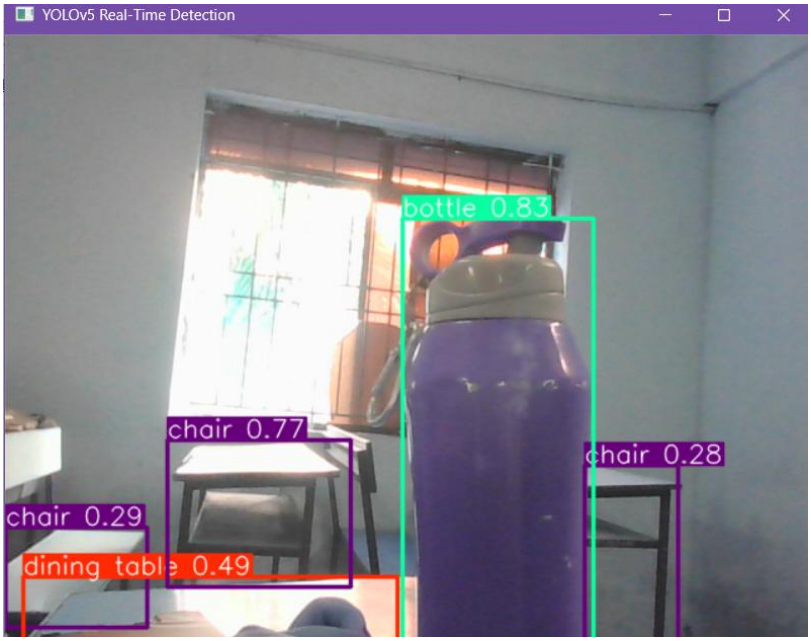
- Test on unseen camera realtime data.
- Preprocess new image
- Use model to predict identity (highest softmax score)

CODE:

```
import torch
import cv2
# Load YOLOv5 model (can change to yolov5s, yolov5m, yolov5l, yolov5x)
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
# Set to eval mode
model.eval()
# Open webcam (0 = default camera)
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("Error: Could not open webcam.")
    exit()
while True:
```

```
        ret, frame = cap.read()
        if not ret:
            break
        # Convert BGR to RGB
        img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        # Inference
        results = model(img)
        # Draw results on the original frame
        annotated_frame = results.render()[0]
        # Show result
        cv2.imshow('YOLOv5 Real-Time Detection', annotated_frame)
        # Exit on 'q' key
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

OUTPUT:



RESULT:

Thus the program has been completed and verified successfully.