```
!pip install ultralytics opencv-python-headless
!wget -O sample_video.mp4 https://filesamples.com/samples/video/mp4/sample_640x360.mp4

from ultralytics import YOLO
import cv2

model = YOLO('yolov8n.pt')

cap = cv2.VideoCapture('sample_video.mp4')
if not cap.isOpened():
    print(" Error: Cannot open video.")
else:
    print(" Video loaded successfully.")

    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    out = cv2.VideoWriter('output_task4.mp4', fourcc, fps, (width, height))

    frame_count = 0
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        results = model(frame)[0]
        boxes = results.boxes.xyxy.cpu().numpy().astype(int)
        classes = results.boxes.cls.cpu().numpy().astype(int)

        for box, cls in zip(boxes, classes):
            x1, y1, x2, y2 = box
            label = model.names[cls]
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, label, (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

        out.write(frame)
        frame_count += 1

    cap.release()
    out.release()
    print(f" Task 4 completed, processed {frame_count} frames. Download 'output_task4.mp4' from the sidebar to view object detection.")
```

```
0: 384x640 1 person, 7.7ms
Speed: 1.8ms preprocess, 7.7ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 person, 7.1ms
Speed: 1.6ms preprocess, 7.1ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)
 Task 4 completed, processed 400 frames. Download 'output_task4.mp4' from the sidebar to view object detection.
```