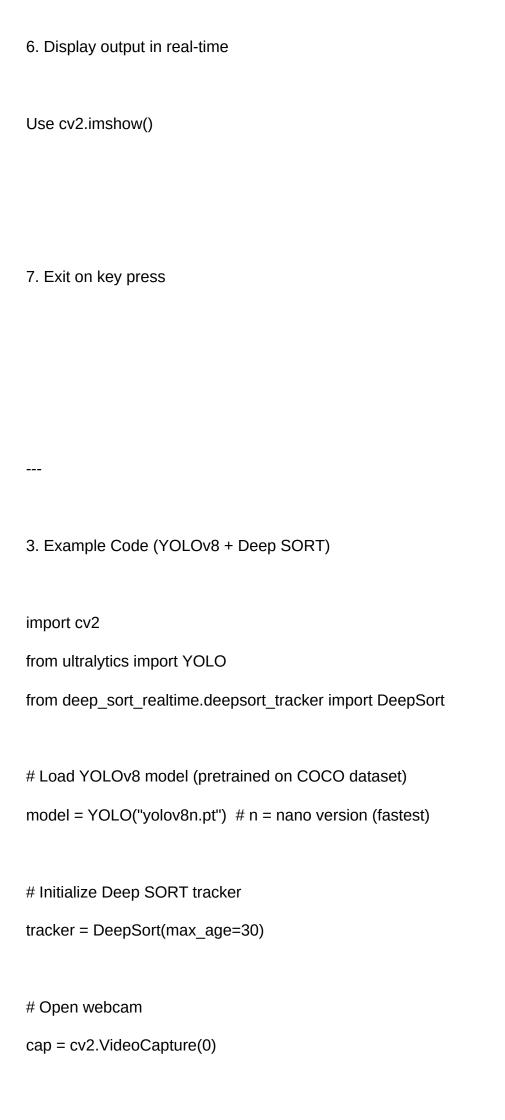
1. Tech Stack
Python
OpenCV → Video capture & drawing
YOLOv8 or Faster R-CNN → Object detection (YOLOv8 is faster for real-time work)
Deep SORT → Multi-object tracking
2. Workflow
1. Set up video input
Use cv2.VideoCapture(0) for webcam
Or load a video file

Implementation Plan

2. Load pre-trained model
YOLOv8 via Ultralytics
Or Faster R-CNN from torchvision
3. Detect objects in each frame
4. Track detected objects
Pass detection results (bounding boxes, confidence, class IDs) into SORT/Deep SORT tracker
Maintain tracking IDs over frames
5. Draw results on each frame
Bounding boxes
Labels (object name + ID)



```
while cap.isOpened():
  ret, frame = cap.read()
  if not ret:
     break
  # Run YOLO detection
  results = model(frame, stream=True)
  detections = []
  for r in results:
     for box in r.boxes:
       x1, y1, x2, y2 = box.xyxy[0]
       conf = float(box.conf[0])
       cls_id = int(box.cls[0])
       label = model.names[cls id]
       # Append in Deep SORT format: [x1, y1, x2, y2, confidence, class]
       detections.append(([x1, y1, x2, y2], conf, label))
  # Update tracker
  tracks = tracker.update_tracks(detections, frame=frame)
  # Draw boxes
  for track in tracks:
     if track.is_confirmed() and track.time_since_update <= 1:</pre>
       x1, y1, x2, y2 = map(int, track.to_ltrb())
```

```
track_id = track.track_id
       label = track.get_det_class()
       cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
       cv2.putText(frame,\ f"\{label\}\ ID:\{track\_id\}",\ (x1,\ y1-10),
               cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
  # Show frame
  cv2.imshow("Object Detection & Tracking", frame)
  if cv2.waitKey(1) \& 0xFF == ord("q"):
     break
cap.release()
cv2.destroyAllWindows()
4. Dependencies
pip install ultralytics deep-sort-realtime opencv-python
5. Notes
```

YOLOv8n is fast but less accurate; you can use yolov8s.pt or yolov8m.pt for better accuracy (slower).
Deep SORT uses appearance + motion to keep consistent IDs even when objects overlap.
If using a video file instead of a webcam:
cap = cv2.VideoCapture("video.mp4")