

YOLOv8 Object Detection – Documentation

Overview

This project demonstrates real-time object detection using the **YOLOv8 (nano)** model on a sample video. The script processes each video frame, detects objects, and displays them with bounding boxes and class labels based on the **built-in YOLOv8 pretrained model**.

Approach & Methodology

- **Model Used:** [YOLOv8n](#) – lightweight, pre-trained on **COCO classes**.
- **Use Case:** Detect and label multiple common objects (e.g., person, car, dog) from a recorded video.
- **Libraries Used:** OpenCV, cvzone, Ultralytics, NumPy, and math.

Input Data & Preprocessing

- **Input Source:** A .mp4 video file loaded via OpenCV.
- **Frame Resizing:** Scales down each frame to 30% of the original dimensions for faster processing.
- **Detection Threshold:** Only displays objects with confidence score > 0.4.

Model Details & Logic

- **Model Source:** Pretrained YOLOv8n model (yolov8n.pt) from Ultralytics (no training was done).
- **Inference:** Frame-wise object detection using stream=True mode.
- **Object Classes:** YOLOv8n is pretrained to detect 80 COCO-based classes (like "person", "car", "dog", etc.).

- Detection Steps:
 - Get bounding box coordinates and class index.
 - Filter by confidence score.
 - Use `cvzone` to draw styled bounding boxes and display labels.

Performance & Results

- Efficiency: Real-time object detection even on modest hardware.
- Detection Accuracy: Performs well for common objects seen in the video (e.g., vehicles, people).
- Visualization: Objects are shown with:
 - Blue bounding boxes
 - Class name (e.g., "car", "person")
 - Confidence score