OBJECTNET

PROBLEM STATEMENT

Object detection is a critical task in computer vision, used in applications like autonomous vehicles, security surveillance, and traffic monitoring. The challenge lies in accurately detecting multiple objects in an image while ensuring real-time performance.

PROJECT OVERVIEW

This project uses Streamlit, YOLOv8 (Ultralytics), TensorFlow.js, and OpenCV to provide a real-time object detection system. Allows seamless model selection and dynamic switching based on input type. Identifies objects with labeled bounding boxes for precise classification.

SOLUTION OFFERED

The solution contains of 3 parts

1) **Image Processing Module :**

   Accepts user-uploaded images and converts them into a format suitable for deep learning models.
   Uses OpenCV and NumPy for preprocessing (resizing, color conversion).

2) **Object Detection Module :**

   Loads YOLOv8 models (Pretrained & Custom-trained) using Ultralytics.
   Runs inference on images and detects multiple objects.
   Applies CNN-based feature extraction to identify objects with confidence scores.

3) **User Interface (UI) Module :**

   Allows model selection and real-time image display.
   Provides detection results in an easy-to-understand format.

WHO ARE THE END USERS?

- Autonomous Vehicles – For detecting objects on roads.
- Security & Surveillance – Detecting intruders in real-time.
- Traffic Analysis – Identifying vehicles, pedestrians, and signs.
- Retail & Smart Cities – Object tracking for automation.

TECHNOLOGY USED TO SOLVE THE PROBLEM

1) **Deep Learning (Object Detection)**

- YOLOv8 (You Only Look Once - Version 8): Works in real-time for detecting objects in images.
- Torch (PyTorch) & Ultralytics Library : Loads and runs pretrained & custom YOLO models. Provides high-speed inference for object detection.

2) **Image Processing**

- OpenCV : Used for image manipulation and visualization.Draws bounding boxes around detected objects.
- NumPy : Converts images into numerical arrays for deep learning models.

3) **Web Development (UI)**

- Streamlit (Python Framework): Provides a web-based UI for easy interaction. Allows image uploads, model selection, and visualization.
- HTML, CSS, JavaScript : JavaScript can be used for real-time detection in the browser with TensorFlow.js.