Real-Time Object Detection and Tracking System

Introduction

The proposed project focuses on developing a robust and innovative real-time object detection and tracking system. The system aims to detect multiple objects within images or video frames, classify them into predefined categories, localize their positions using bounding boxes, and track their movements across consecutive video frames. This project addresses challenges such as diverse environments, dynamic object movement, and real-time processing, making it suitable for applications like surveillance, autonomous driving, and smart city management.

Problem Statement

Core Problem:
The primary objective is to design a system that can:

- Detect and classify objects within an image or video frame.
- Localize detected objects with bounding boxes and confidence scores.
- Continuously track objects across video frames.
- Perform these tasks in real-time or near real-time.

Challenges:
To achieve practical usability, the system must handle:

- Diverse environmental conditions (lighting, occlusions, and angles).
- Real-time performance for efficient processing.
- High precision and robustness to minimize errors and false detections.

Innovative Approach

The solution employs state-of-the-art deep learning techniques for object detection and tracking:

Object Detection Models:

- **YOLO - World(real time open vocabulary object detection)**: Balancing speed and accuracy for real-time applications.
- **EfficientDet**: Optimized for resource-constrained environments.
- **Faster R-CNN**: For high-accuracy detection and localization tasks.

Here we will be using **Ensemble Learning** for more accurate prediction of the Object Detection Models.

Object Tracking Algorithms:

- **SORT (Simple Online and Realtime Tracking)**: For lightweight, real-time tracking.
- **Deep SORT**: Enhanced tracking using appearance-based features for improved accuracy under occlusions.

Performance Optimization:

- **Model Quantization and Pruning**: Reducing model size for faster inference.
- **Edge Computing**: Using platforms like NVIDIA Jetson or Google Coral for low-latency processing.

Key Features

1. **Image and Video Upload**: Supports file uploads for detection and tracking.
2. **Real-Time Video Stream**: Detects and tracks objects in live video streams.
3. **Dynamic Result Display**: Provides bounding boxes, class labels, and confidence scores for each detected object.
4. **Tracking Across Frames**: Tracks moving objects dynamically within video sequences.
5. **User-Friendly Interface**: Intuitive UI with responsive design for seamless user interaction.
6. **Backend Communication**: Efficient communication with the server for detection and tracking tasks.

7. **Edge Device Optimization**: Ensures real-time performance on resource-constrained hardware.

Technical Stack

- **Frameworks and Libraries**: TensorFlow, PyTorch, OpenCV.
- **Model Deployment**: TensorRT, TensorFlow Lite, or ONNX for inference acceleration.
- **Datasets**: COCO for diverse object classes and KITTI for autonomous driving scenarios.

Expected Outcomes

This project will deliver a scalable and robust real-time object detection and tracking system. Key outcomes include:

- High-accuracy detection and tracking of objects in various environments.
- Real-time performance suitable for edge devices.
- A modular and scalable design adaptable to different domains like surveillance, autonomous systems, and smart cities.

By focusing on efficiency, precision, and scalability, this project offers an innovative solution for real-world applications.