Vehicle Counting System Documentation

Introduction

This document provides an overview of a vehicle counting system using the YOLOv8 object detection model and additional tools for video processing and annotation. The system is designed to process input video footage, detect and count vehicles, and annotate the video with the detected vehicles and their trajectories.

System Overview

Input Type

- **Input Format**: Video file.
- **Color Space**: RGB

Preprocessing Techniques

- **Frame Extraction**: Frames are extracted from the input video for individual processing.
- **Fusion of Model Layers**: The YOLOv8 model layers are fused to enhance performance.

Computer Vision Pipeline Components

1. **Frame Generator**: Extracts frames from the input video.
2. **Object Detection**: Uses the YOLOv8 model to detect vehicles in each frame.
3. **Tracking**: Applies BYTETracker to track the detected vehicles across frames.
4. **Annotation**: Annotates frames with bounding boxes, labels, and tracking information.
5. **Line Zone Counting**: Counts vehicles crossing a predefined line in the frame.

Models and Libraries Used

- **YOLOv8**: Pre-trained object detection model for detecting vehicles.
- **BYTETracker**: For tracking detected vehicles across frames.
- **Supervision (sv)**: For video processing and annotation, including frame extraction, detection handling, and annotation.

Challenges Faced

- **Real-Time Processing**: Ensuring the system processes frames quickly enough for real-time use.
- **Accuracy of Detection and Tracking**: Maintaining high accuracy in various lighting and weather conditions.

Performance Assessment

- **Inference Speed**: Measured by the time taken to process each frame.

Success and Failure Scenarios

- **Success**: Accurate detection and counting of vehicles, smooth tracking, and clear annotations.
- **Failure**: overlapping vehicles movements.

System Requirements

- **Compute**: Requires a GPU for real-time performance, but can run on a CPU with reduced speed.
- **Hosting**: Suitable for deployment on cloud platforms or high-performance edge devices.

Use Cases and Applications

- **Traffic Monitoring**: Real-time vehicle counting and monitoring for traffic management.
- **Smart City Infrastructure**: Integration with smart city systems for automated traffic control.
- **Toll Collection**: Vehicle counting and classification for automated toll systems.

System Diagram

Preliminary Performance Analysis

Qualitative Analysis

The system effectively detects and tracks vehicles in various scenarios. Annotations are clear and provide useful information for traffic analysis. However, performance may vary based on video quality, lighting conditions, and camera angles.

- **Strengths**:
  - High accuracy in vehicle detection under normal conditions.
  - Effective tracking of vehicles across frames.
  - Clear and informative annotations.
- **Weaknesses**:
  - Performance degradation in poor lighting or weather conditions.
  - Occasional tracking errors due to occlusions.

Detailed Workflow

- **Loading Pre-Trained Model**
  - The YOLOv8 model is loaded and fused for optimized performance.

- **Frame Extraction**
  - Frames are extracted from the input video using Supervision's frame generator.

- **Object Detection and Annotation**
  - Vehicles are detected using the YOLOv8 model, and results are annotated on frames.

- **tracking and Line Counting**

- BYTETracker is used for tracking vehicles, and a line zone is defined for counting.

➢ **Video Processing and Annotation**

The entire video is processed frame by frame, with detections tracked and annotated.

By following this structured approach, the system efficiently detects, tracks, and counts vehicles, providing valuable insights for various applications.