**Vehicle Detection and Speed Estimation Project using YOLO and OpenCV**

This project was started with usual steps of object detection, fine-tuning @Ultralytics YOLO11 model with @RoboFlow aerial car images dataset, then I applied post-processing steps to estimate the speed of the cars. The project leverages YOLO11 model and video processing using @OpenCV to track moving vehicles and estimate their speeds from aerial footage.

**Logic and Math for Nerds :)**

1. **Object Detection**: Vehicles are tracked in each video frame using fine-tuned YOLO11.
2. **Tracking Movement**: The center point of each vehicle is tracked between consecutive frames. The movement (in pixels) between two frames is calculated as the ***Euclidean distance*** between the current and previous center points.
3. **Speed Estimation**:
   * The pixel movement is converted into real-world distance using a conversion factor (meters per pixel).
   * Meters per pixel factor was calculated using the average length of a car: 3.48 meters and dividing it with the average pixels occupied by the length of the car.
   * Speed is calculated by dividing this real-world distance by the time between frames, resulting speed in meters per second. It's then converted to kilometers per hour (km/h).
4. **Smoothing Speed**: To avoid erratic speed changes, a **cumulative moving average (CMA)** is used to smooth the speed estimates over time to ensure stable change in speed.

This project highlights how computer vision and AI can be applied to tasks like traffic monitoring and vehicle speed analysis!

#ComputerVision #ArtificialIntelligence #YOLO #OpenCV #Python #MachineLearning #TrafficMonitoring #SpeedEstimation #DeepLearning #ObjectDetection #AIApplications #VideoProcessing #AutonomousSystems #AI