PROJECT ABSTRACT

Vehicle Speed Estimation from Traffic Camera using Computer Vision

Per-vehicle speed from a single traffic camera with no radar: YOLOv8 detection plus ByteTrack tracking, converted to km/h through a one-time homography calibration, with overspeed flagging and lane-wise analytics in a live dashboard.

Project type: Software

Availability: Built to order

Suitable branches: AI & Machine Learning

Technologies: Python 3, PyTorch, YOLOv8, ByteTrack, OpenCV homography, NumPy, HTML dashboard

Price: Request a quotation

Prepared by Projectech · projectech.in — for academic project reference.

1. Title

Vehicle Speed Estimation from Traffic Camera using Computer Vision — radar-free speed measurement from pixels and plane geometry.

2. Introduction and Background

Speed enforcement conventionally needs radar guns, inductive loops or LiDAR — expensive to install and maintain per lane. Yet the camera already watching the road captures everything needed: with a one-time geometric calibration mapping image pixels to road-plane metres (homography), tracked vehicle displacements become speeds via distance over time. Combined with modern detection (YOLOv8) and multi-object tracking (ByteTrack), a single ordinary camera becomes a speed sensor, and the calibration procedure itself is excellent viva material.

3. Problem Statement

Traffic studies and enforcement-support need per-vehicle speed measurement without costly roadside hardware. A system is needed that detects and tracks vehicles in monocular traffic video, converts pixel motion to real-world speed through calibrated geometry, flags overspeed events against the posted limit, and reports lane-wise statistics.

4. Objectives

- Detect vehicles per frame with YOLOv8 and maintain identities with ByteTrack.
- Calibrate each camera once via clicked road points (OpenCV getPerspectiveTransform).
- Estimate per-vehicle speed as smoothed displacement over time in km/h.
- Flag overspeed events with snapshots against the posted limit.
- Validate accuracy against handheld radar-gun ground truth (MAE in km/h).
- Present live speeds, readings table and lane analytics in a dashboard.

5. Proposed Methodology

The operator clicks 4+ known road points once per camera; OpenCV computes the homography from image to road-plane metres. Each frame passes through YOLOv8-m; ByteTrack associates boxes into persistent tracks. Each track's bottom-centre is projected to metres per frame; speed is the 5-frame moving-averaged displacement divided by the frame interval. Vehicles above the posted limit are flagged and snapshotted. Evaluation pairs sample clips with radar-gun readings and reports mean absolute error per condition (day, dusk, light rain).

6. System Architecture and Working

Five stages. **Calibration:** click-to-mark tool builds and validates the homography. **Detection:** YOLOv8-m finds vehicles each frame. **Tracking:** ByteTrack keeps persistent IDs. **Speed:** projected displacements are smoothed into km/h readings and compared with the limit. **Presentation:** a single-file HTML dashboard shows the tracked feed with speed labels and motion trails, the live readings table with overspeed highlighting, and analytics (speed distribution histogram, per-lane means, 85th percentiles).

7. Hardware and Software Requirements

- Laptop or desktop (GPU recommended for real-time detection; CPU workable at lower FPS).
- Python 3.10+, PyTorch, Ultralytics YOLOv8, ByteTrack, OpenCV, NumPy, Matplotlib.
- Traffic-camera footage (1080p, 15–30 FPS) or an RTSP stream.
- Tape-measured road reference points for the one-time calibration; handheld radar gun for validation (project-collected).
- Modern web browser for the dashboard.

8. Expected Outcomes

Per-vehicle speed at ±3.1 km/h MAE against radar ground truth (design target), detector mAP@0.5 of 0.84 and under 4 ID-switches per 100 vehicles (design targets). Targets are validated in the built-to-order build run; the report documents achieved figures from the radar-paired evaluation. Deliverables: pipeline, calibration tool, annotated sample clips, evaluation sheet, dashboard, report, PPT, viva Q&A.;

9. Applications

- Traffic-speed studies for urban planning without roadside hardware.
- Overspeed hotspot identification for enforcement planning.
- Campus and gated-community speed monitoring.
- Teaching projective geometry, tracking and applied evaluation.

10. Future Scope

- Automatic calibration from lane markings and known lane widths.
- Night-time operation with IR-illuminated camera models.
- Vehicle-class-specific speed limits (trucks vs cars).
- Integration with ANPR for end-to-end violation processing.