EECS 504 Project Report: GeoVision - Traffic Violation Detection System
Andrew Huang, Jinle Lin, Weirui Wan

Executive Overview

Intersections account for at least one-third of U.S. traffic fatalities, yet violation enforcement still largely relies on manual review of camera footage. We present an end-to-end, user-configurable system for automatic traffic violation detection from a single fixed traffic camera. Through an interactive calibration tool, users annotate traffic signal bulbs, stop zones, stop lines, and regions of interest; the system then detects and tracks vehicles, infers signal state, and evaluates violation logic frame by frame. Vehicle detection uses an attention-based YOLO12 backbone fine-tuned on 2,100 annotated intersection images, paired with ByteTrack for persistent identities across brief occlusions. Traffic signal state is estimated via HSV color analysis with a 10-frame majority vote, and a RANSAC-estimated homography projects tracked vehicles onto a satellite bird's-eye view to enable ground-plane reasoning about trajectories and near-collision prediction. The prototype detects red-light running, stop-sign non-compliance, and near-collision events on real traffic footage. The fine-tuned detector achieves 80.2% precision, 83.6% recall, and 84.9% mAP@50 on a held-out test set.

Background and Impact

Intersections are among the most dangerous locations in road traffic, as vehicles, pedestrians and traffic lights interact in a complex environment. In 2022, at least one third of traffic fatalities occurred at intersections. Violations such as running red lights and not obeying stop signs can lead to serious accidents, so it is crucial to effectively monitor these behaviors. However, traditional traffic law enforcement often relies on manual operations. Manually reviewing traffic surveillance videos is both cumbersome and time-consuming, and it is difficult to achieve large-scale operation when checking thousands of vehicles. Therefore, the development of automated systems for detecting traffic violations was prompted. This system can automatically identify vehicles, track their movement trajectories, and determine whether they have violated red light or stop sign regulations. This can reduce the workload of human workers and improve the efficiency of traffic monitoring. By detecting violations such as red-light running in real time, the system can discourage unsafe driving behavior and enhance road safety. In addition, it can generate valuable data on traffic patterns and violation hotspots, supporting better infrastructure design and policy decisions. The expected impact includes reduced accident rates, improved traffic efficiency, and the advancement of data-driven and intelligent transportation systems.

Despite these benefits, such systems raise important societal concerns. Continuous camera-based monitoring introduces privacy risks, and the system may produce false positives due to occlusion, lighting variation, or homography noise. There is also a risk of limited generalization across different intersections and traffic conditions. To ensure responsible deployment, the system should be used as a decision-support tool with human review required before enforcement actions are taken rather than a fully automated legal authority.

Method

Vehicle Tracking and Detection

The vehicle detector uses Ultralytics' Yolo12 as the base model. It is a single-stage anchor-free detector from the YOLO family.Unlike its predecessors which use a traditional CNN backbone, YOLOv12 introduces an attention mechanism in its backbone to capture long-range spatial dependencies. One of the challenge in our scenario is vehicles at the far end of an intersection may be as small as 30 pixels wide, and this model structure can help improve detection of partially occluded and small vehicles. The model outputs bounding boxes with class labels and confidence scores in a single forward pass. Bytetrack is for detections across frames. It performs two-stage association. In the first stage, high-confidence detections are matched using IoU overlap against existing tracked vehicles, then unmatched vehicles are given a second chance to associate with low-confidence detections. This method can help preserves tracking continuity through brief occlusions. Each vehicle is assigned a persistent ID that is used throughout all of the downstream violation logic.

EECS 504 Project Report: GeoVision - Traffic Violation Detection System
Andrew Huang, Jinle Lin, Weirui Wan

HSV thresholding

Traffic light state is estimated by analyzing the color of each signal bulb. The Hue, Saturation, Value (HSV) color space is used because hue is more robust to illumination intensity changes. A lit bulb exhibits high saturation and high value within a hue range specific to its color. The mean H, S, and V are computed inside a small circular region for each bulb position. The active bulb is the one with the highest mean V whose mean H falls within the expected range. If no bulb passes the check, the system falls back to selecting the brightest bulb by V. A 10-frames majority vote prevents single-frame misclassifications at signal transitions.

Homography Calibration

Homography is used to transform the tracked vehicles from the camera's view onto a bird's eye view satellite image. A 3x3 homography matrix maps image pixel coordinates to a flat ground plane, modeled as the road surface. Given a set of point pairs between the camera and satellite view, the matrix is estimated by solving a Direct Linear Transform system. RANSAC is used to reject outlier pairs from the estimation. Using the estimated matrix, any points on the camera view is projected to its equivalent location on the satellite view.

Prototype

The prototype is implemented in Python using OpenCV, Ultralytics, and NumPy. It contains three tools:

calibrate_H.py	Interactive correspondence picker. Alternates click focus between camera frame and satellite image. Computes H via RANSAC, displays per-point reprojection error.
define_region.py	Region annotation tool. Supports quadrilateral drawing for traffic lights, stop zones, and ROIs. Traffic bulbs are placed as single-click dots. Stop lines are drawn as 2-point segments.
run_detection.py	Main inference pipeline. Loads H and regions files, builds lane pairings, runs the tracker, and evaluates all violation conditions. Annotated camera view and satellite view are rendered.

The pretrained YOLO12s model is fine-tuned on 2100 annotated intersection images resized to 640x640. Stop line crossing is detected using the signed area of the triangle formed by the two endpoints of the stop line segment and the vehicle's position. The sign of the cross product of the segment vector with the vector to the vehicle indicates which side of the line the vehicle is on. A sign change between consecutive frames indicates the vehicle has crossed. The crossing check examines the last five positions. This handles frames where the tracker loses the vehicle momentarily which happens when occlusions occur at stop line boundaries. A stop sign requires a vehicle to come to a complete stop before proceeding. Stationarity is defined in terms of the vehicle's position over a sliding window of N frames. It is stationary if the standard deviation of its x and y coordinates over the window falls below a threshold. If the vehicle accumulates a contiguous run of stationary frames exceeding a minimum of half a second it is considered compliant. It is a violation if the vehicle exits the zone without ever achieving compliance. There is a short grace period after zone exit that prevents premature judgement caused by tracker jitter at the zone boundary. Near collision detection detects the vehicle's ground-plane trajectory using the means of frame-to-frame displacements over a short sliding window to predict the velocity. The velocity is extrapolated linearly forward in time to predict the trajectory. Two vehicles are in collision risk if their predicted positions come within a distance threshold at any future time step. The homography projection amplifies tracker jitter so an exponential moving average smoothing calculation is applied to the satellite view only.

Nuances/ Limitations

ByteTrack ID switches during heavy occlusion can reset the stop sign zone timer. This allows the vehicle to exit without being flagged if its ID changes mid-zone. Homography assumes a flat ground plane where tall vehicles such as trucks and buses are projected incorrectly. This is why bottom-center coordinates are used. Linear extrapolation fails for turning

vehicles so a car turning left is predicted to continue straight for a short moment, generating a false near-collision warning at the apex of the turn. The HSV hue detection does not work as well in direct sunlight when the glare desaturates the bulb below the saturation threshold. The 10-frames majority vote introduces a short delay at signal transitions so vehicles that move immediately on a green change may be briefly and incorrectly flagged. The pixel-based check for stationary checks is perspective-dependent. The same movement produces larger displacement for vehicles closer to the camera, so it demands greater stillness from near vehicles than far ones.It is recommended that the user provide extra dataset on the intersection to fine tune the detection model.

Results

	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
YOLO12s	80.2	83.6	84.9	67.1

Table 1. Evaluation metrics of the fine-tuned YOLO12 detection model

Table 1 shows the performance of the YOLO12 detection model on the test set.

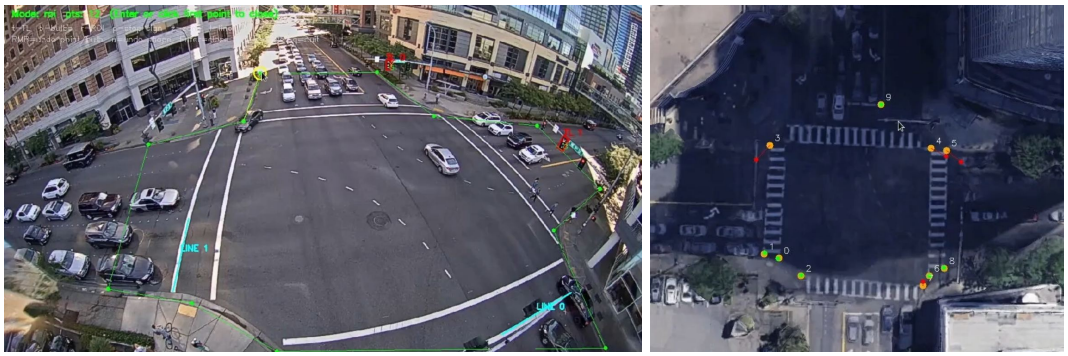


Figure 1. (Left) Defining interest regions, (Right) Selecting points for homography

Figure 2 shows the prototype running on real time traffic camera footage. The camera view displays tracked vehicles with bounding boxes and movement trails. The satellite overlay shows ground-projected trajectories, predicted paths, and near-collision warning. Figure 3 and 4 are examples of red light violation. The tracked vehicle crosses the stop line when the corresponding traffic signal is still red as shown in the figures and its bounding box turns red and indicates a violation occurred. Another traffic violation the system can detect is stop zone violation. The yellow zones in figure 5 are user-defined stop zones and vehicles must remain stationary in the zones for a certain amount of time. The image on the left shows the tracked vehicle violated the stationary requirement. The image on the right is an example where the vehicle complies with the stop requirement. Similarly, all vehicles in the ROI will be detected and projected onto the overhead view. The violation statistics are also shown on the top left corner of the camera view.

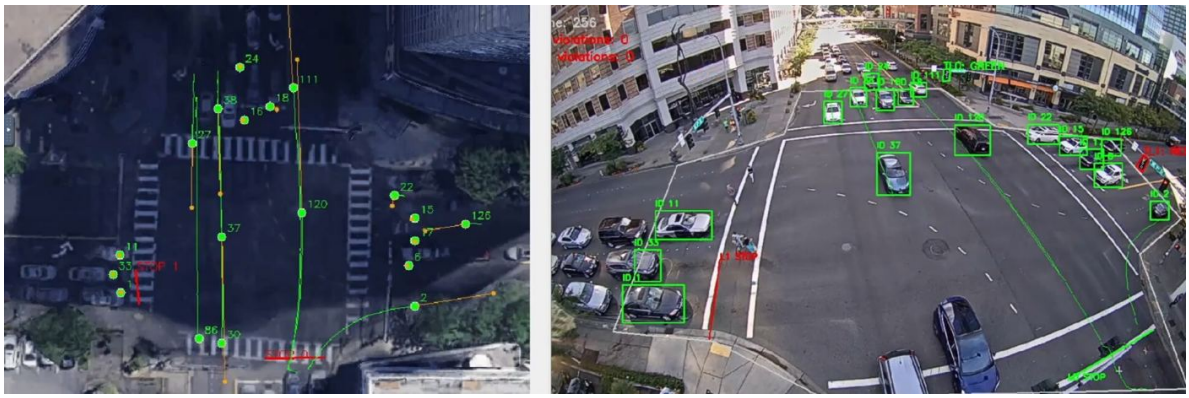


Figure 2. Vehicles detection and tracking

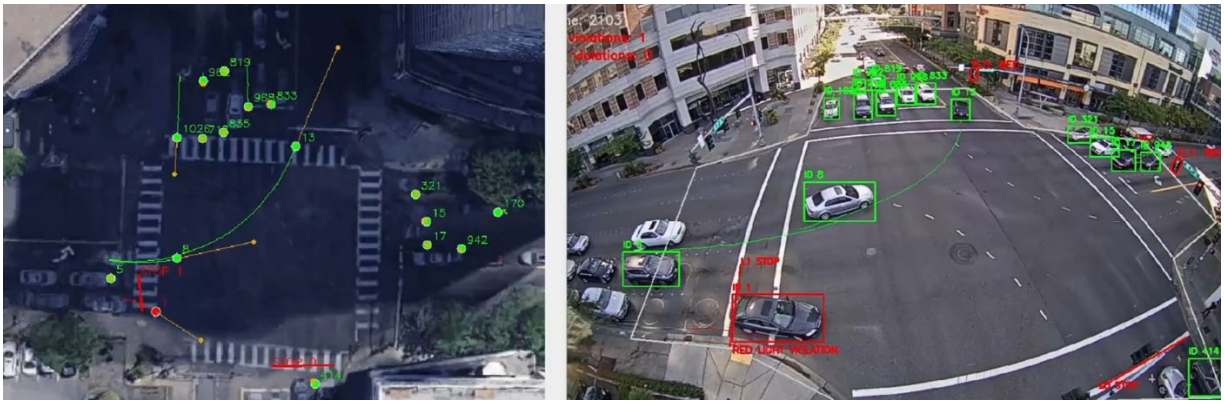


Figure 3. Example of red light violation

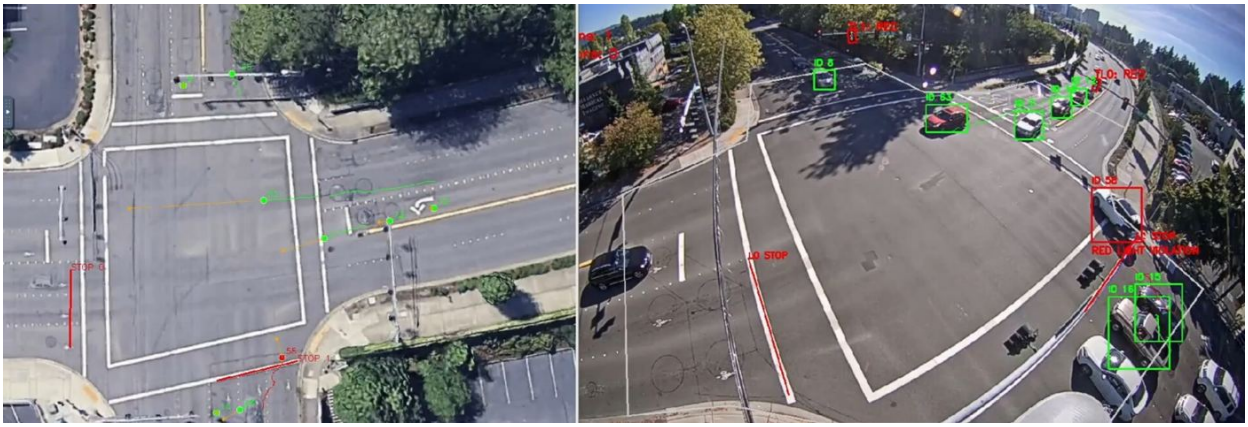


Figure 4. Example of red light violation



Figure 5. (Left) Example of stop sign violation, (Right) Example of stop sign compliance

There is a YouTube video showing the system in action. The code is available on GitHub.

References

[1]City of Coldwater MI, “4 Corners Camera Downtown,” YouTube, Jan. 07, 2026. https://www.youtube.com/watch?v=1H0iTzv2jiQ (accessed Apr. 15, 2026).
[2]Friant Roulette, “Traffic Camera - Car Spotting - Exotic Supercars - Police Scanner - Live from Fresno, California,” YouTube, Feb. 14, 2026. https://www.youtube.com/watch?v=1xl0hX-nF2E (accessed Apr. 15, 2026).
[3]Virtual Railfan, “Trains in the Street at La Grange, Kentucky, USA | LIVE Train Camera (PTZ),” YouTube, May 26, 2024. https://www.youtube.com/watch?v=9SLt3AT0rXk (accessed Apr. 15, 2026).
[4]City-of-Bellevue, “GitHub - City-of-Bellevue/TrafficVideoDataset,” GitHub, 2022. https://github.com/City-of-Bellevue/TrafficVideoDataset (accessed Apr. 15, 2026).
[5]Federal Highway Administration, “About Intersection Safety | FHWA,” Dot.gov, Jul. 26, 2024. https://highways.dot.gov/safety/intersection-safety/about (accessed Apr. 15, 2026).