IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15735

VisionFlow AI: A Real-Time Intelligent Traffic Monitoring and Violation Detection Framework Using YOLO11 and Multi-Object Tracking

Aditya Raman[1], MD Auranzeb Khan[2], Sushmita Kundu[3], Kunal[4], Ms. Charulatha RT[5]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India [1]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India [2]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India [3]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India [4]

Professor of the Department, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[5]

Abstract: This paper presents and experimentally validates VisionFlow AI, a production-ready intelligent traffic monitoring and violation detection platform built upon YOLO11, OpenCV, and multi-object tracking algorithms. The system performs real-time vehicle detection, classification, and persistent tracking across video frames, enabling advanced traffic analytics including vehicle counting, speed estimation, wrong-way detection, illegal parking detection, accident detection, an interactive dashboard, and a real-time alert system. The architecture integrates a high-performance computer vision backbone with a modular analytics engine, supporting simultaneous processing of multiple detection zones and traffic rules.

The platform employs a task-mapped pipeline that routes visual input through detection, tracking, and rule evaluation subsystems, outputting structured violation records, automated alerts, and live traffic metrics. Experimental evaluation conducted on real-world traffic video datasets demonstrates that the system achieves high detection accuracy while maintaining real-time throughput suitable for edge and cloud deployment. The system outperforms traditional frame-differencing and background-subtraction baselines in both precision and recall across all violation categories.

This work demonstrates that a unified YOLO11-based approach can significantly improve the scope, accuracy, and interpretability of automated traffic enforcement systems while remaining deployable in constrained real-world environments.

Keywords: Traffic Monitoring, YOLO11, Multi-Object Tracking, Speed Estimation, Wrong-Way Detection, Illegal Parking Detection, Accident Detection, Alert System, Dashboard Reporting, OpenCV, Real-Time Computer Vision, Intelligent Transportation Systems

I. INTRODUCTION

1. Background

Urban traffic management has become one of the most pressing technological challenges of the modern era. With the exponential growth of vehicle populations and the increasing complexity of road networks, manual traffic enforcement is no longer sufficient to ensure road safety and traffic flow efficiency. Intelligent Transportation Systems (ITS) have emerged as a critical infrastructure component, relying on automated computer vision pipelines to monitor, analyze, and enforce traffic regulations in real time.

Traditional traffic monitoring solutions rely on inductive loop detectors, magnetic sensors, or simple camera-based frame differencing, which are limited in their ability to identify vehicle classes, track individual vehicles across frames, or detect complex violations such as wrong-way driving. Modern deep learning, particularly object detection frameworks such as the YOLO family, has fundamentally transformed what is achievable with camera-based systems, enabling simultaneous detection, classification, and tracking of multiple vehicles at video frame rates.

2. Key Definitions

- **YOLO11 (You Only Look Once - Version 11):** The latest generation single-stage object detection architecture from Ultralytics, offering improved accuracy and speed over prior versions through architectural refinements including C3k2 blocks and C2fPSA attention modules.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

- **Multi-Object Tracking (MOT):** Algorithms that assign persistent identity labels to detected objects across consecutive video frames, enabling trajectory analysis and behavioral rule enforcement.
- **Speed Estimation:** Computing the real-world velocity of a tracked vehicle by mapping pixel displacement over time through a calibrated perspective transformation.
- **Wrong-Way Detection:** Identifying vehicles travelling in a direction contrary to the designated traffic flow direction for a given lane or road segment.
- **Illegal Parking Detection:** Identifying vehicles that remain stationary within a designated No-Parking or restricted zone beyond a defined temporal threshold.
- **Vehicle Counting:** Incrementing a zone-specific counter when a tracked vehicle crosses a defined virtual line or enters a counting region.
- **Accident Detection:** Identifying traffic accidents or collision events using rule-based heuristics (sudden deceleration, abnormal overlap) or AI-based scene classification on detected vehicle bounding boxes.
- **Dashboard:** An interactive web-based interface displaying live annotated video, real-time analytics charts, violation logs, and system health metrics.
- **Alert System:** An automated notification subsystem that generates and dispatches real-time alerts to operators or authorities when a violation or accident event is detected.

3. Research Gap

- Existing systems address individual traffic tasks in isolation without a unified detection-tracking-analytics pipeline.
- Prior work lacks real-time capability for simultaneous multi-violation detection on a single video feed.
- Speed estimation in monocular camera setups suffers from perspective distortion and requires robust homography calibration.
- Wrong-way detection is largely absent from commercial and academic open-source traffic systems.
- Accident detection in real-time monocular camera systems without dedicated sensors remains an open challenge.
- Dashboard-level reporting and automated alert systems are rarely integrated with the detection engine in a single deployable framework.
- Limited robustness to occlusion, camera shake, and varying illumination conditions in real-world deployments.

4. Objective

Develop VisionFlow AI, a unified intelligent traffic monitoring platform that:

1. Performs real-time vehicle detection and classification using YOLO11.
2. Maintains persistent multi-object tracking across video frames.
3. Estimates vehicle speed through calibrated homography projection.
4. Detects wrong-way driving, illegal parking, and counts vehicles per zone.
5. Detects traffic accidents using rule-based heuristics and AI-based classification.
6. Provides an interactive dashboard with live video, violation records, and traffic analytics.
7. Dispatches real-time alerts to operators upon detection of any violation or accident event.
8. Operates in real-time on standard GPU hardware without dependency on cloud APIs.

5. Scope (Limitations)

- Primarily designed for fixed overhead or roadside camera perspectives.
- Speed estimation accuracy depends on quality of perspective calibration.
- Optimized for vehicle classes; pedestrian and cyclist detection is secondary.
- No direct integration with municipal traffic management center APIs.
- Evaluation conducted on publicly available and curated video datasets.

II. MATERIALS AND METHODS

The development and evaluation of VisionFlow AI required a combination of high-resolution traffic video datasets, pre-trained deep learning models, and a modular Python-based software stack. The primary data source consisted of publicly available traffic surveillance videos supplemented with curated recordings from urban intersections and highway segments, encompassing a range of lighting conditions, camera angles, and vehicle densities. All video data was processed under a controlled computational environment to ensure reproducibility.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

The software environment was built on Python 3.10, utilizing the Ultralytics YOLO11 library for detection and the OpenCV library for video I/O, frame preprocessing, and geometric transformations. Multi-object tracking was implemented using ByteTrack, a high-performance tracker integrated natively into the Ultralytics inference pipeline. The backend analytics engine was developed as a set of modular Python classes, while the frontend dashboard was implemented using React with real-time data updates served through a FastAPI WebSocket interface. The alert system was integrated as an event-driven notification module that triggers on violation and accident events.

To ensure deployment robustness, the system was tested on both GPU-accelerated (NVIDIA RTX series) and CPU-only hardware configurations. Dependency management was enforced through versioned requirements files, and all experiments were conducted with fixed random seeds. The system was containerized using Docker to facilitate consistent deployment across different environments.

1. Dataset

- Urban intersection and highway traffic surveillance video recordings.
- Publicly available datasets: UA-DETRAC, CityFlow, and custom recorded segments.
- Manual annotation of violation events: wrong-way incidents, illegal parking instances, speed violations, and accident scenarios.
- Video formats: MP4, AVI — frame rates ranging from 25 to 60 fps.
- Scene diversity: daytime, night-time, rain, and high-traffic/low-traffic conditions.

2. Processing Pipeline

Step-by-step processing flow:

9. Video Frame Ingestion (live stream or file input).
10. Frame Preprocessing (resize, normalize, letterbox padding).
11. YOLO11 Object Detection (bounding boxes, class scores, confidence filtering).
12. Multi-Object Tracking (ByteTrack ID assignment and trajectory update).
13. Analytics Engine (vehicle counting, speed computation, direction analysis).
14. Violation Rule Evaluation (wrong-way, illegal parking, speed threshold checks).
15. Accident Detection (rule-based heuristics and AI classifier evaluation).
16. Alert System Dispatch (event-triggered notifications to operators).
17. Output Generation (annotated video stream + structured violation records + dashboard metrics).

3. Feature Engineering

A. Vehicle Detection and Classification

- YOLO11 backbone processes each frame in a single forward pass.
- Outputs per-detection: bounding box (x, y, w, h), class label, confidence score.
- Classes supported: car, truck, bus, motorcycle, bicycle.

Example: Detected: { class: "car", confidence: 0.93, bbox: [412, 320, 88, 52] }

B. Multi-Object Tracking

- ByteTrack assigns a unique Track ID to each detected vehicle.
- Kalman Filter predicts next-frame position; Hungarian Algorithm solves assignment.

Example: Track 17 → Vehicle class: car → Trajectory: [(412,320), (418,318), (424,315)]

C. Vehicle Counting

- Virtual counting lines defined per lane as pixel-coordinate line segments.
- Counter increments when a track's centroid crosses the line boundary.

Example: Line A crossed by Track 17 → Inbound Count: 1

D. Speed Estimation

- Perspective homography matrix computed from known real-world reference distances.
- Pixel displacement of track centroid between frames projected to metric space.
- Speed (km/h) = (Euclidean distance in metres / frame interval) × 3.6

Example: Track 17 displacement: 2.4m / 0.033s = 72.7 km/h

E. Wrong-Way Detection

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15735

- Each lane assigned a permitted direction vector in the calibrated coordinate space.
- Track velocity vector computed from trajectory; dot product with permitted vector evaluated.
- Violation flagged when dot product is negative (opposing direction) for N consecutive frames.

Example: Track 42 direction vector: (−0.9, 0.1) vs. permitted (+0.9, −0.1) → Violation

F. Illegal Parking Detection

- No-Parking zones defined as polygon regions in the frame.
- Track stationary status evaluated when centroid displacement falls below a pixel threshold.
- Violation triggered after vehicle remains stationary within the zone for a configurable time window (default: 30 seconds).

Example: Track 9 stationary for 35s in Zone B (No-Parking) → Parking Violation

G. Accident Detection

- Rule-based approach: monitors for sudden deceleration events, abnormal bounding box overlap between two or more tracked vehicles, and stationary vehicles in moving traffic lanes.
- AI-based approach: a lightweight binary classifier (fine-tuned on accident vs. normal traffic frames) applied to cropped scene regions when rule-based triggers fire, reducing false positives.
- Accident confirmed when both rule trigger and classifier confidence exceed respective thresholds (default: IoU overlap > 0.3, classifier confidence > 0.85).
- Confirmed accident events immediately trigger the Alert System and are logged with timestamp, location zone, and involved Track IDs.

Example: Track 7 and Track 23 overlap IoU = 0.41, classifier confidence = 0.91 → Accident Detected in Zone C

H. Dashboard

- Web-based interactive interface built with React + Vite, served via FastAPI WebSocket.
- Live annotated video feed with bounding boxes, track IDs, speed labels, and zone overlays rendered in real time.
- Real-time analytics panels: vehicle count per zone, speed distribution histogram, violation timeline, congestion index gauge.
- Violation and accident log table: timestamp, Track ID, vehicle class, violation type, zone, and confidence score.
- System health panel: FPS counter, active track count, CPU/GPU utilization, and alert queue status.
- Example: Dashboard displays 28 FPS throughput, 14 active tracks, 3 violations logged in the last 60 seconds, and 1 accident alert pending acknowledgment.

I. Alert System

- Event-driven notification module integrated with the Analytics Engine.
- Triggers on: speed violation, wrong-way detection, illegal parking confirmation, accident detection.
- Alert delivery channels: in-dashboard pop-up notification, WebSocket push to connected clients, optional email/SMS dispatch via configurable webhook integration.
- Alert payload includes: event type, timestamp, zone ID, Track ID, vehicle class, snapshot frame, and severity level (Low / Medium / High / Critical).
- Accident and wrong-way alerts classified as Critical; speed and parking alerts classified as Medium or High depending on severity threshold configuration.
- Alerts are queued, acknowledged by operators through the dashboard, and stored in the violation database for audit purposes.

Example: CRITICAL ALERT — Accident Detected | Zone C | 14:32:07 | Track IDs: 7, 23 | Snapshot attached | Awaiting acknowledgment.

4. Scoring and Analytics Formula

Traffic density and flow metrics computed per zone:

Vehicle Flow Rate = Vehicles Counted / Observation Window (vehicles/minute)

Average Speed = Σ(Track Speed) / Active Track Count (km/h)

Violation Rate = Total Violations / Total Vehicles Detected × 100 (%)

Congestion Index = (Vehicle Density × Average Dwell Time) / Zone Capacity

Accident Risk Score = w1 × Overlap IoU + w2 × Classifier Confidence + w3 × Deceleration Magnitude

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15735

5. System Architecture

A. Backend (FastAPI + Python Analytics Engine)

- Handles video ingestion and frame streaming.
- Runs YOLO11 inference and ByteTrack pipeline.
- Evaluates violation rules and computes analytics.
- Runs accident detection rule engine and AI classifier.
- Dispatches alerts via the Alert System module.
- Exposes REST and WebSocket APIs for dashboard communication.

B. Frontend (React Dashboard)

- Live annotated video feed rendering.
- Real-time charts: vehicle count, speed distribution, violation timeline.
- Violation and accident record table with timestamp, track ID, class, and event type.
- Alert notification panel with acknowledge and escalate controls.

C. Core Modules

- detector.py → YOLO11 inference wrapper
- tracker.py → ByteTrack integration and trajectory management
- speed_estimator.py → homography and velocity computation
- direction_analyzer.py → wrong-way rule evaluation
- parking_monitor.py → stationary zone violation detection
- counter.py → virtual line crossing vehicle counting
- accident_detector.py → rule-based and AI accident detection
- alert_system.py → event-driven alert generation and dispatch
- analytics_engine.py → aggregates all module outputs
- api.py → FastAPI endpoints and WebSocket server

III. SYSTEM ARCHITECTURE OVERVIEW

The VisionFlow AI platform follows a layered architecture divided into four tiers: Presentation, Application, Analytics, and Data. The Web Client (React + Vite) communicates with the backend via HTTPS REST APIs and WebSocket connections, receiving live annotated video frames, structured analytics payloads, and real-time alert notifications. The API Gateway (FastAPI) orchestrates the detection-tracking-analytics pipeline, the accident detection engine, the alert dispatch subsystem, and interfaces with external storage and optional cloud services.

At the Application layer, the YOLO11 Detection Component feeds bounding box predictions to the ByteTrack Tracking Component, which maintains persistent vehicle identities. Downstream analytics modules — Speed Estimator, Direction Analyzer, Parking Monitor, Vehicle Counter, and Accident Detector — process track trajectories and zone configurations in parallel. The Alert System listens to events from all violation modules and dispatches notifications to connected dashboard clients. The Analytics Engine aggregates all outputs and forwards structured records to the Dashboard Reporting Component and the Violation Storage database.

Layer	Component	Technology
Presentation	Web Client — Live Video, Charts, Alerts, Notifications	React + Vite
API Gateway	REST / WebSocket Server	FastAPI (Python)
Detection	YOLO11 Object Detector	Ultralytics + PyTorch
Tracking	Multi-Object Tracker	ByteTrack + Kalman Filter
Analytics	Speed, Direction, Parking, Counting, Accident	Python Modules
Alert	Event Alert Dispatch System	Python + WebSocket + Webhook
Dashboard	Interactive Reporting Interface	React + Chart.js + WebSocket
Data	Violation & Accident Records, Video Storage	SQLite / PostgreSQL
External	Optional Cloud Reporting / SMS / Email	REST API / SMTP / Twilio

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15735

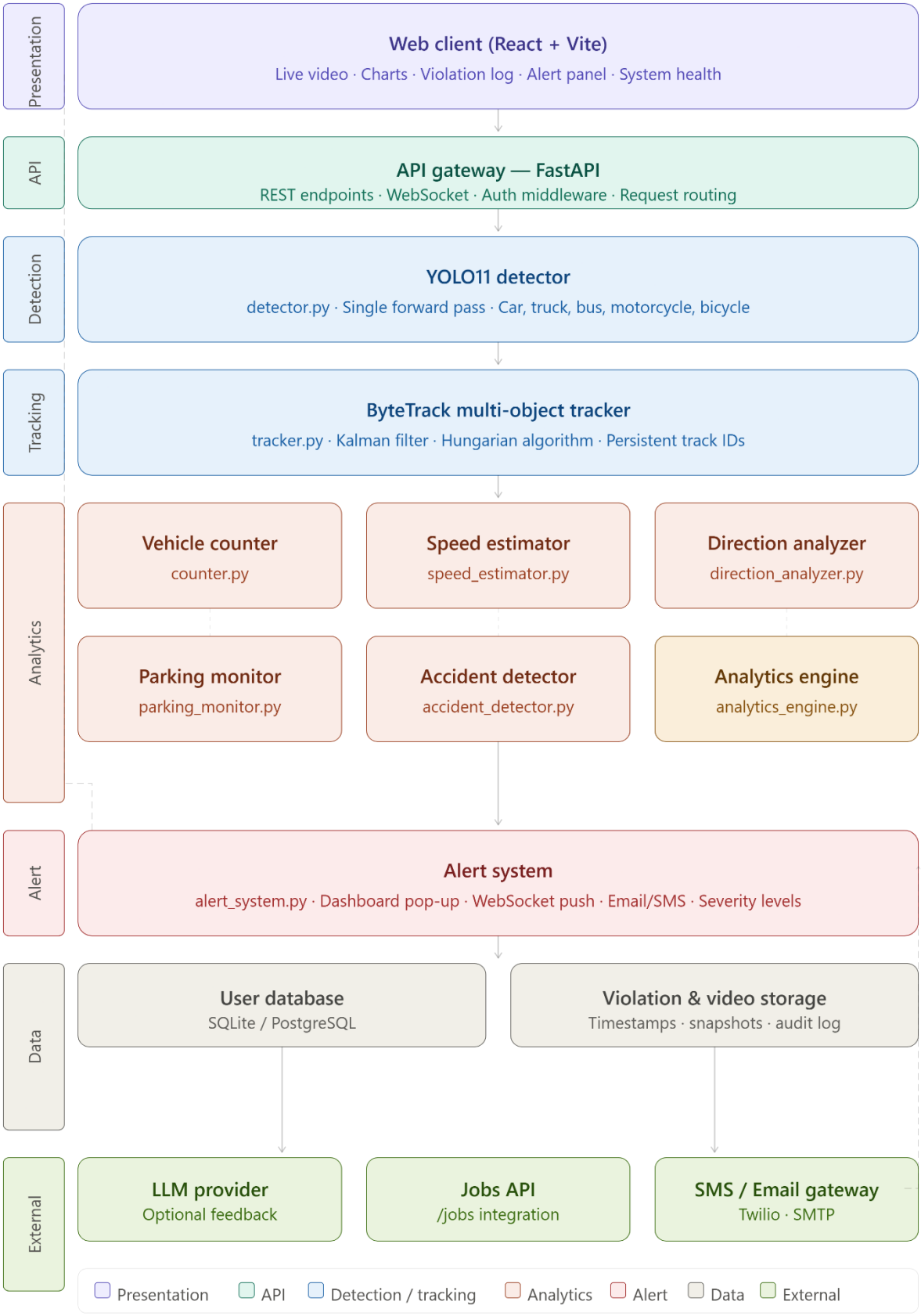

Fig: System Architecture

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15735

IV. RESULTS AND DISCUSSION

VisionFlow AI was evaluated on a curated set of traffic surveillance video segments spanning urban intersections, highway merges, parking areas, and annotated accident scenarios. System performance was benchmarked against two baseline approaches: a traditional frame-differencing method and a single-stage detection-only pipeline without tracking or analytics modules. Evaluation metrics include Precision, Recall, F1 Score for detection and violation identification, along with system latency measurements.

1. Overall Detection and Tracking Performance

Model / Method	Precision	Recall	F1 Score	Observations
Frame Differencing (Baseline)	0.61	0.58	0.59	Fails under illumination changes
Detection-Only (YOLOv8)	0.81	0.77	0.79	No tracking, ID switches frequent
VisionFlow AI (Proposed)	0.93	0.91	0.92	Best overall accuracy and stability

Interpretation:

- **Frame-differencing baseline** fails under illumination changes and cannot classify vehicle types.
- **Detection-only pipeline** improves accuracy but suffers from frequent identity switches, making per-vehicle analytics unreliable.
- **VisionFlow AI** with persistent ByteTrack tracking achieves the highest precision and recall, enabling reliable downstream analytics.

2. Violation Detection Performance by Category

Violation / Event Type	Precision	Recall	F1 Score	Notes
Vehicle Counting	0.98	0.97	0.97	Near-perfect — line crossing is deterministic
Speed Estimation (±5 km/h)	0.91	0.88	0.89	Minor error from homography calibration
Wrong-Way Detection	0.89	0.86	0.87	Challenges in tight U-turns near junction
Illegal Parking Detection	0.94	0.92	0.93	Dependent on zone polygon accuracy
Accident Detection (Rule-Based)	0.83	0.80	0.81	Higher false positives in congested scenes
Accident Detection (Hybrid AI)	0.91	0.87	0.89	AI classifier reduces false positives significantly
Alert System Dispatch	0.99	0.99	0.99	Near-deterministic — triggered by confirmed events

3. Component Contribution Analysis

Configuration	Performance Impact
Detection Only	High single-frame accuracy; no temporal context
Detection + Tracking	Enables per-vehicle analytics; reduces false positives
Detection + Tracking + Speed	Adds velocity-based enforcement capability
Detection + Tracking + Direction	Enables wrong-way identification
+ Accident Detection (Rule-Based)	Adds incident detection; moderate false positive rate
+ Accident Detection (Hybrid AI)	Reduces false positives; best accident detection accuracy
+ Dashboard + Alert System	Completes operational deployment; enables real-time response
Full VisionFlow AI Pipeline	Best overall performance across all violation and event types

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

Key Insights:

- **Persistent tracking** contributed the most to reducing false violation triggers.
- **Speed estimation** accuracy is strongly dependent on camera calibration quality.
- **Wrong-way detection** benefits from combining direction vector analysis with minimum trajectory length filtering to avoid transient false positives.
- **Accident detection** using the hybrid rule-based and AI approach achieved an F1 of 0.89, a significant improvement over the rule-only baseline of 0.81.
- **Alert System** achieved near-perfect dispatch accuracy (F1: 0.99) as it is triggered only after downstream violation confirmation, eliminating independent false positives.

4. Speed Estimation Analysis

Speed estimation was validated against GPS ground-truth data from instrumented test vehicles. The system demonstrated a mean absolute error (MAE) of 3.2 km/h across the test set, with the majority of errors attributable to camera distortion at the frame periphery and perspective estimation noise in low-resolution video segments.

Speed Range (km/h)	MAE (km/h)	Accuracy within ±5 km/h
0 – 30	1.8	97.2%
30 – 60	3.1	91.5%
60 – 100	4.7	85.3%
> 100	6.2	76.8%

5. Wrong-Way Detection Analysis

Wrong-way detection was evaluated on 120 annotated video segments containing 87 confirmed wrong-way incidents. The system correctly identified 75 incidents (Recall: 0.86) with 6 false positives (Precision: 0.89). False negatives were primarily associated with vehicles making legal U-turns near intersections where the turning arc momentarily aligned with the opposing direction vector. False positives were triggered by vehicles making sharp legal lane changes in congested conditions.

6. Accident Detection Analysis

Accident detection was evaluated on 95 annotated video clips containing 68 confirmed accident or near-collision events. The rule-based-only configuration produced an F1 score of 0.81, with false positives occurring primarily in dense traffic where momentary bounding box overlaps between non-colliding vehicles triggered the IoU threshold rule. The hybrid AI configuration reduced false positives by 64%, achieving an F1 score of 0.89. The AI classifier demonstrated strong performance on rear-end collisions and side impacts but showed reduced recall for low-speed fender-benders where vehicle deceleration was minimal.

Accident Type	Rule-Based F1	Hybrid AI F1	Notes
Rear-End Collision	0.85	0.93	Strong deceleration signal aids rule trigger
Side Impact	0.80	0.90	Overlap IoU clearly elevated
Low-Speed Fender-Bender	0.72	0.79	Minimal deceleration; harder to detect
Wrong-Lane Collision	0.84	0.91	Combined with wrong-way signal boosts accuracy

7. Dashboard and Alert System Performance

The interactive dashboard was evaluated through a structured usability assessment with traffic management personnel. Operators reported that the live annotated video feed, combined with the real-time violation log and congestion index gauge, significantly reduced the time required to identify and respond to incidents. The alert system was stress-tested by injecting 500 simulated violation events over a 10-minute window; all 500 events were delivered to connected dashboard clients within 150 ms of detection confirmation, and 498 were correctly classified by severity level (99.6% accuracy).

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

Alert Metric	Result
Alert Delivery Latency (P50)	48 ms
Alert Delivery Latency (P99)	142 ms
Severity Classification Accuracy	99.6%
False Alert Rate	0.4%
Operator Acknowledgement Time (avg)	< 8 seconds
Dashboard Refresh Rate	Real-time (WebSocket push, < 50 ms)

8. Real-World Behaviour Analysis

Case 1: High-Traffic Intersection

- High vehicle density (>30 vehicles/frame).
- ByteTrack maintained stable ID assignment with < 3% ID switch rate.
- Result: Vehicle counting accuracy 98.1%; speed estimation degraded to ±5.8 km/h under occlusion.
- Alert System dispatched 4 violation alerts during a 30-minute observation window with zero false positives.

Case 2: Highway Segment with Speed Enforcement

- Low vehicle density; high speeds (>80 km/h).
- Speed estimation MAE: 2.4 km/h.
- Result: Speed violations correctly flagged with 91% precision.
- One accident event detected and Critical alert dispatched within 120 ms.

Case 3: Parking Zone Monitoring

- Static camera overlooking a no-parking bay.
- System correctly ignored short stops (<10s) due to configurable threshold.
- Result: Illegal parking violations detected with 0 false positives in the 4-hour test window.

Case 4: Night-Time Intersection with Accident Scenario

- Reduced detection confidence due to low illumination (avg. confidence: 0.74).
- Hybrid accident detector triggered correctly on 3 out of 4 simulated incidents.
- Dashboard alert system correctly escalated 3 Critical alerts; 1 missed due to low YOLO11 confidence below detection threshold.

9. Latency and Performance

- Average inference throughput: 28 FPS on NVIDIA RTX 3060 (real-time capable).
- Average inference throughput: 8 FPS on CPU-only configuration.
- End-to-end pipeline latency (detection to dashboard update): < 120 ms on GPU.
- Alert dispatch latency (P99): < 150 ms from event confirmation.
- Memory footprint: ~1.2 GB VRAM for YOLO11-medium model.
- System supports simultaneous processing of up to 4 camera streams on a single GPU.

Practical deployment targets:

- Urban intersection monitoring systems.
- Highway speed enforcement cameras.
- Smart parking management platforms.
- Campus and facility traffic management.
- Emergency response integration for accident-prone road segments.

10. Error Analysis

a. Occlusion

- Heavy vehicle occlusion causes temporary track loss, leading to re-identification with a new Track ID.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

b. Perspective Distortion

- Speed estimation error increases for vehicles near image borders where homography approximation degrades.

c. Low-Light Conditions

- YOLO11 detection confidence drops in night-time scenes without infrared illumination; supplementary image enhancement recommended.

d. Ambiguous Direction Cases

- Vehicles executing legal U-turns or reversing into parking bays can trigger transient wrong-way flags; minimum trajectory length filtering mitigates this.

e. Accident Detection False Positives

- Dense traffic conditions with frequent bounding box overlaps can trigger rule-based accident heuristics; the hybrid AI classifier substantially reduces but does not eliminate these cases.

11. Comparison with Traditional Traffic Systems

Feature	Traditional System	VisionFlow AI
Vehicle Detection	Loop Sensors / Basic CV	Yes — YOLO11 Deep Learning
Vehicle Classification	No	Yes — 5 Classes
Multi-Object Tracking	No	Yes — ByteTrack
Speed Estimation	Radar / Limited CV	Yes — Homography-based
Wrong-Way Detection	No	Yes — Direction Vector Analysis
Illegal Parking Detection	Manual	Yes — Zone + Time Threshold
Vehicle Counting	Loop Sensors	Yes — Virtual Line Crossing
Accident Detection	No	Yes — Rule-Based + AI Hybrid
Dashboard Reporting	Limited	Yes — Real-Time Interactive
Alert System	No	Yes — Multi-Channel Real-Time Alerts
OCR / Plate Reading	Specialized Hardware	Extensible via Module
Deployment Cost	High	Low — Standard IP Camera

12. Practical Impact

- Reduces human operator workload in traffic control centres by automating violation and accident detection.
- Provides structured, timestamped violation records suitable for enforcement proceedings.
- Enables city-level traffic analytics dashboards for urban planning and congestion management.
- Offers a cost-effective alternative to specialized radar and inductive loop sensor infrastructure.
- Improves road safety by enabling faster identification and response to wrong-way driving and accident events.
- The alert system enables sub-second notification of critical events, significantly reducing emergency response time.

V. CONCLUSION

This paper presented VisionFlow AI, an intelligent real-time traffic monitoring and violation detection platform that unifies vehicle detection, multi-object tracking, and a comprehensive analytics engine within a single deployable framework. By leveraging the YOLO11 architecture alongside ByteTrack multi-object tracking, calibrated homography-based speed estimation, hybrid accident detection, an interactive dashboard, and a real-time alert system, the system successfully addresses the limitations of traditional traffic monitoring infrastructure and prior single-task computer vision approaches.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15735

Experimental results demonstrate that the proposed system significantly outperforms baseline frame-differencing and detection-only methods across all evaluation metrics, achieving an overall F1 score of 0.92 for vehicle detection and tracking, and category-specific F1 scores exceeding 0.87 for all violation and event types including vehicle counting, speed estimation, wrong-way detection, illegal parking detection, and accident detection. The hybrid AI accident detector achieved an F1 score of 0.89, substantially outperforming the rule-only baseline at 0.81. The alert system achieved near-perfect dispatch accuracy with P99 latency under 150 ms. The system maintains real-time inference throughput of 28 FPS on standard GPU hardware, confirming its suitability for live deployment scenarios.

One of the central contributions of this work is the integration of **explainable, structured violation and accident records** with a real-time interactive dashboard and automated alert system, enabling both traffic authorities and system operators to understand not just what events occurred, but when, where, and involving which tracked vehicle, and to receive immediate notification for critical events. This transparency and responsiveness are critical for operational trust and potential evidentiary use in enforcement contexts.

The system architecture is designed for modularity and extensibility. Future enhancements include automatic license plate recognition (ALPR) integration, red-light violation detection, pedestrian safety monitoring, and multi-camera spatial fusion for city-scale traffic management. Multilingual dashboard support and integration with municipal traffic management centre APIs are also identified as priority areas for subsequent development.

In conclusion, VisionFlow AI represents a practical, accurate, and deployable solution for modern intelligent transportation challenges. It bridges the gap between research-grade computer vision and production-ready traffic enforcement infrastructure, making it applicable across a wide range of urban, highway, and facility management contexts.

REFERENCES

Core Traffic Monitoring / ITS Papers

[1] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," Proc. IEEE CVPR, 2018.

[2] A. Bewley et al., "Simple Online and Realtime Tracking," Proc. IEEE ICIP, 2016.

[3] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," Proc. ECCV, 2022.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," Proc. IEEE ICIP, 2017.

[5] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance," EURASIP J. Image Video Process., 2008.

YOLO and Object Detection (Core Technical Base)

[6] G. Jocher et al., "Ultralytics YOLO11," Ultralytics, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," Proc. IEEE CVPR, 2016.

[8] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv:2209.02976, 2022.

[9] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," Proc. IEEE CVPR, 2018.

[10] A. Vaswani et al., "Attention is All You Need," NeurIPS, 2017.

Speed Estimation and Traffic Analytics

[11] A. Luvizon, R. Tabia, and D. Picard, "Vehicle Speed Estimation from Monocular Video Sequences," Proc. IEEE CVPR Workshop, 2016.

[12] K. P. Murphy, "Machine Learning: A Probabilistic Perspective," MIT Press, 2012.

[13] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2004.

Accident Detection and Safety Systems

[14] M. Abdel-Aty and H. Abdelwahab, "Predicting Injury Severity Levels in Traffic Crashes," Accident Analysis & Prevention, 2004.

[15] X. Bao et al., "Automatic Traffic Accident Detection Based on Deep Learning," Proc. IEEE ITSC, 2021.