# Impact of INT8 Quantization on YOLOv11n for Ego-Centric Vehicle Collision Risk Estimation Using Dashcam Video


Ashwin Suresh Nair
*Department of Computing Technologies*
*SRM Institute of Science and Technology*
*Kattankulathur – 603203, Chengalapattu District, Tamil Nadu, India*
an6055@srmist.edu.in

Milan P Joseph
*Department of Computing Technologies*
*SRM Institute of Science and Technology*
*Kattankulathur – 603203, Chengalapattu District, Tamil Nadu, India*
mj0754@srmist.edu.in

Arulalan V
*Department of Computing Technologies*
*SRM Institute of Science and Technology*
*Kattankulathur – 603203, Chengalapattu District, Tamil Nadu, India*
arulalav@srmist.edu.in



Abstract—Road traffic accidents remain a major global safety challenge, often caused by delayed hazard perception and insufficient situational awareness in complex driving environments. Modern Advanced Driver Assistance Systems (ADAS) rely heavily on visual perception models for object detection and risk assessment, yet deploying deep learning models on edge devices requires a careful balance between detection accuracy and computational efficiency. This paper presents an ego-centric collision risk estimation framework that integrates object detection, multi-object tracking, trajectory forecasting, and spatiotemporal conflict analysis using monocular dashcam video. The perception module is implemented using the YOLOv11n object detection model, which is evaluated in two configurations: a full-precision FP32 model and an INT8 quantized version to help analyze deployment efficiency. The system estimates the future motion of surrounding agents and the ego vehicle, thereby identifying potential trajectory conflicts, and will compute a continuous collision risk score along with predictive lead-time metrics. Experimental results show that INT8 quantization improves inference speed by approximately 9.5% while introducing a slight reduction in detection confidence and average predictive lead time; however, despite reduced detection completeness, the overall risk signal remains stable across both of the configurations. These findings highlight the trade-off between computational efficiency and predictive safety performance when deploying quantized deep learning models in real-time driving systems.



Index Terms—collision risk estimation, INT8 quantization, YOLOv11n, ego-centric perception, dashcam video, ADAS, trajectory forecasting, multi-object tracking


## I. Introduction

Road traffic accidents account for millions of injuries and fatalities each year. Urban driving environments are particularly challenging due to the density found in traffic, unpredictability in the behavior of pedestrians, and frequent occlusions. Human drivers rely on continuous anticipation of surrounding agents' movements to avoid very dangerous situations. However, delayed perception or misjudgment can result in very serious accidents.

To improve safety, modern Advanced Driver Assistance Systems (ADAS) attempt to detect hazards and warn drivers even before any collisions occur. Many existing systems rely mostly on rule-based warnings which are derived from radar or proximity thresholds. While effective in controlled scenarios, these methods are found out to be often failing to capture complex interactions among multiple dynamic agents.

Recent advances made in deep learning have significantly improved visual perception capabilities in that of driving systems. Convolutional neural networks enable accurate object detection in real time, thereby allowing vehicles to identify surrounding agents such as the likes of pedestrians, bicycles, and other vehicles. Among these models, the YOLO (You Only Look Once) family of detectors has become widely adopted due to its balance between good detection accuracy and inference speed.

However, object detection alone is insufficient for predicting collisions. Risk emerges not from the current spatial distance between agents but from their future motion interactions.

Therefore, collision risk estimation requires integrating detection, tracking, trajectory prediction, and conflict analysis into a unified framework.

Another challenge in deploying deep learning models in real-world systems is that of the computational efficiency. Full-precision neural networks require significant amount of memory and processing power, which can limit real-time deployment on edge devices. Model compression techniques such as quantization can reduce computational cost by converting model weights from 32-bit floating point precision to lower precision formats such as 8-bit integers.

While quantization is commonly evaluated using the likes of detection accuracy metrics, its impact on downstream safety-critical tasks such as that of collision risk prediction remains underexplored.

This work proposes an ego-centric collision risk estimation framework using dashcam video and evaluates the effect of INT8 quantization on detection performance, inference speed, and predictive lead time.

The key contributions of this paper are:

1) A real-time ego-centric collision risk estimation pipeline using monocular dashcam video.
2) Integration of object detection along with tracking then trajectory forecasting, and conflict analysis.
3) Quantization of the YOLOv11n model to INT8 precision for efficient inference.
4) Experimental comparison of FP32 and INT8 models across system-level safety metrics.

## II. Related Works

### A. Object Detection and Tracking in Driving Scenarios

Recent advances in deep learning have significantly improved the performance of object detection systems used in intelligent transportation applications. Early real-time detection frameworks such as YOLO demonstrated that high-speed detection could be achieved using a unified convolutional architecture that predicts bounding boxes and class probabilities in a single network pass [1]. Sub-sequent developments in the YOLO family have further improved things like detection accuracy, efficiency, and robustness.

The YOLOv11 architecture represents a kind of recent evolution of this design philosophy, introducing both the architectural improvements and optimized inference pipelines that enable efficient deployment in real-time perception systems [2]. These properties make YOLO-based detectors particularly suitable for traffic monitoring and driver assistance systems where low latency is considered very critical.

While object detection provides frame-level recognition of traffic participants, many driving applications would be still requiring continuous tracking of objects across frames. Multi-object tracking (MOT) algorithms address this particular challenge by associating detections over time to maintain persistent identities. Modern tracking approaches such as ByteTrack associate detection boxes across frames using motion and confidence cues to help achieve a reliable identity consistency in those crowded scenes [3]. Earlier approaches such as DeepSORT combined motion models with appearance embeddings to help to improve tracking robustness under occlusions and viewpoint changes [4].

Vision-based vehicle detection and behavior analysis have been extensively studied in intelligent transportation research. Surveys of these methods very much highlight the importance of combining detection and tracking to understand traffic dynamics and predict vehicle interactions [5].

### B. Trajectory Prediction and Motion Forecasting

Predicting the future motion of traffic participants is a very critical component of autonomous driving and collision avoidance systems. Early trajectory prediction methods have often relied on simple kinematic models such as constant velocity or constant acceleration assumptions. More recent research only has explored learning-based approaches that leverage neural networks to capture those complex motion patterns. Recurrent neural network models such as Social LSTM incorporate lot of interactions between agents to improve pedestrian trajectory prediction in crowded environments [6], [7].

However, many deep learning trajectory prediction methods have shown to require large annotated datasets and significant computational resources, thereby making them less suitable for real-time deployment in lightweight systems. For practical driving applications, simplified kinematic forecasting models remain attractive because of the fact that they can generate plausible future trajectories with minimal computational overhead while maintaining real-time performance.

### C. Collision Risk Estimation

Collision risk estimation has traditionally been studied most of the time within the context of driver assistance systems and traffic safety analysis. Classical safety metrics such as Time-to-Collision (TTC), Post-Encroachment Time (PET), and minimum distance thresholds are widely used to assess potential hazards between that of the vehicles and pedestrians. Several studies have explored real-time collision risk assessment methods for advanced driver assistance systems (ADAS), focusing mainly on predicting hazardous interactions even much before they occur [8]. Surveys on motion prediction and risk estimation highlight the importance of combining trajectory forecasting with the likes of probabilistic safety evaluation to improve early hazard detection [9].

Computer vision libraries such as OpenCV have played an important role in the enabling of practical implementations of perception-based risk estimation systems [10]. Optical flow algorithms, such as including the Farneback method, provide efficient motion estimation capabilities that can be used to approximate ego vehicle movement directly from video input only [11].

Models of pedestrian dynamics, such as the Social Force Model, have also hugely influenced the design of trajectory prediction systems by modeling interactions between agents in very crowded environments [12]. Similarly, traffic flow models provide insights into vehicle behavior and interaction patterns that are very useful for risk assessment in transportation systems [13].

Recent surveys on pedestrian interaction with the likes of autonomous vehicles emphasize the very importance of predictive perception systems capable of understanding dynamic interactions between vehicles and vulnerable road users [14]. However, evaluating and validating such systems still remains a major challenge due to the very complexity of real-world driving environments [15].

### D. Supporting Tools and Motion Modeling Foundations

Real-time perception and tracking systems are commonly implemented while using optimized computer vision libraries such as the likes of OpenCV, which provide efficient tools for not only detection but also tracking, and visualization pipelines [10]. Ego-motion estimation in monocular video streams is frequently performed whilst using optical flow techniques, with the Farnebäck method being a very widely adopted approach hugely due to its computational efficiency [11]. Foundational models of pedestrian and traffic dynamics, including the likes of social force-based formulations and traffic flow theory, very much often provide analytical insights into collective motion behavior and interaction dynamics [12], [13].

### E. Research Gap

Despite advances in object detection, tracking, trajectory prediction, and collision risk estimation, few systems integrate all components into a cohesive ego-centric frame-work operating on monocular dashcam video. Surveys on autonomous vehicle–pedestrian interaction highlight the lack of interpretable, real-time risk estimation systems suitable for complex urban environments [14]. Additionally, those challenges that are in validating and deploying such systems in safety-critical applications remain an open problem [15].

## III. Description of Methodology

The proposed collision risk estimation framework mostly operates on monocular dashcam video captured from that of a moving vehicle and follows an ego-centric processing paradigm. The system integrates object detection along with multi-object tracking then trajectory forecasting, and spatiotemporal conflict analysis to estimate potential interactions that can happen between the ego vehicle and surrounding traffic participants.

The overall pipeline processes each incoming frame to detect dynamic agents, estimate their motion trajectories, predict possible future paths, and evaluate the likelihood of collision with the ego vehicle's projected trajectory. The system produces both frame-level and object-level risk estimates, along with predictive lead-time metrics indicating how early hazardous interactions are detected. To evaluate the effect of model compression on the perception module, the object detection stage is implemented
using two versions of the same model:

- YOLOv11n FP32 (baseline)
- YOLOv11n INT8 quantized model

Both detector configurations operate within the same risk estimation pipeline so that any observed differences in performance can be attributed solely to quantization effects.

An overview of the ego-centric collision risk estimation pipeline is illustrated in Fig. 1.

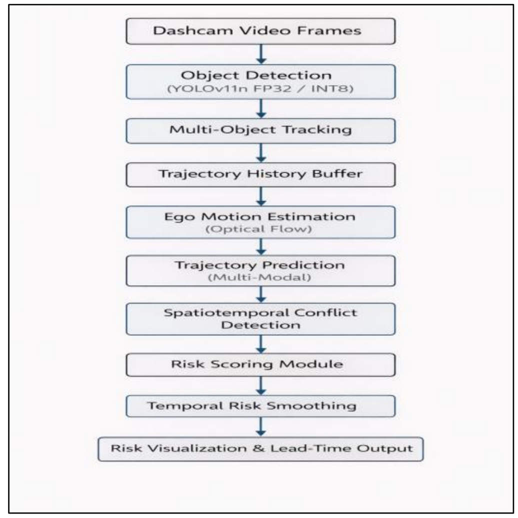


Fig. 1. Overview of the ego-centric collision risk estimation pipeline.

### A. System Overview

For each incoming video frame, the system performs the following processing stages:

1) Detect dynamic agents such as vehicles and pedestrains using a real-time object detection model.
2) Track detected agents across frames to construct motion trajectories.
3) Estimate the ego vehicle's motion using optical flow between consecutive frames.
4) Predict multiple plausible future trajectories for each tracked agent.
5) Project the ego vehicle's future trajectory over a fixed prediction horizon.
6) Detect potential conflicts between predicted agent and ego trajectories.
7) Compute a continuous collision risk score based only on temporal urgency and spatial proximity.
8) Smooth the likes of risk signal over time to ensure stability and interpretability.
9) Record that of frame-level risk values and per-object lead-time metrics.

Each component is designed to remain computationally lightweight, enabling real-time operation using standard vision hardware.

Algorithm 1: Ego-Centric Collision Risk Estimation Pipeline

Input: Dashcam video frames $I_t$

Output: Frame-wise risk score $R_t$, conflict points, per-object lead time

Initialize: Trajectory histories $\mathrm{H}_i \leftarrow \emptyset$ for all agents; Previous risk $R_{t-1} \leftarrow 0$

For each frame $I_t$:

1) Detect dynamic agents using YOLOv11n object detector (FP32 or INT8).
2) Assign persistent object IDs via multi-object track-ing.
3) Update trajectory history $\mathrm{H}_i$ for each tracked agent.
4) Estimate ego vehicle velocity using optical flow.

5) Project ego future path $E(t)$.
6) Generate multi-modal future trajectories $A^k(t)$ for each agent.
7) Detect spatiotemporal conflicts between $E(t)$ and $A^k(t)$.
8) Compute urgency-weighted risk score $R_t$.
9) Smooth risk temporally using exponential averaging.
10) Log risk values and update per-object distance histories.

End For

Compute: Per-object lead time using peak-risk and minimum-distance frames.

### B. Object Detection and Multi-Object Tracking

Object detection is performed using the YOLOv11n model, selected for its balance between detection accuracy and real-time inference speed. The detector identifies bounding boxes and class labels for dynamic agents including pedestrians, bicycles, cars, motorcycles, buses, and trucks.

To maintain consistent object identities across frames, the detection results are very often integrated with that of a multi-object tracking module.

Each detected object is assigned with a unique identifier, thereby allowing its spatial position to be tracked over time. The center point of each bounding box is used as the agent's very image-space position. For each tracked agent $i$, a trajectory history is very much maintained as a sequence of recent positions:

$$H_i = \{(x_{i(t-k)}, y_{i(t-k)}), \ldots, (x_{i(t)}, {}_{i(t)})\}$$

where $k$ is the minimum history length required for trajectory prediction. Tracks with insufficient history are excluded from future prediction to reduce noise and identity-switch artifacts.

### C. Ego Vehicle Motion Estimation

The ego vehicle's motion is estimated directly from the video stream without relying on external sensors. Optical flow is computed between consecutive grayscale frames using the Farneback method. The average vertical com-ponent of the optical flow field is used as an approximation of the ego vehicle's forward velocity in image space.

Let $v_e$ denote the estimated ego velocity. The ego vehicle's current position is assumed to be located at the bottom center of the image frame, defined as:

$$(x_e, y_e) = \left(\frac{W}{2}, H\right)$$

where $W$ and $H$ are the frame width and height, respectively. This ego-centric reference frame very much simplifies downstream trajectory comparison and ensures that all risk computations are relative to that of the ego vehicle.

### D. Ego Trajectory Projection

The ego vehicle's future trajectory is projected over a fixed horizon $T$ assuming constant velocity and straight-line motion. The ego future path is defined as:

$$E(t) = (x_e,\ y_e + t \cdot v_e), \quad t = 1, 2, \ldots, T$$

This approximation is sufficient for short-term risk estimation and avoids introducing complex vehicle dynamics models that require additional sensors or calibration.

### E. Agent Trajectory Forecasting

For each tracked agent with sufficient history, future motion is estimated using a velocity-based kinematic model. The agent's instantaneous velocity is computed from its recent trajectory history using finite differences:

$$v_x = \frac{1}{k-1}\sum_{j=1}^{k-1}\big(x_i(t-j+1) - x_i(t-j)\big)$$

$$v_y = \frac{1}{k-1}\sum_{j=1}^{k-1}\big(y_i(t-j+1) - y_i(t-j)\big)$$

The resulting velocity vector defines the agent's current motion direction and speed. To account for the case of uncertainty in future behavior, multiple plausible future trajectories are generated by perturbing the motion direction with a predefined set of angular offsets. Each offset represents a potential deviation in that of the agent's path.

For each agent $i$ and mode $k$, a future trajectory is generated as:

$$A^k(t) = A_i(0) + t \cdot \|\mathbf{v}_i\| \cdot (\cos\theta_k,\ \sin\theta_k)$$

where $\theta_k$ denotes the perturbed direction angle. This multi-modal trajectory generation enables the system to reason over a set of possible futures rather than committing to a single predicted path.

### F. Spatiotemporal Conflict Detection

Collision risk is evaluated by comparing the predicted agent trajectories with the ego vehicle's projected path. For each agent $i$, each trajectory mode $k$, and each future timestep $t$, the Euclidean distance between the agent and ego positions is computed:

$$d_{i,k}(t) = \|A^k(t) - E(t)|$$

A potential conflict is detected if this distance falls below a predefined threshold $d_c$ after a minimum prediction time

$$d_{i,k}(t) < d_c \quad \text{and} \quad t > t_{\min}$$

This formulation allows the system to identify potential path intersections that may occur several frames into the future.

### G. Risk Scoring Formulation

For each detected conflict, a risk urgency score is computed based on both temporal and spatial factors. Temporal urgency increases as the predicted conflict time approaches, while spatial severity increases as the predicted distance decreases. The urgency score for a conflict is defined as:

$$\boxed{w_{i,k} = \left(\frac{T-t}{T}\right) \cdot \exp\left(-\frac{d_{i,k}(t)}{\lambda}\right)}$$

where $\lambda$ is a distance normalization constant. The overall risk score for the current frame is obtained by averaging urgency scores across all detected conflicts:

$$\boxed{R_t = \frac{1}{N} \sum w_{i,k}}$$

If no conflicts are detected, the risk score is set to zero.

### H. Temporal Risk Smoothing

To reduce noise and ensure stable risk visualization, the raw risk score is smoothed using an exponential moving average:

$$\boxed{\hat{R}_t = \alpha \, \hat{R}_{t-1} + (1-\alpha) \, R_t}$$

where $\alpha \in [0, 1]$ controls the degree of smoothing. This temporal filtering preserves early risk trends while suppressing abrupt fluctuations caused by detection or tracking noise.

### I. Lead-Time Estimation

To evaluate the predictive capability of the system, lead-time analysis is performed on a per-object basis. For each agent involved in a conflict, two key frames are identified:

- The frame at which the risk score associated with the agent reaches its maximum.
- The frame at which the agent reaches minimum distance from the ego vehicle.

The predictive lead time is defined as the difference between these two frames, converted to seconds using the video frame rate. Positive lead times indicate that the system successfully anticipates hazardous interactions before closest approach.

### J. Visualization and Output

The system overlays the following visual cues on the output video:

- Bounding boxes and persistent IDs for tracked objects.
- Historical and predicted future trajectories.
- Ego vehicle predicted path.
- Highlighted conflict points corresponding to maxi-mum urgency.
- A continuously updated ego-centric risk score.

Additionally the likes of frame-wise risk values, minimum distances, and per-object lead times are recorded for the process of quantitative analysis.

## IV. Experimental Method and Setup

### A. Dataset and Evaluation Setup

The proposed collision risk estimation framework is evaluated using the likes of a real-world dashcam video captured from a moving vehicle in urban traffic environments so to say. The recorded scenes contain a diverse set of dynamic agents often including pedestrians, bicycles, cars, motorcycles, buses, and trucks interacting under varying traffic density and motion patterns. Unlike that of curated autonomous driving datasets, the footage contains natural camera motion, illumination variations, and of course unstructured agent behavior, providing a realistic evaluation environment.

All experiments are conducted using monocular RGB video only, thereby without relying on additional sensors such as LiDAR, radar, GPS, or vehicle telemetry. The videos are processed at their native frame rate, and the entire perception and risk estimation pipeline operates in real time only.

To evaluate the impact of model compression on the perception stage, two versions of the same object detection model are very often used:

- YOLOv11n FP32 (baseline model)
- YOLOv11n INT8 (quantized model)

Both models operate within the same ego-centric risk estimation pipeline, including identical tracking, trajectory prediction, conflict detection, and risk computation modules. This controlled setup ensures that any observed differences in performance arise solely from the effects of model quantization only.

Key parameter values used in the experiments are summarized in Table I.

TABLE I
Experimental Parameter Configuration

| Parameter | Value | Description |
|---|---|---|
| FUTURE_STEPS | 30 | Prediction horizon |
| MIN_HISTORY | 8 | Minimum trajectory length |
| CONFLICT_DIST | 50 px | Conflict distance threshold |
| MIN_CONFLICT_TIME | 3 frames | Minimum future offset $\alpha$ |
| | 0.8 | Risk smoothing coefficient |

### B. Qualitative Results

Fig. 2 illustrates representative frames generated by the proposed framework during when the runtime is going on. Detected agents are visualized using bounding boxes along with persistent tracking identifiers that help allow the system to maintain consistent object identities across frames.

For each tracked agent, the system will surely displays both historical trajectories and predicted future motion paths. Historical motion is rendered as polyline traces showing past movement, while future trajectories are projected forward to represent possible agent motion over the prediction horizon. The ego vehicle's predicted path is displayed using a distinct overlay originating from the bottom-center of the frame.

Potential conflict points are highlighted using circular markers placed at the predicted location where the agent trajectory intersects with the likes of the ego trajectory with maximum urgency.

These visual indicators often do make their appearance in several frames before the agent reaches its closest distance to that of the ego vehicle, demonstrating the very predictive nature of the risk estimation framework.

In pedestrian crossing scenarios, the system successfully identifies elevated risk when a pedestrian's predicted trajectory intersects with the likes of the ego vehicle's projected path. Conversely, agents moving parallel to or diverging from the ego trajectory do not at all trigger high risk values even when spatial proximity is small.

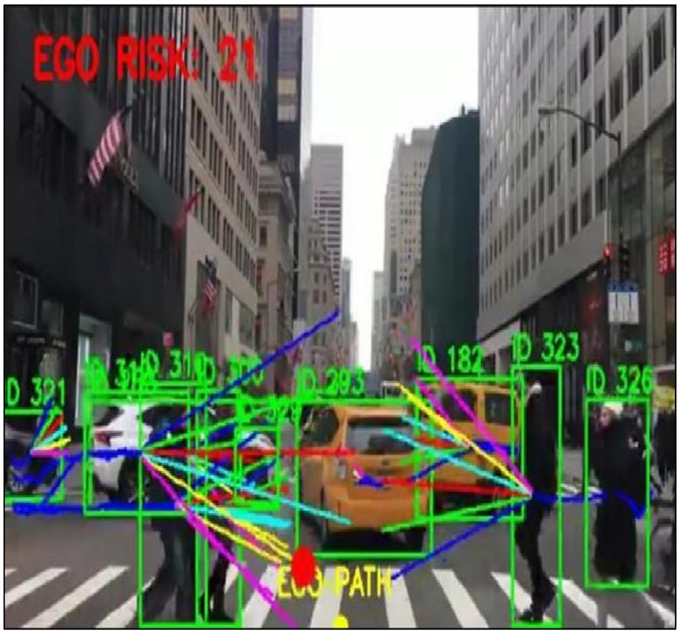


Fig. 2. Representative output frames showing detected agents, tracked trajectories, ego path, conflict points, and risk score overlay.

## C. Risk Signal Behavior

The ego-centric collision risk score evolves continuously as the driving scene also changes over time. Fig. 3 shows the temporal evolution of the risk signal plotted against frame index for both the FP32 and INT8 detector configurations. In non-threatening scenarios, the risk score remains close to zero. As predicted trajectories of surrounding agents begin to intersect with the likes of a projected ego path, the risk score gradually increases. The risk typically peaks shortly before the predicted interaction occurs and decreases once the agent moves away from the ego vehicle's projected trajectory path. Temporal smoothing is applied using an exponential moving average, which stabilizes the risk signal and would also prevents abrupt fluctuations caused by transient detection noise. The risk curves produced by the FP32 and INT8 models are highly similar, indicating that the overall risk estimation behavior remains very much stable even when the perception module is quantized.

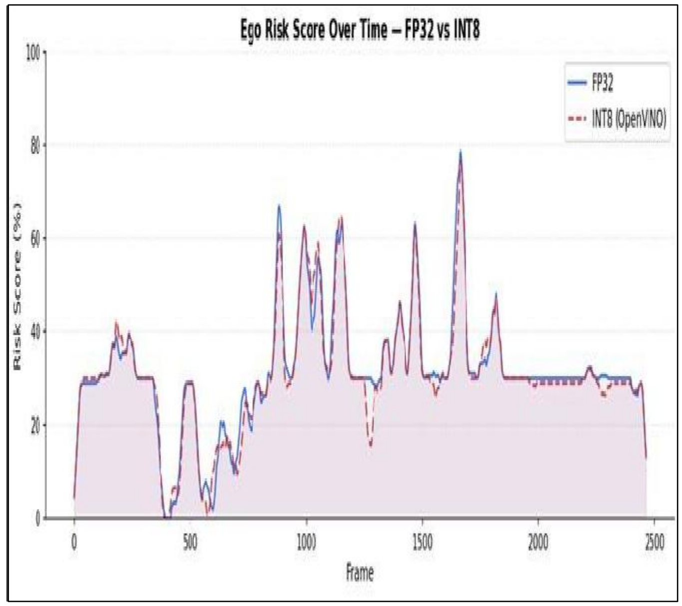


Fig. 3. Temporal evolution of the collision risk signal for FP32 and INT8 detector configurations plotted against frame index.

## D. Lead-Time Analysis

To evaluate the predictive capability of the proposed framework, lead-time analysis is performed for each tracked agent involved that are in a conflict event.

For every interaction, two frames are identified:

- The frame where the risk score associated with the agent reaches its maximum.
- The frame where the agent reaches minimum distance from the ego vehicle.

The predictive lead time is defined as the difference between these two frames converted to seconds using the video frame rate.

Fig. 4 and illustrate the distribution of lead times across three categories:

- Valid predictions (>0.2 s)
- Near-zero predictions (0–0.2 s)
- Negative lead times (<0 s)

The majority of the interactions fall within the valid prediction range, thus very much indicating that the system successfully anticipates potential hazards before closest would even approach. The INT8 model produces slightly fewer near-zero and negative lead-time events, suggesting that quantization does not significantly degrade the very predictive capability of the system.

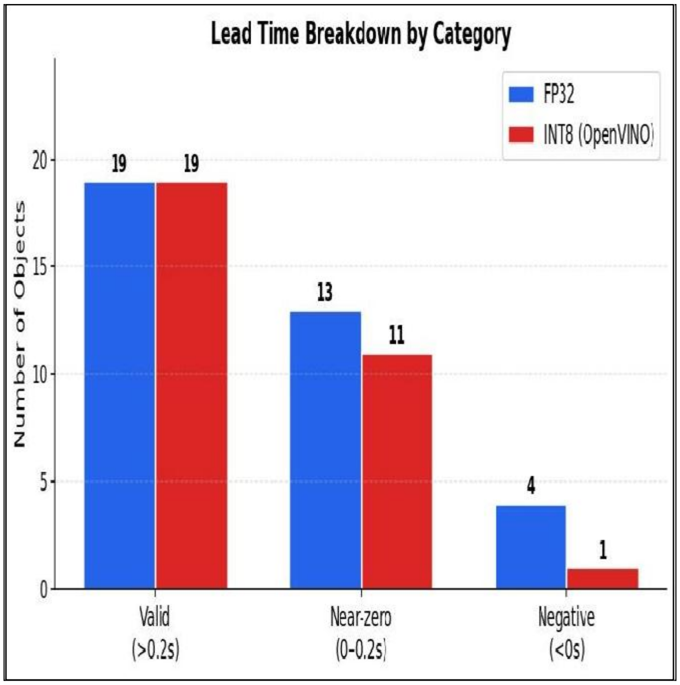


Fig. 4. Lead-time distribution for the FP32 detector configuration.

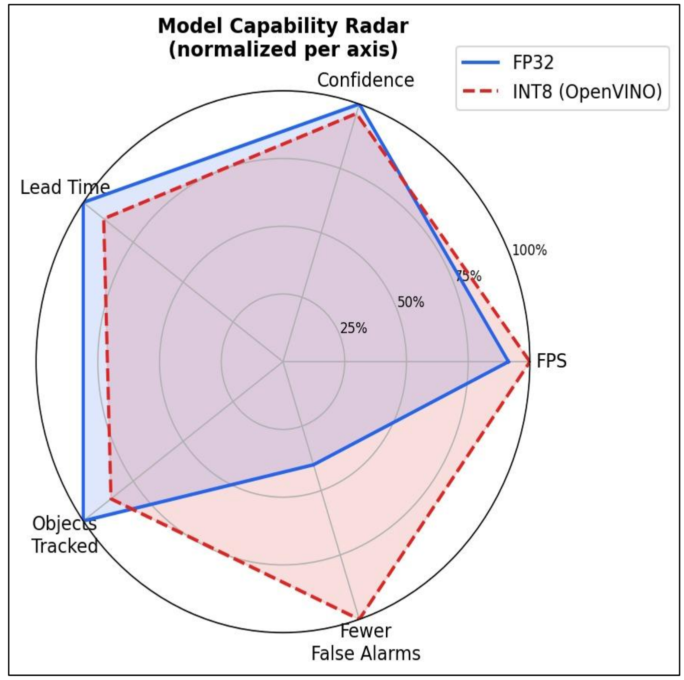


Fig. 5. Model Capability Radar for FP32 and INT8.

## E. Quantitative Summary

To quantify the effect of model quantization, key performance metrics are compared between the FP32 and INT8 detector configurations.

Fig. 6 summarize the comparison across four metrics:

- Average inference speed (FPS)
- Detection confidence
- Number of objects tracked
- Average predictive lead time

The INT8 quantized model very much achieves an inference speed of approximately 6.0 FPS, compared to 5.48 FPS for the FP32 baseline, thereby representing a 9.5% increase in the processing speed. Quantization slightly reduces the average detection confidence from 0.661 to 0.636, which does results in a small decrease in the number of tracked objects which are across the video sequence. This reduction in detection persistence thus leads to a minor decrease in average predictive lead time from 1.152 seconds to 1.034 seconds.

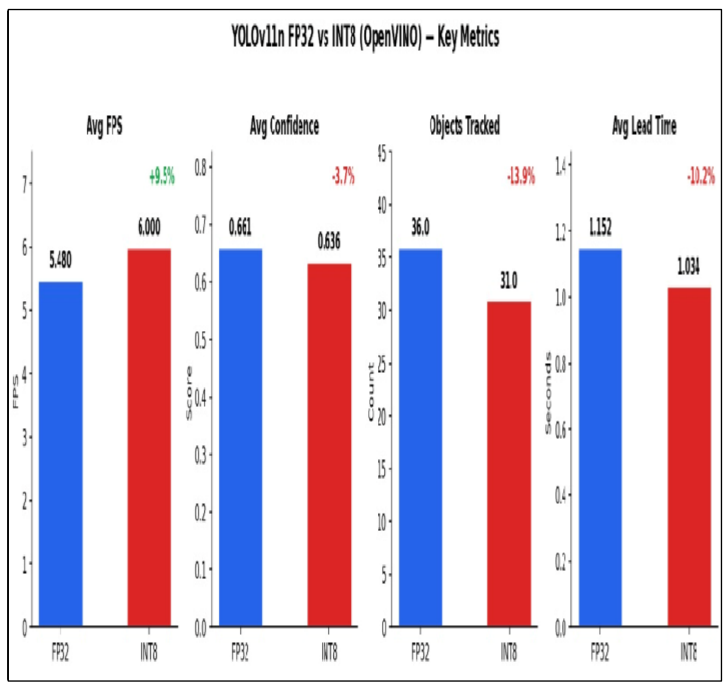


Fig. 6. Quantitative performance metrics for the FP32 baseline and INT8 quantized detector.

# V. Discussion

## A. Interpretation of Results

The experimental results demonstrate that ego-centric trajectory comparison provides an effective mechanism for collision risk estimation. Unlike simple distance-based metrics, the proposed framework evaluates potential hazards in the context of predicted future motion.

The lead-time results very much does indicate that risk scores generally peak before agents reach their closest distance to the ego vehicle, thus confirming that the system provides early warnings of any of the potential conflicts.

The comparison between FP32 and INT8 detector configurations reveals that model quantization introduces a big trade-off between computational efficiency and perception accuracy. While the INT8 model does provide with faster inference, the slight reduction in detection confidence can reduce the number of tracked agents and also shorten the predictive horizon for some of the interactions.

However, the overall shape and behavior of the risk signal remain mostly largely unchanged, suggesting that the proposed framework is robust enough so as to moderate degradation in detection quality.

### B. Strengths of the Proposed Approach

The proposed framework does offer several advantages:

- Predictive risk estimation based on future trajectory interactions only rather than that of instantaneous proximity.
- Model interpretability, as each stage of the pipeline explicitly contributes to the likes of final risk score.
- Lightweight trajectory forecasting, thus avoiding computationally expensive deep learning predictors.
- Edge deployment feasibility, enabled by the very use of quantized perception models.
- Robustness to model compression, as risk estimation performance remains stable under INT8 quantization.

These properties make the proposed system suitable for deployment in real-time driver assistance or embedded perception systems.

### C. Limitations

Despite its effectiveness, the system has several limitations. First, ego vehicle motion is approximated using optical flow, which may be sensitive to camera vibration or lighting variations. Second, agent trajectories are more often predicted using linear the likes of kinematic models that do not explicitly capture complex social interactions or behavioral intent. Additionally, all computations are performed in image space, meaning that distances are measured in pixel units rather than that of real-world coordinates. While this approach is sufficient for relative risk estimation, it often does limits the direct the interpretation of physical safety margins.

Finally, quantization slightly reduces detection reliability, which may affect also the tracking stability in dense traffic scenarios.

### D. Failure Modes

Failure cases typically arise in scenarios involving:

- Abrupt changes in agent direction.
- Partial occlusions that cause track fragmentation.
- Stationary agents that suddenly begin moving.
- Rapid ego vehicle maneuvers that violate the straight-line motion assumption.

These situations highlight potential directions for future improvements, including enhanced ego motion modeling and more advanced trajectory prediction methods.

## VI. Conclusion

This work presented an ego-centric collision risk estimation framework that operates on monocular dashcam video and predicts potential interactions between the ego vehicle and surrounding traffic participants. The proposed system has shown to integrate real-time object detection, multi-object tracking, trajectory forecasting, and spatiotemporal conflict analysis so as to compute continuous collision risk scores. Unlike traditional proximity-based safety metrics, the framework evaluates that of the potential hazards by comparing predicted future trajectories of agents with the likes of projected motion of the ego vehicle. This approach enables the system to detect potential conflicts before even the agents reach their closest approach, providing early warning signals that align very much with intuitive human driving behavior.

To investigate the effect of perception model compression, the study compared a YOLOv11n FP32 baseline detector with an INT8 quantized YOLOv11n model within the same risk estimation pipeline. Experimental results showed that INT8 quantization improved inference speed by approximately 9.5%, increasing processing throughput while maintaining similar overall risk estimation behavior. Although quantization slightly reduced detection confidence and resulted in fewer tracked objects, the resulting impact on predictive lead time was relatively small.

Average lead time decreased from 1.152 seconds to 1.034 seconds, demonstrating that the system remains capable of anticipating hazardous interactions even when using a compressed detector.

Overall, the results indicate that the likes of ego-centric trajectory-based risk estimation is robust enough to moderate perception degradation and can operate effectively with the likes of quantized models. This property makes the framework suitable for deployment in resource-constrained edge environments, such as say for example, embedded driver assistance systems and onboard vehicle perception modules.

### A. Future Work

While the proposed framework have shown to demonstrates promising results, several opportunities does exist for even further improvement.

First, ego vehicle motion estimation could be enhanced by incorporating additional motion cues or sensor fusion techniques. The current implementation relies on optical flow-based motion estimation, which may be sensitive to that of camera vibrations or illumination changes. Integrating with the likes of inertial measurements or visual odometry could improve the accuracy of ego motion prediction.

Second, the current trajectory prediction module does very much assume constant velocity and linear motion. Although this approach is computationally efficient, it may not capture the likes of those complex behavioral patterns such as pedestrian hesitation, sudden direction changes, or socially influenced movement. Future work could explore lightweight learning-based trajectory predictors that will help incorporate interaction-aware models.

Third, risk estimation is currently performed in image space using pixel-based distance measurements as of now. Mapping detected trajectories into a real-world coordinate system using camera calibration or depth estimation could help to enable even more accurate physical risk metrics and also better interpretability of safety margins.

Additionally, further experiments could investigate the impact of model compression techniques beyond INT8 quantization, such as the likes of structured pruning along with knowledge distillation, and mixed-precision inference, to better understand the trade-offs between perception accuracy then computational efficiency, and downstream safety performance.

## References


[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[2] Ultralytics, “YOLOv11: Next-Generation Real-Time Object Detection Architecture,” Ultralytics Documentation, 2024.

[3] Y. Zhang, P. Sun, Y. Jiang et al., “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” in Proc. Euro-pean Conf. Computer Vision (ECCV), 2022.

[4] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” in Proc. IEEE Int. Conf. Image Processing (ICIP), 2017.

[5] S. Sivaraman and M. Trivedi, “Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis,” IEEE Trans. Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773–1795, 2013.

[6] A. Rudenko et al., “Human Motion Trajectory Prediction: A Survey,” Int. J. Robotics Research, vol. 39, no. 8, pp. 895–935, 2020.

[7] A. Alahi et al., “Social LSTM: Human Trajectory Prediction in Crowded Spaces,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[8] J. K. Min, J. H. Lee, and M. S. Kang, “Real-Time Collision Risk Assessment for Advanced Driver Assistance Systems,” IEEE Trans. Vehicular Technology, vol. 68, no. 2, pp. 1271–1282, 2019.

[9] S. Lefèvre, D. Vasquez, and C. Laugier, “A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles,” Robotics and Autonomous Systems, vol. 59, no. 7–8, pp. 444–460, 2011.

[10] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

[11] G. Farnebäck, “Two-Frame Motion Estimation Based on Poly-nomial Expansion,” in Proc. Scandinavian Conf. Image Analy-sis, 2003.

[12] D. Helbing and P. Molnar, “Social Force Model for Pedestrian Dynamics,” Physical Review E, vol. 51, no. 5, pp. 4282–4286, 1995.

[13] M. Treiber and A. Kesting, Traffic Flow Dynamics. Springer, 2013.

[14] S. Rasouli and J. Tsotsos, “Autonomous Vehicles That Interact With Pedestrians: A Survey,” IEEE Trans. Intelligent Transportation Systems, vol. 21, no. 3, pp. 900–918, 2020.

[15] P. Koopman and M. Wagner, “Challenges in Autonomous Vehicle Testing and Validation,” SAE Int. J. Transportation Safety, vol. 4, no. 1, pp. 15–24, 2016.