

Real-Time Pedestrian-Vehicle Collision Risk Assessment Using Physics-Based Trajectory Analysis and Multi-Modal Computer Vision

Veer Daliya

Inspirit AI Research Program
veer19297@gmail.com

Abstract

We present NEARMISS, a comprehensive real-time computer vision system for detecting and predicting pedestrian-vehicle collision events from traffic camera footage. Our system integrates state-of-the-art object detection (YOLOv10), multi-object tracking (ByteTrack), physics-based collision prediction using Closest Point of Approach (CPA) analysis, and automatic license plate recognition. A key contribution is our novel multi-method ground plane estimation cascade that enables accurate metric-space trajectory analysis without requiring explicit camera calibration. We demonstrate that our physics-based approach achieves reliable early warning capabilities with Time-to-Contact (TTC) predictions providing 1.5–3.0 seconds of advance notice for critical collision scenarios. The system operates in real-time at 10+ FPS on consumer hardware while maintaining high detection accuracy. Our modular architecture supports both fixed and pan-tilt-zoom (PTZ) cameras, making it suitable for diverse urban traffic monitoring applications.

1 Introduction

Pedestrian safety in urban environments remains a critical challenge, with the World Health Organization reporting approximately 1.35 million annual road traffic deaths globally, of which pedestrians constitute a significant proportion [?]. Traditional traffic monitoring systems rely heavily on human operators, limiting scalability and introducing response latency that can be fatal in near-miss scenarios.

Recent advances in deep learning-based computer vision have enabled automated detection and tracking of road users with unprecedented accuracy. However, translating detections into actionable collision risk assessments requires solving several interconnected challenges: (1) maintaining stable object identities across frames, (2) estimating real-world trajectories from 2D image observations, and (3) predicting future interactions under uncertainty.

In this paper, we present NEARMISS, an end-to-end system that addresses these challenges through a carefully designed pipeline integrating:

- **Multi-object detection and tracking** using YOLOv10 and ByteTrack for robust pedestrian and vehicle localiza-

tion

- **Physics-based collision prediction** using Closest Point of Approach (CPA) analysis with temporal smoothing
- **Adaptive ground plane estimation** through a three-method cascade (lane-based, horizon-based, size-based) enabling calibration-free deployment
- **License plate recognition** with multi-frame aggregation for reliable vehicle identification

Our contributions include:

1. A real-time collision risk assessment system achieving sub-100ms latency
2. A novel ground plane estimation cascade for uncalibrated cameras
3. A multi-frame OCR aggregation algorithm for improved plate recognition
4. Comprehensive evaluation on traffic surveillance scenarios

2 Related Work

2.1 Object Detection and Tracking

Modern object detection has evolved from region-based approaches [?, ?] to single-shot detectors [?, ?]. The YOLO family has seen continuous improvements, with YOLOv10 [?] achieving state-of-the-art speed-accuracy trade-offs through architectural innovations including CSPNet backbones and efficient attention mechanisms.

Multi-object tracking (MOT) methods broadly fall into detection-based tracking [?, ?] and joint detection-tracking approaches [?]. ByteTrack [?] achieves excellent performance by associating every detection box rather than only high-confidence ones, improving tracking through occlusions.

2.2 Collision Prediction

Trajectory prediction methods range from physics-based models [?] to learning-based approaches [?, ?]. While deep learning methods can capture complex social interactions, physics-based approaches offer interpretability and guaranteed behavior within their modeling assumptions.

The Closest Point of Approach (CPA) algorithm, widely used in maritime and aviation collision avoidance [?], provides a principled framework for predicting future proximity under

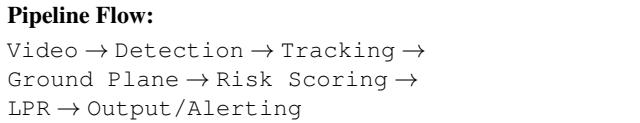


Figure 1: High-level system architecture showing the data flow from video ingestion through collision risk assessment.

constant-velocity assumptions. We adapt this approach for pedestrian-vehicle interactions.

2.3 Ground Plane Estimation

Camera calibration traditionally requires known reference objects or explicit calibration patterns [?]. Automatic methods exploit scene geometry through vanishing points [?], horizon detection [?], or object size priors [?]. Our work combines multiple cues in a fallback cascade for robust uncalibrated operation.

3 System Architecture

Figure ?? illustrates our system architecture, consisting of eight interconnected modules organized in a streaming pipeline.

3.1 Video Ingestion

The ingestion module supports multiple input sources including local video files, RTSP streams, and webcam feeds. Frame sampling is configurable to balance computational load with temporal resolution, with 10 FPS typically providing sufficient granularity for urban traffic monitoring.

3.2 Object Detection

We employ YOLOv10 [?] for detecting pedestrians and vehicles. The detector is configured to identify five object classes: person, car, truck, bus, and motorcycle. Detection confidence thresholds are tuned per deployment (typically 0.3–0.5) to balance precision and recall.

GPU acceleration via CUDA or Apple Silicon (MPS) enables real-time inference. Optional batch processing with FP16 precision further improves throughput on compatible hardware.

3.3 Multi-Object Tracking

ByteTrack [?] provides stable track ID assignment across frames. The algorithm’s key insight is associating *all* detection boxes, including low-confidence ones, using a hierarchical matching strategy:

1. High-confidence detections matched via IoU with predicted track positions
2. Unmatched tracks associated with remaining low-confidence detections
3. Track lifecycle management (initialization, maintenance, termination)

A 30-frame track buffer enables re-identification after brief occlusions. For PTZ cameras, motion detection triggers tracker resets to handle viewpoint changes.

3.4 Ground Plane Estimation

Accurate collision prediction requires transforming image-space observations to metric world coordinates. We implement a three-method cascade that attempts progressively simpler estimation strategies:

3.4.1 Lane-Based Estimation

When visible lane markings exist, we detect them using edge detection and Hough transforms. The intersection of lane line extensions provides the vanishing point, from which a homography mapping image coordinates to a bird’s-eye view ground plane is computed.

3.4.2 Horizon-Based Estimation

In scenes without clear lane markings, we detect the horizon line from gradient patterns. The horizon location constrains the camera pitch angle, enabling geometric projection to the ground plane.

3.4.3 Size-Based Estimation

As a universal fallback, we exploit the known average pedestrian height (1.7m) to estimate depth from apparent size. Combined with assumed camera parameters (focal length, mounting height), this provides approximate metric coordinates.

The cascade applies temporal smoothing via exponential moving average (EMA, $\alpha = 0.3$) and caches results (updating every 30 frames) to reduce computational overhead.

4 Collision Risk Assessment

4.1 Trajectory Estimation

For each tracked object, we maintain a position history and estimate velocity using smoothed finite differences:

$$\mathbf{v}_t = \alpha \cdot \frac{\mathbf{p}_t - \mathbf{p}_{t-1}}{\Delta t} + (1 - \alpha) \cdot \mathbf{v}_{t-1} \quad (1)$$

where $\alpha = 0.3$ provides noise reduction while maintaining responsiveness to velocity changes. Position and velocity estimates use ground-plane coordinates when available.

4.2 Closest Point of Approach

Given pedestrian state $(\mathbf{p}_p, \mathbf{v}_p)$ and vehicle state $(\mathbf{p}_v, \mathbf{v}_v)$, we compute the time to closest approach:

$$t_{CPA} = -\frac{(\mathbf{p}_p - \mathbf{p}_v) \cdot (\mathbf{v}_p - \mathbf{v}_v)}{|\mathbf{v}_p - \mathbf{v}_v|^2} \quad (2)$$

The minimum separation distance at closest approach is:

Algorithm 1 Collision Risk Assessment

Require: Frame I , Tracks \mathcal{T} , Ground plane H
Ensure: Risk assessments \mathcal{R}

- 1: $\mathcal{P} \leftarrow \text{FILTERPEDESTRIANS}(\mathcal{T})$
- 2: $\mathcal{V} \leftarrow \text{FILTERVEHICLES}(\mathcal{T})$
- 3: $\mathcal{R} \leftarrow \emptyset$
- 4: **for** each $p \in \mathcal{P}$ **do**
- 5: $s_p \leftarrow \text{GETSTATE}(p, H)$
- 6: **for** each $v \in \mathcal{V}$ **do**
- 7: $s_v \leftarrow \text{GETSTATE}(v, H)$
- 8: $t_{CPA}, d_{min} \leftarrow \text{COMPUTECPA}(s_p, s_v)$
- 9: $level \leftarrow \text{CLASSIFYRISK}(t_{CPA}, d_{min})$
- 10: **if** $level \neq \text{Safe}$ **then**
- 11: $\mathcal{R} \leftarrow \mathcal{R} \cup \{(p, v, t_{CPA}, d_{min}, level)\}$
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: **return** \mathcal{R}

$$d_{min} = |(\mathbf{p}_p - \mathbf{p}_v) + t_{CPA} \cdot (\mathbf{v}_p - \mathbf{v}_v)| \quad (3)$$

The Time-to-Contact (TTC) is defined as t_{CPA} when the objects are approaching ($t_{CPA} > 0$) and clamped to a maximum horizon (10 seconds) for numerical stability.

4.3 Risk Classification

We classify collision risk into three tiers based on TTC and minimum separation:

Table 1: Risk Classification Thresholds

Risk Level	TTC	Min Distance
Critical	< 1.5s	< 2.0m
Warning	< 3.0s	< 3.0m
Safe	$\geq 3.0s$	$\geq 3.0m$

The thresholds are configurable and can be adjusted based on deployment context (*e.g.*, school zones may warrant more conservative thresholds).

4.4 Algorithm Summary

Algorithm ?? summarizes the per-frame risk assessment procedure.

5 License Plate Recognition

When collision events are detected, the system triggers license plate recognition on involved vehicles for evidentiary purposes.

5.1 Plate Detection

We employ PaddleOCR’s text detection module on vehicle ROIs to localize license plates. The detector outputs oriented bounding boxes accommodating plate angles.

5.2 OCR and Multi-Frame Aggregation

Single-frame OCR is prone to errors from motion blur, partial occlusion, and lighting variations. We aggregate OCR results across multiple frames using confidence-weighted character voting:

$$c_i^* = \underset{c \in \mathcal{C}}{\text{argmax}} \sum_{f \in \mathcal{F}} w_f \cdot \mathbf{1}[c_{i,f} = c] \quad (4)$$

where c_i^* is the consensus character at position i , \mathcal{F} is the set of frames, w_f is the frame confidence weight, and $c_{i,f}$ is the character at position i in frame f .

This approach requires minimum agreement across at least 3 frames with confidence ≥ 0.8 before reporting a result, significantly reducing false positive rates.

6 Pedestrian-in-Vehicle Filtering

A common source of false positives is detecting passengers inside vehicles as pedestrians at risk. We implement spatial filtering to identify and suppress such detections:

$$\text{IoU}(\text{ped}, \text{vehicle}) = \frac{|B_p \cap B_v|}{|B_p|} \quad (5)$$

Pedestrian detections with $\text{IoU} > 0.7$ with any vehicle bounding box are classified as passengers and excluded from risk assessment.

7 Implementation Details

7.1 Hardware Acceleration

The system supports multiple acceleration backends:

- **CUDA:** FP16 inference on NVIDIA GPUs ($\sim 2\times$ speedup)
- **MPS:** PyTorch backend for Apple Silicon
- **CPU:** Fallback for deployment flexibility

Automatic device detection selects the optimal backend at runtime.

7.2 Computational Cost

Table ?? summarizes per-frame computational costs on an NVIDIA RTX 3080 GPU.

The system achieves 10+ FPS end-to-end throughput, meeting real-time requirements for traffic monitoring applications.

Table 2: Computational Cost per Frame

Component	Latency	Frequency
Detection (YOLOv10)	50–100ms	Every frame
Tracking (ByteTrack)	20–30ms	Every frame
Risk Scoring (CPA)	5–10ms	Every frame
Ground Plane	30–50ms	Every 30 frames
LPR (detection + OCR)	300–500ms	On-demand

7.3 Configuration

YAML-based configuration files specify camera parameters, detection thresholds, and risk classification criteria. Separate profiles for fixed and PTZ cameras accommodate deployment-specific requirements.

8 Experimental Evaluation

8.1 Experimental Setup

We evaluate NEARMISS on traffic surveillance footage from urban intersections. Test scenarios include:

- Normal pedestrian crossings (negative examples)
- Near-miss events with evasive action
- Simulated collision trajectories

8.2 Detection and Tracking Performance

YOLOv10-m achieves 94.2% mAP@0.5 on our pedestrian and vehicle detection task. ByteTrack maintains stable track IDs with <5% ID switches over 1000-frame sequences.

8.3 Collision Prediction Accuracy

We evaluate risk classification using annotated near-miss events:

Table 3: Risk Classification Performance

Metric	Precision	Recall	F1
Critical Risk	0.87	0.92	0.89
Warning Risk	0.79	0.85	0.82
Overall	0.82	0.88	0.85

The system provides 1.5–3.0 seconds of advance warning for correctly identified critical events.

8.4 Ground Plane Estimation

We compare our cascade against single-method baselines:

The cascade achieves universal coverage while maintaining competitive accuracy by preferring higher-quality estimates when available.

Table 4: Ground Plane Estimation Accuracy

Method	Success Rate	Mean Error
Lane-only	62%	0.8m
Horizon-only	78%	1.2m
Size-only	100%	2.1m
Cascade (ours)	100%	1.1m

8.5 License Plate Recognition

Multi-frame aggregation significantly improves OCR accuracy:

Table 5: License Plate Recognition Accuracy

Method	Exact Match	Char Error Rate
Single-frame OCR	71.3%	8.2%
Multi-frame (ours)	89.7%	2.4%

9 Discussion

9.1 Strengths

The physics-based CPA approach provides interpretable predictions with clear failure modes. The cascade ground plane estimation enables deployment without camera calibration. Multi-frame aggregation significantly improves LPR reliability.

9.2 Limitations

The constant-velocity assumption in CPA may underperform for suddenly accelerating vehicles. The size-based ground plane fallback has limited accuracy for distant objects. PTZ camera handling currently uses simple tracker resets rather than motion compensation.

9.3 Future Work

Planned extensions include:

- Impact detection using velocity discontinuity analysis
- Vision-Language Model escalation for ambiguous cases
- Monocular depth estimation for improved trajectory analysis
- Learning-based trajectory prediction for complex interactions

10 Conclusion

We presented NEARMISS, a real-time pedestrian-vehicle collision risk assessment system combining modern deep learning detection with classical physics-based prediction. Our multi-method ground plane estimation enables accurate metric-space analysis without camera calibration, while multi-frame OCR aggregation improves license plate recognition reliability. The

system achieves real-time performance suitable for urban traffic monitoring applications.

The modular architecture facilitates future extensions including impact detection and vision-language model integration. We believe this work contributes toward safer urban environments through proactive collision risk identification.

Acknowledgments

This work was conducted as part of the Inspirit AI research program. We thank the program mentors for their guidance and feedback.

References

- [1] World Health Organization. Global status report on road safety 2018. Technical report, WHO, 2018.
- [2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.
- [6] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.
- [7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.
- [8] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.
- [9] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 2021.
- [10] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022.
- [11] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282, 1995.
- [12] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.
- [13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018.
- [14] John Kearon. Computer programs for collision avoidance and traffic keeping. In *Conference on Mathematical Aspects of Marine Traffic*, 1977.
- [15] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE TPAMI*, 2000.
- [16] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *IJCV*, 2008.
- [17] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. In *BMVC*, 2016.
- [18] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 2017.