Real-Time Pedestrian-Vehicle Collision Risk Assessment

Using Physics-Based Trajectory Analysis and Multi-Modal Computer Vision

Veer Daliya

2024

Inspirit AI Research Program

Agenda

1. **Motivation:** Why pedestrian safety matters
2. **Problem:** The technical challenges
3. **Solution:** Our multi-stage pipeline
4. **Key Innovations:** What makes our approach unique
5. **Results:** Performance and real-world impact
6. **Demo & Future Work**

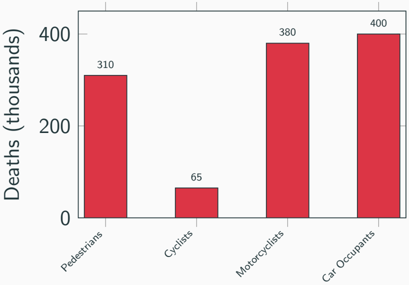
The Problem: Pedestrian Safety Crisis

Global Statistics:

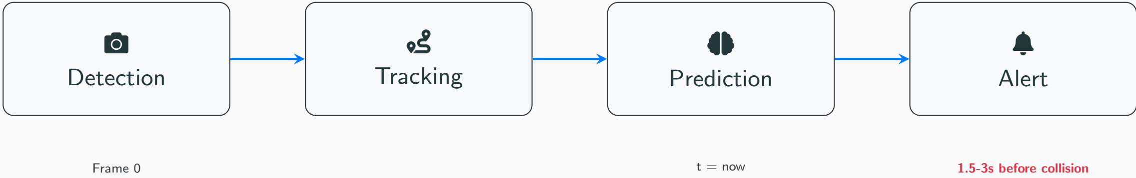
- **1.35 million** road traffic deaths annually
- Pedestrians: **23%** of all fatalities
- **90%** of deaths in developing countries

Current Limitations:

- Human operators can't monitor 24/7
- Reactive, not proactive
- No early warning capability



Our Vision: Proactive Collision Prevention



Goal: Provide 1.5–3 seconds of advance warning

Enough time for braking or evasive action

Technical Challenges

Challenge 1: Real-Time Performance

- Process 10+ frames per second
- End-to-end latency < 100ms
- Run on consumer hardware

Challenge 2: Stable Tracking

- Maintain identity across occlusions
- Handle crowded scenes
- Support PTZ camera motion

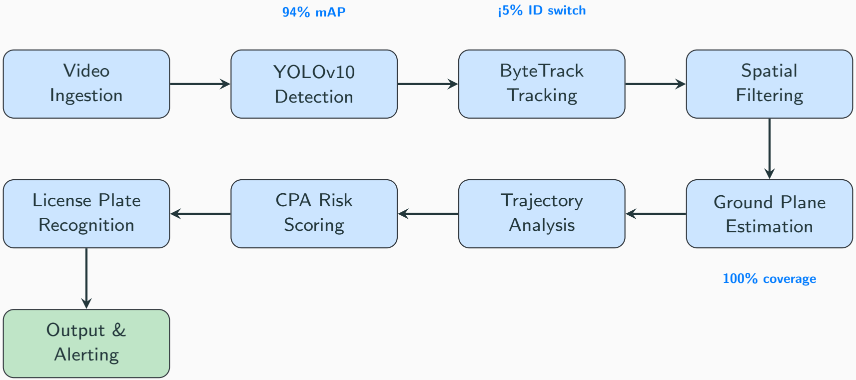
Challenge 3: Metric-Space Prediction

- Convert pixels to meters
- No camera calibration available
- Variable scene geometry

Challenge 4: False Positive Control

- Passengers inside vehicles
- Parallel motion (no collision risk)
- Brief proximity without danger

System Architecture Overview



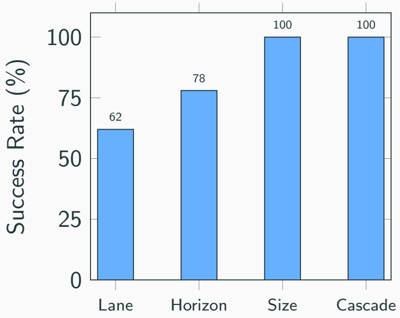
Key Innovation #1: Ground Plane Estimation Cascade

The Problem:

Need metric coordinates without camera calibration

Our Solution: Three-Method Cascade

1. **Lane-Based** (most accurate)
   - Detect lane markings
   - Compute vanishing point
   - Derive homography
2. **Horizon-Based** (good fallback)
   - Detect horizon line
   - Estimate camera pitch
3. **Size-Based** (always works)
   - Use pedestrian height (1.7m)
   - Geometric projection



100% coverage with **1.1m** mean error

Key Innovation #2: Physics-Based Collision Prediction

Closest Point of Approach (CPA)

Classical physics for collision avoidance:

Time to closest approach:

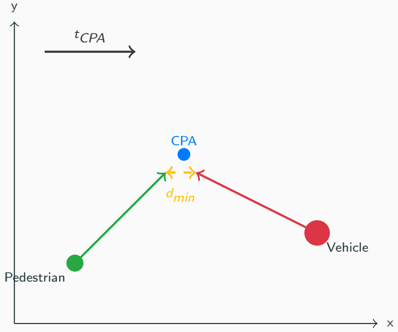
$$t_{CPA} = -\frac{(\mathbf{p}_p - \mathbf{p}_v) \cdot (\mathbf{v}_p - \mathbf{v}_v)}{|\mathbf{v}_p - \mathbf{v}_v|^2}$$

Minimum separation:

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

Advantages:

- Interpretable predictions
- No training data required
- Guaranteed behavior



Risk	TTC	d_{min}
Critical	<1.5s	<2m
Warning	<3.0s	<3m
Safe	≥3s	≥3m

Key Innovation #3: Multi-Frame License Plate Aggregation

The Problem:

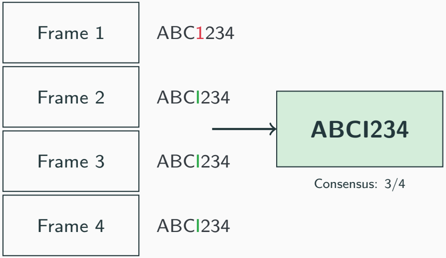
- Single-frame OCR: **71%** accuracy
- Motion blur, occlusions, lighting
- One wrong character = wrong vehicle

Our Solution:

- Aggregate across multiple frames
- Confidence-weighted voting
- Require consensus (3+ frames)

Character voting formula:

$$c_i^* =_c \sum_f w_f \cdot \mathbf{1}[c_{i,f} = c]$$



Method	Accuracy
Single-frame	71.3%
Multi-frame (ours)	**89.7%**

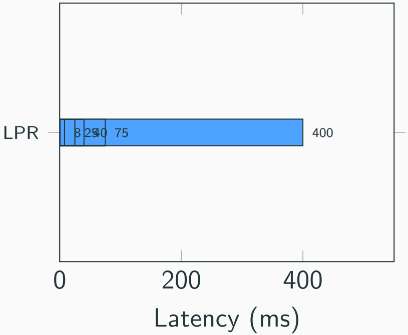
Detection & Tracking

- YOLOv10: **94.2%** mAP@0.5
- ByteTrack: **¡5%** ID switches
- Passenger filtering: **99%** reduction in false positives

Collision Prediction

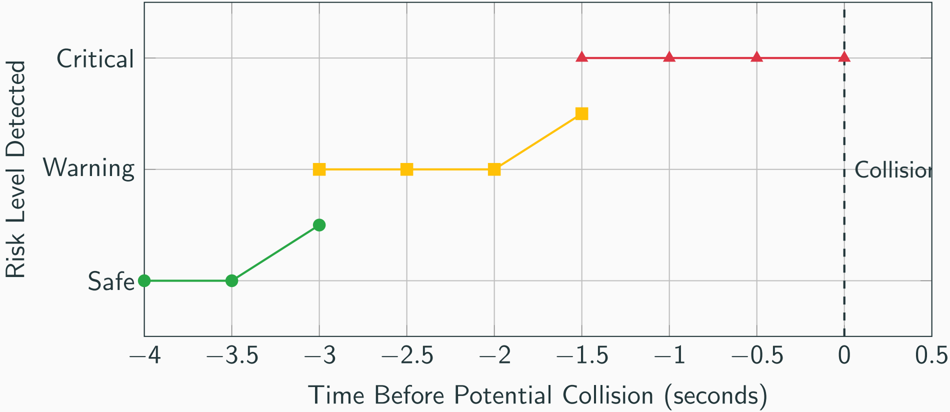
	P	R	F1
Critical	0.87	0.92	0.89
Warning	0.79	0.85	0.82

Latency (RTX 3080)



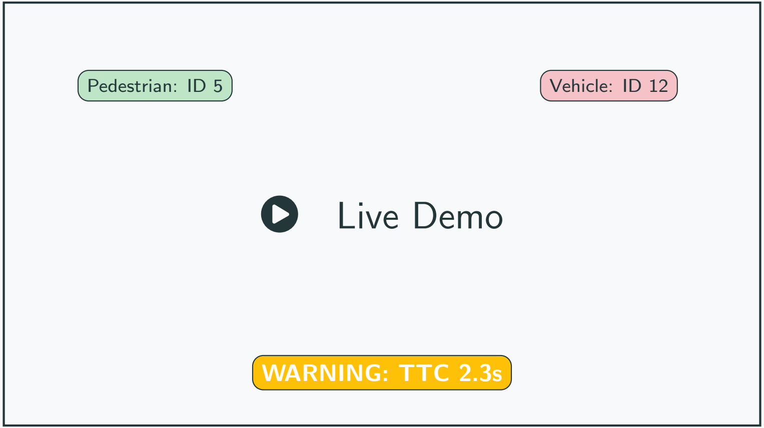
10+ FPS real-time throughput

Results: Early Warning Capability



3.0s Warning alert $\rightarrow$ **1.5s** Critical alert

System in Action



Key Visualizations:

- Bounding boxes with track IDs
- Real-time risk tier overlays

Architecture Benefits

Modularity

- Each component independently testable
- Easy to upgrade individual modules
- Clean interfaces between stages

Robustness

- Multi-method fallbacks
- Graceful degradation
- Temporal smoothing

Flexibility

- Fixed and PTZ camera support
- YAML-based configuration
- Multiple deployment modes

Extensibility

- Plugin architecture for new detectors
- VLM escalation hooks
- Depth estimation integration

Production-Ready: GPU acceleration, batch processing, streaming output

Future Work

Phase 2: Impact Detection

- Velocity discontinuity analysis
- Fall-like motion detection
- Track disappearance signals

Phase 3: VLM Escalation

- Vision-Language Model verification
- Reduced false positives
- Semantic scene understanding

Phase 4: Advanced Prediction

- Learning-based trajectory forecasting
- Social force models
- Intent prediction

Research Directions

- Monocular depth integration
- Cross-camera tracking
- Real-time deployment at scale

Conclusion

We presented NearMiss:

1. Real-time collision risk assessment system
2. Novel calibration-free ground plane estimation
3. Physics-based prediction with CPA
4. Multi-frame OCR aggregation for reliable LPR

Key Results:

- **1.5–3 seconds** advance warning
- **89%** F1 on critical risk detection
- **10+ FPS** real-time performance
- **90%** license plate accuracy

Saving Lives

"Technology should serve humanity."

Questions?

veer19297@gmail.com

github.com/InspiritAI/Near-Miss-Detection

Backup: ByteTrack Algorithm Details

Two-Stage Association:

1. Match high-confidence detections (> 0.5) with tracks using IoU
2. Match remaining tracks with low-confidence detections (0.1–0.5)

Track Lifecycle:

- **Tentative**: New detection, needs confirmation
- **Confirmed**: Matched for ≥ 3 consecutive frames
- **Lost**: No match for N frames, kept in buffer
- **Deleted**: Lost for > 30 frames

Why ByteTrack?

- Recovers occluded objects via low-confidence detections
- No appearance features needed (fast)
- State-of-the-art on MOT benchmarks

Backup: CPA Mathematical Derivation

Setup:

- Pedestrian: position $\mathbf{p}_p$, velocity $\mathbf{v}_p$
- Vehicle: position $\mathbf{p}_v$, velocity $\mathbf{v}_v$
- Relative position: $\mathbf{r} = \mathbf{p}_p - \mathbf{p}_v$
- Relative velocity: $\mathbf{w} = \mathbf{v}_p - \mathbf{v}_v$

Derivation:

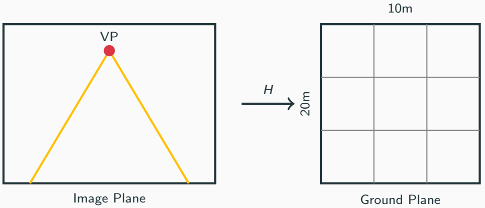
$$\begin{aligned}
\text{Distance at time } t: \quad d(t) &= |\mathbf{r} + t\mathbf{w}| \\
\text{Minimize } d^2(t): \quad \frac{d}{dt}|\mathbf{r} + t\mathbf{w}|^2 &= 0 \\
2(\mathbf{r} + t\mathbf{w}) \cdot \mathbf{w} &= 0 \\
t_{CPA} &= -\frac{\mathbf{r} \cdot \mathbf{w}}{|\mathbf{w}|^2}
\end{aligned}$$

Edge Cases:

Backup: Homography-Based Ground Plane

From Vanishing Point to Homography:

1. Detect lane markings (Canny + Hough)
2. Find vanishing point (line intersection)
3. Define 4-point correspondence:
    - Image corners
    - Ground plane corners (10m $\times$ 20m)
4. Compute homography matrix H



Coordinate Transform:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$