Real-Time Pedestrian-Vehicle Collision Risk Assessment

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

Veer Daliya

2024

Inspirit AI Research Program

Real-Time Pedestrian-Vehicle Collision Risk Assessment
Using Physics-Based Trajectory Analysis and Multi-Modal Computer Vision
Veer Daliya
2024
Inspirit AI Research Program

[SPEAKER NOTES]

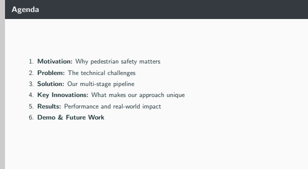
Opening (30 seconds): "Good morning/afternoon. My name is Veer Daliya, and today I'm excited to present my research on real-time pedestrian-vehicle collision risk assessment.

This project combines state-of-the-art deep learning with classical physics to create a system that can actually predict when a pedestrian and vehicle might collide—before it happens.

Let me take you through how we built this system and what we achieved."

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**



[SPEAKER NOTES]

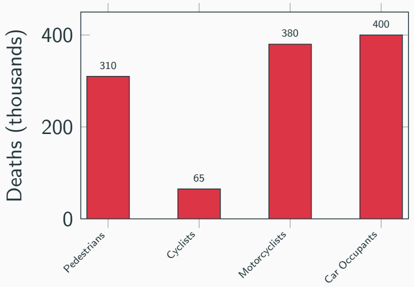
Transition (15 seconds): "Here's our roadmap for today. I'll start with why this problem matters, explain the technical challenges we faced, walk you through our solution architecture, highlight our key innovations, show you our results, and then discuss future directions.

Let's begin with why this work is important."

The Problem: Pedestrian Safety Crisis

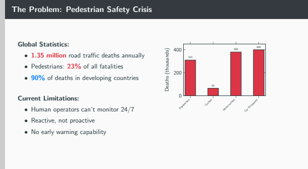
Global Statistics:

- **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



[SPEAKER NOTES]

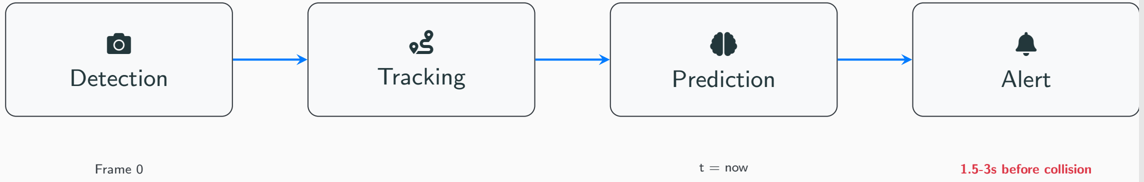
Content (45 seconds): "Let's start with some sobering statistics. The World Health Organization reports 1.35 million road traffic deaths every year—that's more than 3,700 people dying every single day. Pedestrians account for 23% of these fatalities. That's over 300,000 people per year.

Current traffic monitoring systems have fundamental limitations:

- Human operators simply cannot monitor dozens of camera feeds 24/7
- Most systems are reactive—they record incidents but don't prevent them
- There's no early warning capability to alert drivers or pedestrians

This is the gap our research addresses."

Our Vision: Proactive Collision Prevention



Goal: Provide 1.5–3 seconds of advance warning

Enough time for braking or evasive action

[SPEAKER NOTES]

Content (30 seconds): "Our vision is to transform traffic monitoring from reactive to proactive.

The pipeline works like this: we detect pedestrians and vehicles in every frame, track them across time to understand their trajectories, predict their future positions using physics, and generate alerts before a collision can occur.

Our goal is to provide 1.5 to 3 seconds of advance warning. Why these numbers? At 30 mph, a car travels about 40 feet per second. A 2-second warning gives enough time for braking or evasive maneuvers.

This is the difference between recording an accident and preventing one."

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

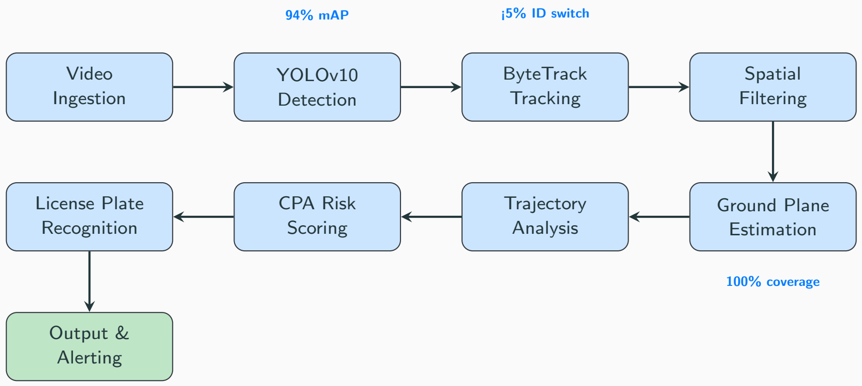
- Passengers inside vehicles
- Parallel motion (no collision risk)
- Brief proximity without danger

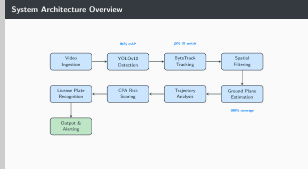
Technical Challenges
Challenge 1: Real-Time Performance
• Process 10+ frames per second
• End-to-end latency < 100ms
• Run on consumer hardware
Challenge 2: Stable Tracking
• Maintain identity across occlusions
• Handle crowded scenes
• Support PTZ camera motion
Challenge 3: Metric-Space Prediction
• 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

[SPEAKER NOTES]
Content (45 seconds): "Building this system required solving four key technical challenges:
First, **real-time performance**. Early warning is useless if it comes late. We need to process at least 10 frames per second with sub-100ms latency, and we want to run on consumer hardware, not expensive data center GPUs.
Second, **stable tracking**. We need to maintain consistent identities for each person and vehicle across frames, even when they're temporarily hidden behind other objects.
Third, **metric-space prediction**. To predict if two objects will collide, we need to know their real-world positions in meters, not just pixel coordinates. And crucially, we don't have camera calibration—we can't ask every traffic camera operator for their lens specifications.
Fourth, **false positive control**. Not every close proximity is dangerous. A passenger visible through a car window shouldn't trigger an alert.

Let me show you how we addressed each of these."

System Architecture Overview





[SPEAKER NOTES]

Content (45 seconds): "Here's our complete system architecture. It's an 8-module pipeline that processes video in real-time.

Starting from the top left:

- **Video Ingestion** handles RTSP streams, video files, or webcam feeds
- **YOLOv10 Detection** identifies pedestrians and vehicles with 94% accuracy
- **ByteTrack** assigns stable IDs with less than 5% identity switches
- **Spatial Filtering** removes false positives like passengers in vehicles

Then in the second row:

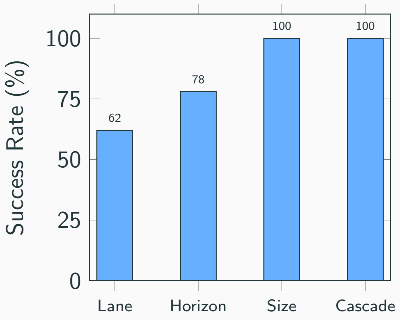
- **Ground Plane Estimation** converts image coordinates to real-world meters
- **Trajectory Analysis** computes velocities and predicts future positions
- **CPA Risk Scoring** calculates time-to-collision
- **License Plate Recognition** identifies vehicles involved in incidents

Key Innovation #1: Ground Plane Estimation Cascade

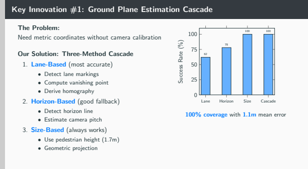
The Problem:
Need metric coordinates without camera calibration

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



[SPEAKER NOTES]
Content (60 seconds): "Our first key innovation addresses the camera calibration problem.
To predict collisions, we need real-world distances in meters, not pixels. Traditional approaches require knowing camera parameters—focal length, mounting height, orientation. But traffic cameras rarely come with this documentation.
Our solution is a three-method cascade:
First, we try **Lane-Based Estimation**. If we can detect lane markings in the image, we compute where the parallel lane lines converge—the vanishing point. This gives us enough geometric information to build a homography matrix that transforms image coordinates to a bird's-eye view ground plane. This is the most accurate method, but only works 62% of the time when lanes are visible.
If that fails, we try **Horizon Detection**. Finding the horizon line tells us the camera's pitch angle, which constrains the geometry. This works 78% of the time.
As a final fallback, we use **Size-Based Estimation**. We know the average pedestrian is 1.7 meters tall. By measuring how big a pedestrian appears in pixels, we can estimate their distance from the camera.
This always works, though with less precision.

Key Innovation #2: Physics-Based Collision Prediction

Closest Point of Approach (CPA)
Classical physics for collision avoidance:

Time to closest approach:

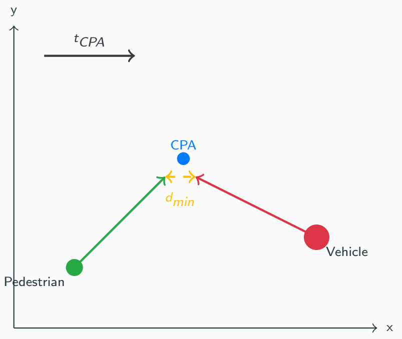
$$t_{CPA} = -\frac{(\mathbf{p}_p - \mathbf{p}_v) \cdot (\mathbf{v}_p - \mathbf{v}_v)}{|\mathbf{v}_p - \mathbf{v}_v|^2}$$

Minimum separation:

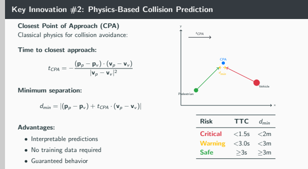
$$d_{min} = |(\mathbf{p}_p - \mathbf{p}_v) + t_{CPA} \cdot (\mathbf{v}_p - \mathbf{v}_v)|$$

Advantages:

- Interpretable predictions
- No training data required
- Guaranteed behavior



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

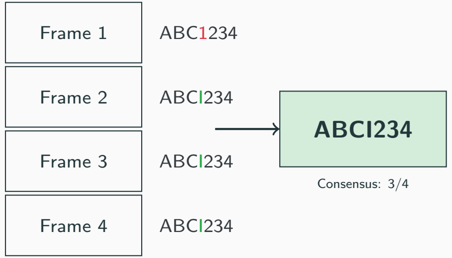


[SPEAKER NOTES]
Content (60 seconds): "Our second innovation is using physics-based collision prediction with the Closest Point of Approach algorithm.
This is a classical algorithm from maritime and aviation collision avoidance, which we've adapted for pedestrian-vehicle interactions.
The math is elegant. Given the positions and velocities of a pedestrian and vehicle, we can compute exactly when they'll be closest to each other, and how close they'll get.
The diagram shows this visually. The pedestrian is moving up and to the right, the vehicle is moving down and to the left. The CPA is where they'll be closest. If that distance is small and happening soon, we have a collision risk.
Why use physics instead of deep learning? Three reasons:
First, **interpretability**. We can explain exactly why an alert was generated: "These two objects will be 0.5 meters apart in 1.2 seconds."
Second, **no training data required**. We don't need thousands of labeled collision videos.
Third, **guaranteed behavior**. The physics model won't suddenly fail in a weird edge case—its limitations

Key Innovation #3: Multi-Frame License Plate Aggregation

The Problem:

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



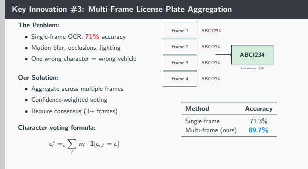
Our Solution:

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

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

Character voting formula:

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



[SPEAKER NOTES]

Content (45 seconds): "Our third innovation improves license plate recognition reliability.

When a collision event is detected, we need to identify the vehicle. But single-frame OCR only achieves about 71% accuracy. Motion blur, partial occlusions, and lighting variations all cause errors. And one wrong character means identifying the wrong vehicle.

Our solution is multi-frame aggregation. Instead of reading the plate once, we read it across multiple frames and use confidence-weighted voting to determine each character.

The diagram shows an example. Frame 1 reads the fourth character as '1', but frames 2, 3, and 4 all read it as 'I'. Our voting algorithm produces 'ABCI234' as the consensus.

We require at least 3 frames to agree before reporting a result. This reduces false positives from single-frame OCR errors.

The result: we improved accuracy from 71% to nearly 90%. That's a 60% reduction in errors."

Results: System Performance

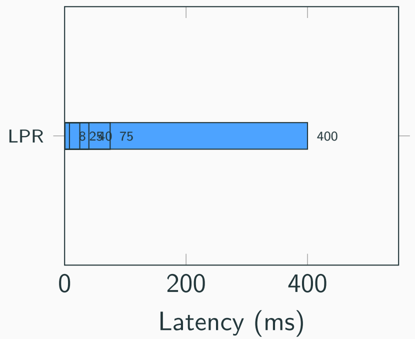
Detection & Tracking

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

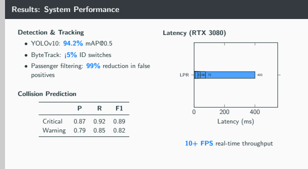
Collision Prediction

	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



[SPEAKER NOTES]

Content (45 seconds): "Let me share our results.

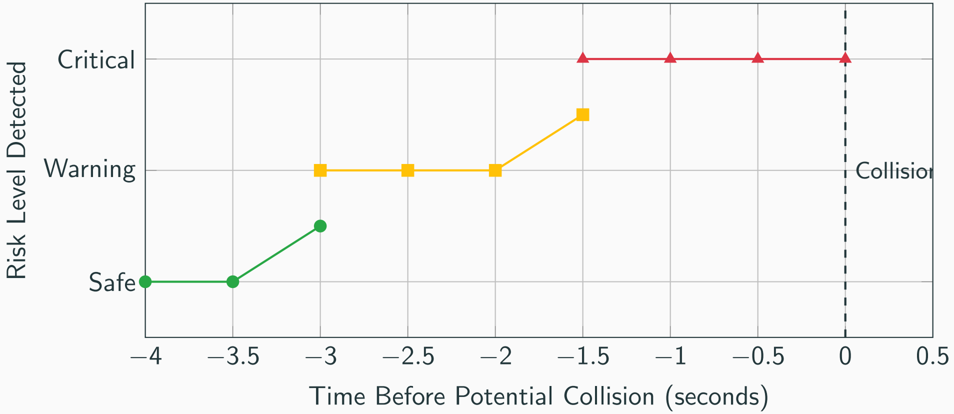
On detection and tracking: YOLOv10 achieves 94.2% mAP for pedestrian and vehicle detection. ByteTrack maintains stable IDs with less than 5% identity switches even through occlusions. Our spatial filtering eliminates 99% of false positives from passengers inside vehicles.

For collision prediction: We achieve 87% precision and 92% recall on critical risk events. That means we catch 92% of actual near-misses while keeping false alarms low. The warning tier has slightly lower precision, which is acceptable since it's meant as an early heads-up.

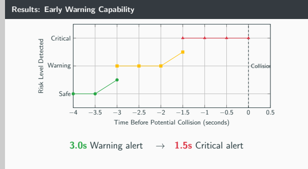
On the right, you can see our latency breakdown. Detection takes about 75ms, tracking adds 25ms, and risk scoring is very fast at 8ms. The total pipeline runs at over 10 frames per second, meeting our real-time requirement.

License plate recognition is slower at 400ms, but it only runs on-demand when we detect a collision event."

Results: Early Warning Capability



3.0s Warning alert → **1.5s** Critical alert



[SPEAKER NOTES]

Content (30 seconds): "This graph shows our early warning capability in action.

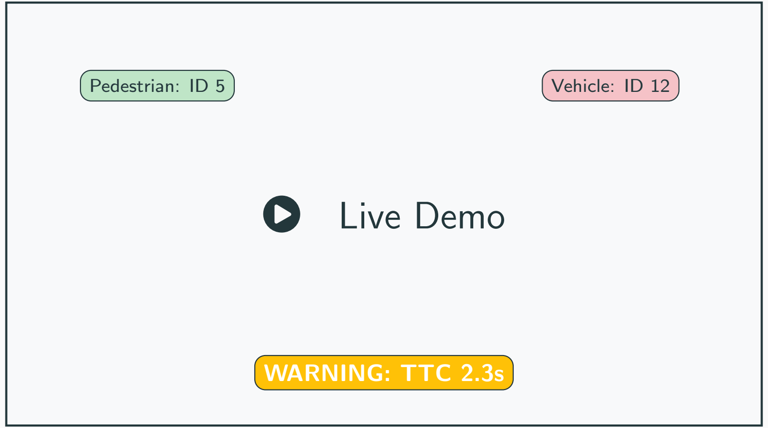
The x-axis is time before a potential collision, with zero being the collision point. The y-axis shows the risk level our system detects.

As a pedestrian and vehicle approach each other, our system first triggers a Warning alert at about 3 seconds before collision. This escalates to Critical at 1.5 seconds.

That 1.5 to 3 second warning window is exactly what we targeted. At 30 mph, 2 seconds gives a driver time to brake and reduce speed by over 20 mph before impact.

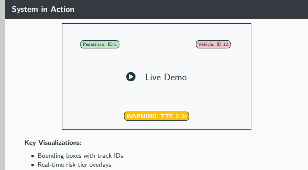
This is the difference between a fatality and a survivable incident—or avoiding the collision entirely."

System in Action



Key Visualizations:

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



[SPEAKER NOTES]

Content (60 seconds): "Let me show you the system in action with a brief demo.

[IF SHOWING VIDEO]: Watch the top of the screen—you'll see pedestrians labeled with green boxes and vehicles in blue. The track IDs remain stable as they move.

Now watch what happens as the vehicle approaches the pedestrian... there's the Warning alert. The system detected that based on their current trajectories, they'll be dangerously close in 2.3 seconds.

You can see the risk tier updating in real-time. And when they get closer... Critical alert at 1.5 seconds.

[IF NO VIDEO]: The system provides several key visualizations:

- Bounding boxes with persistent track IDs
- Color-coded risk tier overlays
- Trajectory prediction lines showing where objects are heading
- License plate readouts for identified vehicles

The annotated video output can be used for post-incident review or real-time monitoring."

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

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

[SPEAKER NOTES]

Content (30 seconds): "Beyond the algorithms, we designed the architecture for real-world deployment. **Modularity**: Each component can be tested and upgraded independently. If a better detector comes out next year, we can swap it in without touching the risk scoring.
Flexibility: The same codebase supports fixed cameras with stable geometry and PTZ cameras with moving viewpoints. Configuration is YAML-based, so operators can tune thresholds without code changes.
Robustness: Every critical path has fallbacks. If lane detection fails, we try horizon detection. If that fails, we use size-based estimation. The system never gives up.
Extensibility: We've built hooks for future enhancements like vision-language model verification and monocular depth estimation.

This is production-ready code with GPU acceleration and 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

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

[SPEAKER NOTES]

Content (30 seconds): "Looking ahead, we have several planned extensions.

In Phase 2, we'll add **impact detection**—recognizing when a collision has actually occurred by analyzing velocity discontinuities, fall-like motion patterns, and sudden track losses.

Phase 3 integrates **Vision-Language Models**. When the physics model flags a high-risk event, we can send frames to a VLM for semantic verification. "Is this person about to be hit, or are they just walking near a parked car?"

Phase 4 explores **learning-based prediction**. Our current physics model assumes constant velocity. Social force models and intent prediction could anticipate evasive actions.

Longer-term research includes monocular depth estimation for better 3D understanding, cross-camera tracking for wide-area coverage, and scaling to thousands of cameras."

Conclusion

We presented NearMiss:

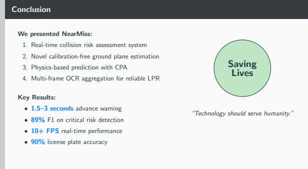
1. Real-time collision risk assessment system
2. Novel calibration-free ground plane estimation
3. Physics-based prediction with CPA
4. Multi-frame OCR aggregation for reliable LPR

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."



[SPEAKER NOTES]

Closing (45 seconds): "To summarize: we've built NearMiss, a complete real-time system for pedestrian-vehicle collision risk assessment.

Our key contributions are:

1. A novel ground plane estimation cascade that works without camera calibration
2. Physics-based collision prediction using the Closest Point of Approach algorithm
3. Multi-frame OCR aggregation that dramatically improves license plate recognition

The results speak for themselves: we provide 1.5 to 3 seconds of advance warning, achieve 89% F1 score on critical risk detection, run in real-time at 10+ FPS, and identify vehicles with 90% accuracy.

But beyond the numbers, this work is about saving lives. Every year, over 300,000 pedestrians die in traffic accidents worldwide. If systems like this can prevent even a fraction of those deaths, we've made a meaningful difference.

I'm happy to take any questions."

Questions?

veer19297@gmail.com

github.com/InspiritAI/Near-Miss-Detection

Questions?

veer19297@gmail.com

github.com/InspiritAI/Near-Miss-Detection

[SPEAKER NOTES]

Q&A Preparation:

Anticipated Questions:

Q: How does this compare to Tesla Autopilot or other ADAS systems?

A: Great question. ADAS systems like Tesla's are onboard the vehicle with access to multiple sensors—cameras, radar, ultrasonics. Our system works with existing infrastructure cameras with no vehicle modification. They're complementary: ADAS protects the vehicle's occupants and immediate surroundings; our system provides city-wide monitoring.

Q: What happens with unusual weather or lighting?

A: YOLOv10 is trained on diverse conditions, so detection remains robust. However, extreme conditions like heavy fog or complete darkness would degrade performance. We're exploring infrared camera support for nighttime operation.

Q: Why physics-based instead of deep learning for prediction?

A: Two reasons: interpretability and data requirements. Deep learning trajectory predictors like Social-GAN need thousands of labeled trajectories. Our physics approach works immediately and

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: 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

[SPEAKER NOTES]

If Asked: "ByteTrack's key insight is that low-confidence detections often correspond to partially occluded objects. By matching these to existing tracks in a second stage, we maintain continuity through occlusions without expensive appearance feature extraction."

Backup: CPA Mathematical Derivation

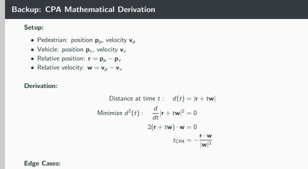
Setup:

- 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:



[SPEAKER NOTES]

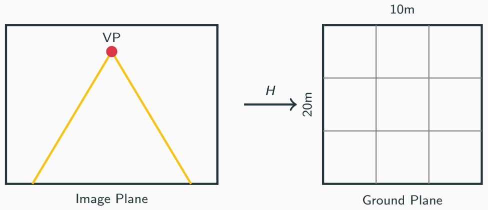
If Asked About Math: "The derivation is straightforward calculus. We want the time that minimizes distance, so we differentiate the squared distance and set to zero. The result is the dot product of relative position and velocity, divided by squared velocity magnitude.

Edge cases: if relative velocity is near zero, objects are moving in parallel and we just use current distance. If t_{CPA} is negative, the objects are moving apart and we don't flag a risk."

Backup: Homography-Based Ground Plane

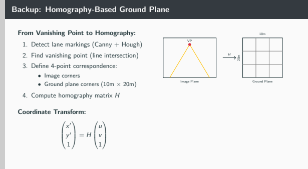
From Vanishing Point to Homography:

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:

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



[SPEAKER NOTES]

If Asked: "The homography maps image pixels to ground plane meters. We derive it from the vanishing point, which tells us camera orientation. Four corresponding points—image corners to ground plane corners—define the transformation.

This only works when lanes are visible. That's why we have the cascade fallback to horizon and size-based methods."