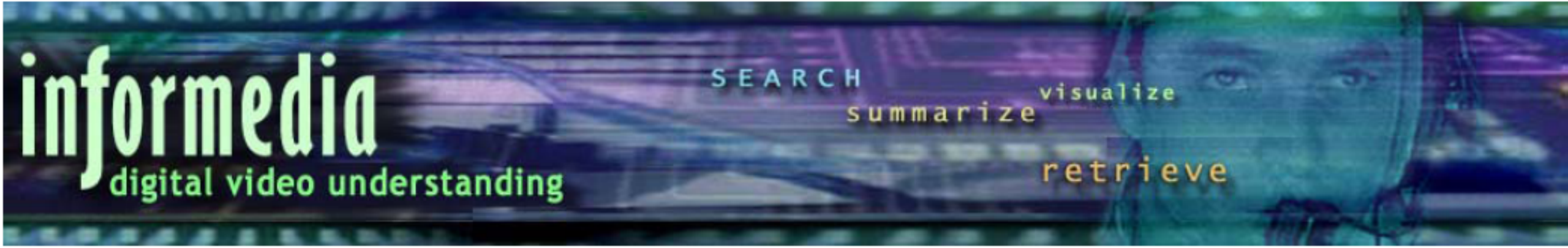



# Training-free Monocular 3D Event Detection System for Traffic Surveillance

Lijun Yu*, Peng Chen[1], Wenhe Liu, Guoliang Kang,
Alexander G. Hauptmann

Carnegie Mellon University
Peking University[1]

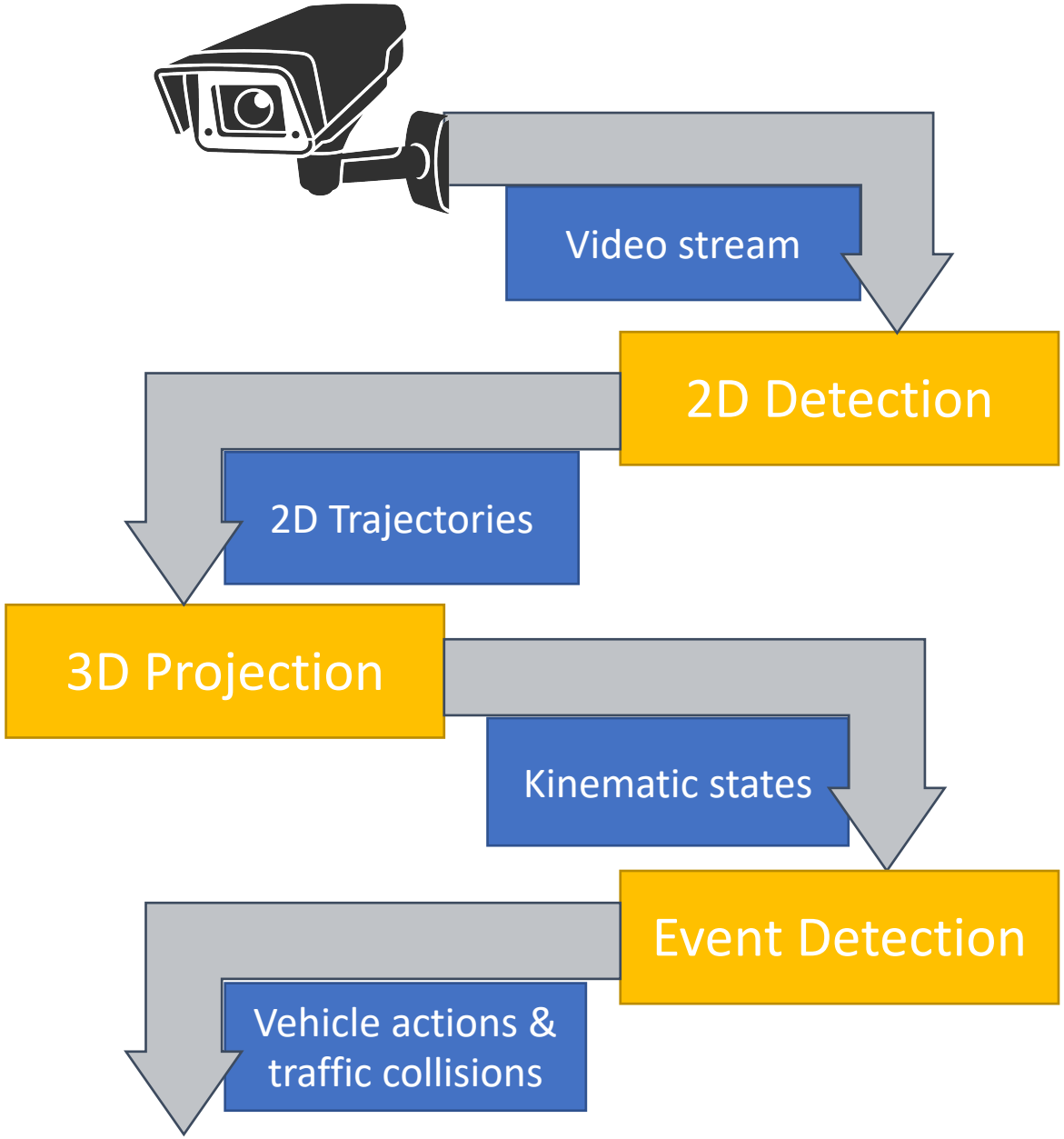
# Introduction

# Motivation

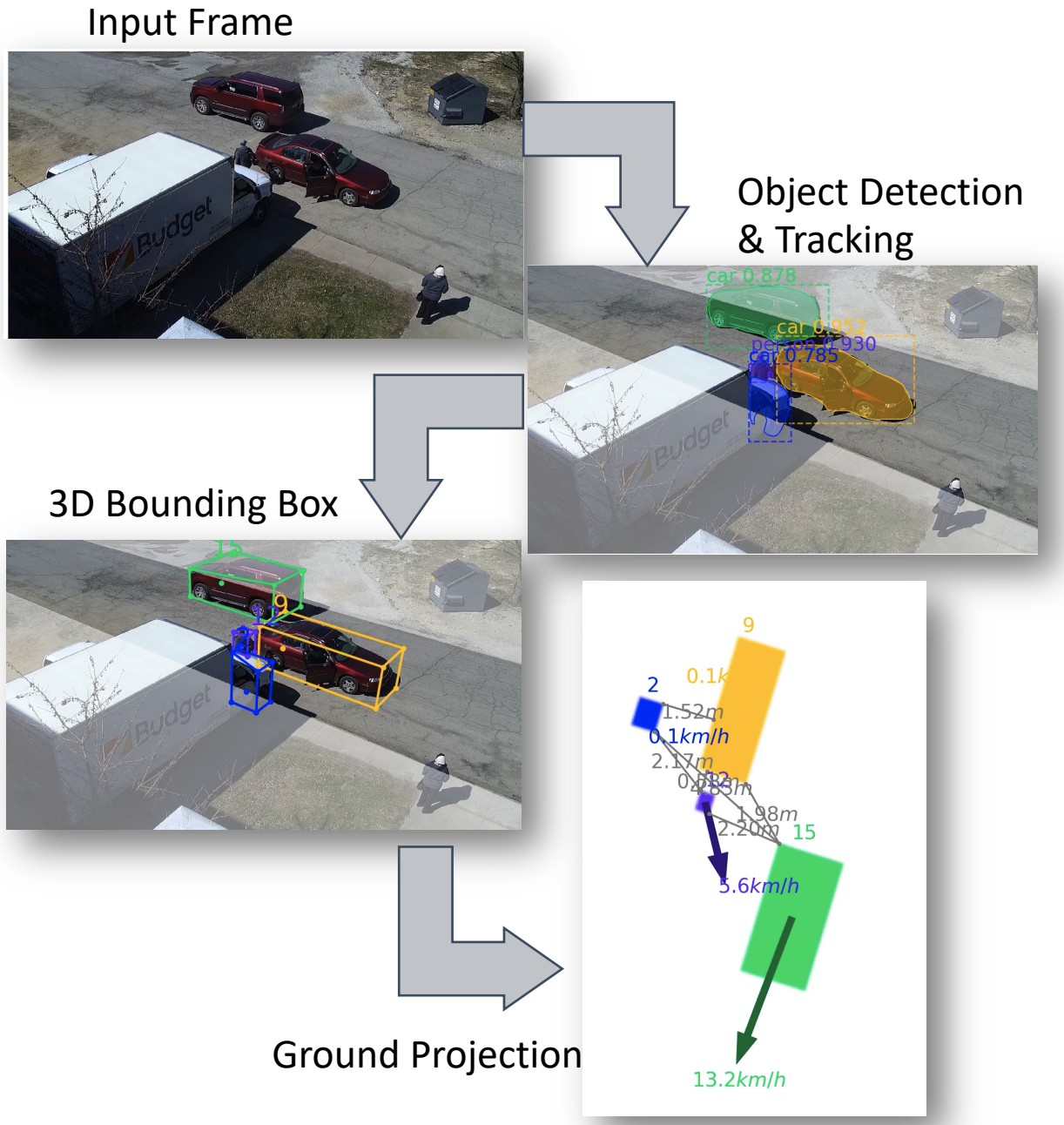
- Large-scale surveillance data to analysis
- Supervised learning systems rely on massive training data
  - Huge investment of time and labor in annotation
  - Challenging to collect rare events like traffic crashes
- Limitations of 2D video analysis
  - Occlusions and different viewing angles
  - Imprecise speed and location
- Expensive to setup 3D/RGB–D cameras

# System Overview

- First attempt of a training-free system for large-scale traffic event detection
- Monocular 3D approach robust to occlusions and camera viewing angles
- Real-time stream processing, outperforms training-based baseline on challenging real-world surveillance dataset

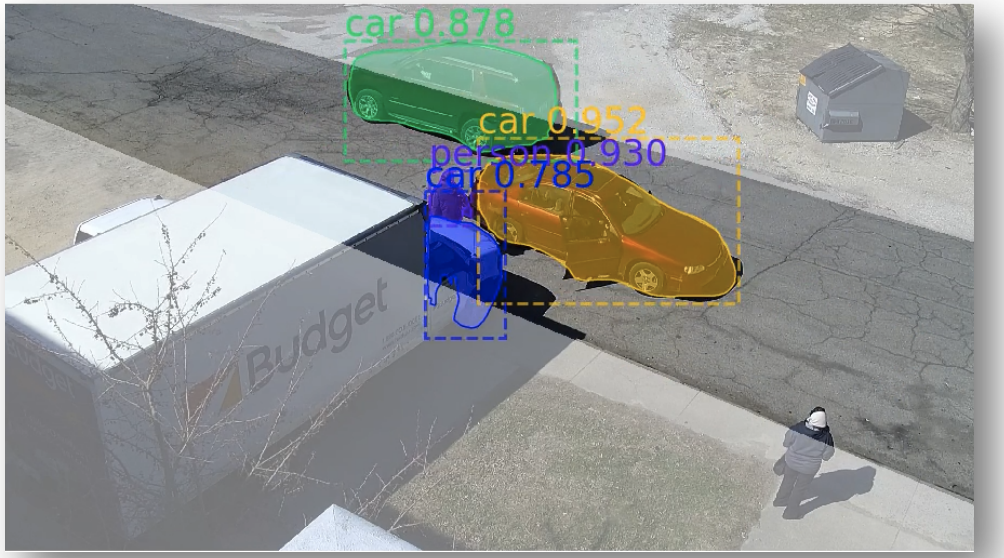


# Monocular 3D Surveillance

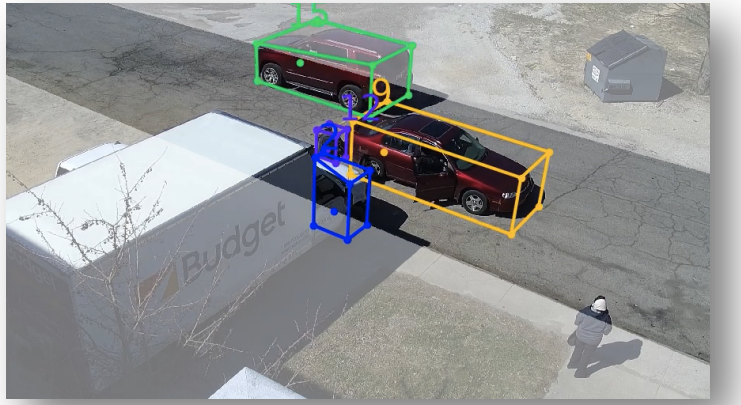


# 2D Detection

- Image Object Detection
  - Mask R-CNN
  - Trained on Microsoft COCO
  - Output: Object type, detection score, 2D bounding box, object mask, ROI feature
- Online Object Tracking
  - Deep SORT using ROI feature
  - Kalman filter with a constant velocity model
  - Output: Vehicle ID, 2D location, speed

# 3D Projection

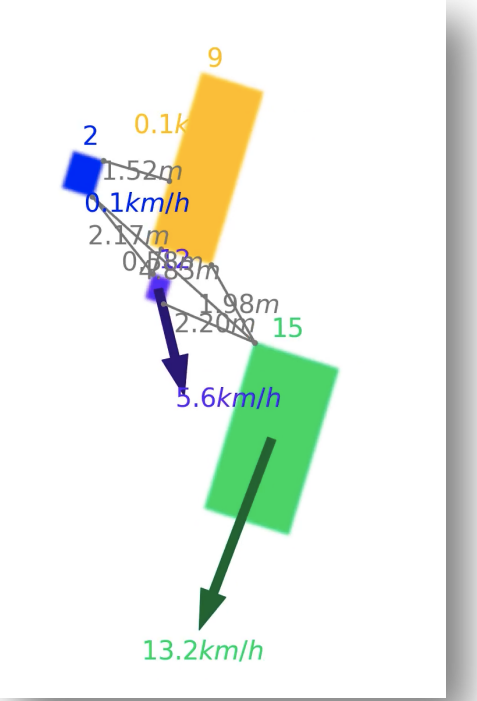


- Camera Calibration
  - Euclidean 3D point $\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$, 2D image point $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$

    Projection matrix $P$

    $$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

  - 1) Provided $K, R, t$
    2) Provided vanishing points
    3) Manual labeling parallel lines
- 3D Bounding Box
  - Contour from 2D mask
  - Speed vector by projecting the states of Kalman filter
  - Ground location by re-projecting the bottom



# Training-free Event Detection

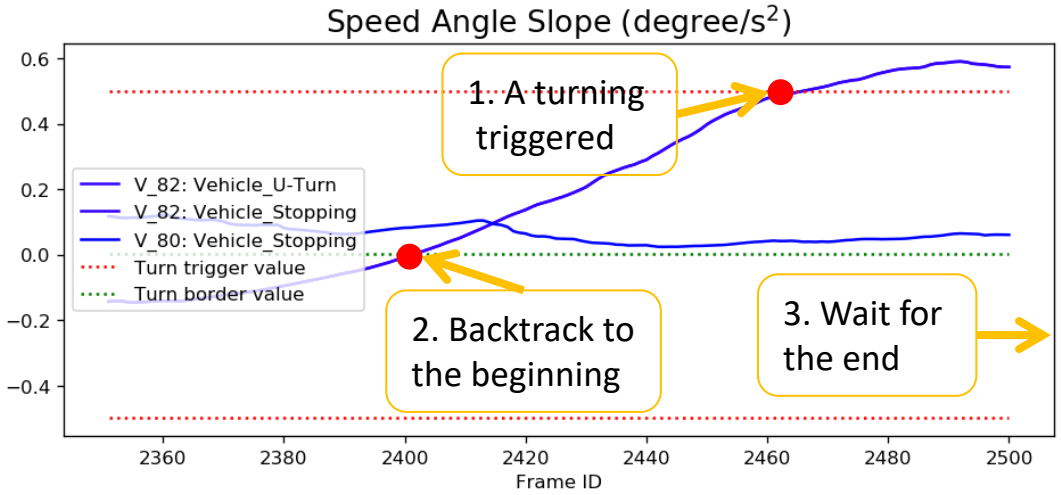
# Vehicle Action Detection

- Event types:
  - Turning events: vehicle turning left, vehicle turning right, vehicle U-turn
  - Linear events: vehicle starting, vehicle stopping
- State Estimation
  - Ground speed vector $\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$ in polar coordinates $\hat{\mathbf{v}} = \begin{pmatrix} v_r \\ v_\theta \end{pmatrix} = \begin{pmatrix} \sqrt{v_x^2 + v_y^2} \\ \operatorname{atan2}(v_y, v_x) \end{pmatrix}$
  - Acceleration $a_{r,t_0}, a_{\theta,t_0}$ estimated by local linear regression
    OLS in a sliding window

$$v_{r,t} = \beta_r + a_{r,t_0} t$$
$$v_{\theta,t} = \beta_\theta + a_{\theta,t_0} t$$

# Vehicle Action Detection

- Trigger-driven Detection Model
  - Training-free and explainable, parameters with physical meaning
  - Frame-level condition of trigger and border (looser), and further classification




*3) Turning Events:* The trigger and border conditions of turning events are

$$|a_\theta| \geq a_{\theta,trigger} \text{ and } v_r \geq v_{turn\_min} \quad (6)$$

$$|a_\theta| \geq a_{\theta,border} \text{ and } v_r \geq v_{turn\_min} \quad (7)$$

where $a_{\theta,trigger} > a_{\theta,border}$. For a turning event started at $t_s$ and ended at $t_e$, the turning angle can be calculated as

$$\theta = v_{\theta,t_e} - v_{\theta,t_s} \quad (8)$$

where $\theta \in (-180, 180]$. The valid condition of turning events is

$$t_e - t_s \geq t_{turn\_min} \text{ and } |\theta| > \theta_{min} \quad (9)$$

Then these events are further classified as

- Vehicle turning left, if $\theta_{min} < \theta < \theta_{max}$
- Vehicle turning right, if $-\theta_{max} < \theta < -\theta_{min}$
- Vehicle U-turn, if $\theta > \theta_{max}$ or $\theta < -\theta_{max}$

*4) Linear Events:* The trigger and border conditions of linear events are

$$|a_r| \geq a_{r,trigger} \quad (10)$$

$$|a_r| \geq a_{r,border} \quad (11)$$

where $a_{r,trigger} > a_{r,border}$. For a linear event started at $t_s$ and ended at $t_e$, the valid condition is

$$\begin{aligned} & t_e - t_s \geq t_{linear\_min} \\ \text{and } & \min(v_{r,t_s}, v_{r,t_e}) \leq v_{stop\_max} \\ \text{and } & \max(v_{r,t_s}, v_{r,t_e}) \geq v_{move\_min} \end{aligned} \quad (12)$$

Then these events are further classified as

- Vehicle starting, if $v_{r,t_s} \leq v_{stop\_max}$ and $v_{r,t_e} \geq v_{move\_min}$
- Vehicle stopping, if $v_{r,t_s} \geq v_{move\_min}$ and $v_{r,t_e} \leq v_{stop\_max}$
- Invalid event, otherwise

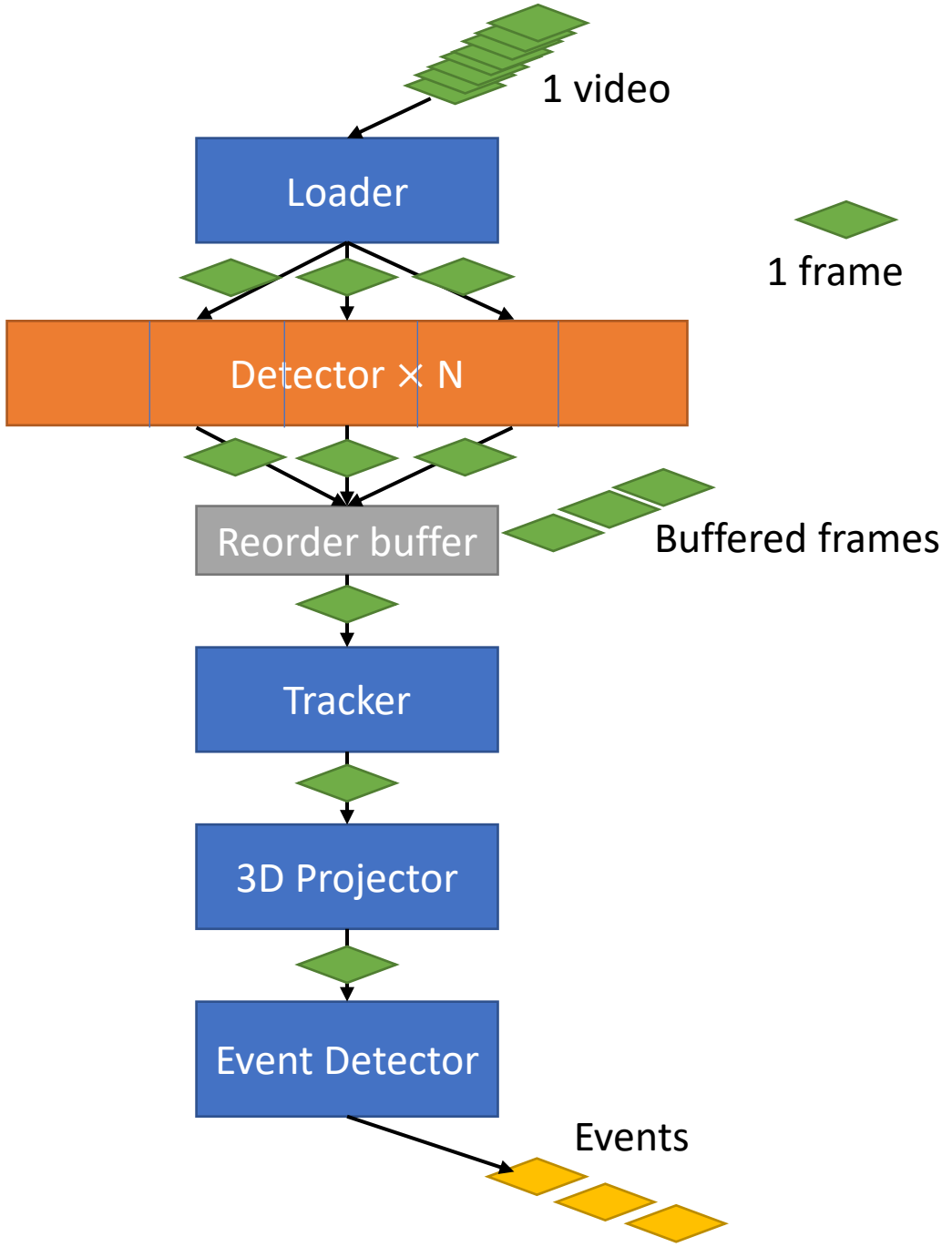
# Traffic Collision Detection

- Distance Measurement
  - Measure minimum distance between each pair of quadrangle vehicles on projected ground location
  - Alert for continuous decline
- Collision Detection
  - Location prediction for very near future
    - Fixed speed kinematic model
  - Check overlap of rigid bodies in predictions
    - Sign of future collision

# Experiments

# Implementation Details

- Realtime stream execution
- Bottleneck: object detection
  - Frame-level parallelism
  - Out-of-order execution and reorder
- System efficiency
  - Stream video at 1080p
  - 4 Nvidia RTX 2080Ti GPUs and 128GB memory
  - Throughput 18fps
  - Latency 0.2s



# Datasets and Baseline

- Multiview Extended Video with Activities (MEVA)
- 2000 videos × 5 minutes, 1080p, 30fps
- Annotated by our team, Train: 4870 minutes; Test: 500 minutes (160 minutes outdoor scenes on calibrated cameras)
- Baseline: training-based system using optical flow features and RNN classifier


- Car Accident Detection and Prediction (CADP)

# Detection and Tracking Result

- Shared backbone of detection and tracking
- Upper bound of event detection performance for both systems
- Event IoU: mean of object IoU over all frames, matched with Hungarian algorithm
- Recall at different IoU threshold

DETECTION AND TRACKING RESULT (RECALL)

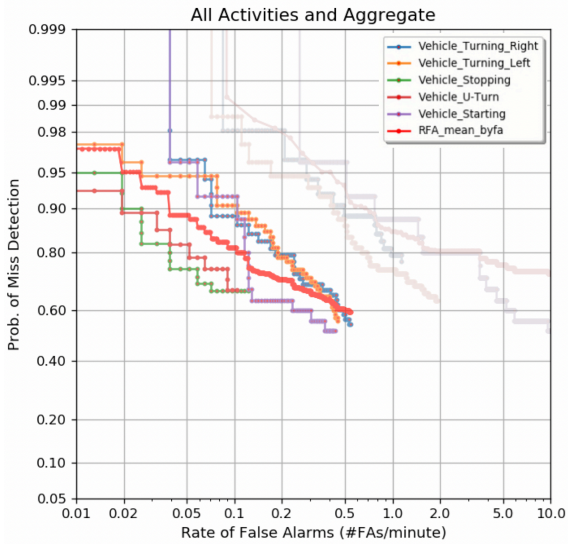
| Event IoU Threshold | 0 | 0.1 | 0.2 | 0.3 |
|---|---|---|---|---|
| Vehicle turning left | 0.95 | 0.78 | 0.59 | 0.62 |
| Vehicle turning right | 0.91 | 0.71 | 0.59 | 0.35 |
| Vehicle U-turn | 0.90 | 0.76 | 0.63 | 0.54 |
| Vehicle starting | 0.95 | 0.89 | 0.79 | 0.73 |
| Vehicle stopping | 0.90 | 0.79 | 0.74 | 0.62 |
| Average | 0.93 | 0.78 | 0.65 | 0.51 |

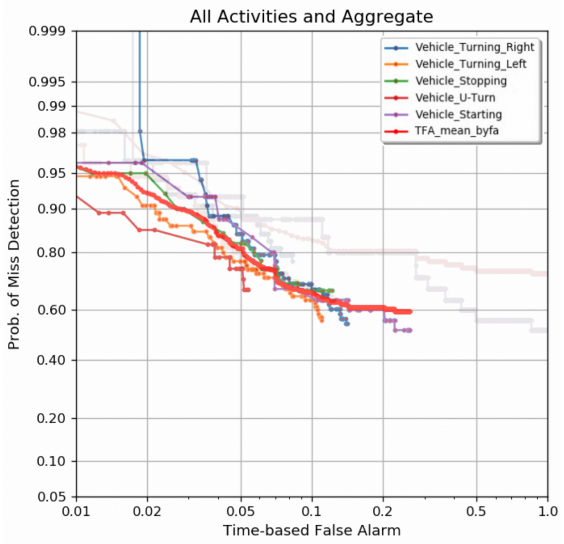
# Event Detection Metrics

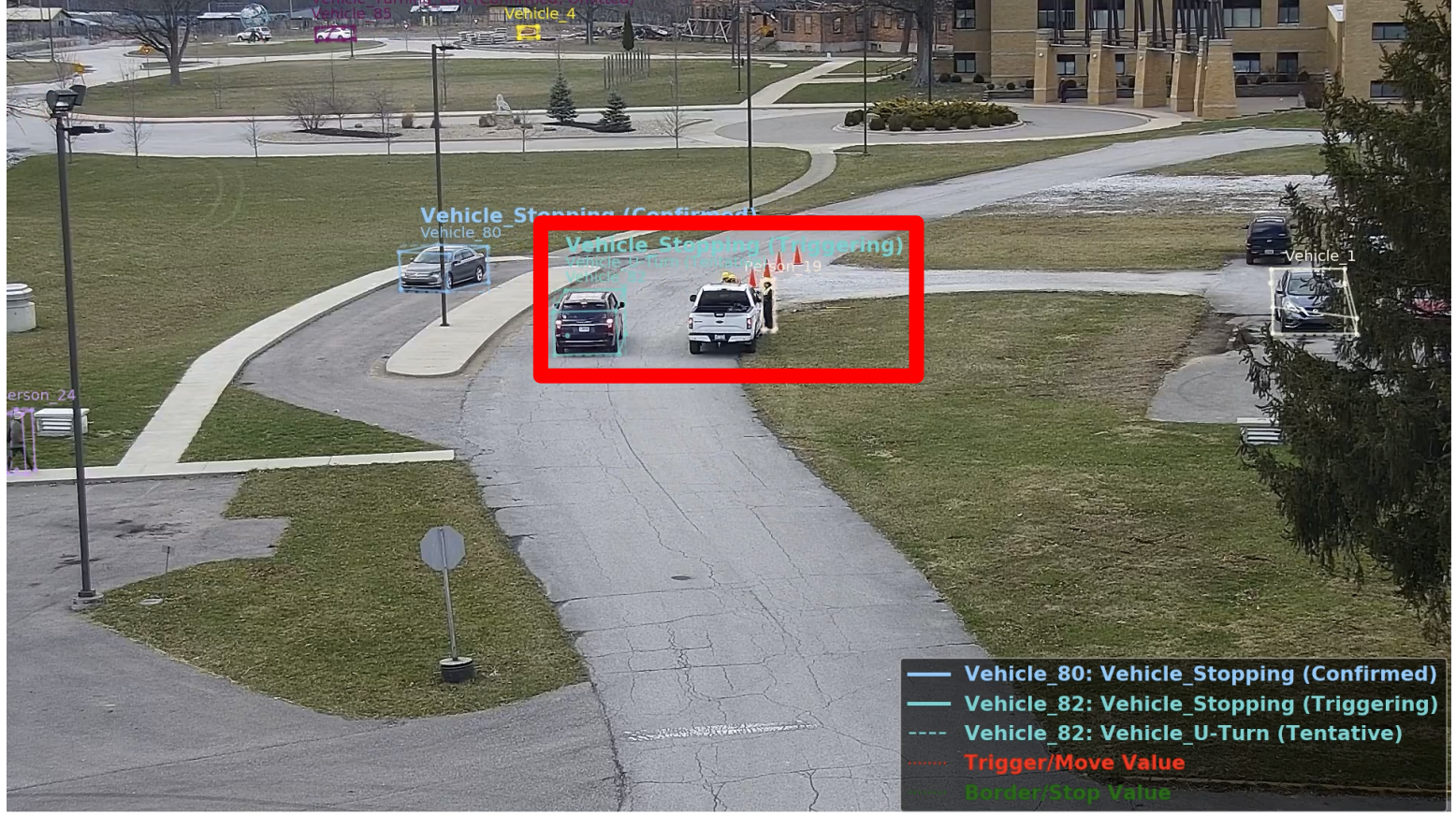
- Official ActEV Scorer (ActEV19_AD_V2) from TRECVID benchmark
- Solid lines: our training-free model
  Transparent lines: training-based model

GENERAL METRICS OVER ALL DETECTED EVENTS
(LOWER IS BETTER FOR ALL METRICS)

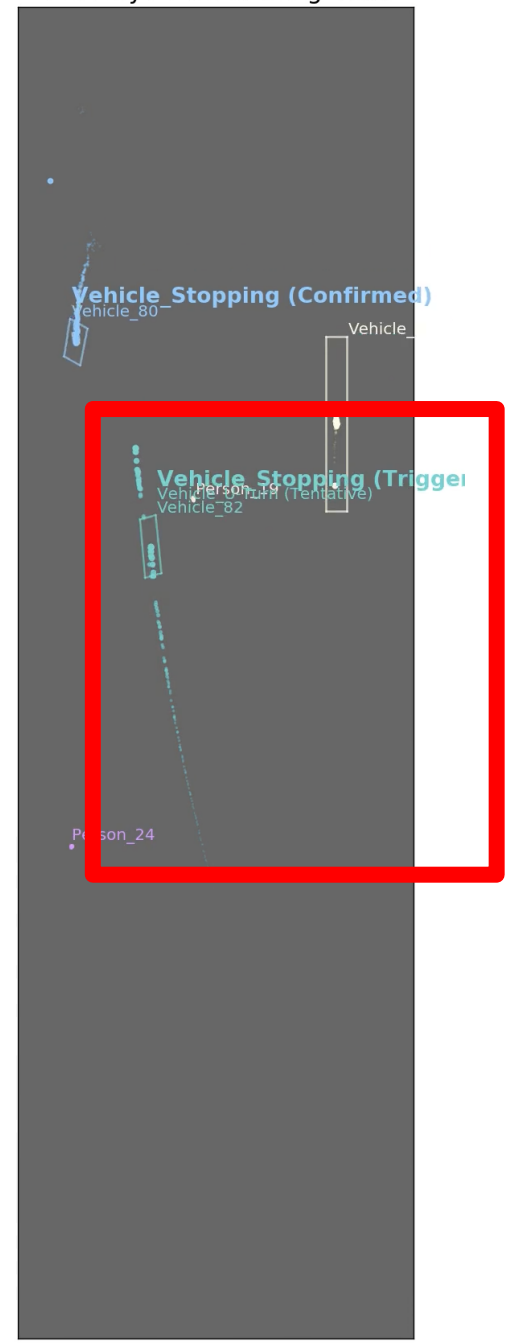
| | Training-free System | | | Training-based System | | |
|---|---|---|---|---|---|---|
| | $P_{miss}$ | $R_{fa}$ | $T_{fa}$ | $P_{miss}$ | $R_{fa}$ | $T_{fa}$ |
| Vehicle turning left | 0.36 | 0.68 | 0.24 | 0.64 | 1.92 | 0.12 |
| Vehicle turning right | 0.47 | 0.85 | 0.25 | 0.77 | 1.14 | 0.08 |
| Vehicle U-turn | 0.53 | 0.98 | 0.40 | 1.00 | 0.03 | 0.00 |
| Vehicle starting | 0.52 | 0.43 | 0.26 | 0.32 | 22.22 | 2.40 |
| Vehicle stopping | 0.68 | 0.12 | 0.12 | 1.00 | 0.00 | 0.00 |
| Mean | **0.60** | - | - | 0.88 | - | - |

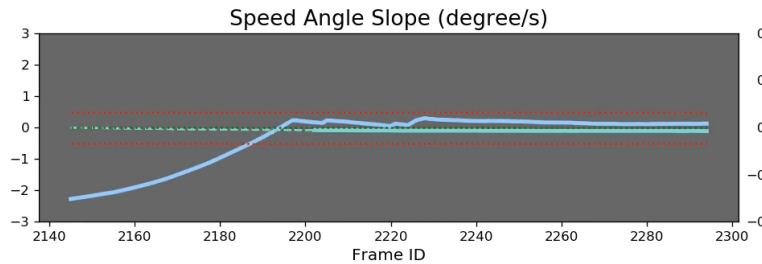


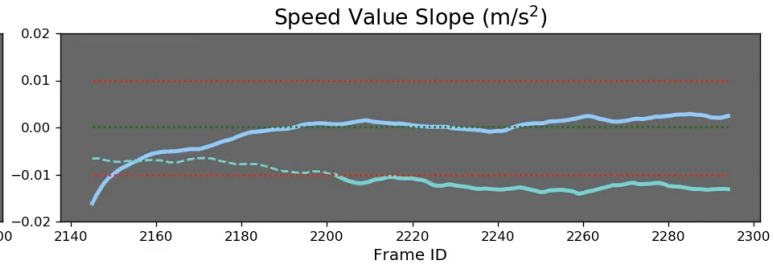



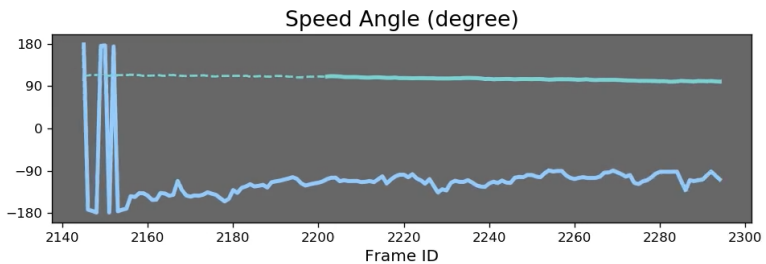


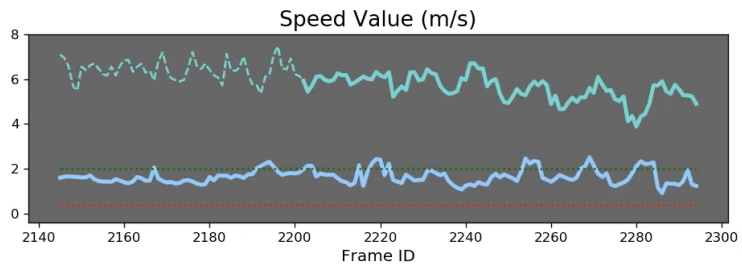

Bird's eye view of the ground



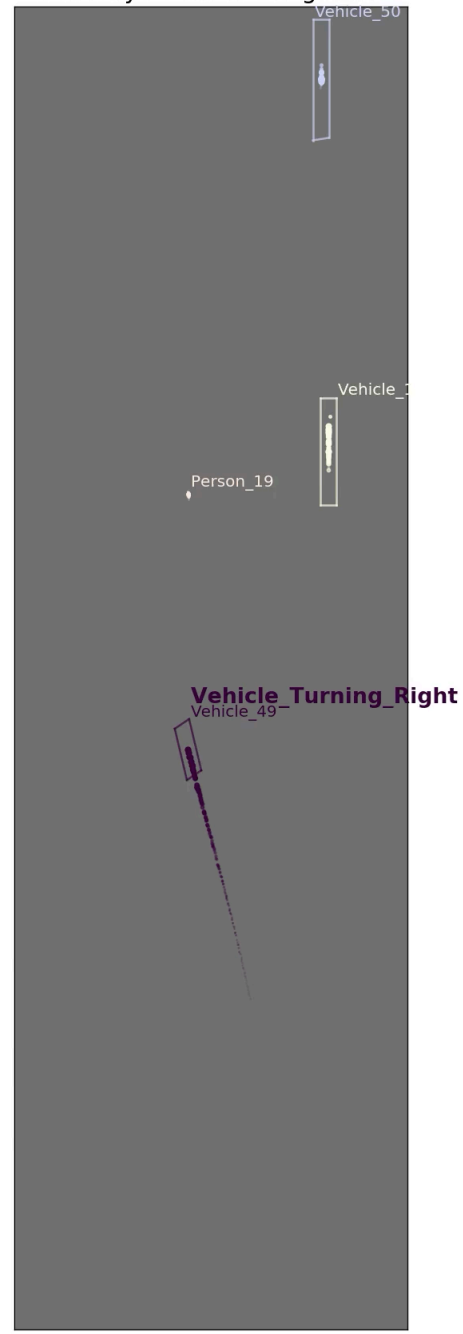



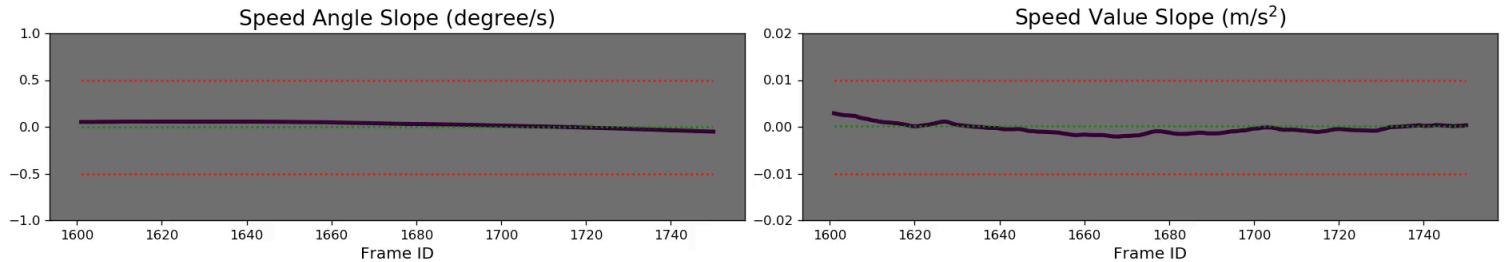



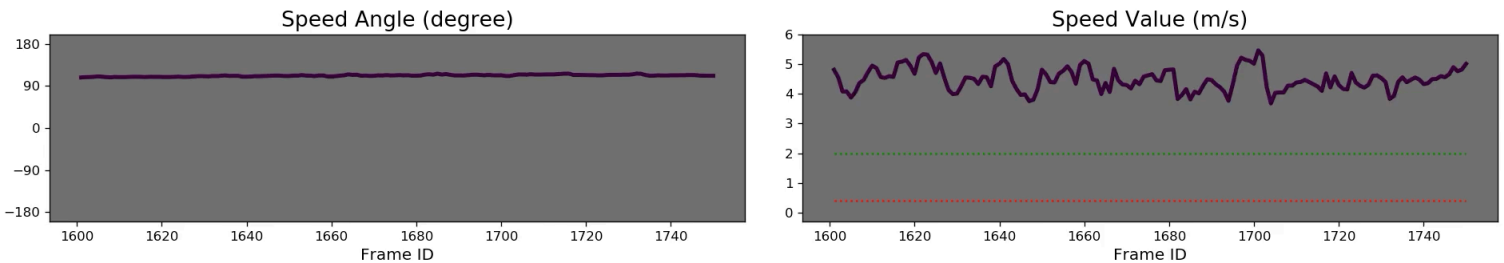

Bird's eye view of the ground



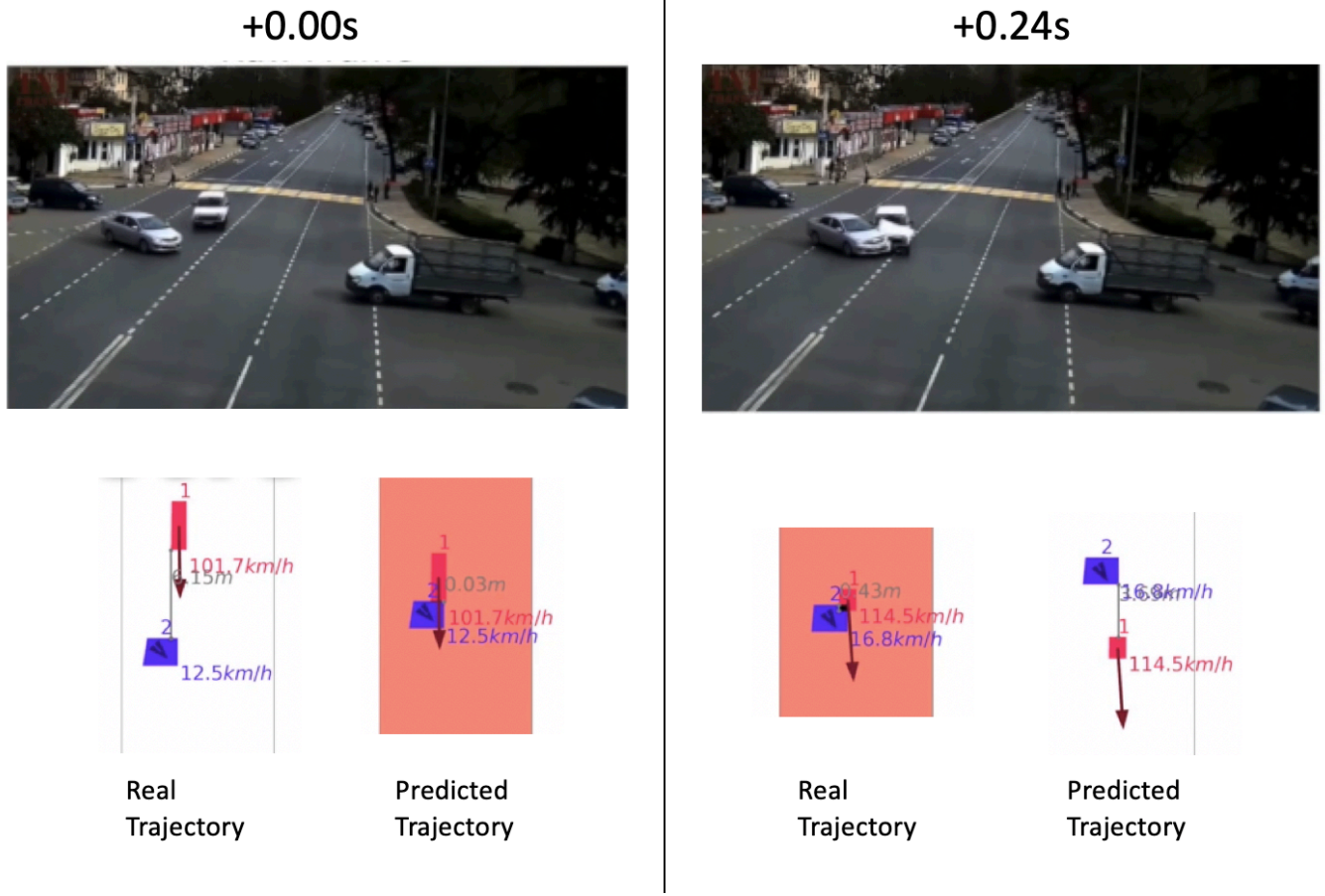




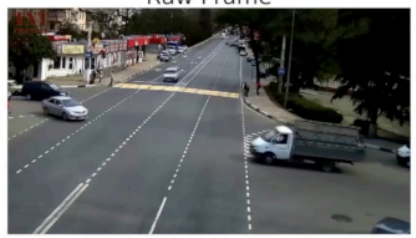


# Traffic Collision Detection

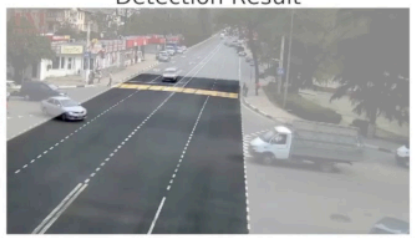
- Still ongoing due to lack of calibration
  - Manually annotated a few
  - Automatic calibration model



Raw Frame



Detection Result



3D Bounding Boxes

Real Time Plane Map

Plane Map for +0.12s

Plane Map for +0.24s

# Conclusion

- Real-time traffic event detection system for traffic surveillance safety
- First attempt of a training-free system for large-scale traffic event detection
- Monocular 3D method to overcome issues of occlusions and camera viewing angles
- Real-time processing for large-scale traffic data
- Significantly outperforms the existing training-based system on real-world surveillance dataset

# Thanks for your attention!