AI VIDEO ANALYTICS FOR MICROMOBILITY
SAFETY ANALYSIS AT TRAIL CROSSINGS

STUDENTS: ANDY CAO, BRIAN KO, MILES CHIN, STEVEN MIAO, WESLEY HUANG

AIWAYSION

Motivation

Why this matters

Micromobility traffic at trail crossings is rising fast, but manual review of safety incidents is slow and inconsistent.
Teams need a repeatable pipeline that turns raw video into trajectories, speeds, and near-miss analyses automatically.

Project Goal

Automate detection, tracking, trajectory prediction, and safety outputs (TTC / PET) from raw trail-crossing video.
Differentiate vulnerable road users —scooters, wheelchairs, pedestrians — for per mode safety analysis.

SYSTEM OVERVIEW

1. DETECTION — *YOLO26*: Find every object in every frame

2. TRACKING — *BoT-SORT*: Link detections into stable IDs over time

3. CALIBRATION — *Homography*: Project 2D pixels to ground plane

4. CLASSIFY — *Per-mode*: Cars · peds · e-bikes scooters · wheelchairs

5. PREDICTION — *Transformer*: Forecast short-horizon motion (DETR-style)

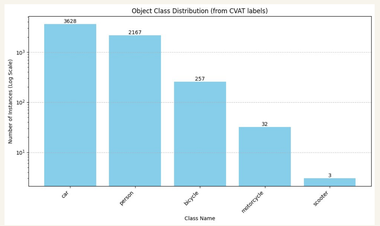
6. NEAR-MISS — *TTC / PET*: Flag risky encounters

Output: per-mode safety insights — trajectories, speeds, near-miss counts for every class of road user.

Classes tracked: cars · trucks · pedestrians · bicycles · e-bikes · scooters · wheelchairs

DATASET & PSEUDO-LABELS

California



Ballard

Easy set (ID 124441) · Hard set (ID 084511) — two 3-min trail-crossing videos

Classes

Car · truck · person · bicycle + fine-grained micromobility: e-bike, scooter, wheelchair

Pseudo-Label Workflow

Detect + Track: YOLO26 + BoT-SORT produces stable IDs across frames.
Alternative path: SAM 3 with noun-phrase prompts as a generalist baseline.
Human-in-the-loop: CVAT review where annotators fix misses, errors, and class labels.
Output: a high-confidence labeled set that drives every downstream experiment.

EXP 1: TRACKING COMPARISON

Key result: SAM 3 preserves identities best on Easy; YOLO26 + BoT-SORT is more stable on Hard.

Easy set

Model	HOTA ↑	mAP ↑	Gaps ↓	Track len ↑
YOLO26x + BoT	0.250	0.375	45	41.0
SAM3 (all)	0.305	0.225	16	79.4
SAM3 (separate)	**0.324**	**0.245**	**14**	**87.2**

Hard set

Model	HOTA ↑	mAP ↑	Gaps ↓	Track len ↑
YOLO26x + BoT	**0.304**	**0.199**	**84**	**54.9**
SAM3 (all)	0.251	0.131	129	64.2
SAM3 (separate)	0.268	0.141	123	68.4

YOLO26 + BoT-SORT on Hard set

SAM 3 (separate) on Hard set

Takeaways

SAM 3 (separate prompting) gives the best HOTA, fewest gaps, and longest tracks on the Easy set.
YOLO26 + BoT-SORT leads on both HOTA and mAP when scenes are crowded and ambiguous (Hard set).

EXP 2: FINE-TUNING YOLO26X

Setup: yolo26x.pt baseline · 100 epochs · 250 frames/class · Colab L4 / H100

Fine-tune on Easy set

Model	mAP50	Precision	Recall
Pretrained	0.6558	0.8314	0.8314
Fine-tuned	**0.9709 (↑48%)**	**0.9742 (↑17%)**	**0.9329 (↑12%)**

Fine-tune on Hard set

Model	mAP50	Precision	Recall
Pretrained	0.3677	0.4629	0.3503
Fine-tuned	**0.8933 (↑53%)**	**0.8923 (↑43%)**	**0.8418 (↑49%)**

Fine-tune on California set

Model	mAP50	Precision	Recall
Pretrained_YOLO	0.6558	0.8314	0.8314
Fine-tuned_YOLO	**0.9709 (↑48%)**	**0.9742 (↑17%)**	**0.8418 (↑49%)**

Result

Fine-tuning lifts mAP50 by ~50% on both sets — biggest gains on the harder, more occluded crossing scenes.
Pure YOLO fine-tuning remained the strongest single model; adding Dino features helped precision but hurt recall.

EXP 3: TRAJECTORY PREDICTION

Homography helps Linear, but Kalman degrades under bad conditioning — so we tried a transformer next.

Baseline trajectory error (ADE / FDE — lower is better)

Method	Coord	Easy ADE	Easy FDE	Hard ADE	Hard FDE
Linear	Direct	9.50	20.91	24.68	67.36
Kalman	Direct	9.44	20.11	23.66	68.37
Linear	Homography	9.05	19.07	22.18	53.90
Kalman	Homography	10.05	18.54	39.66	70.50
Transformer	**Homography**	**8.21**	**18.13**	**20.42**	**47.18**

How our transformer predicts the next 1–2 seconds of motion

1. WHERE & HOW FAST — Past 1s of motion: position · velocity speed · heading

2. UNDERSTAND — Transformer encoder reads the history and finds patterns

3. IMAGINE FUTURE — Decoder queries ask "where next?" at each future step

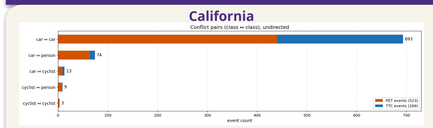
4. PREDICTED PATH — Future positions for 1–2 seconds ahead — all at once

Why it works — two key tricks

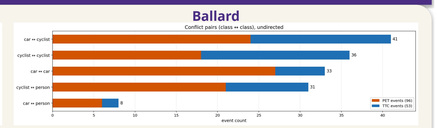
Predict the curve, not the line. The model only learns how a trajectory bends away from a straight-line guess — so it never wastes capacity on the easy case.
Train on synthetic turns. Real trail data is mostly straight; we add rotated, flipped, and synthetically curved tracks so the model actually sees enough turns to learn from.

NEAR-MISS EXAMPLES & CONCLUSION

California



Ballard



Safety signals

Time to Collision (TTC) — if both users keep their current paths, how many seconds until they meet? Small TTC = imminent conflict.
Post-Encroachment Time (PET) — one user crosses a spot, the other arrives later — how big is the gap? Small PET = close call.

TTC example — two cyclists, TTC = 0.13 s

PET example

Conclusion

Fine-tuned YOLO26 + BoT-SORT is the most stable detector-tracker stack for crowded trail scenes.
Homography calibration improves linear prediction; transformer learns curvature on a kinematic baseline.
Pipeline outputs trajectories, speeds, TTC and PET — convertible into per-mode safety insights.
Cross-site validation on California and Ballard confirms generalization across varied trail geometries and user mixes.
Per-site conflict patterns surface actionable risk signals — car ↔ car dominates California, while cyclist interactions lead in Ballard.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: HUNG-MIN HSU (INDUSTRY), JAI JAISIMHA (FACULTY)

SPONSOR: AIWAYSION