Human Motion Analysis Project

Computer Vision 2025

Prof. Nicola Conci

University of Trento

Seyed Morteza Mojtabavi

January 2, 2026

Abstract

Aerial surveillance in shared spaces presents a distinct challenge: targets are often small, densely packed, and frequently occluded, making traditional appearance-based tracking *unreliable*. To address this, I adopted the **Tracking-by-Detection (TBD)** paradigm, selecting **YOLOv8** for its balance of speed and accuracy, and pairing it with a motion-centric association strategy inspired by **ByteTrack** [4]. This approach was chosen to mitigate the limitations of visual re-identification in low-resolution footage by associating low-confidence detections typically discarded by standard algorithms.

I optimized the training pipeline by scaling inputs to **960px** and utilizing the **AdamW** optimizer to maximize sensitivity to small targets. To rigorously assess performance, I employed the **HOTA** metric, which balances detection precision and association stability. Evaluation on the **Stanford Drone Dataset** demonstrated that this high-resolution, motion-based approach *significantly* reduces track fragmentation in complex, high-density scenes. Furthermore, a custom **Grid-Based Path Analysis** validated the system's *semantic reliability*, confirming that inferred traffic flows align closely with Ground Truth patterns despite the chaotic environment.

1. Introduction

Unmanned Aerial Vehicles (UAVs) provide a unique vantage point for analyzing crowd dynamics, simultaneously capturing diverse agents like pedestrians, bikers, and cars [1]. However, this perspective introduces significant challenges, primarily *drastic scale variation*. Targets often appear as **"small objects"** (*<32 pixels*), which standard detectors struggle to identify due to weak feature representation and loss during downsampling [2, 3].

This project analyzes the "little" subset of the **Stanford Drone Dataset** (**Video 0** and **Video 3**). These dense shared spaces present a rigorous test for computer vision: *unstructured movement*, *diverse agents*, and *chaotic occlusion patterns*. To address these challenges, this report presents the following contributions:

- **High-Resolution Training:** Implemented a YOLOv8s model at 960px to prevent feature loss for small objects.
- **Complexity-Based Data Curation:** Developed a scoring algorithm to prioritize dense frames and rare classes (e.g., Skaters, Cars), mitigating dataset imbalance.
- **ByteTrack Association:** Integrated a motion-centric tracker that associates low-confidence detections, recovering objects when the detector yields weak or ambiguous predictions [4].
- **Semantic Path Analysis:** Built a grid-based tool to map raw trajectories into semantic traffic flows, revealing dominant movement patterns.

2. Related Works

The landscape of **Multi-Object Tracking (MOT)** has recently seen a push toward End-to-End Transformers like **MOTR** [17], which unify detection and association. However, these models are notoriously *data-hungry* and *computationally heavy* for edge deployment. Consequently, the **Tracking-by-Detection (TBD)** framework remains the standard in the literature [6]. TBD's *modularity* is critical for aerial surveillance, allowing us to plug in specialized high-resolution detectors to tackle the "*small object*" problem directly without overcomplicating the tracking logic.

Object Detection Within TBD, performance hinges on the detector. Historically, two-stage models like **Faster R-CNN** [10] and **Mask R-CNN** [11] offered high precision but proved too slow for real-time aerial tasks. The **YOLO family [14]** revolutionized this by enabling *single-pass prediction*. Recent methods, such as **YOLOv8** [15] and the UAV-optimized **RLRD-YOLO** [2], have refined this approach with multi-scale feature fusion designed to prevent small aerial targets from vanishing during downsampling. This makes them a more practical choice for drone surveillance than heavy Transformer-based detectors like **DETR** [12] or **DINO** [13].

Data Association Strategies Once detections are secured, the challenge shifts to association. Early heuristics like **SORT** failed when detectors missed frames, while **DeepSORT** attempted to fix this with appearance embeddings. However, as noted in recent surveys [5], visual features are *unreliable* for small, texture-less drone targets. To address this, I adopted the *motion-centric* **ByteTrack [4]**. Unlike predecessors that discard low-confidence detections, ByteTrack utilizes these weak signals to recover objects during partial occlusions. This strategy reduces fragmentation without the massive overhead of graph-based methods like **MPNTrack** [16].

Evaluation Metrics: Benchmarking these systems requires nuanced metrics. The traditional **MOTA** is often criticized for over-emphasizing detection precision while neglecting association stability [8]. While **IDF1** [9] improves on this by focusing on identity consistency, the modern gold standard is **HOTA (Higher Order Tracking Accuracy) [7]**. By *balancing* detection and association accuracy, HOTA provides the fairest assessment of a tracker's stability in dynamic environments.

3. Methodology

The pipeline consists of four integrated modules: **Dataset Management**, **Model Training**, **Object Tracking**, and **Analysis**.

3.1 Dataset Management and Curation Raw SDD annotations were parsed into YOLO format using a *heuristic selection strategy* rather than random sampling. To mitigate class imbalance, I implemented a "Complexity Score" that assigned a **3 x weight** to rare classes (e.g., Skater, Car) versus **1 x** for common agents. Additionally, I enforced a **5-frame gap** to reduce redundancy and scaled inputs to **960px**, significantly larger than the standard 640px, to ensure small features remained detectable.

3.2 Model Training (YOLOv8s) I benchmarked a standard "Old Model" (640px, SGD, 40 epochs) against the optimized "New Model." The Old Model's critical failure stemmed from its *insufficient* *sampling strategy;* it relied on a short **3-frame** gap and lacked any mechanism to prioritize scene density. This approach inadvertently *flooded* the training set with redundant, easy samples from *Video 0*, creating a severe **bias**. Consequently, the model failed to learn from *Video 3,* which is significantly longer, denser, and more complex, simply because the naive sampling didn't account for its chaotic nature.

The "New Model" introduced six critical changes:

1. **Resolution (960px):** Upscaled by **50%** to prevent small object features from *vanishing*.
2. **Data Curation:** Replaced simple filters with a "*Complexity Score*" to prioritize dense *Video 3* frames and increased temporal gaps to reduce redundancy.
3. **Class Coverage:** Expanded detection to **all 6 classes** (including rare agents like Skaters/Buses).
4. **Optimizer (AdamW):** Replaced SGD for better convergence on complex data.
5. **Scheduler:** Implemented **Cosine Annealing** (*cos_lr=True*) for precise weight refinement.
6. **Duration (150 Epochs):** Extended training from 40 to 150 epochs (approx. **2 days** on **CPU**) to ensure convergence on difficult geometries.

Workflow Optimization: To mitigate CPU latency, I implemented a utility script (*cache_detections.py*) that caches raw detections to Parquet files. This decoupled architecture enabled rapid, *offline* fine-tuning of tracking parameters without re-running the computationally expensive inference step.

3.3 Tracking Engine The tracking logic (*tracker_utils.py*) extends ByteTrack with three aerial-specific adaptations:

- **Kalman Filter:** I utilized an 8-dimensional Constant Velocity Model tracking *width/height* directly (rather than aspect ratio) to better handle rapid vertical scale changes.
- **Two-Stage Association:** High-confidence detections (**0.68**) are matched via IoU and strictly gated by **Mahalanobis distance** (rejecting jumps **>70 px**). Low-confidence "*rescue*" detections (**0.45–0.68**) undergo a **Size Consistency** check (rejecting area changes **>2.0x** or **<0.3x**) to prevent noise integration.
- **Stability:** To fix label flickering between lookalike classes (Pedestrian/Biker), I implemented **Class Locking**, which freezes the ID to the majority label once confidence exceeds **0.80** for **10** frames. Additionally, lost tracks are projected up to **100** frames to handle long-term tree occlusions.

3.4 Trajectory Path Analysis To quantify crowd dynamics, I developed a pipeline that partitions the frame into a **3×3 semantic grid**. Raw coordinates are compressed into discrete "*Regional Steps*" (e.g., *Top-Left to Center*), filtering noise to retain significant transitions. Finally, I constructed an **Origin-Destination Matrix**, filtering out static tracks to isolate and quantify the dominant movement patterns in the shared space.

4. Results

4.1 Quantitative Evaluation on 300 *diversified frames* (Video 0 & 3, *stride 50*) confirms the robustness of the high-resolution approach. As shown in **Table 1**, the **New Model** (960px, AdamW) *significantly outperforms* the baseline. While the **Old Model** *degrades rapidly* (max F1 approx **0.61**), the New Model maintains high stability with an F1-score above **0.80**, even at strict confidence thresholds (**0.70**). This proves that upscaling enables the detector to separate small targets from background noise with much higher certainty.

Conf Threshold	Model	Precision	Recall	F1-Score	Accuracy
0.25	Old Model	0.6271	0.5852	0.6054	0.4341
	New Model	**0.7781**	**0.8296**	**0.8030**	**0.6709**
0.55	Old Model	0.6890	0.5485	0.6107	0.4396
	New Model	**0.8277**	**0.8059**	**0.8166**	**0.6901**
0.70	Old Model	0.7273	0.5022	0.5941	0.4226
	New Model	**0.8560**	**0.7703**	**0.8109**	**0.6819**
0.85	Old Model	0.8298	0.1707	0.2831	0.1649
	New Model	**0.9214**	**0.3830**	**0.5411**	**0.3709**

Table 1: **Performance Metrics across Confidence Thresholds.** Evaluation set: 300 frames (Video 0 & 3), Stride: 50 frames, IoU Threshold: 0.5.

4.2 Tracking Performance (HOTA) To validate the pipeline, I evaluated the tracker using HOTA, splitting the analysis by scene to highlight adaptability (**Table 2**). On the simple **Video 0**, the Old Model performed slightly better (HOTA **0.78** vs **0.73**), likely due to *overfitting* on this predictable subset, where it could memorize specific background conditions.

Scene	Model	HOTA	DetA	AssA
Video 0 (Easy)	**Old Model**	**0.7806**	**0.8592**	**0.7093**
	New Model	0.7323	0.8069	0.6647
Video 3 (Complex)	Old Model	0.3714	0.4981	0.2861
	New Model	**0.5618**	**0.6987**	**0.4539**

Table 2: **Tracking Metrics by Scene Complexity.** Comparison of HOTA, Detection Accuracy (DetA), and Association Accuracy (AssA).

However, on the chaotic **Video 3**, the Old Model *collapsed* (HOTA **0.37**). In contrast, the New Model demonstrated superior *generalization*, achieving a **51% improvement** (HOTA **0.56**). This gain was driven largely by *Association Accuracy* (**AssA**), which rose from **0.2861** to **0.4539**, proving the system effectively maintains identities through heavy occlusion.

Figure 1 illustrates this stability visually. While both models perform well on the simple scene (Dashed Lines), the Old Model (Orange Solid) *degrades rapidly* on the complex scene as the IoU threshold increases. The New Model (Blue Solid) maintains a *consistent performance gap*, confirming its robustness in dense, realistic environments.

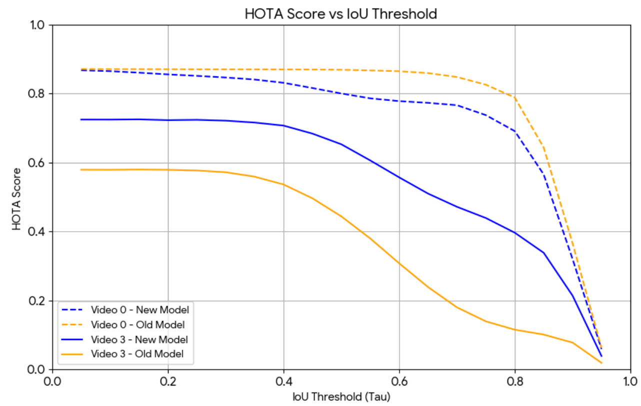

Figure 1: HOTA Score vs. IoU Threshold. The **X-axis** represents localization strictness (IoU Threshold), and the **Y-axis** tracks HOTA performance. **Dashed lines** indicate the simple Video 0, while **Solid lines** indicate the complex Video 3. **Blue** denotes the New Model, and **Orange** denotes the Old Model.

4.3 Path Frequency Analysis To validate *semantic reliability*, I compared the traffic patterns generated by the New Model against Ground Truth (GT) in **Table 3**. I defined "*paths*" by mapping raw coordinates to a **3x3 semantic grid**, compressing noisy trajectories into discrete regional transitions (e.g., *Middle-Left to Middle-Right*).

The results confirm that the system captures the correct crowd behaviors: the top identified paths are **identical** between the Model and GT in both scenes. In **Video 0**, alignment is *near-perfect*; the primary linear route was detected with **100% accuracy** (**6 occurrences**), proving precision in low density.

Scene	Rank	Ground Truth (GT)	New Model (Predicted)	Deviation
Video 0 *(Low Density)*				
(Top 3 Match)	1	Middle-Left → Middle-Right (6)	Middle-Left → Middle-Right (6)	**0**
	2	Middle-Left → Bottom-Middle (6)	Middle-Left → Bottom-Middle (6)	**0**
	3	Bottom-Middle → Middle-Right (5)	Bottom-Middle → Middle-Right (6)	**+1**
Video 3 *(High Density)*				
(Top 5 Match)	1	Middle-Right → Middle-Left (39)	Middle-Left → Middle-Right (38)	**+3**
	2	Middle-Left → Middle-Right (35)	Middle-Right → Middle-Left (37)	**-2**
	3	Bottom-Middle → Middle-Left (26)	Bottom-Middle → Middle-Left (23)	**-3**
	4	Bottom-Middle → Middle-Right (18)	Bottom-Middle → Top-Middle (23)	**+5**
	5	Bottom-Middle → Top-Middle (18)	Bottom-Middle → Middle-Right (18)	**0**

Table 3: **Top Frequent Paths** Comparison of the highest-traffic routes identified by the New Model vs. Ground Truth (GT).

Crucially, the chaotic **Video 3** confirms *robustness*. Despite the density, the **Top 5 paths** identified by the model are identical to the GT. While minor count variations exist due to occlusion, the system correctly prioritizes massive cross-traffic; for instance, the volume error for the top two flows is just **1.3%** (**75 predicted vs. 74 actual**). This proves the tracker provides a highly accurate macroscopic view of crowd dynamics.

5. Discussion

5.1 Impact of Resolution on Stability The link between resolution and stability was clear. While the Old Model frequently "*broke*" tracks as subjects moved, the New Model's high recall (**83%**) provided **continuous measurements** to the Kalman Filter. This minimized prediction *drift*, effectively preserving the original ID throughout the trajectory.

5.2 The "Ground Truth *Paradox*" Evaluation was complicated by significant noise in **the Ground Truth (GT)**, specifically **sparse training data** for rare classes and frequent annotation errors (missing or mislabeled objects). This created a paradox: qualitative checks showed the New Model often detected valid targets (e.g., pedestrians in deep shadows) that were unannotated in the GT. Standard metrics penalized these correct detections as "**False Positives**", meaning the reported numbers likely *underestimate* the model's true real-world performance.

5.3 Limitations of Tracking Logic Despite robust detection, the tracking logic relies on simplified assumptions. First, it uses a uniform **Constant Velocity Model**, failing to differentiate between *erratic* pedestrians and fast, *smooth* bikers. Second, it lacks "**Social Physics**", ignoring the human instinct for collision avoidance. Finally, the system allows ***spontaneous*** **track initialization** anywhere; stricter logic requiring objects to enter from borders or occlusions would better reflect physical reality and reduce false positives.

6. Conclusion

This project demonstrated that solving the "small object" and "occlusion" dilemmas in drone surveillance requires more than just scaling up the input; it demands a cohesive integration of **resolution, tracker design, and rigorous data evaluation**.

Key Findings:

- **Synergy of Resolution and Architecture:** While increasing input to **960px** prevented "*vanishing targets*" (driving a **24% Recall boost**), the **Tracking-by-Detection** paradigm ensured stability. By pairing the detector with motion-centric association (**ByteTrack**), the system *bridged occlusions* where appearance-based trackers would have failed.
- **Data-Driven Validation:** Contrasting the sparse "Video 0" against the chaotic "Video 3" was critical. This stress-test revealed that while the baseline *overfitted* simple patterns, the optimized pipeline possessed *true generalization*, maintaining a **0.56 HOTA** in dense scenes.
- **Semantic Reliability:** Beyond raw metrics, **Grid-Based Path Analysis** confirmed real-world utility. The model accurately captured semantic behavior, replicating the **top 5 dominant traffic flows** with *near-perfect fidelity* compared to Ground Truth.

Future Directions: The current Kalman Filter model lacks "*social*" awareness. Future work should integrate **Social Force Models** to predict collision-avoidance and stricter **Entry/Exit Logic** to prevent physically impossible track initializations, further refining the system for urban analytics.

References

[1] Robicquet, A., et al. (2016). Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. ECCV.

[2] Wang, S., et al. (2025). RLRD-YOLO: An Improved YOLOv8 Algorithm for Small Object Detection from an UAV Perspective. MDPI.

[3] Lou, H., et al. (2025). NSC-YOLOv8: A Small Target Detection Method for UAV-Acquired Images. MDPI Electronics.

[4] Zhang, Y., et al. (2022). ByteTrack: Multi-Object Tracking by Associating Every Detection Box. ECCV.

[5] Wu, X., et al. (2025). Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A Survey. IEEE Geoscience and Remote Sensing Magazine.

[6] Luo, W., et al. (2021). Deep Learning-Based Multi-Object Tracking: A Comprehensive Survey.

[7] Luiten, J., et al. (2021). HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. IJCV.

[8] Bernardin, K., & Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP Journal on Image and Video Processing.

[9] Ristani, E., et al. (2016). Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. ECCV Workshops.

[10] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NeurIPS.

[11] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. ICCV.

[12] Carion, N., et al. (2020). End-to-End Object Detection with Transformers. ECCV.

[13] Zhang, H., et al. (2023). DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. ICLR.

[14] Redmon, J., et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR.

[15] Jocher, G., et al. (2023). Ultralytics YOLOv8.

[16] Brasó, G., & Leal-Taixé, L. (2020). Learning a Neural Solver for Multiple Object Tracking. CVPR.

[17] Zeng, F., et al. (2022). MOTR: End-to-End Multiple-Object Tracking with Transformer. ECCV.