Experiments with Bot-SORT

Neta Mendelbaum & Hadar Margalith*

July 2025

Abstract

This project explores the performance of the BoT-SORT tracking framework for multi-object tracking (MOT) across varied environments. BoT-SORT extends the ByteTrack and SORT architectures with key enhancements, including a refined Kalman filter state representation, camera motion compensation (CMC), and IoU–Re-ID fusion for improved identity tracking.

We assess the impact of different YOLOv11 variants, multiple CMC strategies, and the integration of re-identification on tracking performance. Experiments were conducted on three benchmark datasets representing diverse challenges, from dense urban pedestrian scenes to aerial wildlife footage. Standard MOT evaluation metrics were used to quantify performance. Additionally, a targeted case analysis of tracking failures was conducted using IoU-based frame selection and visualize analysis.

Results indicate that the heavier YOLOv11x model does not consistently outperform the lightweight YOLOv11n, and that Re-ID often fail to yield significant accuracy gains. The optimal CMC configuration remains inconclusive. Notably, substantial performance issues on the BuckTales dataset suggest domain mismatch and insufficient training data. Overall, findings underscore the need for domain-specific tuning to achieve robust and generalizable tracking performance across diverse contexts.

1 Introduction

Multi-Object Tracking (MOT) in computer vision refers to the task of identifying, locating, and following multiple distinct objects over time within a sequence of images or video frames. The key challenge is not just detecting objects in each frame, but also maintaining their unique "identity" as they move, interact, disappear, and reappear. Object positions are typically represented using rectangular annotations known as bounding boxes (BBs).

In [Aharon et al., 2022] was proposed a novel MOT framework called **BoT-SORT**, that handles this task for presidents tracking. This model belongs to the SORT family of tracking algorithms [Wojke et al., 2017] and builds upon the architecture of ByteTrack [Zhang et al., 2021], introducing several key enhancements.

The first enhancement involves a modification to the **Kalman filter** state representation. By that time the costume was to defined the state vector as:

$$x = [x_c, y_c, a, h, \dot{x}_c, \dot{y}_c, \dot{a}, \dot{h}],$$

where (x_c, y_c) denotes the center of the bounding box, a represents its area, and h its height. BoT-SORT reformulates this as:

$$x_k = [x_c(k), y_c(k), w(k), h(k), \dot{x}_c(k), \dot{y}_c(k), \dot{w}(k), \dot{h}(k)],$$

substituting area a with width w and retaining height h. This adjustment improves the clarity of object boundaries , including detailed features like moving legs.

Secondly, BoT-SORT introduces **Camera Motion Compensation (CMC)** by leveraging functionalities provided by the **OpenCV** library [Bradski, 2000]. CMC addresses scenarios where the camera itself is in motion—whether mounted on moving platforms (e.g., vehicles), undergoing systematic motion (e.g., surveillance cameras with fixed rotation patterns), or experiencing minor displacements due to external factors (e.g., wind). Incorporating CMC enhances track continuity and reduces fragmentation.

The third innovation involves the integration of **IoU–Re-ID Fusion**, which increase the Kalman filter's reliance on spatial and velocity information with appearance-based features. By

*Department of Information Systems at the University of Haifa

considering visual similarity between detections across frames, the model achieves more robust identity preservation.

Collectively, these improvements yield measurable enhancements in standard MOT performance metrics, including **MOTA**, **IDF1**, and **HOTA** [Luiten et al., 2020], compared to previous state-of-the-art models.

This project aims to evaluate the impact of the BoT-SORT model's enhancements on **MOT** (Multiple Object Tracking) performance across a range of datasets, including pedestrian tracking and other types of objects. Specifically, we examine the influence of different YOLO versions, various CMC (camera motion compensation) methods, and option of re-identification (ReID) on overall tracking effectiveness.

The methodologies and the conditions used in our experiments are detailed in the following section. Subsequent section present the results, and a last section of discussion including limitations and conclusions.

The code used for our experiments is publicly available at `https://github.com/NettaMendel/Experiments-with-Bot-SORT`.

2 Method and Experiments

2.1 Datasets

In this study, we evaluate our approach using three different datasets. Each dataset contains distinct object types and scene characteristics, which allows us to assess the generalizability and robustness of the tracking models across varying contexts. Due to the differences between them, each dataset is evaluated separately.

2.1.1 MOT17

The MOT17 dataset consists of video sequences captured using both static and moving cameras. Each sequence is provided with three sets of detections: DPM, Faster R-CNN, and SDP. The training set comprises 15,948 frames, with frame rates ranging between 14 and 30 FPS [Dendorfer et al., 2020]. Figure 1 show few randomly selected images form this dataset.

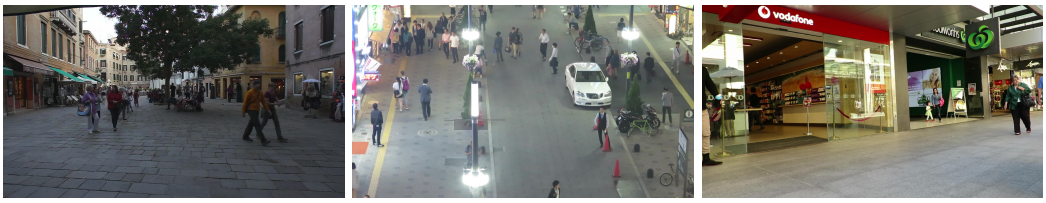
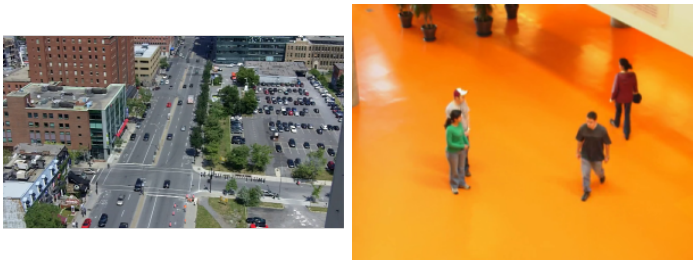


Figure 1: Image from the videos of MOT17 [Dendorfer et al., 2020].

2.1.2 Urban Tracker

The Urban Tracker dataset contains five video sequences of urban traffic scenes. Three of the videos capture a single intersection, while one features three intersections; all depict mixed traffic, including pedestrians, cars, and large vehicles. The fifth is recorded from inside a building, capturing indoor pedestrian activity such as walking, object handling, and crossing paths. The videos are recorded at 25–30 FPS and contain a total of 8,141 frames [Jodoin et al., 2014]. Figure 2 presents one frame from each video.

(a) Sherbrooke (b) Rouen (c) St-Marc

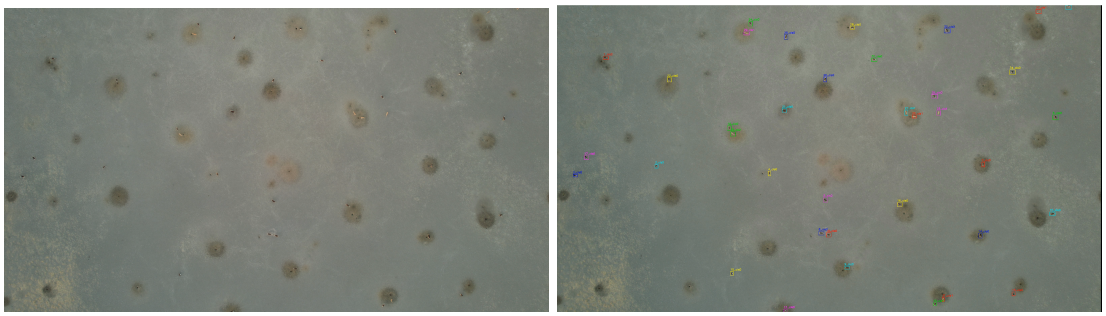


(d) René-Lévesque (e) Atrium

Figure 2: One Image from the videos of Urban Tracker [Jodoin et al., 2014].

2.1.3 BuckTales

The BuckTales dataset consists of 12 aerial videos capturing wild blackbuck antelopes in open grassland environments, recorded using unmanned aerial vehicles (UAVs). The videos are recorded at 30 frames per second (FPS) and vary in length from 10 seconds to 3 minutes [Naik et al., 2024]. Figure 3 shows two frames from this dataset: one raw frame and one with the corresponding annotations, based on the ground truth.



(a) Without annotation (b) With annotation

Figure 3: Examples of frames from BuckTales with and without annotation [Naik et al., 2024].

2.2 YOLO Variants

To examine the impact of model size and architectural complexity on tracking performance, we evaluated two versions of the YOLO (You Only Look Once) object detection framework. These versions represent different trade-offs between speed and accuracy, as they implemented by Ultralytics [Jocher and Qiu, 2024]:

- **YOLOv11n:** A lightweight variant in the YOLOv11 series, consisting of approximately 2.6 million parameters. YOLOv11n is designed for low-latency inference and high efficiency, making it suitable for real-time applications on resource-constrained devices. However, its compact size may limit detection accuracy in more complex scenarios.
- **YOLOv11x:** The extra-large version of the YOLOv11 family, with approximately 56.9 million parameters. YOLOv11x prioritizes detection accuracy over inference speed, offering greater feature extraction capabilities for dense and complex scenes, though it requires significantly more computational resources.

Both models were pre-trained on the COCO dataset [Lin et al., 2015], which comprises 80 object categories. These include general classes such as people, cars, and animals, as well as more specific objects like umbrellas, handbags, and sports equipment.

By comparing these two YOLO variants under consistent experimental conditions, we aim to assess how model size affects multi-object tracking performance when integrated into a shared pipeline. This evaluation helps clarify the practical trade-offs between detection accuracy, inference speed, and hardware requirements, which are essential for selecting the appropriate model in real-world deployments.

2.3 Camera Motion Compensation

As introduced in BoT-SORT [Aharon et al., 2022], camera motion compensation (CMC) plays a critical role in scenarios involving moving cameras. BoT-SORT supports multiple CMC methods; the default approach is based on sparse optical flow. In this study, we also evaluated two alternative methods: one based on Scale-Invariant Feature Transform (SIFT) and another based on Oriented FAST and Rotated BRIEF (ORB) features. This comparison enables us to examine the impact of different motion compensation strategies on tracking robustness and overall performance.

2.4 Re-Identification Module (Re-ID)

BoT-SORT also includes an optional re-identification (Re-ID) module that leverages object appearance features to assist in maintaining consistent identities across frames. This module supplements the traditional IoU-based tracking mechanism by improving robustness in situations with occlusions or missed detections. While the original BoT-SORT implementation reported improvements in tracking metrics with Re-ID enabled, it also noted a decrease in processing speed [Aharon et al., 2022]. As a result, Re-ID is disabled by default. In our experiments, we evaluated the impact of enabling this module on both performance and computational efficiency.

2.5 Evaluation Strategy

To evaluate the performance of our experiments, we adopted a two-stage approach. First, we used standard multi-object tracking evaluation metrics, as outlined in [Aharon et al., 2022].

Second, for a video in each of the datasets, we identified frames where the prediction deviated most significantly from the ground truth. We manually inspected these cases to determine whether they represented special or challenging scenarios that deserve further discussion.

2.5.1 Evaluation Metrics

For the evaluation, we used the metrics defined in [Luiten et al., 2020] and their implementation as provided in the TrackEval toolkit [Jonathon Luiten, 2020].

- **Multiple Object Tracking Accuracy (MOTA)**:
  MOTA measures three types of tracking errors at the detection level: false negatives (FN), false positives (FP), and identity switches (IDSW). The final MOTA score is defined as:

$$\text{MOTA} = 1 - \frac{|\text{FN}| + |\text{FP}| + |\text{IDSW}|}{|\text{gtDet}|}$$

  where $|\text{gtDet}|$ is the total number of ground-truth detections.

- **IDF1**:
  IDF1 evaluates the accuracy of identity preservation by computing a one-to-one mapping between ground-truth trajectories (gtTrajs) and predicted trajectories (prTrajs). It is defined as:

$$\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5|\text{IDFN}| + 0.5|\text{IDFP}|}$$

  where IDTP, IDFN, and IDFP represent true positives, false negatives, and false positives under identity matching, respectively.

- **Higher-Order Tracking Accuracy (HOTA)**:
  HOTA provides a unified measure that balances detection and association accuracy. For further details, see [Luiten et al., 2020].

- **False Positives (FP)**:
  A false positive is defined as any predicted detection (prDet) that does not correspond to a ground-truth detection (gtDet).
- **False Negatives (FN)**:
  A false negative is defined as any ground-truth detection that is not matched by any predicted detection.
- **Identity Switch (IDSW)**:
  An identity switch occurs when the tracker incorrectly swaps object identities or reinitializes a lost track with a different identity.
- **Unique ID Count (IDs):**
  The total number of unique object identities assigned by the tracker during the evaluation.
- **Frames Per Second (FPS)**:
  FPS measures the processing speed of the tracker, i.e., how many frames are processed per second (in Hz). It is highly dependent on the hardware used.

2.5.2 Evaluation of case analysis

We investigated the poorest-performing frames by comparing YOLO-based tracking predictions with ground truth annotations. Through an IoU-based matching process, we identified frames containing a high number of inaccurate matches—defined as predictions with low spatial overlap relative to the ground truth. The count of low-IoU matches served as a practical metric for detecting frames with significant prediction errors. To support visual analysis, we extracted the corresponding images and generated three visualizations per frame: one with ground truth boxes, one with predicted boxes, and one combining both using distinct colors. This enabled a detailed examination of common failure patterns in the model's tracking performance.

2.6 Implementation Details

All experiments were conducted using the PyTorch framework [Paszke et al., 2019] and the Ultralytics package [Jocher and Qiu, 2024]. The models were evaluated on a desktop equipped with an Intel® Core™ Ultra 9 275HX processor (up to 5.4 GHz with Intel® Turbo Boost Technology, 36 MB L3 cache, 24 cores, 24 threads) and an NVIDIA® GeForce RTX™ 5070 Laptop GPU with 8 GB of GDDR7 memory.

We used the default model configurations provided by Ultralytics [Jocher and Qiu, 2024] without any additional training or hyperparameter tuning, except for what we discussed earlier.

3 Results

3.1 Model Evaluation

3.1.1 Results on BuckTales

For the BuckTales dataset, the model failed to detect any relevant objects, except for a few birds that occasionally passed by (Note: the dataset focuses on antelopes). We will discuss several possible reasons for this failure in the discussion section. This issue was observed with both YOLOv11n and YOLOv11x, indicating that the other BoT-SORT parameters would not meaningfully impact the model's performance, except for the FSP, since they all depend on at least one correct detection in the initial frame where the object appears. Therefore, we did not evaluate any CMC metrics beyond the default configuration and haven't preformed case analysis on this dataset.

3.1.2 MOT17 Results

Table 1 presents the results of the evaluated metrics on the MOT17 training set. It can be observed that different configurations performed best for different metrics, indicating no single configuration dominated across all evaluation criteria.

While YOLOv11x, the largest model, showed slight improvements in some cases, it did not demonstrate a consistent advantage over the smaller YOLOv11n variant. Furthermore, the use of the Re-ID option—which is designed to improve identity preservation—did not lead to better tracking accuracy, although it did contribute to lower FPS performance.

Overall, except for FPS, all evaluated metrics underperformed compared to results reported in previous works. These discrepancies and their potential causes will be discussed in the following Discussion and Limitations section. On the positive side, the FPS results outperformed those reported in [Aharon et al., 2022], as further explained in the next section.

YOLO variant	ReID	CMC	HOTA	MOTA	CLR_FN	CLR_FP	IDSW	IDF1	IDs	FPS
YOLO 11 n	w\o	sparseOptFlow	35.6	**23.9**	203,848	**51,078**	1,410	40.6	3,241	35.5
	w	sparseOptFlow	35.8	22.1	198,266	62,103	2,064	41.2	4,519	46.3
	w\o	sift	35.8	23.7	203,524	52,092	1,404	40.8	**3,181**	**63.5**
	w	sift	36.0	22.0	197,975	62,904	2,013	41.4	4,366	47.0
	w\o	orb	35.9	23.8	203,407	51,948	1,365	40.9	3,217	33.3
	w	orb	**36.1**	22.0	198,116	62,823	1,953	**41.6**	4,435	28.5
YOLO 11 x	w\o	sparseOptFlow	35.4	4.5	196,398	124,119	1,194	38.5	4,263	38.9
	w	sparseOptFlow	35.0	2.7	190,078	136,256	1,467	37.8	5,226	29.9
	w\o	sift	35.6	4.4	196,095	124,869	**1,068**	38.6	4,225	27.3
	w	sift	36.0	22.0	197,975	62,904	2,013	41.4	4,366	58.1
	w\o	orb	35.5	4.4	196,212	124,927	1,083	38.5	4,213	38.4
	w	orb	35.0	2.6	**189,911**	136,914	1,341	37.7	5,169	30.3

Table 1: Comparison of the configurations discussed in the Method and Experiments sections on the MOT17 training set. The best results are highlighted in bold.

3.1.3 Urban Tracker Results

The results for the UrbanTracker dataset are shown in Table 2. The first observation is that the MOTA scores obtained are invalid (i.e., negative). This issue will be discussed in detail in the next section. For the remainder of this section, we disregard the MOTA results and focus on the remaining metrics.

Overall, the results for this dataset were generally poor. However, it is worth noting that the YOLOv11x model, especially when used with the CMC shift option (with or without the Re-ID component), achieved better performance compared to the other configurations.

Another notable observation is that, for the HOTA and false positive (FP) metrics, the use of Re-ID resulted in worse performance compared to the same configurations without Re-ID. In contrast, the Re-ID component improved the results for false negatives (FN) and IDF1.

Two specific metrics worth attention: the ID switch (IDSW) count, which was particularly low—especially when using YOLOv11x—and the frames per second (FPS), which remained consistently high in this dataset, especially in YOLOv11n Variant. Both results are encouraging, indicating improved identity consistency and efficient processing speed.

YOLO variant	ReID	CMC	HOTA	MOTA	CLR_FN	CLR_FP	IDSW	IDF1	IDs	FPS
YOLO 11 n	w\o	sparseOptFlow	14.6	-666.4	13,962	215,046	157	9.9	1,000	**45.1**
	w	sparseOptFlow	14.2	-785.6	13,079	251,455	258	9.4	1,353	39.4
	w\o	sift	14.7	-667.3	13,951	215,331	153	10.0	970	32.1
	w	sift	14.3	-785.7	13,063	251,512	252	9.5	1,329	36.1
	w\o	orb	14.5	**-649.8**	14,006	**210,003**	175	9.9	1,630	35.9
	w	orb	14.1	-772.7	13,132	247,540	276	9.4	2,127	36.5
YOLO 11 x	w\o	sparseOptFlow	17.8	-675.2	10,122	221,596	81	13.2	913	36.7
	w	sparseOptFlow	17.3	-750.1	9,571	244,549	78	12.6	1,185	31.3
	w\o	sift	**17.9**	-674.9	10,103	221,525	79	**13.3**	**903**	25.7
	w	sift	17.4	-750.0	**9,552**	244,522	**77**	12.6	1,171	24.7
	w\o	orb	17.7	-672.4	10,163	220,687	92	13.1	1,293	24.5
	w	orb	17.2	-747.1	9,611	243,582	94	12.5	1,557	23.2

Table 2: Comparison of the configurations discussed in the Method and Experiments sections on the Urban Tracker dataset. The best results are highlighted in bold.

3.2 Case Analysis

The case analysis presented in the following subsection was conducted on one video from each dataset, excluding the BuckTales dataset. The selected videos were those that performed the worst within their respective datasets in terms of the HOTA, MOTA, and IDF1 metrics.

3.2.1 MOT17

In multiple Figures analyzed from the *YOLO11n* model (e.g., Figure 4b), several recurring discrepancies were observed between the predicted bounding boxes and the ground truth annotations. The model consistently detected non-human components like backpacks and hands as independent

objects, even though they were not defined separately in the ground truth. Additionally, in several cases, YOLO failed to identify individuals seated in the background or distinguish between a person and the bicycle they were riding, treating them as a single object or omitting the person altogether. Bounding boxes were sometimes notably smaller than the actual object, especially in cases involving partial visibility of the object. These issues were also evident in Figure 4a from the *YOLO11n with ORB and ReID* variant, where distant individuals were missed and accessory items were over-detected. These errors suggest that YOLO11n tends to over-detect accessory items while under-detecting small, distant, or occluded persons, possibly due to differences in labeling conventions or the model's training bias toward prominent foreground objects.

In several frames analyzed from the *YOLO11x* model in the different configuration (e.g., Frames 4d, 4c), it can be seen that identification items that are physically part of a person as independent objects was also present. Distant individuals were often detected only partially, with bounding boxes failing to encompass the full extent of the person.

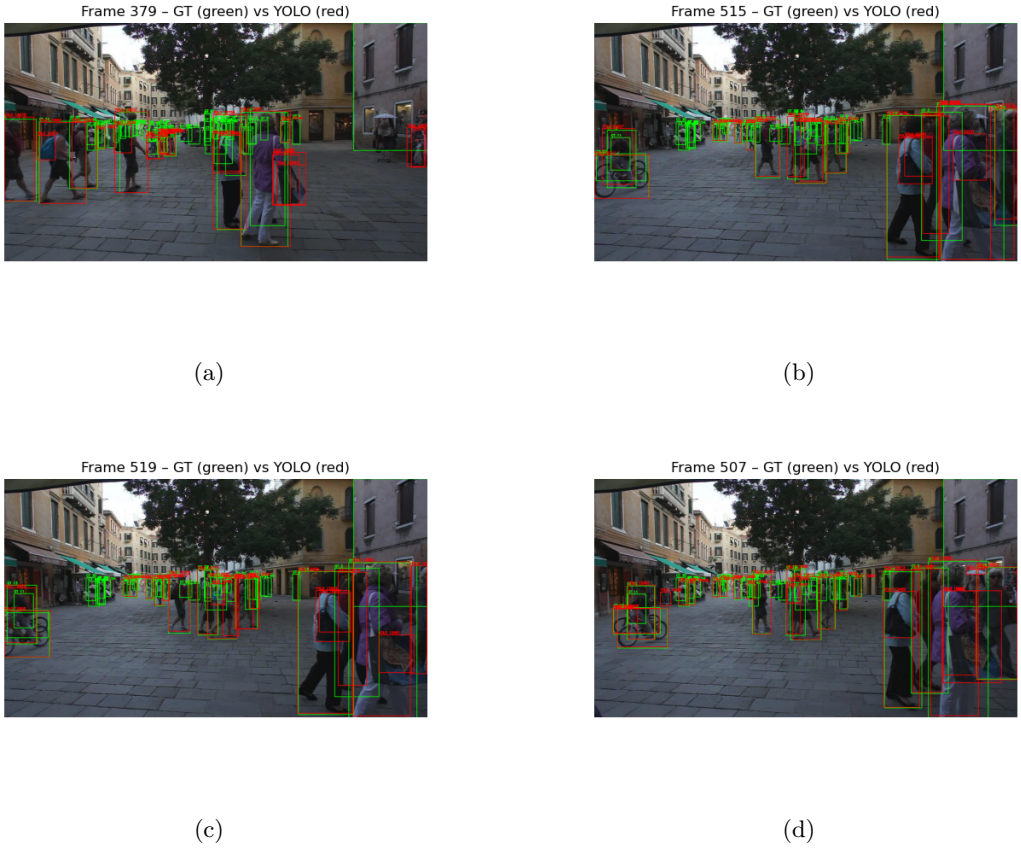


Figure 4: Examples of discrepancies between YOLO-based predictions (red) and ground truth (green) across different frames and model variants. **(a)** Frame 379 using YOLOv11n with ORB and ReID. **(b)** Frame 515 using YOLOv11n with SparseOptFlow and without ReID. **(c)** Frame 519 using YOLOv11x with SIFT and ReID. **(d)** Frame 507 using YOLOv11x with SparseOptFlow and ReID.

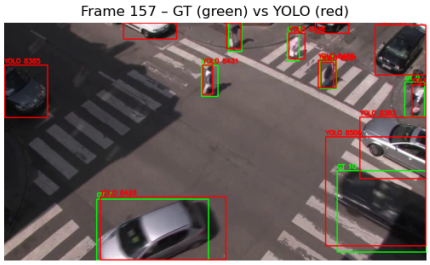
3.2.2 Urban Tracker

Upon examining the results, we identified several noteworthy cases. In some instances, the model detected objects (across most configurations) that were not present in the ground truth annotations but clearly exist in the scene, as illustrated in Figure 5a. Additionally, we observed several cases where both the model and the ground truth annotated the same object, but the bounding boxes were not well aligned (in the same figure for example).

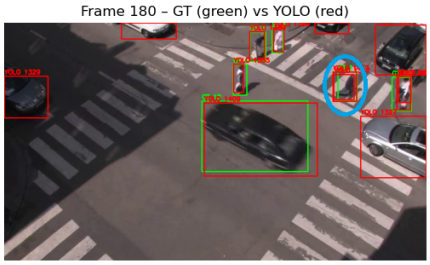
In Figure 5b, the model mistakenly detects the same object twice, as highlighted by a circle. Figure 5c shows a mostly correct detection. However, despite being visually accurate, our evaluation flagged it as suboptimal due to slight misalignment with the ground truth annotations.

In general, YOLOv11x detected more objects compared to YOLOv11n. For example, in Figure 5d, a car is missing from the YOLOv11n detection in most configurations that flagged this

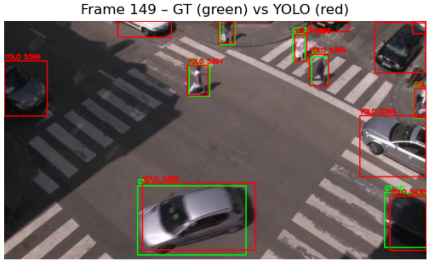
frame. However, as shown in Figure 5e, the car was detected in the subsequent frame, albeit with a significant localization error.



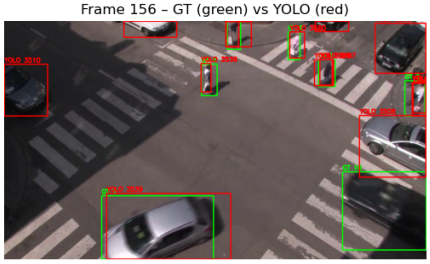
(a)



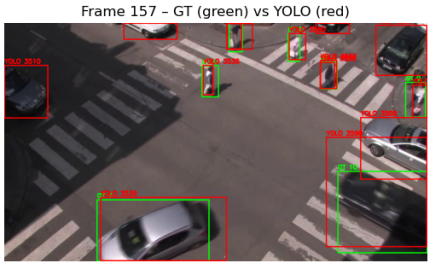
(b)



(c)



(d)



(e)

Figure 5: Results from the Rouen video. Model detections are shown in red, and ground truth in green. **(a)** YOLOv11n with CMC set to ORB and without ReID – Frame 157. **(b)** YOLOv11n with CMC set to SIFT and with ReID – Frame 180. **(c)** YOLOv11x with CMC set to ORB and without ReID – Frame 149. **(d)** YOLOv11n with CMC set to ORB and with ReID – Frame 156. **(e)** YOLOv11n with CMC set to ORB and with ReID – Frame 157.

4 Discussion and Limitations

4.1 Discussion on BuckTales Results

As discussed in the Results section, the models failed entirely on the BuckTales dataset. This failure can be attributed to several factors, or a combination thereof.

The primary reason likely stems from the pretraining dataset used for the YOLO models. As noted earlier, the models were trained on the COCO dataset, which includes 80 object categories;

however, antelopes are not among them. As a result, the models were not equipped to recognize this specific class, which significantly limits their ability to detect and track the blackbuck antelopes present in BuckTales.

A second contributing factor relates to the visual characteristics of the dataset. As shown in Figure **??**, the antelopes appear relatively small in the video frames due to the high-altitude aerial perspective from which the videos were recorded. In addition, their body color closely resembles the grassland background, resulting in low visual contrast. This combination of small object size and background similarity poses a significant challenge for object detection models like YOLO, which tend to struggle in detecting small, low-contrast objects—particularly when those objects are not part of the training classes.

Together, these factors can explain the poor performance observed on this dataset and suggest that domain-specific fine-tuning or additional training on more relevant data would be necessary for effective tracking in this setting.

4.2 Discussion on MOT17

As mentioned earlier, the results we obtained on the MOT17 dataset were not satisfactory, especially when compared to those reported in [Aharon et al., 2022] and to general expectations for this benchmark. Several factors may explain this underperformance.

First, the YOLO variants used in our experiments, as introduced in [Jocher and Qiu, 2024], were trained on images from the COCO dataset, not on video sequences from MOTChallenge. This domain mismatch may have limited the detector's ability to generalize to the specific characteristics of MOT17 sequences.

Second, the MOT17 ground truth annotations include mostly pedestrians who move through the scene, while the YOLO detector returns all detected objects—including non-human classes. This mismatch can lead to an overabundance of predicted detections relative to the ground truth, potentially inflating false positives and harming evaluation metrics such as MOTA and HOTA.

The high FPS performance we observed is likely due to the significantly more powerful hardware used in our setup compared to that used in [Aharon et al., 2022].

Regarding the cases in which parts of an object are detected as separate objects are likely the result of detection errors. These errors may arise from inconsistent motion between different parts of the same object, which can confuse the detector. In some instances, the issue may be due to class misclassification—for example, an umbrella held by a person or a chair that someone is sitting on might be detected as a standalone object, despite not being labeled as such in the ground truth. Future work could aim to further identify and analyze these cases in order to better understand their causes and explore potential solutions to mitigate this problem.

As for cases where distant pedestrians were not detected, several explanations are possible. If the COCO dataset (on which YOLO was trained) contains relatively few examples of small or distant individuals, the model may lack sufficient exposure to such cases. Additionally, distant pedestrians typically appear smaller and less distinguishable, making it more difficult for the detector to separate them from the background or from one another. Future work could address this issue by training YOLO models on datasets with more instances of small and distant pedestrians, thereby improving the model's familiarity with such scenarios.

In cases where the bounding box does not align well with the ground truth because the person is partially concealed by another, this may indicate the model's inability to maintain consistent bounding box sizes throughout a tracked trajectory. Such issues highlight a potential area for improvement in future trackers architectures.

Few more directions that can be review in future works include exploring more advanced re-identification (ReID) modules and integrating temporal cues (e.g., optical flow or temporal attention) could also help improve detection of occluded or small-scale pedestrians. Finally, benchmarking under varying hardware conditions would clarify the trade-off between performance and real-time feasibility.

4.3 Discussion on Urban Tracker

As previously mentioned, the MOTA scores obtained on the Urban Tracker dataset were invalid (i.e., negative). This issue may stem from the fact that [Jonathon Luiten, 2020] does not natively support this dataset, necessitating manual adjustments to enable evaluation. Specifically, the ground truth provided with Urban Tracker was not in the format required by TrackEval, and we attempted to convert it manually. Potential formatting errors in this conversion process may have

contributed to the unreliable MOTA results. Adding annotations in the MOTChallenge format to this dataset could facilitate future efforts to evaluate tracking results more reliably.

Overall, performance across the remaining metrics—excluding IDSW and FPS—was also suboptimal. However, we were unable to find prior studies that reported results using these metrics (apart from MOTA), leaving us without a clear baseline for comparison.

An interesting observation is that the CMC-SIFT option yielded better results on the Urban Tracker dataset compared to both ORB and SparseOptFlow. This outcome was somewhat unexpected, given that SparseOptFlow is the default in both [Aharon et al., 2022] and [Jocher and Qiu, 2024]. Notably, neither source explicitly justifies this choice; the default is simply adopted in the released code. This discrepancy may be attributed to the fact that Urban Tracker was not used for training or evaluation in the development of BoT-SORT, which could explain the superior performance of SIFT in our experiments. Future work should investigate the effectiveness of different CMC options across a wider range of datasets and consider evaluating additional available methods.

On the other hand, one unsurprising outcome was that YOLOv11x generally outperformed YOLOv11n. This can be attributed to the significant difference in model size and capacity, with YOLOv11x having greater representational power.

As with the MOT17 results, the high FPS values observed in this dataset are primarily due to the high-performance hardware used in our experimental setup.

In our case analysis of a video from this dataset, we found that the claims made by the original authors—that their modifications enhance the model's ability to capture peripheral parts of objects (e.g., feet)—did not consistently hold in our evaluation. Specifically, we observed numerous instances in which the detected bounding boxes failed to include such peripheral regions. This discrepancy may again stem from differences in the datasets used in [Aharon et al., 2022], suggesting an opportunity for further exploration and potential improvement.

Additionally, we identified instances where visible objects in the video were missing from the ground truth. Most of these objects remained stationary throughout the majority of the video, which may explain why they were not annotated. However, this omission poses challenges for evaluating the dataset: detectors such as YOLO are likely to flag these objects regardless of their motion, and any attempt to introduce a pipeline component that filters out such stationary objects would undermine real-time applicability. This is because determining that an object has remained static can only be confirmed retrospectively, at the end of the video. This issue has two important implications. First, it suggests that the dataset may require re-annotation to ensure the correctness and completeness of the official ground truth. Second, it indicates that our reported metric scores may not fully reflect the model's true performance. This may also partially explain the anomalously MOTA values observed.

In cases where the model detected the same object twice within a single frame, one possible explanation is that in earlier frames, there were indeed two distinct objects that later became spatially merged. In such scenarios, this behavior could be considered desirable. However, in the specific cases we analyzed, duplicate detections appeared to correspond to the same object, indicating a tracking error. This issue highlights an area for potential improvement in future work.

Lastly, in cases where the tracker failed to detect a car, one possible explanation is the vehicle's speed—particularly considering that BoT-SORT was primarily designed for pedestrian tracking and may not be well-suited for handling fast-moving vehicles. Another explanation is that the object had only recently entered the frame and was subsequently detected in later frames, as observed in some of our cases. Nonetheless, this limitation should be addressed in future studies.

4.4 Conclusion

In this work, we conducted a series of experiments in the field of multiple object tracking (MOT) using the BoT-SORT model. We showed that in some cases, using a larger YOLO model can lead to improved performance; however, this is not always the case. Additionally, some of the improvements introduced by BoT-SORT—such as the use of re-identification (ReID)—did not consistently yield the expected benefits across all settings.

Another important observation relates to the choice of the camera motion compensation (CMC) method. Our results suggest that the optimal CMC configuration remains an open question and warrants further investigation.

We also validated the assumption that stronger hardware leads to faster model performance. In our experiments, we achieved higher FPS compared to those reported in the BoT-SORT paper, which can be attributed to the use of more powerful hardware.

An additional finding worth noting is that although the BoT-SORT model was originally designed for pedestrian tracking, it demonstrated the ability to perform on non-human objects as well, provided that the YOLO detector was trained to recognize those object classes.

References

[Aharon et al., 2022] Aharon, N., Orfaig, R., and Bobrovsky, B.-Z. (2022). Bot-sort: Robust associations multi-pedestrian tracking.

[Bradski, 2000] Bradski, G. (2000). *The Opencv Library*, volume 25.

[Dendorfer et al., 2020] Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I. D., Roth, S., and Leal-Taixé, L. (2020). Motchallenge: A benchmark for single-camera multiple target tracking. *CoRR*, abs/2010.07548.

[Jocher and Qiu, 2024] Jocher, G. and Qiu, J. (2024). Ultralytics yolo11.

[Jodoin et al., 2014] Jodoin, J.-P., Bilodeau, G., and Saunier, N. (2014). Urban tracker: Multiple object tracking in urban mixed traffic. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, Steamboat Springs, Colorado, USA. Accepted for WACV 2014.

[Jonathon Luiten, 2020] Jonathon Luiten, A. H. (2020). Trackeval. https://github.com/JonathonLuiten/TrackEval.

[Lin et al., 2015] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.

[Luiten et al., 2020] Luiten, J., Osep, A., Dendorfer, P., Torr, P. H. S., Geiger, A., Leal-Taixé, L., and Leibe, B. (2020). HOTA: A higher order metric for evaluating multi-object tracking. *CoRR*, abs/2009.07736.

[Naik et al., 2024] Naik, H., Yang, J., Das, D., Crofoot, M. C., Rathore, A., and Sridhar, V. H. (2024). Bucktales : A multi-uav dataset for multi-object tracking and re-identification of wild antelopes.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv:1912.01703. NeurIPS 2019 preprint.

[Wojke et al., 2017] Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402.

[Zhang et al., 2021] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. *CoRR*, abs/2110.06864.

Statement on the Use of Artificial Intelligence

During the preparation of this study, artificial intelligence tools were used for the following purposes:

- **Code writing and development**: Microsoft Copilot and ChatGPT were used.
- **Academic editing and language refinement**: ChatGPT and Gemini were employed.

These tools assisted in improving code quality, clarifying ideas, and enhancing the language and structure of the manuscript. However, the authors take full responsibility for the content, accuracy, and originality of the work.