International Research Journal of Multidisciplinary Scope (IRJMS), 2026;7(3):1348-1362

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i03.011543

A Comparative Evaluation of Lightweight YOLO Detectors with BoT-SORT and ByteTrack for Real-time Multi-object Tracking

Feriantano Sundang Pranata[1]*, Jufriadif Na'am[2], Yulhan[3], Yuke Permata Lisna[1], Rima Agustia Utami[4], Risma Rahmatunisa[4], Indra Saputra[5], Elviza Yeni Putri[4], Melda Mahniza[4], Nurul Inayah Hutasuhut[4]

[1]Department of Tourism, Universitas Negeri Padang, Padang, Indonesia, [2]Informatics, Universitas Nusa Mandiri, Jakarta, Indonesia, [3]Informatics Management, Universitas Mahaputra Muhammad Yamin, Solok, Indonesia, [4]Department of Family Welfare Sciences, Universitas Negeri Padang, Padang, Indonesia, [5]Department of Beauty and Cosmetology, Universitas Negeri Padang, Padang, Indonesia. *Corresponding Author's Email: feriantano@unp.ac.id

Abstract

Real-time multi-object tracking (MOT) requires detector-tracker configurations that balance detection accuracy, association reliability, identity preservation and computational efficiency, particularly in central processing unit (CPU)-only and resource-constrained inference settings. However, previous You Only Look Once (YOLO)-based MOT studies have often evaluated isolated detector-tracker pairs, domain-specific scenarios, or heterogeneous hardware settings, which limits direct comparability across configurations. This study evaluates fourteen detector-tracker configurations using seven lightweight YOLO-based object detectors, namely YOLOv3-tiny, YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n and YOLO12n and two online trackers, BoT-SORT and ByteTrack. Experiments were conducted on MOT17 and MOT20 under a unified CPU-only private-detection protocol. Each configuration was assessed using metrics covering detection accuracy, tracking performance, identity preservation and computational efficiency. No single configuration dominated all evaluation dimensions. BoT-SORT generally showed stronger tracking performance and identity preservation, as indicated by higher tracking and identity-related metrics and fewer identity switches, whereas ByteTrack showed higher throughput and shorter runtime. Matched-detector comparisons supported this trade-off, indicating that BoT-SORT was more favorable for association and identity continuity, whereas ByteTrack was more favorable for computational efficiency. Among the evaluated detectors, YOLOv8n, YOLOv10n and YOLO11n showed relatively balanced accuracy-efficiency profiles across datasets and metrics. These findings support the interpretation that CPU-only MOT performance is shaped by detector-tracker interaction rather than detector accuracy or runtime efficiency alone. Therefore, configuration selection should be aligned with deployment priorities, particularly the trade-off between identity continuity and throughput.

Keywords: BoT-SORT, ByteTrack, CPU-only Inference, Detector-tracker Configuration, Lightweight YOLO Detectors, Multi-object Tracking.

Introduction

Video-based visual understanding plays an important role in modern computer vision because many intelligent systems are expected to interpret dynamic scenes continuously rather than detect objects in isolated frames. In this context, multi-object tracking (MOT) combines object detection, localization, temporal association and identity preservation to maintain target trajectories across video frames (1). MOT supports applications such as intelligent surveillance, traffic monitoring, autonomous driving, robotics and embedded vision systems, where trajectory-level information is required for real-time scene interpretation and decision-making (2). Unlike frame-level object detection, MOT must preserve object identities as targets move, interact, become occluded, disappear, or reappear in complex environments (3). Therefore, MOT performance depends on both accurate detections and reliable association mechanisms that maintain temporal continuity under changing spatial and visual conditions (4, 5). This makes MOT a deployment-relevant task that extends beyond frame-level detection because a useful tracking system must preserve trajectories

(Received 01st April 2026; Accepted 15th July 2026; Published 23rd July 2026)

while satisfying real-time processing constraints. Real-world MOT remains challenging because detection errors and association errors are strongly interconnected. In crowded or uncontrolled scenes, partial occlusion, illumination changes, camera motion, appearance similarity and overlapping trajectories can reduce detection accuracy and increase uncertainty in frame-to-frame association (6, 7). Missed detections may fragment trajectories, false detections may introduce incorrect tracks and ambiguous associations may cause identity switches when multiple targets move in close proximity (8). These challenges suggest that MOT performance should be understood as an interaction among detection accuracy, association reliability, identity preservation and computational efficiency. This interaction becomes more constrained in central processing unit (CPU)-only and resource-constrained environments, where limited computational capacity, memory availability, inference speed and latency constraints require careful control over detector and tracker complexity (9, 10). These constraints are particularly relevant for low-cost surveillance systems, embedded monitoring devices and field deployments where dedicated accelerators are unavailable or not feasible for routine deployment. Practical real-time MOT systems must therefore balance accuracy and efficiency because a highly accurate configuration may be computationally impractical, whereas a fast configuration may fail to maintain stable identities (11).

Most real-time MOT pipelines follow a tracking-by-detection paradigm, in which object detections are generated in each frame and associated across consecutive frames to form trajectories (12). You Only Look Once (YOLO)-based object detectors are widely used in this paradigm because they provide a practical balance between detection accuracy, inference speed and deployment feasibility (13). Recent YOLO families have introduced design changes related to programmable gradient information, end-to-end detection, architectural optimization and attention-oriented detection (14–17). However, the usefulness of a detector in MOT is not determined only by frame-level detection accuracy, but also by whether detector outputs support stable downstream association. A high-precision detector may reduce false positives, but low recall can interrupt trajectories. Conversely, a detector that produces more candidate boxes may improve target continuity while increasing association ambiguity. ByteTrack is designed to handle detection uncertainty by associating both high-confidence and low-confidence detection boxes (18). Bag of Tricks for Simple Online and Realtime Tracking (BoT-SORT) is designed to improve association through motion information, camera-motion compensation and refined state estimation, with optional appearance cues in its standard formulation (19). These tracker design differences indicate that detector choice and tracker design should be evaluated together under the same experimental setting.

Previous YOLO-based MOT studies have shown the feasibility of combining YOLO detectors with tracking algorithms across applied domains. YOLO-based pipelines have been used for pedestrian tracking and crowded-scene MOT (20–22), vehicle tracking and traffic management (23–25) and domain-specific applications such as bird and unmanned aerial vehicle tracking, dairy cow monitoring, fish tracking, maritime tracking and edge-oriented driver-assistance systems (26–30). Although these studies demonstrate the adaptability of YOLO-based MOT pipelines, many evaluations remain limited to a single detector version, a fixed tracker, a specific application domain, or hardware settings that are not directly comparable. This fragmentation limits the ability to determine whether observed performance differences are caused by detector behavior, tracker association strategy, dataset difficulty, or computational setting. Consequently, existing evidence remains fragmented regarding how earlier and recent lightweight YOLO detectors behave when evaluated with alternative real-time trackers under the same evaluation protocol. More specifically, limited systematic evidence is available on how lightweight YOLO detectors interact with BoT-SORT and ByteTrack in terms of detection accuracy, tracking performance, identity preservation, throughput and runtime efficiency under a unified CPU-only MOT setting.

To address this gap, this study evaluates fourteen detector-tracker configurations using seven lightweight YOLO-based object detectors, namely YOLOv3-tiny, YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n and YOLO12n and two online trackers, BoT-SORT and ByteTrack. The evaluation uses MOT17 and MOT20 to examine detector-tracker behavior under crowd density, occlusion, camera motion, identity ambiguity and visual variation (31). This study addresses the gap by using a unified CPU-only private-detection protocol to evaluate earlier and recent lightweight YOLO detectors with BoT-SORT and ByteTrack under the same evaluation setting. This design supports a controlled assessment of how detector choice and tracker design jointly affect detection accuracy, tracking performance, identity preservation and computational efficiency. This study contributes by providing empirical evidence on detector-tracker trade-offs in CPU-limited inference conditions. Specifically, it provides a controlled CPU-only comparison, evaluates detector-tracker behavior across detection, tracking, identity and efficiency dimensions and provides deployment-oriented evidence for selecting MOT configurations under constrained inference conditions. Rather than ranking configurations solely by detection accuracy or runtime, this study interprets performance as a multidimensional deployment trade-off across detection, tracking, identity and efficiency metrics.

Methodology

This study used a controlled comparative experimental design to evaluate detector-tracker configurations for real-time MOT under a unified CPU-only private-detection protocol. Each configuration consisted of one lightweight YOLO-based object detector and one online tracker, either BoT-SORT or ByteTrack. The evaluated detectors were YOLOv3-tiny, YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n and YOLO12n. Each detector was used as a frame-level object detector to generate person-class detections from the selected MOT17 and MOT20 benchmark sequences. The resulting detections were processed independently by BoT-SORT and ByteTrack to produce trajectories and identity assignments.

All configurations were evaluated using the same benchmark sequences, detector inference settings, comparable tracker parameters where applicable and evaluation protocol. This procedure was used to reduce variation caused by inconsistent detection sources, heterogeneous hardware settings, or different tracking inputs. The evaluation followed a private-detection setting, meaning that detections were generated by the evaluated YOLO detectors rather than taken from public MOTChallenge detections. Therefore, the comparison focused on how detector choice and tracker design jointly affected tracking performance under the same CPU-only inference condition.

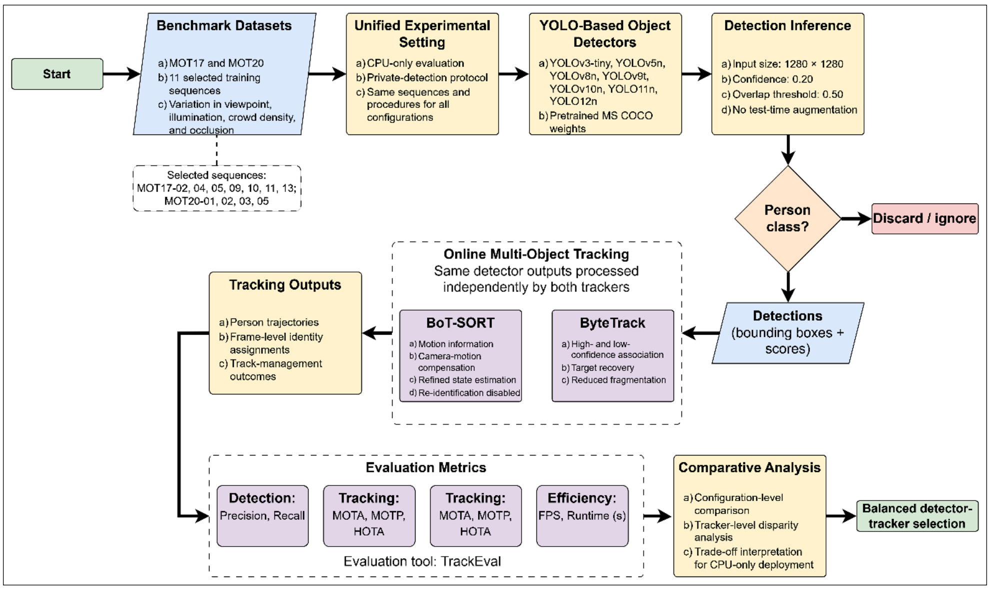

Figure 1: Experimental Workflow for CPU-only Private-detection Multi-object Tracking Evaluation

Figure 1 summarizes the experimental workflow used to evaluate the fourteen detector-tracker configurations under the CPU-only private-detection protocol. The workflow began with the selection of MOT17 and MOT20 sequences. Each video frame was passed through one of the evaluated lightweight YOLO-based object detectors to obtain bounding boxes, confidence scores and class predictions. A person-class filtering step was then applied because MOT17 and MOT20 are pedestrian tracking benchmarks. Detections classified as person were retained and formatted as tracking inputs, whereas non-person detections were discarded. The retained person detections from each detector were processed separately by BoT-SORT and ByteTrack using equivalent detector outputs. The tracker outputs consisted of frame-level bounding boxes, trajectory identifiers, identity assignments and track-management results. These outputs were evaluated using metrics covering detection accuracy, tracking performance, identity preservation and computational efficiency. The quantitative evaluation was complemented by matched-detector performance disparity analysis and qualitative error analysis to interpret the observed trade-offs across all fourteen detector-tracker configurations.

Benchmark Datasets

This study used MOT17 and MOT20 from the MOTChallenge initiative as pedestrian MOT benchmarks. MOT17 was selected because it contains diverse pedestrian scenes with variations in viewpoint, camera motion, illumination, pedestrian density and background complexity. Although MOT17 provides public detections generated by Deformable Part Model, Faster region-based convolutional neural network and scale-dependent pooling, these detections were not used because this study adopted a private-detection protocol in which all detections were generated by the evaluated lightweight YOLO detectors. MOT20 was included to represent more crowded scenes with heavier occlusion, higher pedestrian density and more frequent target overlap (31). The use of both datasets allowed evaluation across selected sequences with different density and occlusion characteristics.

The experiments used 11 selected training sequences: MOT17-02, MOT17-04, MOT17-05, MOT17-09, MOT17-10, MOT17-11, MOT17-13, MOT20-01, MOT20-02, MOT20-03 and MOT20-05. The training sequences were used for evaluation because their ground-truth annotations are available. These sequences were selected to provide variation in crowd density, occlusion, viewpoint, camera movement, illumination and background complexity while maintaining computational feasibility for CPU-only evaluation. The same detection, tracking and evaluation procedures were applied to all selected sequences to support a controlled within-study comparison across detector-tracker configurations. Figure 2 shows representative frames from the selected MOT17 and MOT20 sequences to illustrate variations in scene density, occlusion, viewpoint, illumination and background complexity.

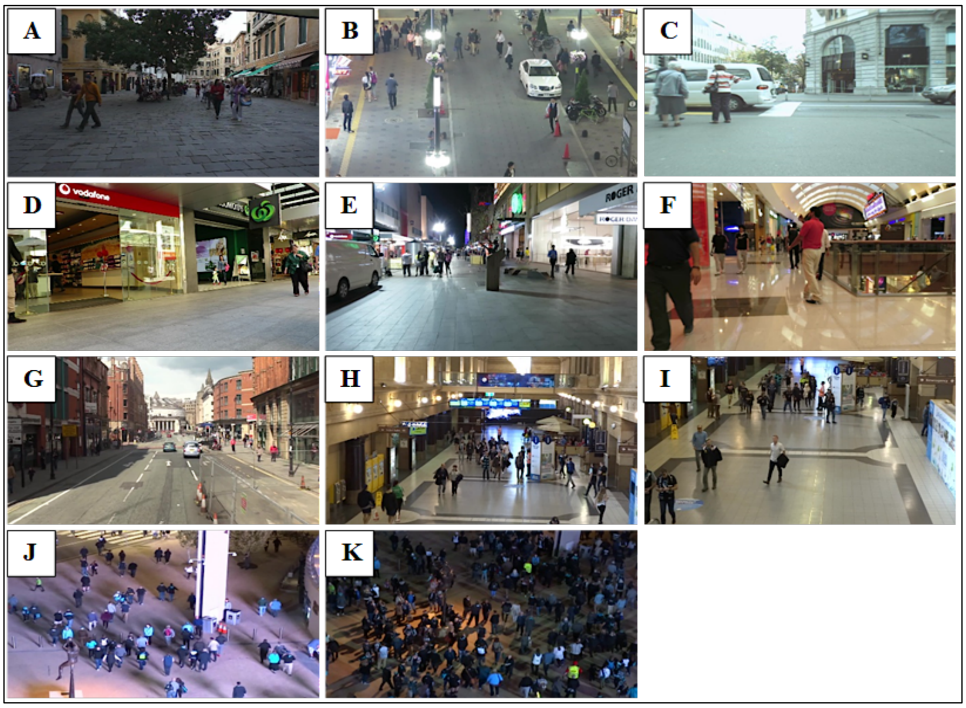

Figure 2: Representative Frames from the Selected MOT17 and MOT20 Sequences Used in the Evaluation. (A) MOT17-02, (B) MOT17-04, (C) MOT17-05, (D) MOT17-09, (E) MOT17-10, (F) MOT17-11, (G) MOT17-13, (H) MOT20-01, (I) MOT20-02, (J) MOT20-03, (K) MOT20-05

YOLO-based Object Detectors

Seven lightweight YOLO-based object detectors were used as frame-level detection modules in the experimental MOT pipeline: YOLOv3-tiny, YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n and YOLO12n. In this study, lightweight YOLO-based object detectors refer to compact-scale detector choices within their respective YOLO families, represented by tiny-scale or nano-scale models and characterized by relatively low parameter counts and model sizes. The selected detectors were included to cover compact detector choices from earlier and recent YOLO families under the same MOT evaluation pipeline. This selection enabled comparison across different lightweight YOLO detector designs while keeping the detection stage, tracking inputs and evaluation procedure consistent across configurations (13–17).

In the detection stage, each YOLO detector was applied independently to every video frame to generate bounding boxes, confidence scores and class predictions. Only person-class detections were retained because MOT17 and MOT20 are pedestrian tracking benchmarks (31). The retained person detections were then used as private detections for the tracking stage, meaning that tracking inputs were generated by the evaluated YOLO detectors rather than by public benchmark detections. The same frame-level detection workflow, including input sequences, preprocessing, confidence filtering, non-maximum suppression, person-class filtering and output formatting, was applied across all detector-tracker configurations. This procedure reduced variation caused by inconsistent detection sources and supported a controlled comparison of detector-tracker configurations.

Table 1 summarizes the model size, parameter count and layer count of the evaluated lightweight YOLO detectors. These specifications are reported as implementation descriptors to characterize the relative compactness of the selected detectors under the experimental framework.

Table 1: Technical Specifications of the Evaluated Lightweight YOLO Detectors

Detectors	Size (MB)	Parameters (M)	Layers
YOLOv3-tiny	23.31	12.17	82
YOLOv5n	5.31	2.65	262
YOLOv8n	6.25	3.16	225
YOLOv9t	4.74	2.13	917
YOLOv10n	5.59	2.78	385
YOLO11n	5.35	2.62	319
YOLO12n	5.34	2.60	465

Note: Detectors refer to the lightweight YOLO model evaluated in this study. Size (MB) indicates the model size in megabytes; Parameters (M) indicates the number of model parameters in millions; Layers refers to the total number of layers in each detector architecture.

Multi-object Tracking

BoT-SORT and ByteTrack were used as online tracking algorithms to associate YOLO-generated person detections across consecutive frames. ByteTrack was selected because its association strategy considers both high-confidence and low-confidence detection boxes (18). BoT-SORT was selected because its design uses motion information, camera-motion compensation and refined state estimation for association (19). Although the standard BoT-SORT framework can incorporate appearance-based re-identification, the re-identification module was disabled in this study to maintain CPU-only comparability and reduce additional computational overhead. For each YOLO detector, the same person-class detector outputs were processed independently by BoT-SORT and ByteTrack. This design supported the evaluation of tracker-level differences under equivalent detection inputs rather than different detection sources. The resulting tracker outputs consisted of frame-level bounding boxes, trajectory identifiers, identity assignments and track-management outcomes, which were used for tracking performance and identity-preservation evaluation.

Experimental Setup

The experiments were conducted in a CPU-only computing environment with an Intel Core i5-11400H processor at 2.70 GHz, 6 cores and 12 threads, 16 GB memory and Windows 11. The software environment used Python 3.10.11, PyTorch 2.1.0 in CPU mode and Ultralytics YOLO version 8.3.214. No graphics processing unit acceleration was used. The detectors were evaluated using pretrained Microsoft Common Objects in Context (MS COCO) weights without dataset-specific fine-tuning (32). This setting provided consistent detector initialization across all evaluated detectors and preserved a deployment-oriented evaluation condition.

All detectors were tested using the same inference configuration: an input size of 1280 × 1280 pixels, a confidence threshold of 0.20, an intersection-over-union (IoU) threshold of 0.50 for non-maximum suppression (NMS), no test-time augmentation and detections restricted to the person class. Commonly shared tracking thresholds were aligned where applicable across ByteTrack and BoT-SORT, with track_high_thresh = 0.25, track_low_thresh = 0.10, new_track_thresh = 0.25, match_thresh = 0.80, track_buffer = 30 and fuse_score = True. In the BoT-SORT configuration, sparse optical flow was used for camera-motion compensation, while the re-identification module was disabled to maintain a comparable CPU-only setting.

Each YOLO detector output was processed separately by ByteTrack and BoT-SORT on the selected MOT17 and MOT20 sequences. Tracking performance was evaluated using the TrackEval library. Computational efficiency was measured using frames per second (FPS) and total end-to-end runtime for detection and tracking. The evaluation followed a private-detection setting and the results were used for within-study comparison rather than direct comparison with the MOTChallenge public leaderboard.

Evaluation Metrics

Performance was evaluated using a MOTChallenge-compatible protocol implemented with the TrackEval toolkit. The evaluation covered four dimensions: detection accuracy, tracking performance, identity preservation and computational efficiency. Detection accuracy was measured using precision and recall, which quantify the correctness and completeness of predicted detections relative to ground-truth annotations. Tracking performance was assessed using Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) from the CLEAR MOT framework, while Higher Order Tracking Accuracy (HOTA) was included to capture both detection and association quality (33, 34).

Identity preservation was evaluated using identity F1 score (IDF1) and the number of identity switches (IDsw), which represent identity-assignment stability and identity-discontinuity errors across frames (35). Computational efficiency was measured using FPS and total end-to-end runtime, where runtime included detection and tracking time for each evaluated sequence. For threshold-based matching procedures, an IoU threshold of 0.50 was applied where required. These metrics were selected to capture the trade-off between detection accuracy, tracking performance, identity preservation and CPU-only processing efficiency.

Performance Disparity Analysis

A post-hoc performance disparity analysis was conducted using aggregate results from each detector-tracker configuration. For each dataset, BoT-SORT and ByteTrack were compared under matched detector settings across the seven lightweight YOLO detectors. Mean differences and percentage differences were calculated for the main tracking, identity-preservation and computational-efficiency metrics, including MOTA, HOTA, IDF1, IDsw, FPS and total runtime.

Because the number of matched detector settings was small, unadjusted two-sided Wilcoxon signed-rank tests with alpha = 0.05 were used as non-parametric matched-sample comparisons to examine whether tracker-level differences followed consistent directions across detectors (36). The p-values were interpreted together with mean and percentage differences and were not treated as standalone evidence of broad statistical generalization. This analysis was used to support within-study interpretation of tracker-level performance disparities and was not intended for direct MOTChallenge leaderboard comparison.

Qualitative Error Analysis Protocol

To support qualitative interpretation, an error analysis protocol was applied to selected tracking outputs from MOT17 and MOT20. The analysis focused on common MOT failure patterns reflected in standard MOT evaluation metrics, including missed detections, false-positive detections, fragmented trajectories, identity switches, bounding-box inconsistency and target confusion under occlusion or high crowd density (33–35). Illustrative frames were selected from sequences with visible occlusion, crowd density, identity ambiguity, or detection inconsistency to show common MOT failure patterns under different scene conditions.

Selected frames were examined to compare how detector-tracker configurations behaved in selected MOT17 and MOT20 scenes with different density and occlusion characteristics. This qualitative analysis was used to complement the quantitative metrics and to support interpretation of why certain configurations produced stronger identity preservation or higher computational efficiency. The qualitative observations were not used as a substitute for benchmark metrics, but as supporting evidence to interpret the observed tracking and identity-preservation patterns.

Table 2: Performance Comparison of Detector-tracker Configurations on MOT17

Detector	Tracker	Recall	Precision	MOTA	MOTP	HOTA	IDF1	IDsw	FPS	Runtime (s)
YOLOv3-tiny	BoT-SORT	0.512	0.766	0.341	0.776	0.394	0.471	92	4.23	180.79
YOLOv3-tiny	ByteTrack	0.494	0.770	0.329	0.770	0.369	0.441	111	4.73	161.54
YOLOv5n	BoT-SORT	0.547	0.785	0.390	0.782	0.412	0.499	73	6.82	111.48
YOLOv5n	ByteTrack	0.523	0.788	0.372	0.777	0.391	0.471	101	7.96	95.79
YOLOv8n	BoT-SORT	0.560	0.777	0.385	0.787	0.414	0.499	75	6.40	118.66
YOLOv8n	ByteTrack	0.537	0.787	0.374	0.780	0.394	0.474	100	7.62	101.39
YOLOv9t	BoT-SORT	0.565	0.771	0.387	0.789	0.426	0.513	68	5.23	145.11
YOLOv9t	ByteTrack	0.543	0.773	0.369	0.782	0.400	0.476	97	6.07	126.20
YOLOv10n	BoT-SORT	0.548	0.824	0.422	0.790	0.419	0.505	74	6.35	119.88
YOLOv10n	ByteTrack	0.527	0.828	0.405	0.784	0.397	0.476	99	7.66	100.07
YOLO11n	BoT-SORT	0.576	0.766	0.391	0.782	0.424	0.514	80	6.28	120.75
YOLO11n	ByteTrack	0.553	0.774	0.378	0.775	0.397	0.477	110	7.51	101.97
YOLO12n	BoT-SORT	0.563	0.787	0.400	0.785	0.418	0.502	71	4.43	172.87
YOLO12n	ByteTrack	0.538	0.790	0.382	0.779	0.398	0.483	89	5.10	151.63

Note: Detector and Tracker indicate the paired detector–tracker configurations evaluated in this study. MOTA = Multiple Object Tracking Accuracy; MOTP = Multiple Object Tracking Precision; HOTA = Higher Order Tracking Accuracy; IDF1 = Identity F1 Score; IDSw = identity switches; FPS = frames per second; Runtime (s) = processing time measured in seconds. Higher values are better for Recall, Precision, MOTA, MOTP, HOTA, IDF1 and FPS, whereas lower values are better for IDSw and Runtime (s).

Table 3: Performance Comparison of Detector-tracker Configurations on MOT20

Detector	Tracker	Recall	Precision	MOTA	MOTP	HOTA	IDF1	IDsw	FPS	Runtime (s)
YOLOv3-tiny	BoT-SORT	0.471	0.833	0.370	0.718	0.308	0.406	1861	3.87	593.50
YOLOv3-tiny	ByteTrack	0.461	0.832	0.361	0.719	0.296	0.391	2069	4.34	528.41
YOLOv5n	BoT-SORT	0.478	0.855	0.390	0.718	0.321	0.426	1669	6.17	371.77
YOLOv5n	ByteTrack	0.469	0.853	0.380	0.719	0.309	0.405	1839	7.19	316.74
YOLOv8n	BoT-SORT	0.508	0.840	0.405	0.723	0.336	0.441	1686	5.89	388.66
YOLOv8n	ByteTrack	0.498	0.838	0.395	0.725	0.327	0.428	1925	6.72	347.42
YOLOv9t	BoT-SORT	0.394	0.866	0.330	0.737	0.298	0.394	716	5.01	451.77
YOLOv9t	ByteTrack	0.388	0.865	0.325	0.738	0.291	0.383	806	5.55	411.29
YOLOv10n	BoT-SORT	0.420	0.896	0.369	0.734	0.315	0.428	659	5.90	387.84
YOLOv10n	ByteTrack	0.414	0.893	0.362	0.736	0.303	0.407	739	6.88	337.83
YOLO11n	BoT-SORT	0.490	0.852	0.399	0.727	0.333	0.443	1068	5.75	396.47
YOLO11n	ByteTrack	0.481	0.852	0.391	0.729	0.318	0.422	1229	6.76	341.68
YOLO12n	BoT-SORT	0.454	0.871	0.381	0.732	0.319	0.426	834	4.08	569.74
YOLO12n	ByteTrack	0.447	0.871	0.374	0.733	0.313	0.416	924	4.53	523.04

Note: Detector and Tracker indicate the paired detector–tracker configurations evaluated in this study. MOTA = Multiple Object Tracking Accuracy; MOTP = Multiple Object Tracking Precision; HOTA = Higher Order Tracking Accuracy; IDF1 = Identity F1 Score; IDSw = identity switches; FPS = frames per second; Runtime (s) = processing time measured in seconds. Higher values are better for Recall, Precision, MOTA, MOTP, HOTA, IDF1 and FPS, whereas lower values are better for IDSw and Runtime (s).

Results

The quantitative results are organized according to four evaluation dimensions: detection accuracy, tracking performance, identity preservation and computational efficiency. Table 2 presents the average performance of each detector-tracker configuration on the selected MOT17 sequences, while Table 3 reports the corresponding results on the selected MOT20 sequences. Higher values indicate better performance for recall, precision, MOTA, MOTP, HOTA, IDF1 and FPS. Lower values indicate better performance for IDsw and runtime. These results should be interpreted as a within-study comparison under a unified CPU-only private-detection setting rather than as direct MOTChallenge leaderboard results.

Performance on MOT17 and MOT20

The MOT17 results suggest that detector-tracker configurations differed across detection accuracy, tracking performance, identity preservation and computational efficiency. For detection accuracy, YOLO11n with BoT-SORT produced the highest recall, whereas YOLOv10n with ByteTrack produced the highest precision. This pattern suggests that recall and precision should be interpreted together with downstream tracking behavior because trajectory continuity is also related to how detections are associated across frames.

For tracking performance on MOT17, YOLOv10n with BoT-SORT produced the highest MOTA and MOTP, while YOLOv9t with BoT-SORT produced the highest HOTA. In terms of identity preservation, BoT-SORT showed higher IDF1 values and fewer IDsw than ByteTrack across all evaluated detectors. YOLO11n with BoT-SORT achieved the highest IDF1, while YOLOv9t with BoT-SORT produced the lowest IDsw. These results suggest that BoT-SORT provided stronger association-sensitive tracking and identity preservation on MOT17.

The MOT20 results suggest more difficult tracking conditions than MOT17. The selected MOT20 sequences contain denser crowds, heavier occlusion and more frequent target overlap. Compared with MOT17, the MOT20 results showed lower overall recall, lower HOTA and substantially higher IDsw. This pattern suggests that dense pedestrian scenes can make trajectory continuity and identity preservation more difficult, even when precision remains relatively high.

On MOT20, YOLOv8n with BoT-SORT showed the strongest tracking profile among the evaluated configurations, with the highest recall, MOTA and HOTA. YOLO11n with BoT-SORT produced the highest IDF1, whereas YOLOv10n with BoT-SORT produced the lowest IDsw and the highest precision. These findings suggest that the preferred detector-tracker configuration depends on the evaluation priority. YOLOv8n with BoT-SORT was more favorable for combined tracking performance on MOT20, while YOLO11n and YOLOv10n were more favorable for IDF1 and IDsw reduction, respectively.

Across both datasets, ByteTrack showed higher FPS and shorter runtime than BoT-SORT for every evaluated detector. Higher-throughput configurations were generally observed for YOLOv5n, YOLOv8n, YOLOv10n and YOLO11n with ByteTrack. However, this higher throughput was generally accompanied by lower identity preservation than BoT-SORT. Therefore, the results suggest a deployment trade-off: BoT-SORT is preferable when identity continuity and trajectory stability are prioritized, whereas ByteTrack is preferable when throughput and runtime efficiency are the main deployment constraints.

Overall, YOLOv8n, YOLOv10n and YOLO11n showed relatively balanced accuracy-efficiency profiles across the two datasets. YOLOv8n provided strong tracking performance on MOT20, YOLOv10n achieved high precision and strong MOTA performance and YOLO11n produced the strongest IDF1 values across datasets. These results support the interpretation that real-time MOT performance under CPU-only inference is shaped by the interaction between detector choice and tracker design rather than by detector accuracy or processing speed alone.

Tracker-level Performance Disparity Analysis

Table 4 summarizes the matched-detector performance disparities between BoT-SORT and ByteTrack across the evaluated YOLO detectors. The analysis was performed separately for MOT17 and MOT20 using aggregate configuration results. Unadjusted two-sided Wilcoxon signed-rank tests were used as supporting matched-sample comparisons. Because each dataset included only seven matched detector settings, p-values were interpreted together with mean and relative differences rather than as standalone evidence of broad statistical generalization.

Table 4: Matched-detector Performance Disparity Between BoT-SORT and ByteTrack

Dataset	Metric	BoT-SORT Mean	ByteTrack Mean	Favorable Mean Difference	Relative Difference (%)	Wilcoxon p-value	Direction
MOT17	MOTA	0.388	0.373	+0.015	4.10	0.016	BoT-SORT higher
MOT17	HOTA	0.415	0.392	+0.023	5.86	0.016	BoT-SORT higher
MOT17	IDF1	0.500	0.471	+0.029	6.22	0.016	BoT-SORT higher
MOT17	IDsw	76.14	101.00	-24.86	24.61	0.016	BoT-SORT lower
MOT17	FPS	5.68	6.66	+0.99	17.39	0.016	ByteTrack higher
MOT17	Runtime (s)	138.51	119.80	-18.71	13.51	0.016	ByteTrack lower
MOT20	MOTA	0.378	0.370	+0.008	2.16	0.016	BoT-SORT higher

MOT20	HOTA	0.319	0.308	+0.010	3.38	0.016	BoT-SORT higher
MOT20	IDF1	0.423	0.407	+0.016	3.93	0.016	BoT-SORT higher
MOT20	IDsw	1213.29	1361.57	-148.29	10.89	0.016	BoT-SORT lower
MOT20	FPS	5.24	6.00	+0.76	14.45	0.016	ByteTrack higher
MOT20	Runtime (s)	451.39	400.92	-50.48	11.18	0.016	ByteTrack lower

Note: MOTA = Multiple Object Tracking Accuracy; HOTA = Higher Order Tracking Accuracy; IDF1 = Identity F1 Score; IDSw = identity switches; FPS = frames per second; Runtime (s) = processing time measured in seconds.

On MOT17, BoT-SORT showed higher mean MOTA, HOTA and IDF1 than ByteTrack, with relative differences of 4.10%, 5.86% and 6.22%, respectively. BoT-SORT also showed 24.61% lower IDsw than ByteTrack. Conversely, ByteTrack showed 17.39% higher FPS and 13.51% lower runtime than BoT-SORT. On MOT20, a similar directional pattern was observed. BoT-SORT showed 2.16%, 3.38% and 3.93% higher MOTA, HOTA and IDF1, respectively and 10.89% lower IDsw. ByteTrack showed 14.45% higher FPS and 11.18% lower runtime. These results support the within-study interpretation that BoT-SORT was more favorable for tracking performance and identity preservation, whereas ByteTrack was more favorable for computational efficiency.

Qualitative Error Analysis

Qualitative tracking outputs were examined to support interpretation of the quantitative patterns in terms of common MOT failure modes, including missed detections, false-positive detections, fragmented trajectories, bounding-box inconsistency, target confusion and identity switches. Figure 3 presents selected ByteTrack outputs to illustrate detector-level differences in selected MOT17 and MOT20 scenes with different density and occlusion characteristics. The examples compare outputs generated using YOLOv3-tiny, YOLOv5n and YOLOv8n.

Figure 3: ByteTrack Tracking Examples on MOT17 and MOT20 Under Different Scene Conditions. (A) MOT17-05 with YOLOv3-tiny, (B) MOT17-05 with YOLOv5n, (C) MOT17-05 with YOLOv8n, (D) MOT20-05 with YOLOv3-tiny, (E) MOT20-05 with YOLOv5n, (F) MOT20-05 with YOLOv8n

The MOT17 examples in Figure 3 show relatively clearer pedestrian separation, where the tracker maintained more stable bounding boxes and more complete target coverage in the selected frames. In contrast, the selected MOT20 examples show denser pedestrian interactions, heavier overlap between targets and more partially visible pedestrians. Under these selected conditions, missed detections, bounding-box inconsistency and fragmented trajectories were more visible.

The qualitative comparison in Figure 3 suggests that detector output behavior was associated with the completeness of tracked targets. In the selected crowded scenes, YOLOv3-tiny showed less stable target coverage, which is consistent with its lower quantitative tracking and identity-preservation metrics. YOLOv5n showed higher computational efficiency but lower identity-preservation performance than the relatively balanced recent detectors. In the illustrated frames, YOLOv8n showed more complete detections and more stable tracking outputs than YOLOv3-tiny and YOLOv5n, which is consistent with its relatively strong trade-off profile in the quantitative results. These visual patterns suggest that tracking failures were associated not only with tracker design but also with detector output behavior under occlusion and crowd density. Figure 4 compares selected BoT-SORT and ByteTrack outputs using YOLO11n to illustrate tracker-level differences in identity preservation under selected visually ambiguous conditions.

Figure 4: Identity Preservation Comparison Using YOLO11n on MOT17-10. (A-C) BoT-SORT Output, (D-F) ByteTrack Output

In the selected frames shown in Figure 4, the BoT-SORT output shows more stable identity continuity, whereas the ByteTrack output shows greater identity ambiguity when visually similar targets interact or pass through partially occluded regions. This pattern is consistent with the quantitative results, where BoT-SORT showed higher IDF1 and HOTA values and fewer IDsw, while ByteTrack showed higher FPS and shorter runtime.

Overall, the qualitative error analysis provides illustrative support for the quantitative results. In the selected MOT17 scenes, tracking outputs were generally more stable when pedestrians were more separable and severe overlap was less frequent. In the selected MOT20 scenes, tracking failures were more frequent under occlusion, overlapping pedestrians and target reappearance, which increased association ambiguity. Missed detections and false-positive detections were associated with trajectory fragmentation, while target confusion under occlusion was associated with a higher likelihood of IDsw. These error patterns help interpret why BoT-SORT showed higher IDF1 and HOTA values and lower IDsw, whereas ByteTrack showed higher FPS and shorter runtime. Therefore, the results support the interpretation that real-time MOT performance under CPU-only inference should be understood as a multidimensional trade-off among detection accuracy, tracking performance, identity preservation and computational efficiency.

Discussion

The results indicate that CPU-only multi-object tracking performance is governed by a multidimensional trade-off rather than by a single universally superior detector-tracker configuration. This interpretation is consistent with the broader MOT literature, which emphasizes that tracking performance depends on the interaction among detection quality, data association, identity consistency and occlusion handling (1). Across MOT17 and MOT20, no configuration achieved the best performance across all evaluation dimensions. BoT-SORT generally provided stronger tracking performance and identity preservation, as reflected by higher MOTA, HOTA and IDF1 values and lower IDsw, whereas ByteTrack consistently achieved higher FPS and shorter runtime. The matched-detector disparity analysis further supports this pattern by showing consistent directional differences between the two trackers across the evaluated YOLO detectors. Rather than indicating a single optimal configuration, these findings show that each detector-tracker combination reflects a different balance between detection completeness, association stability, identity preservation and computational efficiency. This interpretation is also aligned with HOTA, which was introduced to evaluate tracking performance by jointly consider-ing detection, association and localization quality (34).

These findings extend previous YOLO-based MOT studies by providing controlled CPU-only evidence across multiple lightweight YOLO detectors and two online trackers under the same benchmark sequences, inference setting and private-detection protocol. Previous YOLO-based pedestrian tracking research reported that improvements in detector design can support more stable tracking outcomes, although the evaluation was centered on a specific detector-tracker configuration (21). In traffic-oriented tracking, YOLO-based detector-tracker pipelines have also been shown to support practical real-time tracking performance, but the reported outcomes were specific to vehicle scenarios and did not examine multiple lightweight detector families under a unified CPU-only setting (25). Edge-device evaluations further indicate that deployment feasibility depends not only on detection accuracy but also on device-level runtime constraints, which is consistent with the present finding that stronger tracking performance and higher throughput did not necessarily occur in the same configuration (30). Compared with these prior outcomes, the present study provides a more controlled comparison by evaluating seven lightweight YOLO detectors and two online trackers using the same experimental protocol. This design allows the observed differences in MOTA, HOTA, IDF1, IDsw, FPS and runtime to be interpreted more directly as detector-tracker trade-offs rather than as effects of different datasets, application domains, or hardware settings.

The detection results also show that frame-level detection accuracy alone is insufficient to determine tracking performance. On MOT20, precision values were generally higher than on MOT17, whereas recall values were lower. This pattern may reflect a more conservative retained-detection pattern under denser and more occluded scenes, which is consistent with the crowded-scene characteristics of MOT20 (32). Although higher precision can reduce false-positive detections, lower recall may increase missed targets and interrupt trajectory continuity. This helps explain why configurations with the highest precision did not always produce the strongest tracking outcomes. For example, YOLOv10n achieved the highest precision on MOT20, while YOLOv8n with BoT-SORT produced the strongest MOTA-HOTA profile on the same dataset. This result indicates that strong tracking performance requires not only accurate localization but also sufficient temporal continuity to support stable association across frames. This interpretation is consistent with the central role of data association in multi-object tracking performance (7).

The performance contrast between BoT-SORT and ByteTrack reflects their different tracker design priorities. ByteTrack associates both high-confidence and low-confidence detection boxes to recover trajectories that might otherwise be lost (18). In the present CPU-only evaluation, ByteTrack was observed to achieve higher throughput and shorter runtime across all evaluated detectors. BoT-SORT, in contrast, incorporates motion information, camera-motion compensation and refined state estimation, which may support more stable association and identity preservation while introducing additional computational cost (19). Because the re-identification module in BoT-SORT was disabled to maintain CPU-only comparability, the observed advantage of BoT-SORT in identity preservation was obtained without additional appearance-embedding computation. Therefore, BoT-SORT's motion-based association and camera-motion compensation mechanisms may be associated with its stronger identity-preservation performance in this setting, although further ablation analysis would be required to isolate the contribution of each component.

The qualitative error analysis provides illustrative support for the quantitative findings. The selected MOT17 examples showed more stable tracking outputs when pedestrians were more visually separable, whereas the selected MOT20 examples showed more frequent missed detections, incomplete boxes, fragmented trajectories and identity ambiguity under dense crowd conditions. These observations are consistent with the characteristics of MOT20 as a crowded-scene benchmark with heavier occlusion and more frequent target overlap (32). The qualitative comparison also suggests that detector output behavior contributes to downstream tracking stability. In the selected frames, incomplete detector outputs were associated with missed targets and fragmented trajectories, whereas more complete detector outputs provided more stable inputs for association. The tracker-level comparison further illustrated that BoT-SORT maintained more stable identity continuity than ByteTrack in selected visually ambiguous frames. These observations should be interpreted as illustrative evidence rather than as a substitute for benchmark metrics. Nevertheless, they help explain the reduction in HOTA under denser conditions, since HOTA evaluates detection, association and localization components jointly (34). They also support the observed differences in IDF1 and IDsw, because IDF1 is designed to evaluate identity preservation in multi-target tracking (35).

The results further suggest that model recency alone should not be used as the primary criterion for deployment selection. YOLO architectures have undergone continuous development and previous reviews have shown that YOLO-based detectors are widely used because they offer a practical balance between detection accuracy and inference speed (13). However, practical CPU-only MOT performance is still determined by the interaction among model structure, input processing, detector output behavior and tracker complexity. In the evaluated setting, YOLOv5n, YOLOv8n, YOLOv10n and YOLO11n generally achieved higher throughput than YOLOv3-tiny and YOLO12n, but higher throughput did not automatically correspond to better identity preservation or stronger tracking performance. When detection, tracking, identity and efficiency metrics were considered together, YOLOv8n, YOLOv10n and YOLO11n showed relatively balanced profiles among the evaluated detectors. These findings indicate that candidate detector-tracker configurations should be evaluated empirically on the intended hardware and expected scene conditions before deployment, particularly because resource-constrained vision systems require a practical balance between accuracy, inference efficiency and implementation feasibility (10).

From a deployment perspective, the findings provide practical guidance for CPU-limited MOT systems. BoT-SORT-based configurations are more appropriate when identity continuity, trajectory stability and reduced IDsw are prioritized, such as in crowded surveillance, pedestrian monitoring, post-event trajectory analysis and applications where stable identity assignment supports downstream behavioral interpretation. ByteTrack-based configurations are more suitable when real-time responsiveness, shorter runtime and higher FPS are the main operational constraints, such as low-latency monitoring, embedded vision and real-time alerting systems. For balanced CPU-only deployment, YOLOv8n, YOLOv10n and YOLO11n appear to be the most promising detector candidates under the evaluated setting, as they combine competitive tracking performance and identity preservation with relatively efficient runtime. However, the final configuration should be selected according to the specific deployment priority: identity continuity favors BoT-SORT, whereas throughput and runtime efficiency favor ByteTrack.

Several methodological limitations should be acknowledged. First, the results represent a within-study comparison under a unified private-detection protocol and should not be interpreted as direct MOTChallenge leaderboard results. Second, the experiments were conducted on a single CPU-only hardware configuration, which may limit generalization to other processors, embedded boards, GPU-based systems, or accelerator-based deployments. Third, the evaluated YOLO detectors used pretrained Microsoft COCO weights without dataset-specific fine-tuning, so the reported results reflect deployment-oriented transfer performance rather than optimized benchmark-specific performance (32). Fourth, the re-identification module was disabled to preserve CPU-only comparability and reduce computational overhead, although this choice may limit the identity-preservation potential of trackers that benefit from appearance embeddings. Fifth, because the evaluation focused on pedestrian MOT using MOT17 and MOT20, the findings may not directly generalize to non-pedestrian domains such as vehicles, animals, maritime objects, or aerial targets. Sixth, memory usage, power consumption, energy efficiency, thermal behavior and hardware-specific acceleration were not measured. Finally, the qualitative error analysis was used only as supporting evidence for interpreting benchmark metrics and should not be treated as a substitute for quantitative evaluation.

Future work should examine whether the observed detector-tracker trade-offs remain consistent across different CPU architectures, embedded devices and edge accelerators. Further studies should also evaluate optimized deployment formats, including quantized models, pruned detectors, Open Neural Network Exchange conversion, OpenVINO acceleration, TensorRT deployment and other model-compression strategies. Dataset-specific fine-tuning and re-identification-enabled tracker configurations should be examined to assess whether identity preservation can be improved without substantial increases in runtime. In addition, future work should include adaptive threshold optimization, memory and power profiling and evaluation across non-pedestrian tracking domains to determine the broader generalizability of these findings in deployment-oriented MOT scenarios.

Conclusion

This study evaluated fourteen detector-tracker configurations using seven lightweight YOLO-based object detectors, namely YOLOv3-tiny, YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n and YOLO12n, with BoT-SORT and ByteTrack under a unified CPU-only private-detection protocol on MOT17 and MOT20. The results show that no single configuration dominated all evaluation dimensions. BoT-SORT generally showed stronger tracking performance and identity preservation, as indicated by higher MOTA, HOTA and IDF1 values and lower IDsw, whereas ByteTrack showed higher FPS and shorter runtime. Among the evaluated detectors, YOLOv8n, YOLOv10n and YOLO11n showed relatively balanced accuracy-efficiency profiles across datasets and metrics. These findings support the interpretation that CPU-only MOT performance is shaped by the interaction between detector choice and tracker design rather than by detector accuracy or runtime efficiency alone. From a practical perspective, BoT-SORT-based configurations may be more suitable when identity continuity and trajectory stability are prioritized, while ByteTrack-based configurations may be more appropriate for throughput-oriented applications. Therefore, configuration selection should be guided by deployment priorities, particularly the trade-off between identity preservation and processing speed. However, this study is limited by its CPU-only evaluation setting, use of pretrained COCO weights without dataset-specific fine-tuning, disabled re-identification module and focus on pedestrian MOT using MOT17 and MOT20. Therefore, the findings should be interpreted as within-study evidence under a unified private-detection protocol rather than as direct MOTChallenge leaderboard results. Future work should examine whether these trade-offs are consistent across different edge devices, optimized model formats, re-identification-enabled configurations, hardware acceleration settings and non-pedestrian tracking domains.

Abbreviations

BoT-SORT: Bag of Tricks for Simple Online and Realtime Tracking, CPU: central processing unit, FPS: frames per second, HOTA: Higher Order Tracking Accuracy, IDF1: identity F1 score, IDsw: identity switches, MOT: multi-object tracking, MOTA: Multiple Object Tracking Accuracy, MOTP: Multiple Object Tracking Precision, YOLO: You Only Look Once.

Acknowledgement

None.

Author Contributions

Feriantano Sundang Pranata: Conceptualization, Methodology, Software, Formal Analysis, Writing – Original Draft, Jufriadif Na'am: Methodology, Supervision, Validation and Writing – Review & Editing, Yulhan: Software, Validation, Yuke Permata Lisna: Investigation, Visualization, Rima Agustia Utami: Resources, Validation, Risma Rahmatunisa: Data Curation, Project Administration, Indra Saputra: Formal Analysis, Writing – Review, Editing, Elviza Yeni Putri: Writing – Review, Editing, Melda Mahniza: Writing – Review, Editing, Nurul Inayah Hutasuhut: Writing –Review, Editing. All authors have read and approved the final manuscript.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability

The datasets analyzed in this study are publicly available in the MOTChallenge repository at https://motchallenge.net.

Declaration of Artificial Intelligence (AI) Assistance

The authors declare that generative AI and AI-assisted technologies were used only to improve the clarity, grammar and language quality of the manuscript. The research concept, experimental design, data processing, analysis, interpretation and conclusions are the authors' original work. The authors reviewed, edited and verified all AI-assisted outputs and take full responsibility for the accuracy, integrity and content of the manuscript.

Ethics Approval

Not Applicable.

Funding

This research received no external funding.

References

1. Luo W, Xing J, Milan A, *et al.* Multiple object tracking: a literature review. Artificial Intelligence. 2021; 293:103448. doi: 10.1016/j.artint.2020.103448
2. Ciaparrone G, Luque Sánchez F, Tabik S, *et al.* Deep learning in video multi-object tracking: a survey. Neurocomputing. 2020; 381:61–88. doi: 10.1016/j.neucom.2019.11.023
3. Jiménez-Bravo DM, Lozano Murciego Á, Sales Mendes A, *et al.* multi-object tracking in traffic environments: a systematic literature review. Neurocomputing.2022; 494:43–55. doi: 10.1016/j.neucom.2022.04.087
4. Hassan S, Mujtaba G, Rajput A, *et al.* multi-object tracking: a systematic literature review. Multimedia Tools and Applications. 2024;83(14):43439–43492. doi:10.1007/s11042-023-17297-3
5. Du C, Lin C, Jin R, *et al.* Exploring the state-of-the-art in multi-object tracking: a comprehensive survey, evaluation, challenges and future directions. Multimedia Tools and Applications. 2024;83(29):73151–73189. doi:10.1007/s11042-023-17983-2
6. Guan Z, Wang Z, Zhang G, *et al.* Multi-object tracking review: retrospective and emerging trend. Artificial Intelligence Review. 2025;58(8):235. doi:10.1007/s10462-025-11212-y
7. Rakai L, Song H, Sun S, *et al.* Data association in multiple object tracking: a survey of recent techniques. Expert Systems with Applications. 2022;192:116300. doi: 10.1016/j.eswa.2021.116300
8. Ma LV, Nguyen TTD, Shim C, *et al.* Visual multi-object tracking with re-identification and occlusion handling using labeled random finite sets. Pattern Recognition. 2024;156:110785. doi: 10.1016/j.patcog.2024.110785
9. Agrawal H, Halder A, Chattopadhyay P. A systematic survey on recent deep learning-based approaches to multi-object tracking. Multimedia Tools and Applications. 2024;83(12):36203–36259. doi:10.1007/s11042-023-16910-9
10. Kamath V, Renuka A. Deep learning-based object detection for resource-constrained devices: systematic review, future trends and challenges ahead. Neurocomputing. 2023; 531:34–60. doi: 10.1016/j.neucom.2023.02.006
11. Park J, Hong J, Shim W, *et al.* multi-object tracking on SWIR images for city surveillance in an edge-computing environment. Sensors. 2023;23(14):6373. doi:10.3390/s23146373
12. Park Y, Dang LM, Lee S, *et al.* Multiple objects tracking in deep learning approaches: a survey. Electronics. 2021;10(19):2406. doi:10.3390/electronics10192406
13. Terven J, Córdova-Esparza DM, Romero-González JA. A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 2023;5(4):1680–1716. doi:10.3390/make5040083
14. Ali ML, Zhang Z. The YOLO framework: a comprehensive review of evolution, applications and benchmarks in object detection. Computers. 2024;13(12):336. doi:10.3390/computers13120336
15. Ramos LT, Sappa AD. A decade of You Only Look Once (YOLO) for object detection: a review. IEEE Access. 2025;13:192747–192794. doi:10.1109/ACCESS.2025.3630988
16. Murat AA, Kiran MS. A comprehensive review on YOLO versions for object detection. Engineering Science and Technology, an International Journal. 2025; 70:102161. doi: 10.1016/j.jestch.2025.102161
17. Kim J, Kim G, Yoshitoshi R, *et al.* Real-time object detection for edge computing-based agricultural automation: a case study comparing the YOLOX and YOLOv12 architectures and their performance in potato harvesting systems.Sensors. 2025;25(15):4586. doi:10.3390/s25154586
18. You L, Chen Y, Xiao C, *et al.* multi-object vehicle detection and tracking algorithm based on improved YOLOv8 and ByteTrack. Electronics. 2024;13(15):3033. doi:10.3390/electronics13153033
19. Xing B, Sun M, Liu Z, et al. Sonar fish school detection and counting method based on improved YOLOv8 and BoT-SORT. Journal of Marine Science and Engineering. 2024;12(6):964. doi:10.3390/jmse12060964
20. Razzok M, Badri A, El Mourabit I, *et al.* Pedestrian detection and tracking system based on Deep-SORT, YOLOv5 and new data association metrics. Information. 2023;14(4):218. doi:10.3390/info14040218
21. Xiao X, Feng X. Multi-object pedestrian tracking using improved YOLOv8 and OC-SORT. Sensors. 2023;23(20):8439. doi:10.3390/s23208439
22. Sheng W, Shen J, Huang Q, *et al.* multi-objective pedestrian tracking method based on YOLOv8 and improved DeepSORT. Mathematical Biosciences and Engineering. 2024;21(2):1791–1805. doi:10.3934/mbe.2024077

23. Lin L, He H, Xu Z, *et al.* Realtime vehicle tracking method based on YOLOv5 + DeepSORT. Computational Intelligence and Neuroscience. 2023; 2023:7974201. doi:10.1155/2023/7974201
24. Kumar S, Singh SK, Varshney S, *et al.* Fusion of Deep SORT and YOLOv5 for effective vehicle detection and tracking scheme in real-time traffic management sustainable system. Sustainability. 2023;15(24):16869. doi:10.3390/su152416869
25. Wang Y, Mariano VY. A multi-object tracking framework based on YOLOv8s and ByteTrack algorithm. IEEE Access. 2024;12:120711–120719. doi:10.1109/ACCESS.2024.3450370
26. Zhu J, Ma C, Rong J, *et al.* Bird and UAVs recognition detection and tracking based on improved YOLOv9-DeepSORT. IEEE Access. 2024; 12:147942–147957. doi:10.1109/ACCESS.2024.3475629
27. Zheng Z, Li J, Qin L. YOLO-BYTE: an efficient multi-object tracking algorithm for automatic monitoring of dairy cows. Computers and Electronics in Agriculture. 2023;209:107857. doi: 10.1016/j.compag.2023.107857
28. Xu X, Hu J, Yang J, *et al.* A fish detection and tracking method based on improved interframe difference and YOLO-CTS. IEEE Transactions on Instrumentation and Measurement. 2024;73:1–13. doi:10.1109/TIM.2024.3476529
29. Yasir M, Shanwei L, Mingming X, *et al.* YOLOv8-BYTE: ship tracking algorithm using short-time sequence SAR images for disaster response leveraging GeoAI. International Journal of Applied Earth Observation and Geoinformation. 2024;128:103771. doi: 10.1016/j.jag.2024.103771
30. Azevedo P, Santos V. Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices. Robotics and Autonomous Systems. 2024;171:104558. doi: 10.1016/j.robot.2023.104558
31. Dendorfer P, Ošep A, Milan A, *et al.* MOTChallenge: a benchmark for single-camera multiple target tracking. International Journal of Computer Vision. 2021;129(4):845–881. doi:10.1007/s11263-020-01393-0
32. Liu L, Ouyang W, Wang X, *et al.* Deep learning for generic object detection: a survey. International Journal of Computer Vision. 2020;128(2):261–318. doi:10.1007/s11263-019-01247-4
33. Bernardin K, Stiefelhagen R. Evaluating multiple objects tracking performance: the CLEAR MOT metrics. EURASIP Journal on Image and Video Processing. 2008;2008:246309. doi:10.1155/2008/246309
34. Luiten J, Ošep A, Dendorfer P, *et al.* HOTA: a higher order metric for evaluating multi-object tracking. International Journal of Computer Vision. 2021;129(2):548–578. doi:10.1007/s11263-020-01375-2
35. Zhang Y, Wang C, Wang X, *et al.* FairMOT: on the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision. 2021;129(11):3069–3087. doi:10.1007/s11263-021-01513-4
36. Wilcoxon F. Individual comparisons by ranking methods. Biometrics Bulletin. 1945;1(6):80–83. doi:10.2307/3001968

How to Cite: Pranata FS, Na'am J, Yulhan, Lisna YP, Utami RA, Rahmatunisa R, Saputra I, Putri EY, Mahniza M, Hutasuhut NI. A Comparative Evaluation of Lightweight YOLO Detectors with BoT-SORT and ByteTrack for Real-time Multi-object Tracking. Int Res J Multidiscip Scope. 2026;7(3):1348-1362.
DOI: 10.47857/irjms.2026.v07i03.011543