





## Article

# Enhancing Pedestrian Tracking in Autonomous Vehicles by Using Advanced Deep Learning Techniques

Majdi Sukkar <sup>1</sup>, Madhu Shukla <sup>2</sup>, Dinesh Kumar <sup>3</sup>, Vassilis C. Gerogiannis <sup>4,\*</sup>, Andreas Kanavos <sup>5</sup>  
and Biswaranjan Acharya <sup>2,\*</sup>

<sup>1</sup> Department of Computer Engineering, Marwadi University, Rajkot 360003, Gujarat, India; majdi.sukkar106169@marwadiuniversity.ac.in

<sup>2</sup> Department of Computer Engineering-AI and BDA, Marwadi University, Rajkot 360003, Gujarat, India; madhu.shukla@marwadieducation.edu.in

<sup>3</sup> School of Computer Science Engineering and Technology, Bennett University, Greater Noida 201310, Uttar Pradesh, India; dinesh.kumar2@bennett.edu

<sup>4</sup> Department of Digital Systems, University of Thessaly, 41500 Larissa, Greece

<sup>5</sup> Department of Informatics, Ionian University, 49100 Corfu, Greece; akanavos@ionio.gr

\* Correspondence: vgerogian@uth.gr (V.C.G.); biswaranjan.acharya@marwadieducation.edu.in (B.A.)

**Abstract:** Effective collision risk reduction in autonomous vehicles relies on robust and straight-forward pedestrian tracking. Challenges posed by occlusion and switching scenarios significantly impede the reliability of pedestrian tracking. In the current study, we strive to enhance the reliability and also the efficacy of pedestrian tracking in complex scenarios. Particularly, we introduce a new pedestrian tracking algorithm that leverages both the YOLOv8 (You Only Look Once) object detector technique and the StrongSORT algorithm, which is an advanced deep learning multi-object tracking (MOT) method. Our findings demonstrate that StrongSORT, an enhanced version of the DeepSORT MOT algorithm, substantially improves tracking accuracy through meticulous hyperparameter tuning. Overall, the experimental results reveal that the proposed algorithm is an effective and efficient method for pedestrian tracking, particularly in complex scenarios encountered in the MOT16 and MOT17 datasets. The combined use of YOLOv8 and StrongSORT contributes to enhanced tracking results, emphasizing the synergistic relationship between detection and tracking modules.

**Keywords:** pedestrian tracking; object detection; multi-object tracking (MOT); autonomous vehicles; deep learning; DeepSORT; StrongSORT; You Only Look Once (YOLO) object detection algorithm



**Citation:** Sukkar, M.; Shukla, M.; Kumar, D.; Gerogiannis, V.C.; Kanavos, A.; Acharya, B. Enhancing Pedestrian Tracking in Autonomous Vehicles by Using Advanced Deep Learning Techniques. *Information* **2024**, *15*, 104. <https://doi.org/10.3390/info15020104>

Academic Editor: Zahir M. Hussain

Received: 2 January 2024

Revised: 28 January 2024

Accepted: 6 February 2024

Published: 9 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, the significance of pedestrian tracking in autonomous vehicles has garnered considerable attention due to its pivotal role in ensuring pedestrian safety. Existing state-of-the-art approaches for pedestrian tracking in autonomous vehicles predominantly rely on object detection and tracking algorithms, including Faster R-CNN (Region-based Convolutional Neural Network), YOLO (You Only Look Once) and SORT algorithms [1]. However, these methods encounter issues in scenarios where pedestrians are occluded or only partially visible [2]. These challenges serve as the impetus for our research, as we strive to enhance the reliability and effectiveness of pedestrian tracking, particularly in complex scenarios.

More specifically, the challenges in pedestrian tracking are multifaceted, ranging from crowded urban environments to unpredictable pedestrian behavior [3]. Existing algorithms often struggle to handle scenarios where individuals move behind obstacles, cross paths, or exhibit sudden changes in direction. Furthermore, adverse weather conditions, low lighting, and dynamic urban landscapes pose additional hurdles for accurate pedestrian tracking [4]. These issues underscore the need for advanced pedestrian tracking solutions that can adapt to diverse and complex real-world scenarios.

In response to these challenges, our research takes inspiration from recent advancements in deep learning and object detection. Deep learning techniques, with their ability to learn intricate patterns and representations from data, have shown promise in overcoming the limitations of traditional tracking methods. Leveraging the capabilities of the YOLOv8 algorithm for object detection, our approach aims to enhance the accuracy and robustness of pedestrian tracking in dynamic and challenging environments [5]. By addressing the limitations of existing algorithms, we aspire to contribute to the development of pedestrian tracking systems that are both reliable and adaptable.

The quest for improved pedestrian tracking is not solely confined to the domain of autonomous vehicles [6]. The relevant applications extend to various fields, such as surveillance, crowd management, and human–computer interaction. Accurate pedestrian tracking is crucial for ensuring public safety, optimizing traffic flow, and enhancing the overall efficiency of smart city initiatives [7,8]. The effectiveness of pedestrian tracking is also integral for a myriad of applications in the context of urban environments, and in smart cities, where the integration of technology aims to enhance the quality of life, with pedestrian tracking playing a crucial role. Efficient tracking systems can contribute to optimized traffic management, improved public safety, and enhanced urban planning [9]. Beyond traffic applications, pedestrian tracking finds applications in surveillance, where monitoring and analyzing pedestrian movement are essential for security [10]. Additionally, in human–computer interaction scenarios, accurate tracking is pivotal for creating responsive and adaptive interfaces, offering a wide array of possibilities for innovative applications [11]. Therefore, the advancements in pedestrian tracking have far-reaching implications, influencing various aspects of our daily lives and the development of smart city ecosystems.

While existing algorithms have made strides in pedestrian tracking [12], the demand for more robust and adaptable solutions remains. Our research addresses this demand by leveraging advanced deep learning techniques and integrating them into a unified framework, StrongSORT. By amalgamating the strengths of state-of-the-art tracking algorithms and deep learning methods, we aim to overcome the challenges posed by occlusion, varied pedestrian behavior, and complex urban environments. Through meticulous fine-tuning and integration with YOLOv8, our proposed approach seeks to push the boundaries of pedestrian tracking accuracy. Particularly, the proposed method leverages a genetic algorithm to automatically improve tracking parameter settings, and it also utilizes manual tuning for high-impact parameters to achieve better performance in pedestrian tracking.

The outcomes of our research contribute not only to the field of autonomous vehicles but also to broader applications that rely on precise and efficient pedestrian tracking. In particular, we propose a pedestrian tracking algorithm built upon the StrongSORT framework. This methodology amalgamates the strengths of both SORT and the deep learning techniques employed in DeepSORT, resulting in a robust solution for pedestrian tracking, especially in challenging scenarios [13]. The urgency to develop improved pedestrian tracking methods that seamlessly operate in real-world situations—commonly fraught with obstacles and partial visibility—underlies the motivation for this research [14].

The structure of the paper is as follows: In Section 2, we provide an overview of related work in pedestrian tracking. Section 3 outlines our proposed methodology, detailing the StrongSORT algorithm and the integration with YOLOv8 for enhanced object detection. Section 4 presents the experimental results and discussion, highlighting the performance improvements achieved. Finally, in Section 5, we draw conclusions from our findings and discuss avenues for future work.

## 2. Related Work

Recent advancements in pedestrian tracking encompass a diverse array of methods and algorithms, broadly classified into the following categories:

1. Multi-Object Tracking (MOT) Methods:

- These methods are designed to concurrently track multiple pedestrians within a scene. Traditional approaches often employ the Hungarian algorithm for association, linking detections across frames [5].
  - Recent advancements leverage deep neural networks, such as TrackletNet [15] and DeepSORT, to learn features and association scores, enhancing tracking accuracy.
2. Re-identification-based Methods:
    - This category integrates appearance-based and geometric feature-based techniques to identify and track pedestrians across different camera views. Notably, Siamese networks are employed to learn a similarity metric between pairs of images [16].
    - Contemporary methods incorporate attention mechanisms to emphasize discriminative regions of pedestrians in order to improve tracking precision [17].
  3. Online Multi-Object Tracking: Online tracking methods dynamically track multiple pedestrians in real-time. Techniques include deep learning-based re-identification [18] and online multiple hypothesis tracking [19].
  4. Multi-Cue Fusion: Strategies in this category amalgamate various cues, such as color, shape, and motion, to enhance tracking robustness in complex scenes. For instance, the multi-cue multi-camera pedestrian tracking (MCMC-PT) method integrates color, shape, and motion cues from multiple cameras for comprehensive pedestrian tracking [18].

This diverse landscape of pedestrian tracking methodologies underscores the ongoing efforts to address challenges in occlusion, partial visibility, and dynamic scenarios. Each approach brings its unique strengths, contributing to the advancement of pedestrian tracking in autonomous vehicle applications. The subsequent sections of this paper delve into the differentiation between algorithms, detailed technology introductions, and present the proposed StrongSORT algorithm's effectiveness in addressing these challenges.

In the research presented in [20], three frameworks for multi-object tracking were evaluated: tracking-by-detection (TBD), joint-detection-and-tracking (JDT), and a transformer-based tracking method. DeepSORT and StrongSORT were classified under the TBD framework. The front-end detector's performance significantly impacts tracking, and enhancing it is crucial. The transformer-based framework excels in MOTA (multiple object tracking accuracy) but has a large model size, while the JDT framework balances accuracy and real-time performance.

In [21], an occlusion handling strategy for a multi-pedestrian tracker is proposed, capable of retrieving targets without the need for re-identification models. The tracker can manage inactive tracks and cope with tracks leaving the camera's field of view, achieving state-of-the-art results on three popular benchmarks. However, the performed comparison does not include DeepSORT or StrongSORT algorithms.

The authors of [13] propose a framework model based on YOLOv5 and StrongSORT tracking algorithms for real-time monitoring and tracking of workers wearing safety helmets in construction scenarios. The use of deep learning-based object detection and tracking algorithms improves accuracy and efficiency in helmet-wearing detection. The study suggests that changing the box regression loss function from CIOU to Focal-EIOU can further improve detection performance. However, the study's limitation is its evaluation on a specific dataset, potentially limiting generalization to other datasets or scenarios.

The VOT2020 challenge assessed different tracking scenarios, introducing innovations like using segmentation masks instead of bounding boxes and new evaluation methods. Most trackers relied on deep learning, particularly Siamese networks, emphasizing the role of AI and deep learning in advancing object tracking for future improvements [22]. In [23], the utilization of the YOLOv5 model and the StrongSORT algorithm for ship detection, classification, and tracking in maritime surveillance systems is examined. The practical results demonstrate high accuracy in ship classification and the capability to track at a speed

approaching real-time. The influence of StrongSORT contributes to enhancing tracking speed, confirming its effectiveness in maritime surveillance systems.

In [24], fine-tuning plays a crucial role in adapting the pre-trained YOLOv5 model for brain tumor detection, significantly enhancing the model's performance in identifying specific brain tumors within radiological images. Thus, this study provides a valuable tool for medical image analysis. In another study [25], a direct comparison with the performance of StrongSORT in its advanced version was conspicuously omitted. While utilizing the KC-YOLO approach based on YOLOv5 in their detection algorithm, the study did not surpass the capabilities of more recent versions like YOLOv8. Introducing newer updates could potentially enhance StrongSORT++, as illustrated in [4], showcasing its substantial outperformance across various metrics, including HOTA and IDF1, on the MOT17 and MOT20 datasets.

The impact of fine-tuning is evident in a machine learning study [26], where deep learning models identified diseases in maize leaves. Fine-tuning significantly improved the performance of pre-trained models, resulting in disease classification accuracy rates exceeding 93%. VGG16, InceptionV3, and Xception achieved accuracy rates surpassing 99%, demonstrating the effectiveness of transfer learning, as previously shown in [27], and the positive impact of fine-tuning on disease detection in maize leaves [26].

In Table 1, a comparison is presented between StrongSORT and other multi-object tracking methods with a focus on the advantages and disadvantages of its method.

**Table 1.** Comparison overview between StrongSORT and other multi-object trackers.

Method	Advantages	Disadvantages
StrongSORT [4]	<ul style="list-style-type: none"> <li>Simple and efficient</li> <li>Robustness to occlusions</li> <li>Robustness to identity switches</li> </ul>	<ul style="list-style-type: none"> <li>Runs relatively slowly</li> <li>Negligible extra computational cost</li> </ul>
DeepSORT [28]	Simple and efficient	Limited robustness to occlusions and identity switches
TrackletNet [15]	Robust to occlusions and identity switches	Complex architecture and not real-time
DMAN [29]	Robust to occlusions and identity switches	Not real-time
SORT [30]	Simple and efficient	Limited robustness to occlusions and identity switches
ATOM [31]	Accurate tracking	Complex architecture and not real-time
IVDM [25]	Enhanced handling of occlusions and identity switches	Needs evaluation against real-time performance

Relevant studies affirm the positive impact of YOLOv5 on tracking, aligning seamlessly with our detection-based tracking algorithm. The general aim of our research is to leverage the enhanced capabilities of the YOLOv8 version. The ongoing research underscores StrongSORT's inherent advantages—simplicity, efficiency, and proficiency in effectively managing occlusions and identity switches. Our results further showcase StrongSORT's suitability across various metrics, and particularly in the MOTA metric.

In recognizing the identified limitations within StrongSORT, such as a slightly slower runtime in specific cases, it becomes imperative to strike a balance in utilizing its features while simultaneously addressing challenges in diverse implementation environments. Future research endeavors will delve into a meticulous examination of the algorithm's efficiency concerning occlusions and identity switches. Our ongoing efforts are dedicated to enhancing these aspects, with a particular emphasis on improving performance in pedestrian scenarios.

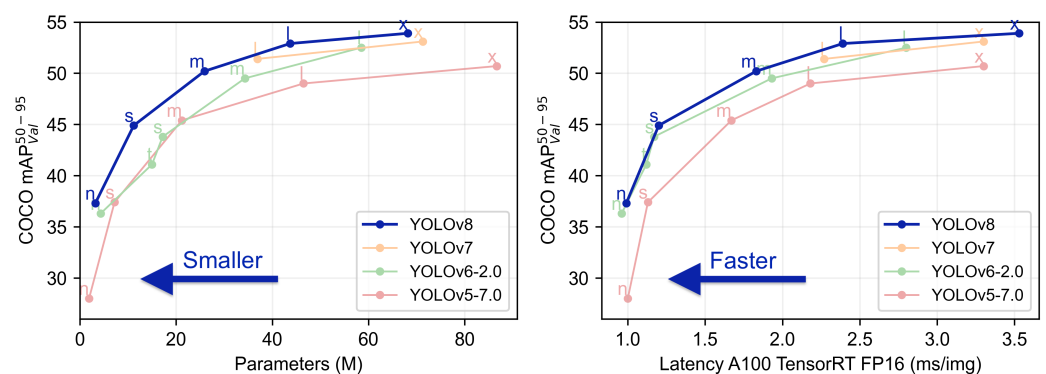


## 2.1. YOLOv8

YOLOv8 (You Only Look Once version 8) represents a significant advancement over its predecessors, YOLOv7 and YOLOv6, incorporating multiple features that enhance both speed and accuracy. A notable addition is the spatial pyramid pooling (SPP) module, enabling YOLOv8 to extract features at varying scales and resolutions [32]. This facilitates the precise detection of objects of different sizes. Another key feature is the cross stage partial network (CSP) block, reducing the network's parameters without compromising accuracy, thereby improving both training times and overall performance [33].

Evaluation of prominent object detection benchmarks, including COCO and Pascal VOC datasets, showcases YOLOv8's exceptional capabilities. The COCO dataset, with 80 object categories and complex scenes, saw YOLOv8 achieving the highest-ever mean average precision (mAP) score of 55.3% among single-stage object detection algorithms. Additionally, it achieved a remarkable real-time speed of 58 frames per second on an NVIDIA GTX 1080 Ti GPU. In summary, YOLOv8 stands out as an impressive object detection algorithm, delivering high accuracy and real-time performance. Its ability to process the entire image simultaneously makes it well suited for applications such as autonomous driving and robotics. With its innovative features and outstanding performance, YOLOv8 is poised to remain a preferred choice for object detection tasks in the foreseeable future [34].

Figure 1 illustrates the performance of YOLOv8 in real-world scenarios [35].



**Figure 1.** YOLOv8 performance.

## 2.2. DeepSORT

DeepSORT is an advanced real-time multiple objects tracking algorithm that leverages deep learning-based feature extraction coupled with the Hungarian algorithm for assignment. The code structure of DeepSORT comprises the following key components [36]:

1. Feature Extraction: Responsible for extracting features from input video frames, including bounding boxes and corresponding features.
2. Detection and Tracking: Detects objects in each video frame and associates them with their tracks using the Hungarian algorithm.
3. Kalman Filter: Predicts the location of each object in the next video frame based on its previous location and velocity.
4. Appearance Model: Stores and updates the appearance features of each object over time, facilitating re-identification and appearance updates.
5. Re-identification: Matches the appearance of an object in one video frame with its appearance in a previous frame.
6. Output: Generates the final output—a set of object tracks for each video frame.

These components collaboratively form a comprehensive tracking algorithm. DeepSORT can undergo training using a substantial dataset of video frames and corresponding object bounding boxes. The training process of DeepSORT fine-tunes the various algorithm's components, such as the appearance model and feature extraction, aiming to

enhance the algorithm's overall performance. Despite its strengths, DeepSORT also faces unique challenges, including:

- Blurred Objects: Tracking difficulties due to image artifacts caused by blurred objects [37].
- Intra-object Changes: Challenges in handling changes in the shape or size of objects [38].
- Non-rigid Objects: Difficulty in tracking objects appearing for short durations [39].
- Transparent Objects: Challenges in detecting objects made of transparent materials.
- Non-linear Motion: Difficulty in tracking irregularly moving objects [40].
- Fast Motion: Challenges posed by quickly moving objects.
- Similar Objects: Difficulty in differentiating objects with similar appearances [41].
- Occlusion: Tracking challenges when objects overlap or obstruct each other [18].
- Scale Variation: Difficulty when objects appear at different scales in the image [42].

To address these challenges, the StrongSORT algorithm has been proposed aiming to offer solutions which enhance tracking robustness and efficiency [4]. The StrongSORT algorithm is briefly reviewed in the following subsection.

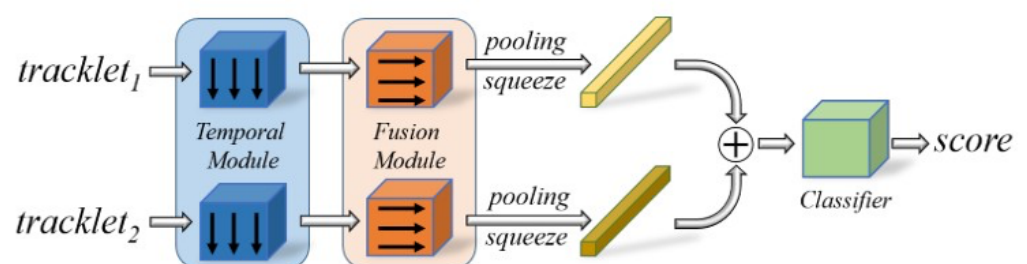
### 2.3. StrongSORT

The StrongSORT algorithm enhances the original DeepSORT by introducing the AFLink algorithm for temporal matching to person tracking and the GSI algorithm for temporal interpolation of matched individuals. New configuration options have been incorporated to facilitate these improvements [4]:

- AFLink: A flag indicating whether the AFLink algorithm should be utilized for temporal matching.
- Path\_AFLink: The path to the AFLink algorithm model to be employed.
- GSI: A flag indicating whether the GSI algorithm should be employed for temporal interpolation.
- Interval: The temporal interval to be applied in the GSI algorithm.
- Tau: The temporal interval to be used in the GSI algorithm.

The original DeepSORT application is updated through the integration of the AFLink and GSI algorithms, providing enhanced capabilities for temporal matching and interpolation of tracked individuals.

Figure 2 presents the framework of the AFLink model. It adopts the spatio-temporal information of two tracklets as input and predicts their connectivity [4].



**Figure 2.** Framework of the AFLink model.

## 3. Proposed Method

### 3.1. Proposed Approach

To enhance the tracking performance of StrongSORT, the approach proposed in the present paper involves a multi-faceted strategy that encompasses diverse appearance model training, precise parameter tuning, incorporation of additional information, and integration with other tracking algorithms. Each strategy aims to address specific challenges in pedestrian tracking, thereby contributing to the overall improvement of the algorithm:

1. **Diverse Appearance Model Training:** Our approach begins with training the appearance model on a diverse set of data. By exposing the network to a wide range of visual appearances, we aim to enhance its ability to effectively track objects. This diversity helps the model generalize better to various scenarios, including challenging conditions, such as occlusions, variations in lighting, and diverse pedestrian characteristics.
2. **Precise Parameter Tuning:** Recognizing the critical role of parameter tuning in tracking accuracy, our approach involves the fine-tuning of various parameters. This includes parameters related to the Kalman filter, the Hungarian algorithm, and the confidence threshold used for tracklet merging. The optimization of these parameters is essential for achieving optimal tracking results, and our methodology systematically explores the parameter space for performance enhancement.
3. **Incorporating Additional Information:** To further improve tracking accuracy, we explore the integration of additional information about the tracked objects. This supplementary information may include object size, shape, or velocity. By incorporating these cues into the tracking process, we aim to provide the algorithm with richer contextual information, enabling more accurate predictions and reducing instances of tracking failures.
4. **Integration with Other Tracking Algorithms:** Our approach investigates the collaborative integration of StrongSORT with other tracking algorithms. This synergistic approach allows us to leverage complementary information from different sources. By combining the strengths of multiple algorithms, we aim to enhance the overall robustness and reliability of pedestrian tracking in diverse and challenging scenarios [20,43].

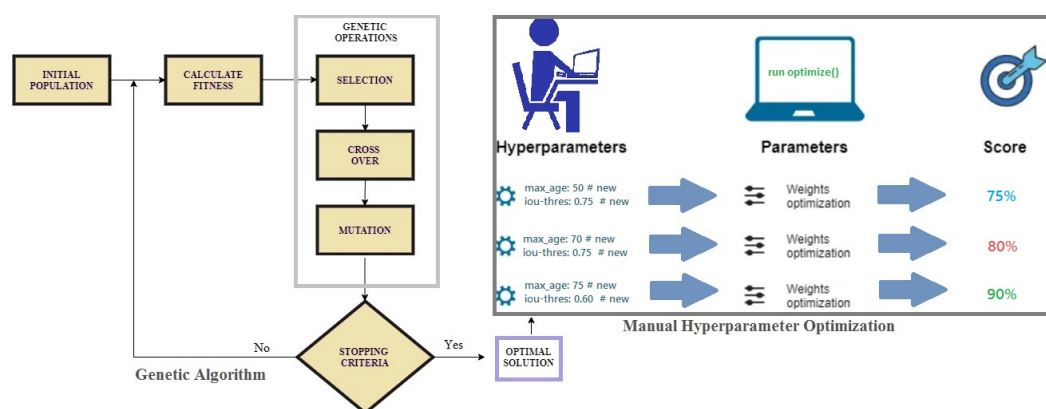
While these strategies present significant potential for enhancing StrongSORT's tracking capabilities, achieving optimal performance is a complex task that requires experimentation with various techniques. In our work, we particularly focus on two primary strategies for parameter tuning: the use of a genetic algorithm and manual tuning of high-impact parameters.

1. **Genetic Algorithm for Parameter Tuning:**
  - **Random Parameter Generation:** In this phase, a diverse set of parameters for the StrongSORT model is randomly generated. The randomness introduces variability, exploring different regions of the parameter space. This diversity is crucial as it helps prevent the algorithm from converging to local optima and promotes exploration of the broader solution space.
  - **Evaluation and Comparison:** The generated parameters undergo evaluation using the tracking dataset. This evaluation involves running the StrongSORT model with the randomly generated parameters and measuring its performance against predefined goals. The goals serve as benchmarks, providing clear criteria to determine the success or failure of a set of parameters. This step is vital for identifying the parameters that contribute to improved tracking accuracy.
  - **Natural Selection:** Natural selection is a key principle inspired by biological evolution. Parameters that exhibit superior performance, as measured by the predefined goals, are selected. This mimics the biological concept of favoring traits that contribute positively to survival and reproduction. The selected parameters become the foundation for the next generation, ensuring that successful traits are passed on and refined over successive iterations.
  - **Iteration:** The entire process is repeated for several generations. In each iteration, the algorithm refines and evolves the parameters based on the success of the previous generations. Over time, this iterative approach leads to the emergence of parameter sets that exhibit superior performance, demonstrating the adaptability and efficiency of the genetic algorithm in finding optimal solutions within the solution space.

## 2. Manual Tuning of High-Impact Parameters:

- **Impact Analysis:** A comprehensive analysis is conducted for understanding the impact of various parameters on the model's performance. This involves studying how each parameter influences the tracking algorithm and its interactions with other parameters and the dataset. Impact analysis provides valuable insights into the complex relationships within the algorithm, guiding subsequent tuning efforts.
- **Precise Manual Tuning:** Based on the insights gained from impact analysis, high-impact parameters are precisely and manually adjusted. This precision involves a deep understanding of the algorithm's behavior and the specific effects of parameter changes. Manual tuning allows researchers to exert fine-grained control over critical aspects, such as the Kalman filter parameters or tracklet merging thresholds, ensuring that adjustments align with the goals of improving tracking accuracy.
- **Experiments and Measurements:** After manual parameter tuning, a series of experiments are conducted to measure the model's performance. These experiments involve running the StrongSORT algorithm with the manually adjusted parameters on the tracking dataset. The goal is to quantify the improvements achieved through manual tuning, providing empirical evidence of the impact of parameter adjustments on tracking accuracy.
- **Reiteration:** The tuning and measurement processes are iterative, allowing researchers to refine parameters further based on experimental results. Reiteration is essential for optimizing the model progressively. Researchers can repeat the manual tuning cycle as necessary to achieve the maximum potential of the algorithm, continually refining the parameter values and ensuring they align with the specific requirements of the tracking scenario.

In summary, the proposed method uses a genetic algorithm to automatically improve tracking parameter settings, and it also utilizes manual tuning for high-impact parameters to obtain better performance in pedestrian tracking (see Figure 3). The synergy of these approaches aims to strike a balance between automated optimization and expert-guided fine-tuning, resulting in an enhanced StrongSORT algorithm that excels in challenging tracking scenarios. The effectiveness of our proposed method lies in its ability to systematically explore the parameter space, combining automated techniques with expert knowledge to achieve robust and adaptable pedestrian tracking in complex real-world scenarios. This hybrid approach allows for a comprehensive optimization strategy, ensuring that the algorithm performs optimally across a diverse range of tracking challenges.



**Figure 3.** Method of tuning hyperparameters.

### 3.2. Datasets

After a thorough evaluation of various tracking algorithms, including Botsort, BYTE-track, Ocsort, and StrongSORT (an upgraded version of DeepSORT), we selected specific

datasets for our study. The chosen datasets were MOT16 and MOT17 Challenge [44], specifically designed for pedestrian tracking.

The MOT16 dataset serves as a benchmark, featuring 14 challenging video sequences. It comprises seven training sets and seven test sets captured in unconstrained environments using both static and moving cameras. The tracking and evaluation in MOT16 are conducted in image coordinates, and all sequences have been accurately annotated following a well-defined protocol. In contrast, the MOT17 Challenge builds upon the MOT16 sequences, providing a more precise ground truth. Each sequence in the MOT17 Challenge is equipped with three sets of detections: DPM, Faster-RCNN, and SDP. These diverse detections are particularly suitable for our project, which focuses on enhancing pedestrian tracking in complex landscapes.

### 3.3. Improving Performance Using StrongSORT

Our primary focus is on fine-tuning the hyperparameters, involving the adjustment of parameters that significantly impact algorithm performance. Key hyperparameters subject to adjustment include the intersection over union (IoU) threshold, maximum cosine distance, inactivity threshold, and maximum age. In this study, we concentrate on fine-tuning the hyperparameters of StrongSORT, an enhanced model derived from DeepSORT, known for delivering superior tracking results.

In terms of computational efficiency, StrongSort\_P demonstrates significant improvements over StrongSORT. The reduction in inference time from 21.3 ms to 19.5 ms indicates a notable enhancement in the speed of the detection phase. Additionally, the update time of StrongSORT experiences a decrease from 104.7 ms to 94.0 ms, reflecting a notable improvement in the processing speed of the tracking algorithm. These enhancements underscore the model's capacity to achieve superior performance without compromising accuracy, showcasing a commendable balance between processing speed and precision.

To facilitate object detection within the tracker, we leveraged various versions of YOLOv8 as our foundational tool. Subsequently, we conducted a comparative analysis using essential criteria, namely IDF1, MOTA, and HOTA, which are thoroughly explained in the next subsection. Our observations revealed that the MOTA and HOTA metrics are well-suited for evaluating improvements in real-time tracking systems, while IDF1 is particularly valuable for assessing enhancements in detection accuracy, an equally crucial aspect of the overall performance evaluation.

### 3.4. Evaluation Metrics

The evaluation of the performance of detection and tracking systems in computer vision relies on several key metrics. In our work, we utilize the following metrics to assess the effectiveness of our tracking algorithm:

**IDF1 Criterion:** IDF1 stands for “Identification F1 Score” and is a criterion for determining detection accuracy. IDF1 is calculated by dividing the number of items correctly identified by the system (True Positives) by the sum of the items correctly identified by the system (True Positives), the items incorrectly identified by the system (False Positives), and the items that were not identified (False Negatives). IDF1 can be defined as follows:

$$IDF1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (1)$$

**MOTA Criterion:** MOTA stands for “Multiple Object Tracking Accuracy” and is a criterion for determining tracking accuracy. MOTA is calculated by dividing the number of items tracked correctly (True Positives) by the sum of the items tracked correctly (True Positives), the items tracked incorrectly (False Positives), the items that were not tracked (False Negatives), and the items that were tracked incorrectly (Mismatches). MOTA can be defined as follows:

$$MOTA = 1 - \frac{FN + FP + Mismatches}{TP + FN + Mismatches} \quad (2)$$



**HOTA Criterion:** HOTA stands for “Higher Order Tracking Accuracy” and is a criterion for determining tracking accuracy and distinguishing between overlapping objects and erroneous movements. This criterion depends on calculating the probability of false transitions and mismatched movements between overlapping objects. HOTA is calculated by dividing the items tracked correctly (True Positives) minus the mismatches and false transitions by the sum of the items tracked correctly, the items tracked incorrectly, the items that were not tracked, and the mismatched movements and false transitions. HOTA can be defined as follows:

$$HOTA = \frac{TP - \text{Mismatches} - FT}{TP + FP + FN - \text{Mismatches} - FT} \quad (3)$$

where in these equations, TP represents True Positives, FP is False Positives, FN is False Negatives, Mismatches refers to instances of incorrect tracking, and False Transitions are cases where the system incorrectly predicts transitions between objects.

In general, IDF1 is considered a tool for measuring the accurate detection of targeted objects in the image, while MOTA is a tool for measuring the temporal tracking accuracy of moving objects. HOTA is considered an extension of MOTA, focusing on distinguishing between overlapping objects and erroneous movements. Both MOTA and HOTA criteria are commonly used in evaluating real-time automated tracking systems, but HOTA is considered more specific and better in critical and complex tracking environments [45].

## 4. Results

### 4.1. Quantitative Evaluation of Tracking Algorithms

The hyperparameters of StrongSORT were tuned using a genetic algorithm, but we did not observe a significant enhancement in the results. Thus, we focused on the most influential parameters in pedestrian tracking. The results indicated that the IOU threshold had a weak effect on performance in StrongSORT\_I, while the maximum age hyperparameter had a clear effect on performance, leading to superior results in StrongSORT\_P. The use of this hyperparameter had a positive impact on all the evaluation metrics. The results are presented in Tables 2 and 3. In particular, these tables present a comprehensive comparison of tracking algorithms, including BYTEtrack, Ocsort, StrongSORT\_I, Botsort, StrongSORT, and the proposed StrongSORT\_P. The metrics evaluated are Higher Order Tracking Accuracy (HOTA), Multiple Object Tracking Accuracy (MOTA), and IDentity F1 (IDF1).

**Table 2.** MOT16 trackers comparison with YOLOv8n.

#	Method	HOTA	MOTA	IDF1
1	BYTEtrack	37.574	33.36	43.946
2	Ocsort	40.852	38.206	48.678
3	StrongSORT_I	42.611	38.46	51.805
4	Botsort	42.644	41.654	51.91
5	StrongSORT	42.698	38.416	51.899
6	StrongSORT_P	42.859	38.467	52.29
Increase		0.161	0.051	0.391
Percentage Increase		0.38%	0.13%	0.75%

In Table 4, we present the performance metrics of our tracker using various YOLOv8 models. The evaluation encompasses five YOLOv8 models, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Each model was examined to gauge its impact on tracking performance, revealing nuanced differences in key metrics. The calculated HOTA (Higher Order Tracking Accuracy), MOTA (Multiple Object Tracking Accuracy), and IDF1 (ID F1 Score) metrics provide a comprehensive overview of the tracker’s effectiveness across different YOLOv8 configurations.

**Table 3.** MOT17 trackers comparison with YOLOv8s.

#	Method	HOTA	MOTA	IDF1
1	BYTetrack	41.073	38.41	48.973
2	Ocsort	44.389	42.394	53.121
3	Botsort	45.087	43.498	53.865
4	StrongSORT_I	45.299	42.414	54.856
5	StrongSORT	45.338	42.384	54.96
6	StrongSORT_P	45.55	42.413	55.338
Increase		0.212	0.029	0.378
Percentage Increase		0.47%	0.07%	0.69%

**Table 4.** Performance metrics with different YOLOv8 models.

Method	HOTA (%)	MOTA (%)	IDF1 (%)
YOLOv8n	0.49	0.15	0.92
YOLOv8s	0.47	0.07	0.69
YOLOv8m	0.60	0.27	0.65
YOLOv8l	0.51	0.46	0.69
YOLOv8x	0.44	0.39	0.61
Average percentage increase (%)	0.50	0.27	0.71

These results indicate varying degrees of improvement across metrics. Notably, an average percentage increase of 0.50% in HOTA, 0.27% in MOTA, and 0.71% in IDF1 was observed. These enhancements underscore the tracker's adaptability to different YOLOv8 models and its consistent performance improvement. The findings presented in Table 4 further reinforce the robustness and versatility of our proposed tracking algorithm in diverse scenarios.

Moreover, these results complement the comparisons with other state-of-the-art trackers presented in Tables 2 and 3. While we acknowledge the importance of diverse tests, the inclusion of various YOLOv8 models provides valuable insights into the algorithm's performance nuances and its ability to adapt to different object detection configurations.

The results in Tables 2 and 3 demonstrate the performance of different tracking algorithms on the MOT16 and MOT17 datasets. The comparison was made using three evaluation metrics: HOTA, MOTA, and IDF1. The proposed enhanced algorithm, StrongSORT\_P, outperformed all other algorithms in all metrics, including the original StrongSORT, validating its effectiveness in enhancing pedestrian tracking accuracy. The percentage increases in HOTA, MOTA, and IDF1 signify the algorithm's substantial improvement over the MOT16 and MOT17 datasets. Notably, the maximum age hyperparameter, a focus of our fine-tuning efforts, emerges as a key contributor to the superior performance of StrongSORT\_P. This emphasizes the significance of precise parameter tuning in optimizing tracking algorithms for real-world scenarios, confirming the importance of our approach in addressing the challenges of pedestrian tracking.

Overall, the results suggest that the proposed enhanced algorithm, StrongSORT\_P, is an effective and efficient method for pedestrian tracking, particularly in complex scenarios such as those found in the MOT16 and MOT17 datasets. The use of YOLOv8 for detection with StrongSORT further improves its performance. Thus, the tuning of the hyperparameters in StrongSORT confirms its importance for optimizing tracking algorithms for specific scenarios.

It is noticeable in Figures 4 and 5, that StrongSORT\_P had the ability to track the same person number 3 after hiding behind the obstacle, while in the regular StrongSORT, it did not consider it the same person and the number changed to 15, indicating weakness in tracking for the normal algorithm in the presence of obstacles. These scenes confirm the results of the previous comparison tables, which gave priority to the algorithm after

improvement in tracking ability. In Figures 4 and 5, we observe the improved tracking ability of StrongSORT\_P, especially in handling scenarios with obstacles.

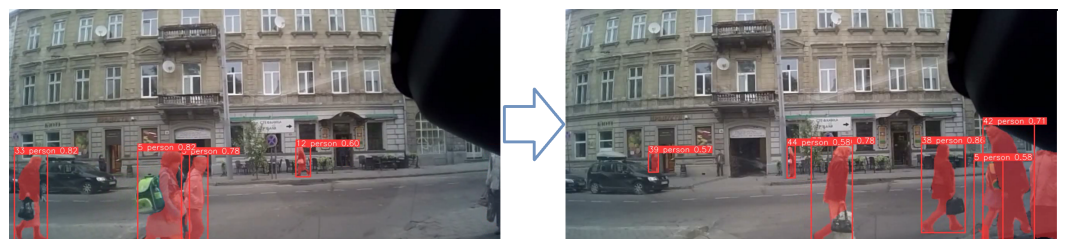


**Figure 4.** Tracking by StrongSORT.

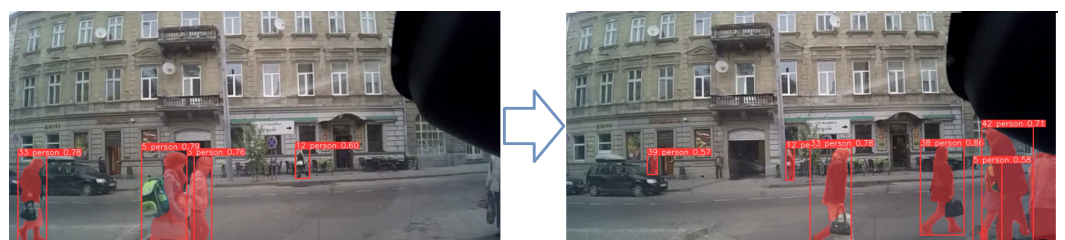


**Figure 5.** Tracking by StrongSORT\_P.

In the more intricate scene depicted in Figures 6 and 7, the enhanced version, StrongSORT\_P, demonstrates improved performance by avoiding identity switching for the person walking away. Conversely, the original tracker version exhibits a switch to key number “44” under similar conditions.

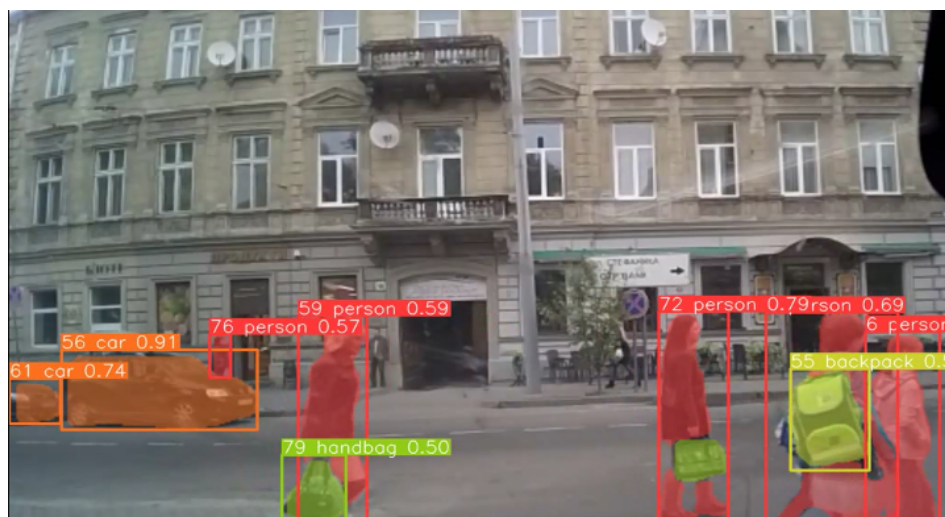


**Figure 6.** Tracking by StrongSORT in a crowded scenario.



**Figure 7.** Tracking by StrongSORT\_P in a crowded scenario.

Figures 8 and 9 showcase the detection and tracking of 80 objects and pedestrians using YOLOv8n-seg along with StrongSORT and StrongSORT\_P, respectively.



**Figure 8.** Detection and tracking of 80 objects using YOLOv8n-seg and StrongSORT.



**Figure 9.** Detection and tracking of pedestrians using YOLOv8n-seg and StrongSORT\_P.

These visualizations provide insights into the real-world performance of the tracking algorithms, reinforcing the quantitative results obtained through the evaluation metrics in Tables 2 and 3.

The limitations of the present study include the size of the studied datasets. Due to limited resources, we were unable to experiment with larger data. Our future work could involve the evaluation of larger datasets, such as MOT20, MOT20Det, and others, and exploring alternative optimization methodologies beyond hyperparameter tuning.

Based on the outcomes of the present comparison, a correlation between the present study's objective and our previous research can be established, enhancing the detection and tracking of pedestrians through an investigation of various versions of the YOLO algorithm and optimizing the StrongSORT algorithm specifically for pedestrian tracking [27]. As a result, more effective methodology for detecting and tracking pedestrians can be developed, surpassing previous findings.

#### 4.2. Discussion

In the quantitative evaluation of tracking algorithms, the optimization of StrongSORT hyperparameters, particularly the focus on the maximum age parameter, revealed significant improvements in pedestrian tracking performance. The results, presented in Tables 2 and 3, showcase the effectiveness of the proposed optimized algorithm, Strong-



**StrongSORT\_P.** This model outperformed other algorithms, including the original StrongSORT, across multiple metrics, indicating its robustness in handling complex scenarios encountered in the MOT16 and MOT17 datasets.

The observed weak effect of the IOU threshold in StrongSORT\_I suggests that certain hyperparameters may have varying impacts on tracking performance. The clear influence of the maximum age parameter underscores the importance of tuning specific parameters tailored to the tracking context. The use of Yolov8 for detection in conjunction with StrongSORT further contributed to enhanced tracking results, emphasizing the synergistic relationship between detection and tracking modules.

While our research study demonstrates promising results, it is crucial to acknowledge its limitations, primarily, the size of the datasets under investigation. The constrained resources prevented experimentation with larger datasets, such as MOT20 and MOT20Det. Future research could expand the evaluation to larger datasets and explore alternative optimization methodologies beyond hyperparameter tuning.

The correlation established between this study's objective and previous research underscores the significance of optimizing tracking algorithms for pedestrian detection. The improved methodology, featuring a combination of YOLO algorithm versions and fine-tuned StrongSORT, surpasses previous findings, indicating a positive direction for advancements in real-world pedestrian tracking scenarios. Figures 4 and 5 visually demonstrate the tracking capabilities of StrongSORT and StrongSORT\_P, emphasizing the latter's improved ability to track individuals, especially in the presence of obstacles. Visualizations in Figures 8 and 9 further reinforce the efficacy of the algorithms, providing a comprehensive understanding of their real-world applicability.

## 5. Conclusions and Future Work

This research systematically compared various tracking algorithms, with a particular focus on the proposed optimized DeepSORT, denoted as StrongSORT\_P. The evaluation, conducted on the MOT16 and MOT17 datasets using three key metrics, unveiled the consistent superiority of StrongSORT\_P over other algorithms, including the baseline DeepSORT. The integration of Yolov8 for object detection notably elevated the overall performance of the tracking algorithms.

To enhance the tracking capabilities of StrongSORT, we meticulously fine-tuned its hyperparameters using a genetic algorithm, with specific emphasis on the maximum age parameter. Our in-depth evaluation revealed that this parameter significantly influenced the algorithm's performance, resulting in the remarkable enhancements seen in StrongSORT\_P. These findings collectively affirm that the proposed algorithm presents a highly effective and efficient approach to pedestrian tracking, particularly in complex scenarios marked by occlusions and partial visibility.

StrongSORT\_P demonstrated substantial improvements in the HOTA, MOTA, and IDF1 metrics over the MOT17 Challenge dataset. These enhancements, as indicated in Table 4, underscore the algorithm's adaptability and suitability for real-world tracking applications. The consistent improvements across various YOLOv8 models (0.50%, 0.27%, 0.71%) further emphasize the robustness of the algorithm, reaffirming its potential in diverse tracking scenarios.

With respect to future work, we recognize the potential for further advancements. Expanding the evaluation to larger datasets, such as MOT20 and MOT20Det, would provide a broader perspective on algorithm performance. Additionally, exploring alternative optimization methodologies beyond hyperparameter tuning could contribute to the ongoing refinement of tracking algorithms in diverse scenarios. Moreover, there is a need for future research to delve deeper into the intricate balance between processing speed and accuracy, especially considering the added complexity introduced by the fine-tuning process and integration with YOLOv8. Uncovering the algorithm's approach to achieving this balance will be crucial for optimizing efficiency without compromising tracking accuracy.



**Author Contributions:** Conceptualization, M.S. (Majdi Sukkar), M.S. (Madhu Shukla), D.K., V.C.G., A.K. and B.A.; Data Curation, M.S. (Majdi Sukkar), M.S. (Madhu Shukla), D.K., V.C.G., A.K. and B.A.; Writing—Original Draft, M.S. (Majdi Sukkar), M.S. (Madhu Shukla), D.K., V.C.G., A.K. and B.A.; Methodology, M.S. (Majdi Sukkar), M.S. (Madhu Shukla), D.K., V.C.G., A.K. and B.A.; Review and Editing, M.S. (Majdi Sukkar), M.S. (Madhu Shukla), D.K., V.C.G., A.K. and B.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Razzok, M.; Badri, A.; Mourabit, I.E.; Ruichek, Y.; Sahel, A. Pedestrian Detection and Tracking System Based on Deep-SORT, YOLOv5, and New Data Association Metrics. *Information* **2023**, *14*, 218. [\[CrossRef\]](#)
2. Bhola, G.; Kathuria, A.; Kumar, D.; Das, C. Real-time Pedestrian Tracking based on Deep Features. In Proceedings of the 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1101–1106.
3. Li, R.; Zu, Y. Research on Pedestrian Detection Based on the Multi-Scale and Feature-Enhancement Model. *Information* **2023**, *14*, 123. [\[CrossRef\]](#)
4. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. StrongSORT: Make DeepSORT Great Again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737. [\[CrossRef\]](#)
5. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.D.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A Benchmark for Multi Object Tracking in Crowded Scenes. *arXiv* **2020**, arXiv:2003.09003.
6. Xiao, C.; Luo, Z. Improving Multiple Pedestrian Tracking in Crowded Scenes with Hierarchical Association. *Entropy* **2023**, *25*, 380. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Myagmar-Ochir, Y.; Kim, W. A Survey of Video Surveillance Systems in Smart City. *Electronics* **2023**, *12*, 3567. [\[CrossRef\]](#)
8. Tao, M.; Li, X.; Xie, R.; Ding, K. Pedestrian Identification and Tracking within Adaptive Collaboration Edge Computing. In Proceedings of the 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Rio de Janeiro, Brazil, 24–26 May 2023; IEEE: Piscataway, NJ, USA 2023; pp. 1124–1129.
9. Son, T.H.; Weedon, Z.; Yigitcanlar, T.; Sanchez, T.; Corchado, J.M.; Mehmood, R. Algorithmic Urban Planning for Smart and Sustainable Development: Systematic Review of the Literature. *Sustain. Cities Soc.* **2023**, *94*, 104562. [\[CrossRef\]](#)
10. AL-Dosari, K.; Hunaiti, Z.; Balachandran, W. Systematic Review on Civilian Drones in Safety and Security Applications. *Drones* **2023**, *7*, 210. [\[CrossRef\]](#)
11. Vasiljevas, M.; Damaševičius, R.; Maskeliūnas, R. A Human-Adaptive Model for User Performance and Fatigue Evaluation during Gaze-Tracking Tasks. *Electronics* **2023**, *12*, 1130. [\[CrossRef\]](#)
12. Alhafnawi, M.; Salameh, H.A.B.; Masadeh, A.; Al-Obiedollah, H.; Ayyash, M.; El-Khazali, R.; Elgala, H. A Survey of Indoor and Outdoor UAV-Based Target Tracking Systems: Current Status, Challenges, Technologies, and Future Directions. *IEEE Access* **2023**, *11*, 68324–68339. [\[CrossRef\]](#)
13. Li, F.; Chen, Y.; Hu, M.; Luo, M.; Wang, G. Helmet-Wearing Tracking Detection Based on StrongSORT. *Sensors* **2023**, *23*, 1682. [\[CrossRef\]](#)
14. Abdulghafoor, N.H.; Abdullah, H.N. A Novel Real-time Multiple Objects Detection and Tracking Framework for Different Challenges. *Alex. Eng. J.* **2022**, *61*, 9637–9647. [\[CrossRef\]](#)
15. Wang, G.; Wang, Y.; Zhang, H.; Gu, R.; Hwang, J. Exploit the Connectivity: Multi-Object Tracking with TrackletNet. In Proceedings of the 27th ACM International Conference on Multimedia (MM), Nice, France, 21–25 October 2019; pp. 482–490.
16. Li, R.; Zhang, B.; Kang, D.; Teng, Z. Deep Attention Network for Person Re-Identification with Multi-loss. *Comput. Electr. Eng.* **2019**, *79*, 106455. [\[CrossRef\]](#)
17. Jiao, S.; Wang, J.; Hu, G.; Pan, Z.; Du, L.; Zhang, J. Joint Attention Mechanism for Person Re-Identification. *IEEE Access* **2019**, *7*, 90497–90506. [\[CrossRef\]](#)
18. Guo, W.; Jin, Y.; Shan, B.; Ding, X.; Wang, M. Multi-Cue Multi-Hypothesis Tracking with Re-Identification for Multi-Object Tracking. *Multimed. Syst.* **2022**, *28*, 925–937. [\[CrossRef\]](#)
19. Kang, W.; Xie, C.; Yao, J.; Xuan, L.; Liu, G. Online Multiple Object Tracking with Recurrent Neural Networks and Appearance Model. In Proceedings of the 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Chengdu, China, 17–19 October 2020; IEEE: Piscataway, NJ, USA 2020; pp. 34–38.
20. Guo, S.; Wang, S.; Yang, Z.; Wang, L.; Zhang, H.; Guo, P.; Gao, Y.; Guo, J. A Review of Deep Learning-Based Visual Multi-Object Tracking Algorithms for Autonomous Driving. *Appl. Sci.* **2022**, *12*, 10741. [\[CrossRef\]](#)
21. Stadler, D.; Beyerer, J. Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 10958–10967.

22. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.P.; Kämäräinen, J.; Danelljan, M.; Zajc, L.C.; Lukežić, A.; Drbohlav, O.; et al. The 8th Visual Object Tracking VOT2020 Challenge Results. In Proceedings of the Workshops on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12539, pp. 547–601.
23. Pham, Q.H.; Doan, V.S.; Pham, M.N.; Duong, Q.D. Real-Time Multi-vessel Classification and Tracking Based on StrongSORT-YOLOv5. In Proceedings of the International Conference on Intelligent Systems & Networks (ICISN); Lecture Notes in Networks and Systems; Springer: Berlin/Heidelberg, Germany, 2023; Volume 752, pp. 122–129.
24. Shelatkar, T.; Bansal, U. Diagnosis of Brain Tumor Using Light Weight Deep Learning Model with Fine Tuning Approach. In Proceedings of the International Conference on Machine Intelligence and Signal Processing (MISP); Lecture Notes in Electrical Engineering; Springer: Berlin/Heidelberg, Germany, 2022; Volume 998, pp. 105–114.
25. Li, J.; Wu, W.; Zhang, D.; Fan, D.; Jiang, J.; Lu, Y.; Gao, E.; Yue, T. Multi-Pedestrian Tracking Based on KC-YOLO Detection and Identity Validity Discrimination Module. *Appl. Sci.* **2023**, *13*, 12228. [\[CrossRef\]](#)
26. Subramanian, M.; Shanmugavadivel, K.; Nandhini, P.S. On Fine-Tuning Deep Learning Models Using Transfer Learning and Hyper-Parameters Optimization for Disease Identification in Maize Leaves. *Neural Comput. Appl.* **2022**, *34*, 13951–13968. [\[CrossRef\]](#)
27. Sukkar, M.; Kumar, D.; Sindha, J. Improve Detection and Tracking of Pedestrian Subclasses by Pre-Trained Models. *J. Adv. Eng. Comput.* **2022**, *6*, 215. [\[CrossRef\]](#)
28. Kapania, S.; Saini, D.; Goyal, S.; Thakur, N.; Jain, R.; Nagrath, P. Multi Object Tracking with UAVs using Deep SORT and YOLOv3 RetinaNet Detection Framework. In Proceedings of the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems, Linz, Austria, 8–10 July 2020; pp. 1–6.
29. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M. Online Multi-Object Tracking with Dual Matching Attention Networks. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11209, pp. 379–396.
30. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.T.; Upcroft, B. Simple Online and Realtime Tracking. In Proceedings of the International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; IEEE: Piscataway, NJ, USA 2016; pp. 3464–3468.
31. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Computer Vision Foundation; IEEE: Piscataway, NJ, USA, 2019; pp. 4660–4669.
32. Guo, M.; Xue, D.; Li, P.; Xu, H. Vehicle Pedestrian Detection Method Based on Spatial Pyramid Pooling and Attention Mechanism. *Information* **2020**, *11*, 583. [\[CrossRef\]](#)
33. Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* **2023**, *11*, 677. [\[CrossRef\]](#)
34. Sirisha, U.; Praveen, S.P.; Srinivasu, P.N.; Barsocchi, P.; Bhoi, A.K. Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 126. [\[CrossRef\]](#)
35. Ultralytics YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 28 January 2024).
36. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3645–3649.
37. Guo, Q.; Feng, W.; Gao, R.; Liu, Y.; Wang, S. Exploring the Effects of Blur and Deblurring to Visual Object Tracking. *IEEE Trans. Image Process.* **2021**, *30*, 1812–1824. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. *Found. Trends Comput. Graph. Vis.* **2020**, *12*, 1–308. [\[CrossRef\]](#)
39. Meimetis, D.; Daramouskas, I.; Perikos, I.; Hatzilygeroudis, I. Real-Time Multiple Object Tracking Using Deep Learning Methods. *Neural Comput. Appl.* **2023**, *35*, 89–118. [\[CrossRef\]](#)
40. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12356, pp. 107–122.
41. Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep Learning in Video Multi-Object Tracking: A Survey. *Neurocomputing* **2020**, *381*, 61–88. [\[CrossRef\]](#)
42. Song, S.; Li, Y.; Huang, Q.; Li, G. A New Real-Time Detection and Tracking Method in Videos for Small Target Traffic Signs. *Appl. Sci.* **2021**, *11*, 3061. [\[CrossRef\]](#)
43. Alikhanov, J.; Kim, H. Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods. *IEEE Access* **2023**, *11*, 68079–68092. [\[CrossRef\]](#)
44. Multiple Object Tracking Benchmark. Available online: <https://motchallenge.net/> (accessed on 28 January 2024).
45. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.H.S.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 548–578. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.