Improvising Object Tracking Algorithm SORT for Long-Term Trajectory Extraction

Rutul Patel*, Saumil Patel†, Kunj Kanzariya‡, and Ayush Patel§
*†‡§Computer Science and Engineering, Ahmedabad University, Ahmedabad, India
{rutul.p, saumil.p, kunj.k1, ayush.p3}@ahduni.edu.in

***Abstract*—This study focuses on enhancing the SORT (Simple Online and Realtime Tracking) algorithm, particularly addressing its weakness in handling occlusions, where objects temporarily disappear and reappear. Our improvement involves a feature comparison technique that reassigns the same IDs to objects when they reemerge, making the tracker more robust to occlusions. By using advanced feature extraction methods, our approach significantly improves tracking performance and ID consistency across frames. Experimental results show notable enhancements in performance, offering a promising solution to occlusion-related challenges in multi-object tracking systems.**

***Index Terms*—Tracking, Detection, Occlusion, features, Similarity.**

I. Introduction

Object tracking is an important aspect of computer vision, focusing on locating and following objects of interest across consecutive frames of a video or a sequence of images. The goal is to identify and monitor objects as they move within a scene, despite changes in scale, orientation, illumination, occlusion, and other challenges.

A. Background

Traditionally, object tracking relied on handcrafted features and algorithms, such as correlation filters, mean-shift, and Kalman filters. However, with advancements in deep learning, especially convolutional neural networks (CNNs), the field has witnessed a paradigm shift. Deep learning-based approaches have shown remarkable performance improvements in object tracking tasks, leveraging the ability of CNNs to learn discriminative features directly from data.

B. Motivation

Object tracking helps autonomous cars avoid collisions by keeping an eye on bikers, cars, and pedestrians[1]. Object tracking in gesture recognition systems facilitates intuitive instructions based on hand or body motions, which is beneficial for human-computer interaction[1]. Applications for virtual and augmented reality depend on object tracking to smoothly blend virtual and physical content, increasing user immersion[1]. Sports analysts, broadcasters, and coaches may better strategize and engage their audience by using object tracking in sports analytics to monitor player motions and performance[1]. Furthermore, object tracking in medical imaging supports diagnosis and therapy by offering insights into dynamic physiological processes including blood flow and organ movements.

II. Data Description

A. Multiple Object Tracking Dataset - 17[2]

The MOT17 dataset is a benchmark collection designed for evaluating multi-object tracking algorithms. It comprises several video sequences capturing diverse, real-world scenarios, each accompanied by precise annotations of pedestrian positions across frames. For the SORT algorithm, which focuses on tracking objects based on motion and appearance, MOT17 provides a valuable testing ground. The dataset includes both raw video and precomputed detection files, allowing assessment of tracking performance under varied conditions. With its mix of crowded scenes and varying lighting, MOT17 challenges SORT's capabilities in data association and maintaining object identities over time, making it an ideal resource for validating and enhancing tracking methodologies.

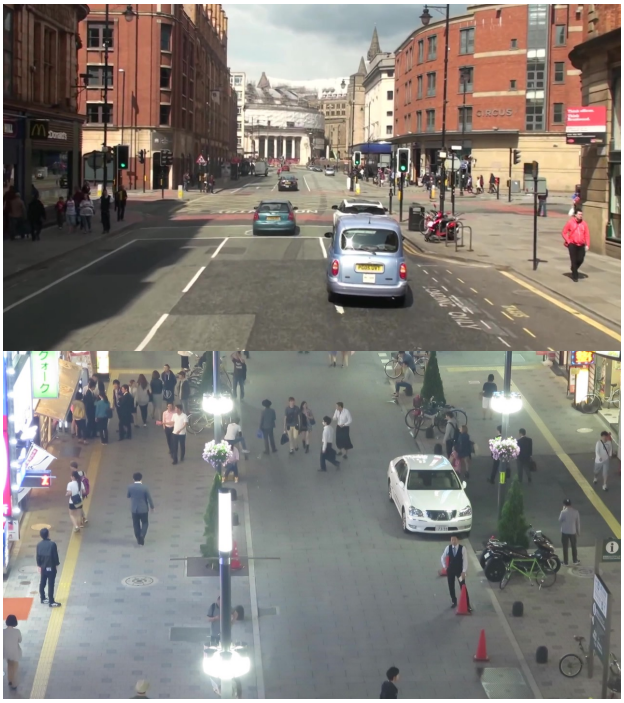

Fig. 1: MOT17 Dataset Example.

B. SORT

The Simple Online and Realtime Tracking (SORT) algorithm represents a pragmatic approach to multi-object tracking (MOT) that prioritizes speed and simplicity. At its core, SORT relies on two main steps: object detection in each frame, followed by a straightforward tracking mechanism that associates detected objects across frames using the Hungarian algorithm based on predicted object locations. Object detection can be performed by any state-of-the-art detection model, and tracking is facilitated by a Kalman filter, which predicts object positions in new frames based on their past position.

SORT's primary advantage lies in its computational efficiency, enabling real-time tracking in various applications. However, it faces challenges in scenarios with frequent occlusions or significant appearance changes, where its simple association mechanism might lead to identity switches or track loss. Despite these limitations, SORT serves as a foundational method for MOT, offering a balance between performance and speed that is crucial for many real-time applications.

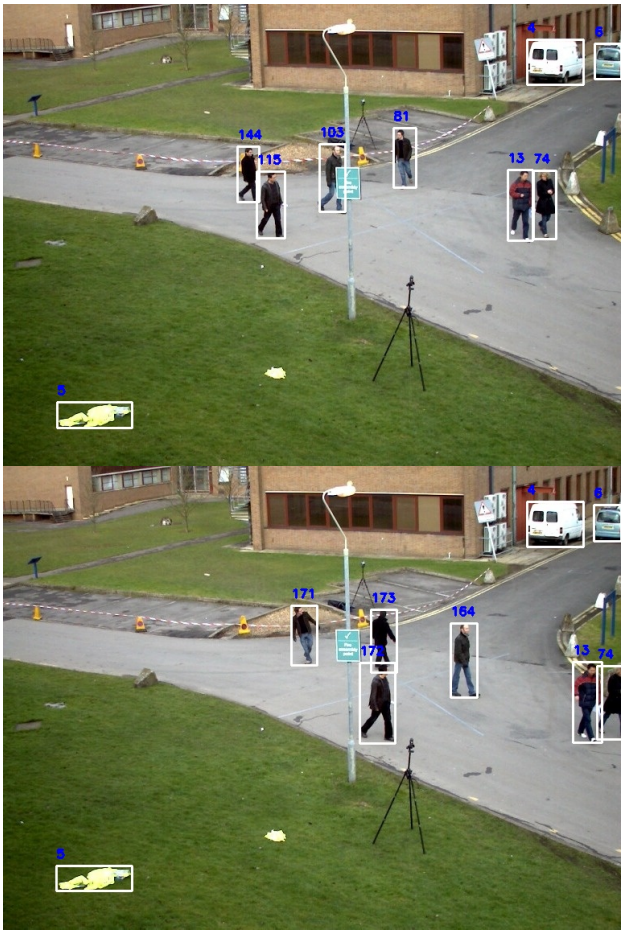

Fig. 2: Object Tracking examples

C. Drawback of SORT

The SORT (Simple Online and Realtime Tracking) algorithm, while advantageous for its speed and efficiency in multi-object tracking tasks, encounters significant limitations in handling occlusions. Specifically, SORT's reliance on simple motion models and the Hungarian algorithm for frame-to-frame association makes it less adept at distinguishing between objects when they overlap or obscure each other. Figure-2 compares two time instances of a MOT-15 dataset scenario and shows fluctuation in the IDs of the pedestrians. This limitation is due to its minimal use of appearance information to re-identify objects post-occlusion. Consequently, in scenarios where objects frequently intersect or occlude each other, SORT may incorrectly merge tracks or lose track of objects altogether. This drawback significantly impacts its performance and reliability in dense, complex environments, limiting its applicability in scenarios requiring precise object tracking through occlusions.

III. Methodology

Object tracking is composed three main steps detection, feature extraction, inter-frame comparison of features. The last step is the most important step as it keeps track of the object by assigning each object a unique ID. We start the object tracking pipeline by selecting an appropriate detection algorithm.

A. Object Detection

Object detection is used to initialize object tracking algorithms by providing initial bounding boxes around the objects of interest. These bounding boxes serve as the starting point for tracking the objects in subsequent frames. object detection is crucial for tracking multiple objects simultaneously. By detecting all objects in a scene initially, the tracking algorithm can assign unique identities to each object and track them individually across frames. We start by detecting objects in each frame of a video using **YOLOv5[4]**. This gives us bounding box coordinates for each detected object.

B. Feature Extraction

The features are the unique characteristics or attributes that help us understand what an object looks like. In our case, we're dealing with images or frames from a video. Each frame might have several objects, like people, cars, or animals. To understand and track these objects, we need to find features that are unique to each object. We use the ResNet50 model to extract these features. For each detected object, you extract features using a pre-trained ResNet50 model. These features are high-dimensional embeddings that represent the visual characteristics of the object.

C. Cosine Similarity Matching

Multi-Object trackers robust to occlusion must reassign the same ID to the object, if it reappears before certain threshold time. Tracker needs to keep track of which object already appeared in order to identify it again. The extracted features which uniquely describe each detected object enables tracker to re-identify the objects. Cosine similarity compares the feature vectors and give the similarity between them. If tracker encounters any object having high cosine-similarity with any previously occured object, it will re-assign the same ID.

D. Object Tracking

The methodology is structured around three core components: object detection, feature extraction, and cosine similarity matching. Object detection is pivotal as it initializes the tracking process by identifying objects in video frames using YOLOv5, generating bounding box coordinates. Feature extraction then captures the distinctive visual attributes of each object using the ResNet50 model, producing high-dimensional embeddings. Finally, cosine similarity matching compares these feature vectors between frames, facilitating the re-identification of objects and enabling the tracker to assign consistent IDs to objects that reappear within a certain threshold time, even amidst occlusion. This comprehensive approach ensures robust and accurate object tracking in complex scenarios.

Fig. 3: Tracking results for frame 11(top) and 12(bottom)

IV. Results and Inferences

This study focused on evaluating the improved tracking performance when a new object enters the scene, using frames 11 and 12 from the MOT15 dataset[Fig. 3]. The analysis showed that our method reliably tracked objects across frames, effectively managing instances where objects became temporarily obscured. The bounding box and Object ID data[Table-1] confirmed that our system maintained consistent identification numbers for each object.

TABLE I: Object IDs and Bbox data of 11th and 12th Frame

Frame ID	Object ID	Bbox: X	Bbox: Y	Bbox: H	Bbox: W
000011	6	734	49	33	44
000011	5	66	496	94	31
000011	4	652	44	69	56
000011	3	464	164	36	82
000011	2	315	207	36	86
000011	1	547	239	36	85
000012	6	734	48	33	45
000012	5	66	496	94	31
000012	4	652	44	69	56
000012	3	460	166	36	81
000012	2	319	206	37	87
000012	1	538	240	32	83

TABLE II: Comparison of MOTA Values for Different Tracking Methods

Tracking Method	MOTA
SORT (SDP)	23
SORT (DPM)	11
Our Approach	34.15

Our results indicate that our enhanced tracking algorithm effectively handles the challenge of occlusion, where objects disappear and reappear. By comparing features to confirm object identities, our approach proved more effective than the standard SORT algorithm. Table-2 describes the Multi-Object Tracking Accuracy comparison where our approach performed significantly better than SORT. This demonstrates the potential of our method to improve the accuracy and reliability of multi-object tracking systems.

V. Acknowledgement

We acknowledge Dr. Mehul Raval and his team for their support and guidance throughout this study.

References

[1] K. Granstrom, M. Baum, and S. Reuter, "Extended Object Tracking: Introduction, Overview and Applications," 2017. [Online]. Available: arXiv:1604.00970 [cs.CV].

[2] Du, Yunhao, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. "Strongsort: Make Deepsort Great Again." *arXiv.org*, February 22, 2023. https://arxiv.org/abs/2202.13514.

[3] Maggiolino, Gerard Ahmad, Adnan Cao, Jinkun Kitani, Kris. (2023). Deep OC-Sort: Multi-Pedestrian Tracking by Adaptive Re-Identification. 3025-3029. 10.1109/ICIP49359.2023.10222576.

[4] P. Dendorfer et al., "MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking." *arXiv*, 2020. doi: 10.48550/ARXIV.2010.07548.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003. keywords: Target tracking;Detectors;Benchmark testing;Kalman filters;Visualization;Complexity theory;Computer Vision;Multiple Object Tracking;Detection;Data Association,

[6] N. Wojke, A. Bewley and D. Paulus, ”Simple online and realtime tracking with a deep association metric,” 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 3645-3649, doi: 10.1109/ICIP.2017.8296962. keywords: Kalman filters;Tracking;Extraterrestrial measurements;Standards;Uncertainty;Cameras;Computer Vision;Multiple Object Tracking;Data Association,