9-25-2024

Enhancing Multiple Object Tracking for Autonomous Vehicles: Integrating Improved Deep SORT with Occlusion Management

Kimia Gholizadeh
University of Windsor

Enhancing Multiple Object Tracking for Autonomous Vehicles: Integrating Improved Deep SORT with Occlusion Management

By

Kimia Gholizadeh

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2024

Enhancing Multiple Object Tracking for Autonomous Vehicles: Integrating Improved Deep SORT with Occlusion Management

by

Kimia Gholizadeh

APPROVED BY:

R. Ruparathna
Department of Civil and Environmental Engineering

M. Hassanzadeh
Department of Electrical and Computer Engineering

N. Zhang, Advisor
Department of Electrical and Computer Engineering

August 16, 2024

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Multi-object tracking (MOT) is a chore task in various applications, including autonomous driving and surveillance systems. Accurate and reliable MOT is essential for these systems to operate safely and efficiently, especially in dynamic and cluttered environments. Despite significant advancements in the field, challenges such as obstacles causing occlusions and background interference persist, often leading to false negatives, reduced accuracy, and reliability.

Traditional MOT methods, such as the widely-used Deep SORT, face significant challenges in handling occlusions and minimizing false negatives. When objects are temporarily obscured, these methods can lose track of them, resulting in inconsistent object identities and increased false negatives. Addressing these challenges is vital for improving tracking performance in complex scenarios.

This thesis enhances Deep SORT by incorporating memory management for occluded items and utilizing the Sørensen-Dice coefficient for better similarity measurement. Our approach re-identifies occluded objects using motion features, enabling more robust tracking even when objects are temporarily obscured.

Evaluated on the MOT16 dataset, our method significantly improves key performance metrics, achieving a MOTA of 61.84 and a recall of 69.8, compared to the baseline Deep SORT performance of 61.40 and 68.9, respectively.

The contributions of this thesis to the field of MOT are significant, providing a reliable method for tracking objects in challenging situations. By addressing the limitations of existing methods, particularly in handling occlusions and reducing false negatives, this work paves the way for more reliable and accurate tracking systems in real-world applications, ultimately enhancing the performance and resilience of autonomous driving systems.

DEDICATION

To my mother, Maryam, whose incredible love and unwavering support have been the wings that lifted me higher. She had faith in me, even when I didn't believe in myself, encouraging me to reach for my dreams. Her selfless dedication to our family has been a constant source of inspiration and strength.

To my father, Alireza, whose wisdom and strength have guided me through life's challenges. His encouragement has been my foundation, giving me the courage to strive for excellence. To my brothers, Kamran and Kambiz, for their unwavering support and belief in my potential, always standing by my side.

To my late uncle Mohsen, who shared my field of interest but couldn't achieve his goals. Following his path is an honor, and his intelligence and passion continue to inspire me. To my boyfriend, Iman, who has always stood by my side, supporting me with his constant encouragement and love. And to my friends, for their encouragement and companionship, providing the emotional support that has been essential during this journey

ACKNOWLEDGEMENTS

Firstly, I want to extend my gratitude to my supervisor, Dr. Ning Zhang, for providing guidance and advice throughout my academic journey. His invaluable insights and support have been crucial to my progress. Dr. Zhang was not just an amazing academic supervisor but also a great person to work with, providing wisdom and encouragement at every step.

Moreover, I wish to convey my heartfelt thanks to my committee members, Dr. Rajeev Ruparthna and Dr. Mohammad Hassanzadeh, for their openness in sharing their knowledge and invaluable support in my research. Their guidance and feedback have played an essential role in the development of this project.

Furthermore, I am sincerely thankful to my friend, Ali Abedi, a PhD candidate, who has guided me significantly along my path. His advice and support have been incredibly helpful.

Lastly, I extend my gratitude to the Department of Electrical and Computer Engineering and everyone who has provided assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Autonomous Vehicles

Vehicular technology, especially autonomous driving, has become more popular and important, helping to develop smart transport systems. These systems need accurate technologies to handle uncertainties like pedestrian actions, random objects, and various road conditions. Autonomous vehicles are expected to drastically change transport around the world, improving human quality of life and making roads safer by reducing traffic accidents significantly.

1.1.1 Levels of Autonomy

The Society of Automotive Engineers (SAE) outlines five distinct levels of vehicle autonomy [15]. There is no automation in Level 0, therefore the driver does all duties with the assistance of safety features. At Level 1, the vehicle can control one function, like steering or braking, but the driver must still pay full attention. Level 2 handles two functions but also requires the driver to be ready to intervene. Level 3 vehicles can drive themselves in some situations but will alert the driver if they need to take over. Level 4 vehicles can drive on their own in many situations without needing the driver to do anything, but the driver can still intervene if they choose to. Lastly, Level 5 vehicles are fully autonomous and can drive themselves in all situations without any help from the driver.

	SAE level	Name	Description
The human driver monitors the environment	0	No Automation	Human driver is responsible for steering, throttle and breaking.
	1	Driver Assistance	The vehicle can perform some control function but not everywhere.
	2	Partial Automation	The vehicle can handle steering, throttle and breaking but the driver is expected to monitor the system and take over in case of faults.
The driving system monitors the environment	3	Conditional Automation	The vehicle monitors the surroundings and notifies the driver if manual control is needed.
	4	High Automation	The vehicle is fully autonomous but only in defined use cases
	5	Full Automation	The driver has only to set the destination. The vehicle will handles any surrounding and make any kind of decision on the way.

Fig. 1.1.1: Levels of Autonomous Vehicles by Sae [15]

1.1.2 Autonomous Vehicles Modules

Autonomous vehicles are built with five essential components: localization, perception, decision-making, planning, and control [35]. The perception component uses a range of sensors, including Radar, camera, and LiDAR, to help the vehicle understand its environment [41]. Localization and mapping algorithms determine the global and local positions of the self-driving vehicle and create environmental maps using data from sensors and other outputs from perception systems [27]. The decision-making component controls actions like braking, acceleration, and maneuvering through lanes [8]. The planning component assists the vehicle in determining the best paths from one location to another. Lastly, the control component ensures that the vehicle's internal mechanisms accurately execute these plans, performing the necessary maneuvers. The primary focus of this study is on autonomous vehicles' perception and localization features.

Fig. 1.1.2: Autonomous Vehicles' Development Stages [35]

1.1.3 Perception

Perception is a critical module for enabling autonomous vehicles to understand their surrounding environment, including the locations, velocities, and future state predictions of pedestrians and objects. This task can be reached using various sensors. Additionally, traditional methods may incorporate short- and long-range radars and ultrasonic sensors, which assist autonomous vehicles' ineffective path-planning and decision-making modules.

1.1.4 Sensor Technology in Autonomous Vehicles

In Autonomous Vehicles (AVs), three main types of sensor technologies are crucial:

LiDAR operates by emitting lasers to map out the vehicle's surroundings in three dimensions.It calculates the time required for the lasers to reflect, allowing for precise mapping and distance calculations [33]. Despite its precision, LiDAR can be limited by its narrower field of coverage compared to other sensors and can experience issues like glare in rainy conditions [33]. Though traditionally expensive, the costs associated with LiDAR have been reduced, making it more accessible.

Cameras in AVs, on the other hand, are characterized into two main types: visible (VIS) and infrared (IR). VIS cameras operate similarly to human eyes, and are valued for their affordability and high-resolution color imaging capabilities [37]. These cam-

eras enable stereo vision, which can generate a 3D view of the environment, though they offer less depth accuracy compared to other sensors.

Infrared cameras exhibit less sensitivity to adverse conditions and fluctuations in brightness [28]. They are particularly useful for detecting warm objects like pedestrians and animals [11]. Additionally, NIR cameras can measure distances using the Time of Flight (ToF) principle, applicable across various settings.

RADAR is essential in autonomous vehicles [4]. This system utilizes the Doppler effect to directly measure vehicle speeds, offering crucial velocity data that helps improve the accuracy of sensor fusion algorithms.

There are various types of RADAR systems distinguished by their operational range. Long-range RADAR operates at 77 GHz and can detect objects up to 200 meters away, although it is limited by low resolution. Short to medium-range RADARs, functioning at 24 GHz and 76 GHz, are more mature and cost-effective, but their resolution is affected by the width of their beams and the length of their wavelengths. RADAR is particularly effective in poor weather conditions as it does not require the processing of data-heavy streams like video, allowing for lower computational demands.

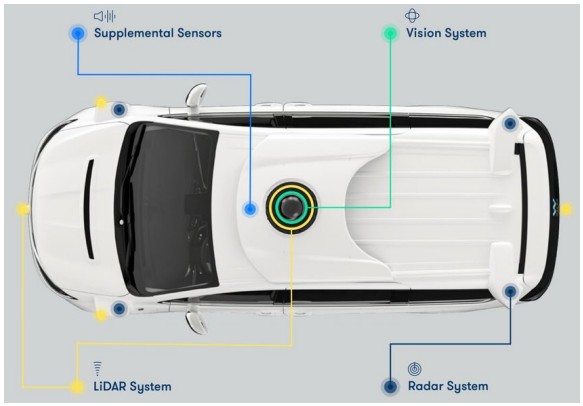


Fig. 1.1.3: Autonomous Vehicles' Perception Sensors[1]

1.2 Object Detection

Over the past twenty years, there has been a swift advancement in object-detecting technologies. The main step in object detection involves identifying instances of specific visual object classes in images. The goal is to create computational models and methods that precisely identify the location (object localization) and category (object classification) of these objects, providing critical information for various computer vision applications.

The effectiveness of object detection systems is typically measured by two key metrics: accuracy, which includes both the precision of object classification and the accuracy of localization, and the speed of detection.

1.2.1 Object Detection in Autonomous Vehicles

Object detection is an essential step in autonomous vehicle technology, crucial for classifying and localizing objects within the vehicle's environment to ensure safe and efficient navigation. This deep learning task enables the vehicle to identify and monitor various objects, using exteroceptive sensors including Cameras, LiDAR , RADAR, and GPS[36]. These sensors collectively help the vehicle understand its surroundings, localize itself, and track both stationary and moving objects.

1.3 Multiple Object Tracking

Multiple Object Tracking (MOT) is a fundamental and indispensable aspect of computer vision. It entails the prediction of trajectories for all objects within a video sequence.

A major benefit of video-based multi-object tracking (MOT) is to precisely detect targets in continuous videos while maintaining their identities, even when there are changes in their appearance or surroundings. This ability to produce complete motion trajectories for tracked objects has attracted considerable interest from researchers. [44]

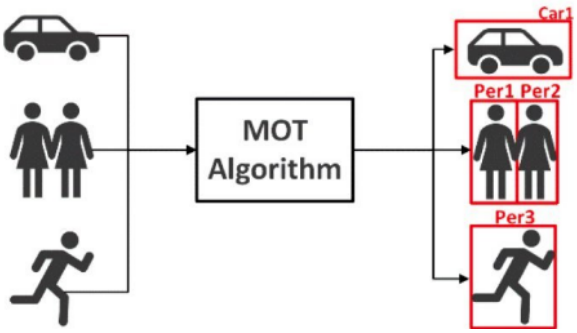


Fig. 1.3.1: Tracking Multiple Objects Using TBD Approach [12]

1.3.1 Multiple Object Tracking Applications

Multi-Object Tracking (MOT) is a sophisticated task with numerous practical applications across various fields. It is employed extensively in video surveillance [6], human behavior recognition [13], and autonomous driving [6]. Beyond these, MOT is also crucial for visual surveillance [38], and human-computer interaction [5], demonstrating its versatility. These diverse real-world applications have significantly driven advancements in MOT research, making it a key focus area in the field.

1.3.2 Multiple Object Tracking Categories

Luo et al.[24] categorized MOT approaches into three distinct groups. Firstly, They categorized the methods according to their initialization approach, differentiating between tracking by detection(TBD) and tracking without decetion. Detection-based approach, often referred to as tracking-by-detection, involves associating detected objects with their paths in subsequent frames based on similarities in appearance or movement In contrast, detection-free tracking necessitates manually locating objects in the initial frame, which are then followed through the subsequent frames. This method is less preferred, especially when new objects appear.

Secondly, the approaches were categorized based on their processing mode, distinguishing between online and offline tracking. Online tracking is a fast tracking, which is more suitable for applications like autonomous driving. In contrast, offline tracking processes a batch of frames at a lower frame rate.

Lastly, the authors classified approaches based on the type of output they produce, which can be either stochastic or deterministic. Stochastic tracking results in variations in tracking at different times, while deterministic tracking results in consistent tracking.

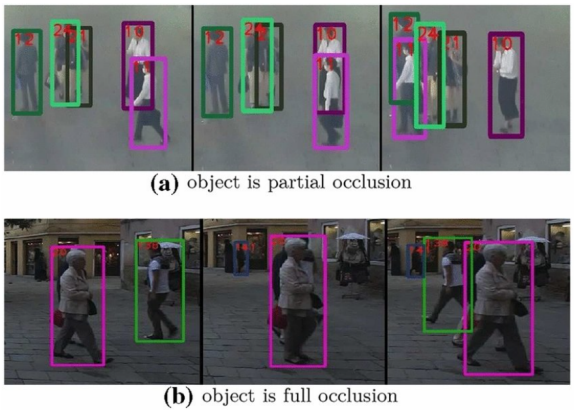
Table 1.3.1: Comparison of Different Tracking Methods

Category	**Method**	**Application**	**Limitation**
Initialization Approach	Detection-based [44]	Reliable detections in clear conditions	May fail in cluttered scenes
	Detection-free [17]	Useful where detection is challenging	Struggles with object variations
Processing Mode	Online Tracking [45]	Suitable for real-time applications	May trade off accuracy for speed
	Offline Tracking [39]	Ideal for detailed tracking	Not suitable for real-time
Output Type	Stochastic [25]	Accounts for randomness	Outcomes can be unpredictable
	Deterministic [18]	Provides precise data	May not always account for random and unforeseen changes

1.4 Multiple Object Tracking Challenges

The monitoring of objects in autonomous vehicles is an ever-evolving field, with numerous challenges that impact the effectiveness of tracking systems. Among these, background interference [21] and obstacles causing occlusion [42] are particularly significant. Background interference occurs when non-target objects or complex scenes disrupt the tracking process, leading to inaccuracies. However, the main focus of this research is on the challenge of occlusion.

Occlusion happens when an object in the view is partially or fully blocked by



(a) object is partial occlusion

(b) object is full occlusion

Fig. 1.4.1: Examples of Full and Partial Occlusions [42]

another object, making it difficult for tracking systems to keep an accurate track of it. This problem is especially common in crowded environments where many objects are present, increasing the likelihood of occlusions. When occlusions occur, tracking systems may lose sight of the object or track it incorrectly, which can greatly reduce the system's overall performance and reliability.

Handling occlusions is complex, particularly when considering both full and partial occlusions, as illustrated in Figure 1.4.1. Partial occlusion means only a part of the object is hidden, which can lead to tracking errors, while full occlusion, where the object is completely hidden, presents an even greater challenge and often results in the object being temporarily lost from the tracking system. Many existing methods struggle to effectively manage these situations, highlighting the need for more advanced techniques that can handle both types of occlusion in real-time.

1.5 Motivation

This research is motivated by the need to improve how tracking systems deal with occlusions in autonomous vehicles. Effectively managing occlusions is crucial for ensuring that tracking systems remain accurate and reliable, which directly affects the safety and efficiency of autonomous driving. By focusing on methods that can handle both full and partial occlusions while also being computationally efficient, this research aims to enable real-time tracking in complex environments. Successfully

tracking objects despite occlusions is essential for the advancement of autonomous vehicle technologies and making them practical for real-world use.

1.6 Research Objectives

The primary objective of this research is to develop a more robust object-tracking system that can effectively manage both full and partial occlusions in real-time scenarios. To achieve this, the system aims to significantly reduce the number of false negatives (FNs) by enhancing its ability to track objects accurately, even when they are partially or fully occluded. Additionally, the research focuses on designing methods that ensure high tracking accuracy while remaining computationally efficient, allowing the system to operate effectively in real time. Ultimately, this work seeks to demonstrate that the proposed methods can improve the resilience of object-tracking systems in autonomous vehicles, ensuring greater reliability in complex, real-world environments.

CHAPTER 2

Related Works

2.1 Object Detection Methods

Detecting objects is essential in multi-object tracking as it provides crucial positional data about objects in an image. These days, many target identification algorithms use Deep Convolutional Neural Networks (CNNs) to achieve a high level of accuracy [14]. These deep learning approaches leverage CNNs to extract object features, thereby boosting both the efficiency and accuracy of recognition. The hierarchical learning capability of CNNs enables the recognition of features at various levels of abstraction, enhancing the precision of feature extraction. As shown in Figure 2.1.1, there are generally two types of object detection techniques: traditional methods and modern methods (based on CNNs). The modern methods are further subdivided into One-stage Detection, Two-stage Detection, and Transformers.

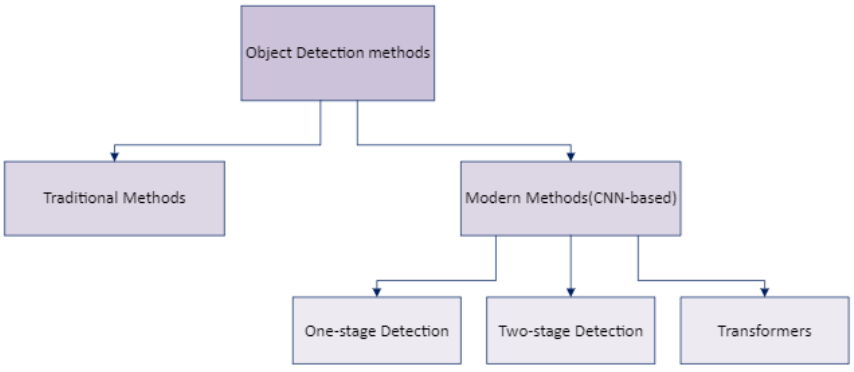


Fig. 2.1.1: Overview of Object Detection Techniques.

2.1.1 Two-stage Detectors

Two-stage detectors involve creating candidate regions and categorizing them in two steps. These approaches, such as the R-CNN series, provide greater detection accuracy compared to traditional methods because they allow for more detailed analysis of each candidate region. However, this comes at the cost of higher computational demands and slower processing times, which can be a limitation for real-time applications.

2.1.1.1 R-CNN series

- **Significant Evolution in Deep Neural Networks**: The R-CNN series signifies a major advancement in deep neural network architectures specifically designed for object detection, with each iteration improving efficiency and precision over its predecessor.

- **R-CNN**: Within computer vision, R-CNN [10] represented a major jump in object detection. This method uses a multi-stage pipeline that consists of three primary stages: the creation of region proposals, the extraction of features, and the refinement and categorization of bounding boxes. Initially, a segmentation process called selective search is used to develop region proposals. This algorithm splits the image into several smaller segments, which are then repeatedly merged to form bigger regions. Bounding boxes are then drawn around these merged areas. Subsequently, each proposed region is processed individually by a CNN to extract feature representations; several CNN architectures can be employed. In the original R-CNN implementation, Support Vector Machines (SVMs) classify each region proposal into predefined object categories, and bounding box regression refines the initial bounding box estimates to enhance localization accuracy.

- **Fast R-CNN**: Fast R-CNN [9] enhances the original R-CNN by passing the entire image through the CNN once to create a feature map, rather than processing each region proposal separately. This approach significantly reduces

computational requirements. A softmax layer identifies objects within these regions and predicts bounding box adjustments after a Region of Interest (RoI) pooling layer extracts and resizes region suggestions.

- **Faster R-CNN**: Faster R-CNN [32] builds upon previous advancements by integrating the Fast R-CNN architecture with a Region Proposal Network (RPN). The RPN, a fully convolutional network, calculates object boundaries and objectness scores at each location to efficiently generate high-quality region proposals. This integration streamlines the object detection process, enhancing the accuracy of region proposals and improving overall performance.

2.1.2 One-stage Detectors

One-stage detection algorithms predict class probabilities and object positions directly, bypassing the region proposal phase. Notable examples include the YOLO (You Only Look Once) series [31], Single-Shot MultiBox Detector (SSD) [23], and RetinaNet [2]. One-stage detectors are known for their speed, making them suitable for real-time applications like object tracking and video analysis. However, this speed often comes at the expense of accuracy, particularly in detecting smaller objects or in complex scenes.

2.1.2.1 YOLO (You Only Look Once)

YOLO is a well-known single-stage detection method that does not use the traditional region proposal phase [31]. It converts each pixel in the image directly into bounding box coordinates and class probabilities, processing the entire image in one step. YOLO's output layer is designed to forecast class probabilities, bounding box coordinates, and confidence scores. This allows it to recognize many objects in a single neural network pass. Compared to more conventional detection techniques, this methodology yields far faster processing speeds by enabling end-to-end object detection, as illustrated in Figure 2.1.2.

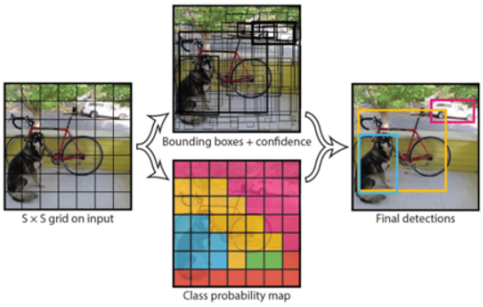


Fig. 2.1.2: Illustration of How YOLO Works [31]

2.1.3 Transformers

Transformers, which became popular in natural language processing, have been adapted for image classification with the introduction of models like Google's Vision Transformer (ViT) [7]. Transformers leverage a wide receptive field to capture long-range dependencies within an image, leading to superior feature extraction capabilities compared to traditional CNNs. However, transformers generally require large datasets and significant computational resources, which can be a limitation in real-time or resource-constrained environments. Additionally, the field of transformers in computer vision is still evolving, and their full potential and drawbacks are areas of ongoing research.

2.2 Multiple Object Tracking Process

MOT is essential for identifying and following multiple objects over time in video sequences. Object Detection and Object Tracking are the two primary stages of the MOT process. Figure 2.2.1 illustrates these stages, which involve several key steps:

1. **Video Sequence**: The system receives a video sequence with various objects that need to be tracked as input.

2. **Object Localization**: In the object detection stage, the system identifies and

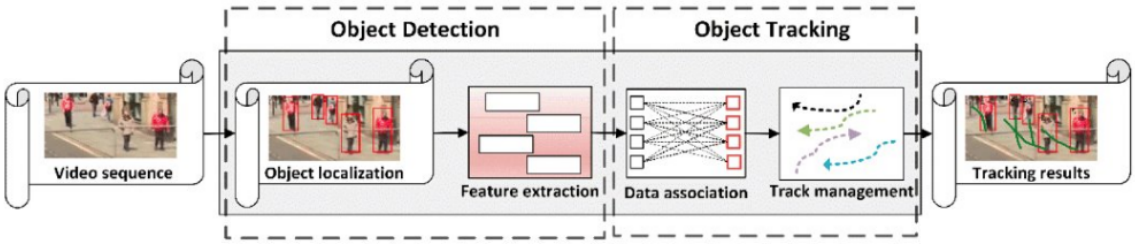


Fig. 2.2.1: The Process of Multiple Object Tracking (MOT)[12].

localizes objects within each frame of the video. This is achieved using techniques like selective search to generate region proposals.

3. **Feature Extraction**: A convolutional neural network (CNN) gathers features for every object it detects,and generates a detailed vector representation for each proposed region.

4. **Data Association**: The features and predicted positions of objects are then used to associate detected objects across frames. This is where techniques like the Hungarian Algorithm come into play, enabling the system to maintain consistent object identities over time.

5. **Track Management**: Tracks that have not been matched for a specific amount of frames must be deleted, new tracks must be created for unmatched detections, and current tracks must be updated with new detections.

6. **Tracking Results**: The final output is a set of tracking results, where each object is consistently identified and followed throughout the video sequence.

2.2.1 Multiple Object Tracking Methods

Several techniques are employed to achieve MOT effectively. The Kalman Filter (KF) relies on motion prediction to estimate future object positions. SORT (Simple Online and Realtime Tracking) enhances the Kalman Filter by using the Hungarian Algorithm for associating predicted positions with the detections [3], enabling real-time performance. Deep SORT further improves tracking by incorporating appearance data from convolutional neural networks (CNNs)..

2.2.1.1 Kalman Filter

The Kalman Filter (KF) is a widely used algorithm in multi-object tracking (MOT) for predicting and updating the positions of objects based on their motion. It is particularly effective in estimating future positions by minimizing the mean square error, making it a popular choice in tracking systems to determine an object's location in the current frame based on its previous state [22].

The Kalman Filter operates in two main phases: prediction and update.

Prediction Phase: In this phase, the Kalman Filter predicts the current state of the object and the uncertainty of this prediction using information from the previous state. This prediction relies on a model that describes how the object's state changes over time.

Update Phase: After making the prediction, the Kalman Filter refines it by incorporating new measurements. The refinement is done by calculating a gain that determines how much the predicted state should be adjusted based on the new measurement. This updated state provides a more accurate estimate of the object's position.

The recursive nature of the Kalman Filter allows it to continuously update its estimates as new data becomes available, making it well-suited for real-time tracking applications, where accurate and timely updates are crucial [22].

2.2.1.2 Simple Online and Realtime Tracking (SORT)

Using the Tracking-by-Detection (TBD) approach, SORT [3] is an effective online method for multiple object tracking (MOT). This approach highlights the usefulness of the algorithm and established a benchmark in the MOT domain at the time of its launch. To identify items in SORT, the authors combine the traditional pedestrian detection model ACF with the CNN-based Faster R-CNN network. Additionally, the program uses the Hungarian algorithm and the Kalman filter to handle prediction and data association.

As the first algorithm in the SORT family, it puts efficiency and speed first. Com-

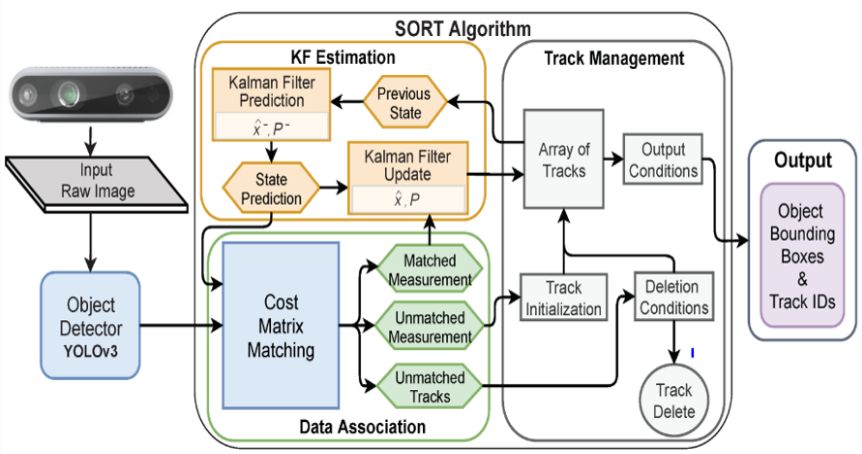


Fig. 2.2.2: SORT Algorithm Overview [29]

pared to the state-of-the-art algorithms of 2016, its processing speed can reach 260 Hz, making it 20 times faster than those earlier algorithms [43]. However, one significant limitation of this technique is that IDs can only be updated by new detections; it is unable to reacquire lost targets. To solve this problem, Deep SORT was developed in response to this weakness.

Using a Kalman Filter (KF), SORT iteratively ascertains the states of tracked objects. The technique uses the Hungarian algorithm [19] to reliably link tracked and identified objects. The Data Association module in SORT is particularly crucial because it compares the measured bounding boxes provided by the object detector with the ones predicted by the KF.

As illustrated in Figure 2.2.2, the process involves inputting raw images into an object detector, followed by the application of the SORT algorithm, which includes KF estimation, data association through cost matrix matching, and track management. The output consists of object bounding boxes and track IDs.

2.2.1.3 Simple Online and Realtime Tracking with a Deep Association Metric (Deep SORT)

By merging appearance data with traditional tracking techniques, Deep SORT improves multiple object tracking [40]. The algorithm forecasts object positions using a Kalman Filter and uses the Hungarian algorithm to associate data between frames

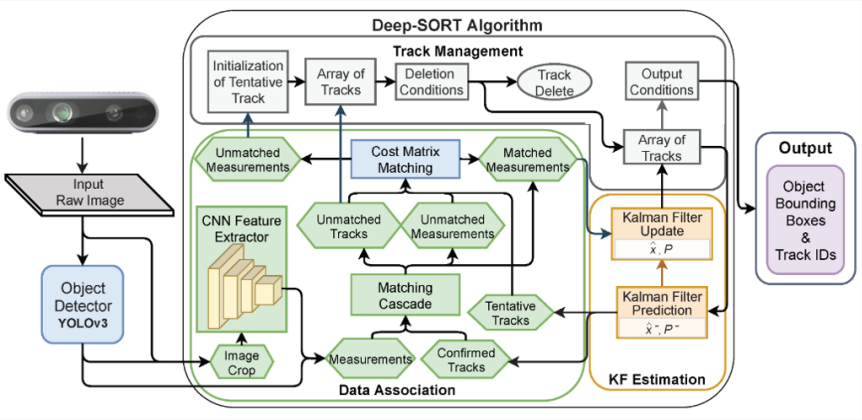


Fig. 2.2.3: Deep SORT Algorithm Overview

by analyzing bounding box overlaps. By extracting both motion and appearance features, a trained convolutional neural network (CNN) enhances the tracker's ability to manage misidentifications and occlusions while maintaining real-time processing capabilities.

In order to improve track associations, Deep SORT [40] integrates appearance data into the SORT algorithm.

As illustrated in Figure 2.2.3, in Deep SORT, the Hungarian algorithm handles the association between tracks and detected bounding boxes through a two-phase matching cascade. Initially, it uses motion and visual criteria to match detections with existing tracks. Afterward, it matches detections to new and tentative tracks using the same approach as SORT.

To incorporate motion data, the squared Mahalanobis distance between the detected states and the anticipated states is calculated by the algorithm. Additionally, the appearance features of detections, obtained from a pre-trained CNN model, are compared using the shortest cosine distance metric. This combination ensures robust and accurate tracking by considering both motion and appearance information.

2.2.2 Similarity Metrics

2.2.2.1 Intersection over Union (IoU)

Intersection over Union (IoU) is a standard metric in multi-object tracking, primarily used for associating detected objects with their corresponding tracklets. It calculates the overlap between predicted positions, typically estimated using a Kalman filter, and the detected bounding boxes. The IoU is defined as the ratio of the intersection area to the union area of two bounding boxes. This metric is particularly effective in scenarios where objects have minimal movement between frames, ensuring accurate associations. Algorithms like SORT [3] rely on IoU for precise object tracking. However, IoU can be less effective in situations involving partial occlusions and small objects, where the overlap between bounding boxes may not be substantial enough for accurate association.

2.2.2.2 Sørensen-Dice Coefficient

The Sørensen-Dice Coefficient, commonly known as the Dice similarity measure, serves as an alternative to IoU, especially in contexts where partial occlusion and small object sizes are prevalent. The Dice coefficient is calculated by dividing twice the area of overlap by the total area of both bounding boxes. Unlike IoU, the Dice coefficient is more sensitive to partial overlaps, which is advantageous in complex tracking environments with frequent occlusions.Figure 2.2.4 illustrates the difference between IoU and the Dice coefficient.

Studies, such as those referenced in [30], have demonstrated that the Dice coefficient can be more effective than IoU in handling scenarios where objects are partially visible or smaller in size. This is because the Dice coefficient accounts for the degree of overlap more robustly, allowing for better detection and tracking in challenging conditions. Specifically, the Dice coefficient has been shown to improve tracking performance in environments where objects frequently enter and exit partial occlusion, providing a more reliable measure for object association under these circumstances.

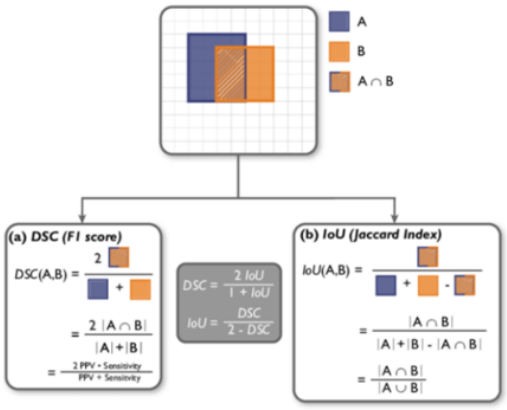


Fig. 2.2.4: Comparison of Intersection over Union and Sørensen-Dice Coefficient [26]

CHAPTER 3

Methodology

3.1 Proposed Methodology

In this section, we introduce the methodology used for enhancing multiple object tracking (MOT) in autonomous vehicles. While pre-detections are utilized to ensure an accurate comparison between the original Deep SORT and our proposed method, our primary focus lies in the tracking component within a tracking-by-detection approach. The method is divided into several key sections, including the use of Deep SORT with the Sørensen-Dice coefficient as a similarity metric, and the implementation of memory management for occluded objects. Each of these methods and algorithms addresses specific challenges in MOT, as outlined in the following sections.

3.1.1 Using Deep SORT with Sørensen-Dice Coefficient and Memory Management for Occluded Objects

Our approach employs an enhanced version of Deep SORT during the tracking phase. This method, illustrated in Figure 3.1.1, integrates several crucial elements designed to enhance tracking efficiency, particularly in managing partial occlusions and small objects. The enhancements include incorporating the Sørensen-Dice coefficient for improved similarity measurement and implementing a memory management system to handle occluded objects. These innovations aim to provide more robust and accurate re-identification of objects, even when they are temporarily obscured.

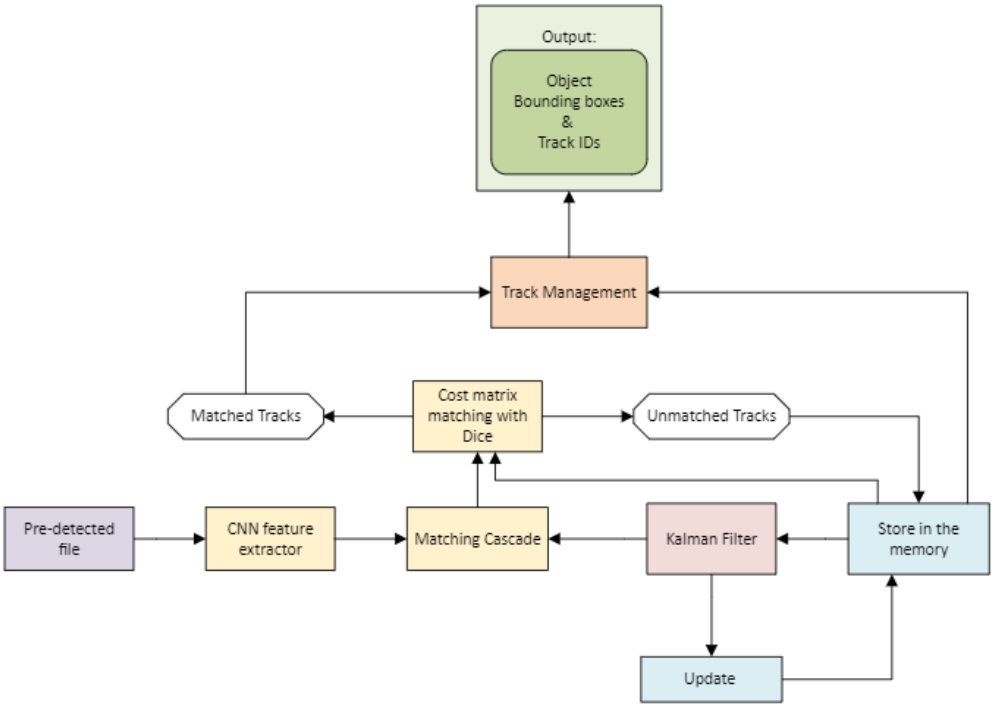


Fig. 3.1.1: Our Proposed Method Process Diagram

3.1.2 Key Components of Deep SORT

3.1.2.1 Kalman Filter

The Kalman Filter plays a pivotal role in estimating the future position of an object based on its previous state. This predictive approach enables the estimation of the current state (e.g., position and velocity) of objects, even amidst noise and uncertainty. The Kalman Filter contributes to smooth and accurate position predictions across frames, which is crucial for effective tracking.

3.1.2.2 Sørensen-Dice Coefficient

To enhance the handling of partial occlusions and small objects, we replace the traditional Intersection over Union (IoU) with the Sørensen-Dice coefficient as the similarity metric. The Sørensen-Dice coefficient offers improved sensitivity to smaller and partially occluded objects, providing a more reliable similarity measure in challenging tracking scenarios.

3.1.2.3 Hungarian Algorithm

The Hungarian Algorithm is employed for data association, efficiently solving the assignment problem. It matches the predicted object states with new detections, ensuring accurate and consistent object tracking across frames.

3.1.2.4 Matching Cascade

The Matching Cascade utilizes a CNN-based method as a matching strategy. It leverages appearance information extracted by a pre-trained CNN to re-identify objects following occlusions. This process helps maintain the consistency of object identities across frames, thereby improving the overall robustness and accuracy of the tracking system.

By integrating these components, Deep SORT enhances the tracking system's performance in several ways:

- **Improved Occlusion Handling**: The inclusion of the Sørensen-Dice coefficient enhances the system's ability to track partially occluded objects, thereby reducing identity switches and false negatives.
- **Robust Data Association**: The combination of the Hungarian Algorithm and Matching Cascade ensures accurate matching of detections with tracks, maintaining the tracking system's consistency and reliability.
- **Accurate Predictions**: The Kalman Filter provides precise predictions of object positions, contributing to smoother and more accurate tracking.

3.1.2.5 Memory Management

A crucial aspect of our methodology is the implementation of memory management for occluded objects. This step ensures that objects that become temporarily undetectable due to occlusions can be effectively re-identified when they reappear, maintaining the integrity of the tracking process.

- **Step 1: Storing Missed Detections**

– **Description**: When the algorithm cannot match some tracks with the detections, it assumes these tracks are occluded objects. To handle these occlusions, the algorithm stores the occluded objects in memory. This memory includes key features such as the last seen position, last seen time, and velocity of each object. This step is crucial for ensuring that temporarily undetected objects can be re-identified when they reappear. The memory storage mechanism uses a dictionary to keep track of these occluded objects, which allows for efficient look-up and updates.

Algorithm 3.1.1 Storing Missed Detections

Track list $\mathcal{T} = \{1, \ldots, N\}$, Memory $\mathcal{M} = \emptyset$ each track $t \in \mathcal{T}$ track t is not detected in the current frame Store the track t in memory $\mathcal{M}$ with features: - Last seen position P_t, Last seen time T_t, and Velocity V_t estimated by Kalman Filter Updated memory $\mathcal{M}$

- **Step 2: Updating Positions of Objects in Memory**

  – **Description**: After storing the occluded objects in memory, the next step is to update their features and predicted positions. This is achieved through a function that uses the Kalman Filter to estimate the new positions of these objects. By continuously updating the objects' features, the algorithm maintains accurate predictions, which are essential for re-identifying the objects when they reappear. The Kalman Filter helps in predicting future positions by considering the object's last known velocity and position, thus ensuring a more precise update of the object's state.

Algorithm 3.1.2 Updating Positions of Objects in Memory

Memory $\mathcal{M}$, Current time $T_{current}$ each object $m \in \mathcal{M}$ Predict future position $P_{predicted}$ of m using Kalman Filter Update position of m to $P_{predicted}$ Update last seen time of m to $T_{current}$ Updated memory $\mathcal{M}$

- **Step 3: Continuous Re-identification of Occluded Objects**

  – **Description**: This step involves continuously re-identifying the occluded objects by matching new detections with the objects stored in memory.

This re-identification process occurs in every frame, ensuring that the algorithm consistently attempts to track the occluded objects. The matching process involves comparing the predicted positions of the occluded objects with new detections using a similarity metric, such as the Sørensen-Dice coefficient, and updating the tracks if a match is found.

Algorithm 3.1.3 Continuous Re-identification of Occluded Objects

Detection list $\mathcal{D} = \{1, \ldots, M\}$, Memory $\mathcal{M}$, Track list $\mathcal{T}$ each detection $d \in \mathcal{D}$ Predict future positions of occluded objects in memory $\mathcal{M}$ using Kalman Filter each object $m \in \mathcal{M}$ detection d matches predicted position of m using matching strategy Update track t with detection d Remove m from memory $\mathcal{M}$ Updated track list $\mathcal{T}$ and memory $\mathcal{M}$

- **Step 4: Memory Cleanup and Performance Efficiency**
  - **Description**: To maintain system efficiency, objects that remain in memory without being re-detected for a certain period are removed. This time-limited deletion prevents the memory from becoming overloaded and keeps the tracking system performant. The cleanup process involves checking the last seen time of each object and removing those that have not been updated within a specified maximum age, ensuring that the system remains efficient and responsive.

Algorithm 3.1.4 Memory Cleanup and Performance Efficiency

Memory $\mathcal{M}$, Current time $T_{current}$, Maximum age A_{max} each object $m \in \mathcal{M}$ $T_{current} - T_m > A_{max}$ Remove m from memory $\mathcal{M}$ Updated memory $\mathcal{M}$

3.2 Implementation Details

The implementation details for our improved Deep SORT model are presented in this section. The steps outlined below cover the setup of the model, the integration of various components, and the execution of the tracking algorithm.

3.2.0.1 Software and Libraries

Our implementation is based on the following software and libraries:

- **Python 3.7**: The primary programming language used for our implementation.
- **NumPy**: For numerical computations and array operations.
- **OpenCV**: For image processing and video frame handling.
- **TensorFlow/Keras**: For implementing and using the pre-trained CNN model for appearance feature extraction.
- **scikit-learn**: For implementing the Kalman Filter and other machine learning utilities.

3.2.0.2 Pre-trained CNN Model

The CNN model used for extracting appearance features is pre-trained using an ample human re-identification dataset. The framework utilized to implement this model is TensorFlow/Keras. The CNN is meant to extract high-dimensional feature vectors from the identified objects, which are then used for matching and re-identification.

3.2.0.3 Data Preparation

We use the MOT16 dataset for benchmarking our model. The dataset includes video sequences with annotated bounding boxes for pedestrians. The data preparation involves the following steps:

- **Frame Extraction**: Extract frames from the video sequences using OpenCV.
- **Bounding Box Annotations**: Load the bounding box annotations provided in the MOT16 dataset.
- **Feature Extraction**: Extract appearance features from the bounding boxes that have been recognized by using the pre-trained CNN model.

CHAPTER 4

Results and Discussion

4.1 Dataset and Evaluation Metrics

4.1.1 MOT16 Dataset

An essential benchmark for assessing tracking algorithms, particularly those aimed at tracking pedestrians, is the Multiple Object Tracking (MOT16) dataset. The dataset includes a wide range of sequences that were taken in different settings, each of which presents a different set of difficulties, such as situations with dense populations, varied camera angles, and lighting conditions that fluctuate.

There are fourteen video sequences in the MOT16 dataset. Ground truth boundary boxes and individual pedestrian identities are painstakingly annotated into every sequence to enable thorough tracking performance assessment. Additionally, the annotations offer details on visibility ratios and occlusion levels, which are critical for determining how well an algorithm can handle objects that are partially or completely obscured.

4.1.2 Challenges in MOT16 Sequences

Each sequence in the MOT16 dataset introduces specific challenges that test the robustness of tracking algorithms:

- **MOT16-02**: Captured in a crowded outdoor market, this sequence presents significant occlusions and frequent interactions between pedestrians, making consistent tracking challenging.

- **MOT16-04**: Filmed at a busy pedestrian crossing, this sequence includes varied lighting conditions and shadows that affect detection accuracy.
- **MOT16-05**: An indoor shopping mall environment with complex background textures and reflections, posing difficulties in distinguishing between pedestrians and background elements.
- **MOT16-09**: features a public space with little pedestrian activity yet quickly changing appearances as a result of people's varied outfits and body positions.
- **MOT16-10**: Recorded at a train station, this sequence has significant scale variations as pedestrians move closer to and further from the camera.
- **MOT16-11**: An outdoor street scene with moving vehicles and varying pedestrian densities, presenting challenges in differentiating between pedestrians and other moving objects.
- **MOT16-13**: A pedestrian street with heavy foot traffic and varying occlusion levels, requiring robust tracking to maintain identities across frames.

4.1.3 Evaluation Metrics

Several common criteria are used to thoroughly assess tracking algorithm performance on the MOT16 dataset:

4.1.3.1 Multiple Object Tracking Accuracy (MOTA)

MOTA is an all-inclusive statistic that accounts for identity shifts, false positives, and false negatives. By taking into account different tracking faults that could happen, it offers a comprehensive assessment of tracking performance. MOTA is computed as:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{1}$$

Where The frame index is indicated by t,
The number of false negatives is represented by FN,

The number of false positives is represented by FP,
IDSW stands for identity switches, and
The number of ground truth objects is represented by GT.

4.1.3.2 Recall

The percentage of true positives that the tracking algorithm accurately identifies is called recall. It's an important metric to know how well the algorithm finds every object in the scene. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Where

TP represents the true positives, and

FN represents the false negatives.

4.1.3.3 False Negatives (FN)

When an object that is present in the ground truth is not detected by the tracking algorithm, this is known as a false negative. Enhancing the accuracy and dependability of the tracking system requires lowering the quantity of false negatives.

4.1.3.4 ID Switches (IDSW)

When the identify of a tracked item is erroneously altered between frames, it is known as an ID switch.

When taken as a whole, these indicators offer a comprehensive evaluation of the tracking algorithm's performance, facilitating a full comparison with alternative approaches. Our methodology intends to overcome the inherent issues in multi-object tracking by utilizing the MOT16 dataset and these assessment measures, with a focus on assuring high recall and MOTA values, as well as enhancing the management of occlusions.

4.2 Results

We evaluated the effectiveness of our strategy using the MOT16 training set. It is very important to emphasize that our methodology relies on a pre-trained model, and our approach was validated using both the training set and testing set of MOT16.

4.2.1 Results in Datasets with Occlusion

Table 4.2.1: Performance Comparison on MOT16 Datasets

Dataset	Method	MOTA	Recall	IDs	FN
MOT16-02	Deep SORT	34.8	36.8	122	18457
	Our Method	35.1	37.0	134	18389
MOT16-09	Deep SORT	39.0	41.3	15	5184
	Our Method	39.6	42.0	20	5121
MOT16-13	Deep SORT	36.0	44.5	197	10683
	Our Method	37.6	44.7	182	10660

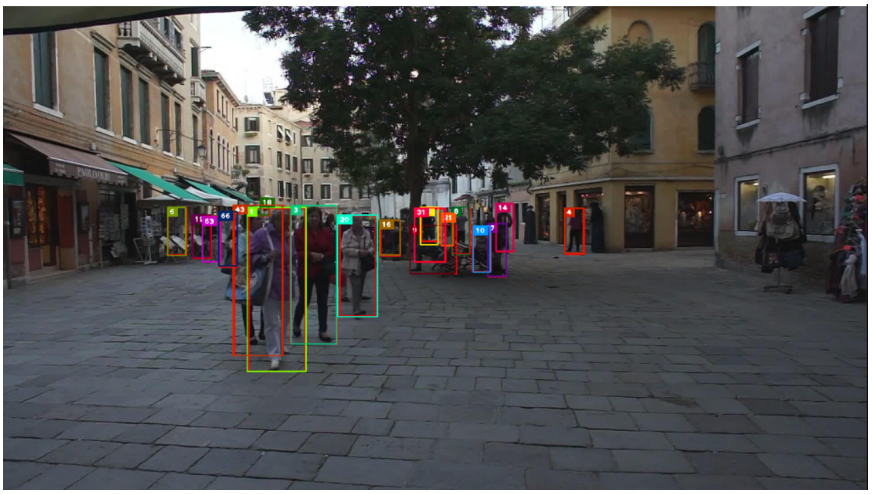


Fig. 4.2.1: Example from MOT02 Dataset

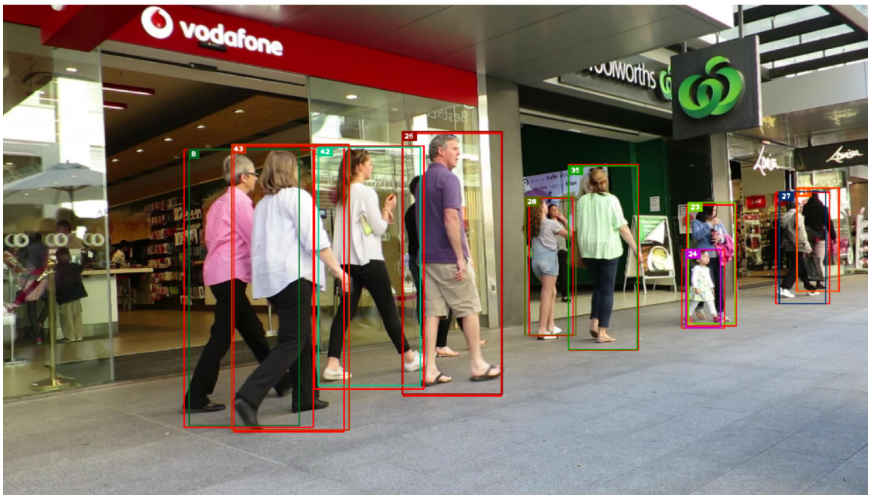


Fig. 4.2.2: Example from MOT09 Dataset

Fig. 4.2.3: Example from MOT13 Dataset

4.2.2 Analysis

MOTA, recall, and False Negative improvements demonstrate the effectiveness of integrating the Sørensen-Dice coefficient and memory management for occluded objects. These enhancements are particularly significant in datasets like MOT16-02 and MOT16-13, characterized by frequent occlusions as shown in the table 4.2.1, and figure 4.2.4.

By addressing this challenge, our method shows a marked improvement in tracking performance, with fewer false negatives and better overall accuracy.

To demonstrate the ability of our tracker to handle partial occlusions, we present

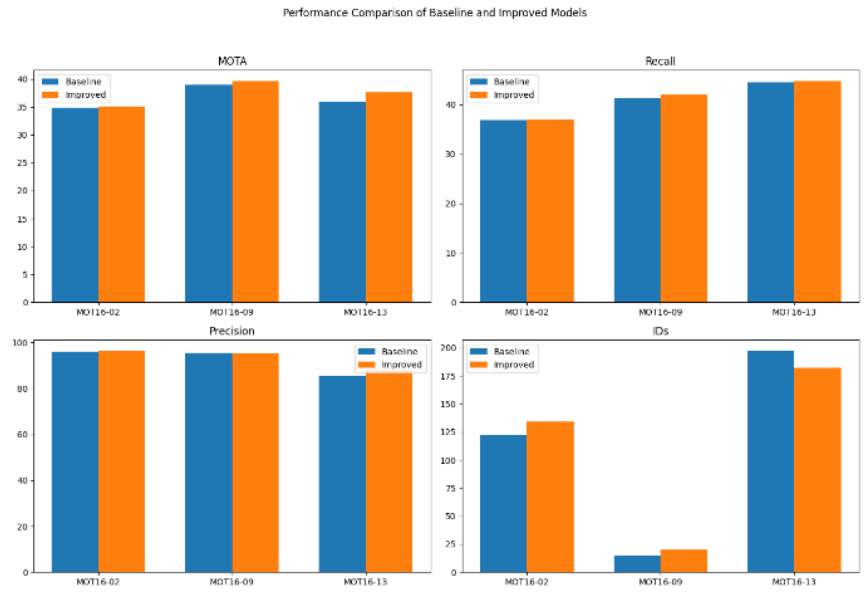


Fig. 4.2.4: Result Diagram in Details on MOT-02, MOT-09, and MOT-13

examples from the MOT-02, MOT-09, and MOT-13 dataset. The figures 4.2.1, 4.2.2, and 4.2.3 illustrate that our tracker can effectively manage partial occlusions, which are prevalent in these datasets. While it is not possible to present full occlusions in images, it is important to note that our methods are also designed to handle full occlusions effectively.

4.2.3 Performance Comparison of Different Models

As illustrated in the table 4.2.2, and figure 4.2.5 the integration of the Dice coefficient alone yielded only modest improvements in specific datasets. However, combining the Dice coefficient with memory management resulted in more substantial enhancements, particularly in handling occlusions. This is evident from the increased MOTA, and recall, as well as the reduction in false negatives (FN) in several datasets. The improvements in these key metrics underscore the efficacy of our proposed methodology in enhancing multi-object tracking performance in complex scenarios.

Table 4.2.2: Performance Comparison of Different Models

Dataset	**Model**	**MOTA**	**Recall**	**FN**	**IDs**
MOT16-02	Deep SORT	34.8	36.8	18457	122
	Deep SORT with Dice	34.8	36.8	18454	124
	Deep SORT with Dice and Memory	35.1	37.0	18389	134
MOT16-04	Deep SORT	31.1	33.4	71921	53
	Deep SORT with Dice	31.1	33.4	71919	55
	Deep SORT with Dice and Memory	31.3	33.9	71433	60
MOT16-05	Deep SORT	48.2	54.1	3524	48
	Deep SORT with Dice	48.4	54.3	3507	48
	Deep SORT with Dice and Memory	48.4	55.0	3453	53
MOT16-09	Deep SORT	39.0	41.3	5184	28
	Deep SORT with Dice	39.0	41.3	5183	28
	Deep SORT with Dice and Memory	39.6	42.0	5121	28
MOT16-10	Deep SORT	50.8	58.2	7078	154
	Deep SORT with Dice	50.9	58.3	7061	148
	Deep SORT with Dice and Memory	51.0	58.0	7114	144
MOT16-11	Deep SORT	62.0	66.5	3375	32
	Deep SORT with Dice	61.8	66.5	3374	33
	Deep SORT with Dice and Memory	62.1	67.4	3289	33
MOT16-13	Deep SORT	36.0	44.5	10683	197
	Deep SORT with Dice	36.5	45.1	10580	201
	Deep SORT with Dice and Memory	37.6	44.7	10660	182

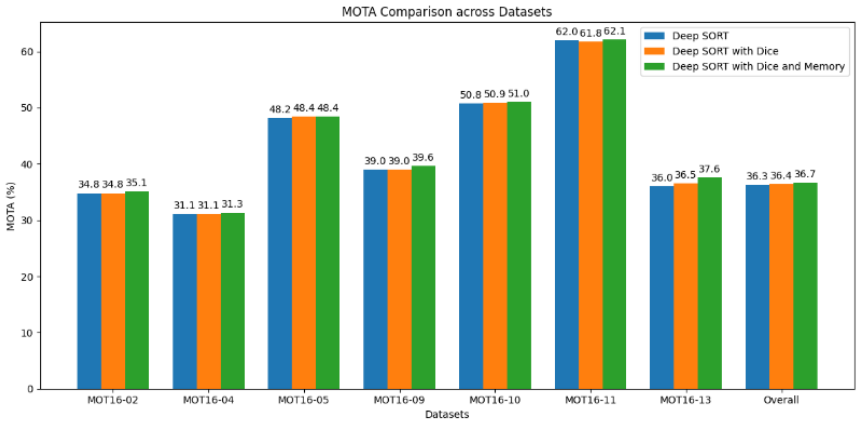


Fig. 4.2.5: MOTA Comparison of Baseline Deep SORT, Deep SORT with Dice Coefficient, and Deep SORT with Dice and Memory

4.2.4 Overall Performance on MOT16 Testing Set

It is important to highlight that these results were obtained by submitting our model's output to the MOTChallenge website for evaluation. The MOTChallenge website provides a standardized platform for evaluating tracking algorithms on the MOT16 testing set, ensuring consistent and reliable comparisons.

Table 4.2.3: Overall Performance Comparison on MOT16 Testing Set

Model	**MOTA**	**Recall**	**IDs**	**FN**
Deep SORT	61.40	68.9	781	56,668
Improved Method	61.84	69.8	894	55,153

4.2.5 Discussion

The comparison of overall performance in Table 4.2.3 demonstrates the improvements achieved by our method over the baseline Deep SORT. Our method shows an increase in MOTA and a significant increase in recall, as well as a reduction in false negatives (FN).

These improvements indicate that our integration of the Dice coefficient and memory management for occluded objects effectively enhances the tracking performance,

especially in complex scenarios with frequent occlusions. Handling occlusions, both partial and full, impacts these metrics by ensuring that occluded objects are correctly identified and re-identified. This reduces the chances of missing objects (thereby improving recall and reducing false negatives) and maintains accurate tracking over time (improving MOTA). The results underscore the efficacy of our proposed methodology in providing a robust solution for multi-object tracking.

CHAPTER 5

Conclusion and Future Work

5.1 Summary of Contributions

In this thesis, we introduced several enhancements to the Deep SORT algorithm aimed at improving its performance in multi-object tracking (MOT) tasks, particularly in challenging scenarios involving occlusions. Our principal contributions are summarized as follows:

- **Integration of the Sørensen-Dice Coefficient**: We replaced the traditional Intersection over Union (IoU) similarity metric with the Sørensen-Dice coefficient, which is more sensitive to smaller and partially occluded objects. This change demonstrated modest improvements in specific datasets.
- **Memory Management for Occluded Objects**: We introduced a memory management system to track occluded objects. By storing missed detections and updating their positions using the Kalman Filter, we effectively re-identified occluded objects when they reappeared.

5.2 Performance Evaluation

Our experimental evaluation demonstrated significant performance gains in multi-object tracking due to the proposed improvements. Key metrics such as MOTA, recall, and the number of false negatives showed notable improvements, especially in datasets characterized by frequent occlusions. The integration of the Sørensen-Dice

coefficient alone yielded modest improvements, while combining the Dice coefficient with memory management resulted in substantial enhancements.

5.3 Conclusion

The improvements observed in MOTA, recall, and the reduction of false negatives underscore the efficacy of our approach in enhancing multi-object tracking performance. The memory management system proved particularly effective in handling occlusions. By storing information about occluded objects and re-identifying them when they reappear, the system reduces the number of false negatives. This ensures that objects are consistently detected and tracked, even when temporarily obscured, thereby improving recall. Additionally, maintaining object continuity enhances tracking accuracy, which is crucial in scenarios with frequent occlusions.

The integration of the Sørensen-Dice coefficient provided a more reliable similarity metric for small and partially occluded objects. The Dice coefficient is more sensitive to differences in overlap compared to traditional IoU metrics, making it better suited for handling cases where objects are partially visible. This sensitivity allows for more accurate matching and re-identification of objects, contributing to the observed improvements in recall and overall accuracy.

However, the increase in the number of identity switches in some datasets suggests that further refinement is needed to balance tracking accuracy and identity preservation. This challenge arises because our method prioritizes high recall and low false negatives, which can sometimes lead to incorrect associations, particularly in densely populated scenes or when objects have similar appearances. The memory management system, while effective in many cases, might occasionally reassign IDs if the appearance features of occluded and reappearing objects are not distinctive enough. This results in an increase in identity switches.

Future work could focus on optimizing the trade-off between these two aspects to achieve even better performance. This could involve refining the feature extraction process to enhance the distinctiveness of appearance features, improving data associ-

ation algorithms, or developing more sophisticated strategies for handling occlusions. Additionally, incorporating advanced techniques like transformer-based models for object detection could provide further improvements. By addressing these issues, it will be possible to reduce identity switches while maintaining or improving recall and accuracy, thereby enhancing the robustness and reliability of multi-object tracking systems.

5.4 Limitations

While our method demonstrated significant improvements in several key metrics, it also resulted in an increase in identity switches (IDs) in some datasets. This rise in IDs can be attributed to the fact that by reducing false negatives (FNs), our method successfully tracks a greater number of objects, which inherently increases the likelihood of identity switches. This is a normal consequence when more objects are tracked, as the system needs to handle more complex scenarios with overlapping and similar-looking objects.

Additionally, while our approach effectively reduced FNs and enhanced overall tracking accuracy, it did expose some limitations in managing variations in object appearances. These variations further contributed to the increase in IDs. Therefore, while our method significantly improves tracking performance, particularly in reducing FNs, it also highlights the need for more advanced techniques to manage identity consistency, especially in complex and diverse environments.

5.5 Future Work

Several avenues for future research could build on the findings of this thesis:

- **Refinement of Memory Management**: Enhancing the memory management system to further reduce identity switches and improve re-identification accuracy. This could involve developing more sophisticated algorithms for managing occluded objects and refining the criteria for storing and retrieving track

information.

- **Feature Extractor for Occlusion Management**: Incorporating a feature extractor to extract appearance features, in addition to motion features, for re-identifying objects. This approach aims to handle occlusion management more effectively and make the re-identification function of our method more robust. Integrating deep learning models that can adaptively learn appearance changes over time could significantly enhance the system's resilience to occlusions.

- **Advanced Kalman Filter Techniques**: Exploring the use of advanced Kalman filter techniques, such as adaptive or interacting multiple model (IMM) filters, to improve the prediction of object movements and reduce the errors in trajectory estimation.

- **Larger Dataset and Transformer Models**: Planning to work on a more extensive dataset to validate and enhance the robustness of our method. Additionally, leveraging transformer-based models for object detection could provide significant improvements in accuracy and handling complex scenes, given their ability to capture long-range dependencies and context in the data.

By focusing on these areas, we aim to further enhance the performance of multi-object tracking systems and address the current limitations, particularly the challenge of identity switches.

REFERENCES

[1] Ahlberg, M. (2019). Optimization based trajectory planning for autonomous racing.

[2] Alhasanat, M. N., Alsafasfeh, M. H., Alhasanat, A. E., and Althunibat, S. G. (2021). Retinanet-based approach for object detection and distance estimation in an image. *International Journal on Communications Antenna and Propagation (IRECAP)*, 11(1):1–9.

[3] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, number 38, pages 3464–3468. IEEE.

[4] Bilik, I., Longman, O., Villeval, S., and Tabrikian, J. (2019). The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE signal processing Magazine*, 36(5):20–31.

[5] Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., and Kasturi, R. (2009). Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, 11(20):206–224.

[6] Chandrakar, R., Raja, R., Miri, R., Sinha, U., Kushwaha, A. K. S., and Raja, H. (2022). Enhanced the moving object detection and object tracking for traffic surveillance using rbf-fdlnn and cbf algorithm. *Expert Systems with Applications*, 191:116306.

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner,

T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, (34).

[8] Evans, K., de Moura, N., Chauvier, S., Chatila, R., and Dogan, E. (2020). Ethical decision making in autonomous vehicles: The av ethics project. *Science and engineering ethics*, 26:3285–3312.

[9] Girshick, R. (2015). Fast r-cnn. *arXiv*, (28).

[10] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number 49, pages 580–587, Columbus, OH, USA. IEEE.

[11] González, A., Fang, Z., Socarras, Y., Serrat, J., Vázquez, D., Xu, J., and López, A. M. (2016). Pedestrian detection at day/night time with visible and fir cameras: A comparison. *Sensors*, 16(9):820.

[12] Guo, S., Wang, S., Yang, Z., Wang, L., Zhang, H., Guo, P., Gao, Y., and Guo, J. (2022). A review of deep learning-based visual multi-object tracking algorithms for autonomous driving. *Applied Sciences*, 12(21):10741.

[13] Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(16):334–352.

[14] Hu, X., Cao, D., Li, L., Xu, D., Zhou, Y., Yang, D., and Liu, Y. (2019). Sinet: A scale-insensitive convolutional neural network for fast vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 20(26):1010–1019.

[15] International, S. (2018). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE international*, 4970(724):1–5.

[16] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[17] Kim, M., Kim, I., Yong, J., and Kim, H. (2023). Scheduling framework for accelerating multiple detection-free object trackers. *Sensors*, 23(22):3432.

[18] Krieger, E. W., Sidike, P., Aspiras, T., and Asari, V. K. (2017). Deterministic object tracking using gaussian ringlet and directional edge features. *Optics & Laser Technology*, 95:133–146.

[19] Kuhn, H. W. (1955a). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(39):83–97.

[20] Kuhn, H. W. (1955b). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(45):83–97.

[21] Li, T., Ding, F., and Yang, W. (2021). Uav object tracking by background cues and aberrances response suppression mechanism. *Neural Computing and Applications*, 33(48):3347–3361.

[22] Li, X., Wang, K., Wang, W., and Li, Y. (2010). A multiple object tracking method using kalman filter. In *The 2010 IEEE International Conference on Information and Automation*, number 36, pages 1862–1866.

[23] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, volume 9905 of *Lecture Notes in Computer Science*, pages 21–37. Springer International Publishing, Cham.

[24] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K. (2021a). Multiple object tracking: A literature review. *Artificial intelligence*, 293:103448.

[25] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K. (2021b). Multiple object tracking: A literature review. *Artificial Intelligence*, 293(24):103448.

[26] Maier-Hein, L., Menze, B., et al. (2022). Metrics reloaded: Pitfalls and recommendations for image analysis validation. *arXiv. org*, (2206.01653).

[27] Maurer, M., Gerdes, J. C., Lenz, B., and Winner, H. (2016). *Autonomous driving: technical, legal and social aspects*. Number 4. Springer Nature.

[28] Olmeda, D., de la Escalera, A., and Armingol, J. M. (2011). Far infrared pedestrian detection and tracking for night driving. *Robotica*, 29(8):495–505.

[29] Pereira, R., Carvalho, G., Garrote, L., and Nunes, U. J. (2022). Sort and deep-sort based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(40):1319.

[30] Razzok, M., Badri, A., El Mourabit, I., Ruichek, Y., and Sahel, A. (2023). Pedestrian detection and tracking system based on deep-sort, yolov5, and new data association metrics. *Information*, 14(46):218.

[31] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *arXiv*, (30).

[32] Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(29):1137–1149.

[33] Royo, S. and Ballesta-Garcia, M. (2019). An overview of lidar imaging systems for autonomous vehicles. *Applied sciences*, 9(19):4093.

[34] Song, Y.-M., Yoon, K., Yoon, Y.-C., Yow, K. C., and Jeon, M. (2019). Online multi-object tracking with gmphd filter and occlusion group management. *IEEE access*, 7(47):165103–165121.

[35] Van Brummelen, J., O'brien, M., Gruyer, D., and Najjaran, H. (2018). Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies*, 89(2):384–406.

[36] Vargas, J., Alsweiss, S., Toker, O., Razdan, R., and Santos, J. (2021). An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(12):5397.

[37] Wang, C., Wang, X., Hu, H., Liang, Y., and Shen, G. (2022). On the application of cameras used in autonomous vehicles. *Archives of Computational Methods in Engineering*, 29(6):4319–4339.

[38] Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(19):3–19.

[39] Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, volume 12356 of *Lecture Notes in Computer Science*, pages 107–122. Springer International Publishing, Cham.

[40] Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, number 41, pages 3645–3649. IEEE.

[41] Yang, C., Shi, Y., Li, L., and Wang, X. (2020). Efficient mode transition control for parallel hybrid electric vehicle with adaptive dual-loop control framework. *IEEE Transactions on Vehicular Technology*, 69(3):1519–1532.

[42] Zhang, X., Wang, X., and Gu, C. (2021). Online multi-object tracking with pedestrian re-identification and occlusion processing. *The Visual Computer*, 37(5):1089–1099.

[43] Zhang, Y., Chen, Z., and Wei, B. (2020a). A sport athlete object tracking based on deep sort and yolo v4 in case of camera movement. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, number 37, pages 1312–1316.

[44] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Bao, F., Li, Z., Zhang, X., Wei, Y., and

Sun, J. (2022). Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, number 14, pages 1–21.

[45] Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. (2020b). Ocean: Object-aware anchor-free tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 771–787. Springer.

VITA AUCTORIS

NAME:	Kimia Gholizadeh
PLACE OF BIRTH:	Iran
YEAR OF BIRTH:	1998
EDUCATION:	
	University of Windsor, M.ASc in Electrical Engineering, Windsor, Ontario, 2024