UNIVERSITY KASDI MERBAH OF OUARGLA

FACULTY of New Technologies of Information and Communication

DEPARTMENT of Electronics and Telecommunications

ACADEMIC MASTER Dissertation

Domain : Science and Technology

Specialty : Telecommunication Systems

Presented by :

❑ Siham Babkar

Topic

Enhancing Autonomous Vehicles Perception : Multi-Object Tracking With An Occlusion Handling Approach

Submit Date : June 2025

before the jury :

❑ Ms.Nadia Dahraoui	President	UKM OUARGLA
❑ Mr.Khaled Bensid	Examiner	UKM OUARGLA
❑ Mr.Azeddine Benlamoudi	Supervisor	UKM OUARGLA
❑ Ms. Chahira Kezzal	Co-Supervisor	UKM OUARGLA

University year : 2024/2025

Abstract

Our thesis presents an enhanced multi-object tracking framework built upon the OC-SORT pipeline, designed to improve robustness in challenging autonomous driving scenarios, particularly under occlusions and missed detections. The proposed method integrates state-of-the-art object detectors—YOLOv8, YOLOv9, and the latest YOLOv11. Experimental evaluation was conducted on the KITTI . The system's performance was assessed using standard tracking metrics such as HOTA, MOTA, IDF1, MT, and ML. Among all configurations, the combination of YOLOv11 with OC-SORT achieved the best results, demonstrating superior detection quality, identity preservation, and long-term tracking reliability. Specifically, it reached 61.07% HOTA, 62.51% MOTA, and 75.67% IDF1, outperforming existing state-of-the-art trackers.

Keywords— Autonomous Vehicles, Perception, Multi-Object Tracking, YOLOv11, OC-SORT.

ملخص

تقدم رسالتنا إطارًا محسّنًا لتتبع الأجسام المتعددة يعتمد على خط أنابيب OC-SORT، مصمم لتعزيز المتانة في سيناريوهات القيادة الذاتية الصعبة، لا سيما في حالات التغطية الجزئية وفقدان الكشف. تدمج الطريقة المقترحة أحدث كاشفات الأجسام مثل YOLOv8 وYOLOv9 وأحدثها YOLOv11.

تم إجراء التقييم التجريبي على مجموعة بيانات KITTI. وتم قياس أداء النظام باستخدام مقاييس تتبع معيارية مثل HOTA وMOTA وIDF1 وMT وML. بين جميع التكوينات، حقق الدمج بين YOLOv11 وOC-SORT أفضل النتائج، مما يدل على جودة كشف متفوقة، والحفاظ على الهوية، وموثوقية تتبع طويلة الأمد. حيث بلغ الأداء 61.07% في HOTA، و62.51% في MOTA، و75.67% في IDF1، متفوقًا على الطرق الحديثة الأخرى في التتبع.

كلمات مفتاحية- المركبات ذاتية القيادة ،إدراك البيئة ،تتبع الأجسام المتعددة ، YOLOv11 ، OC-SORT.

ACKNOWLEDGMENTS

First of all, I thank **ALLAH**
for granting me patience and endurance in the face of the challenges I faced throughout my journey. I have felt His guidance every day of my life.

I would like to extend my gratitude and appreciation to my supervisor, Mr. AZEDDINE BENLAMOUDI, for his academic guidance, and to Ms. CHAHIRA KEZZAL and SELSABIL BENDERRADJI for their support and guidance.

I also thank the committee members, Ms. NADIA DAHRAOUI (President) and Mr. KHALED BENSID (Examiner). Every word from you was the heartbeat of this thesis, and every piece of advice was a building block in the construction of this humble edifice. Without your support and generous guidance, this step would not have been completed in this way. Thank you from the heart, and may your knowledge remain a beacon illuminating the paths of your students, and may your giving continue to be a witness to generations that are proud that you are part of their journey.

I also extend my deepest appreciation to all the professors in the Communications Department who taught us and whose skills aided us in our studies.

DEDICATIONS

To those who planted in my heart a love of knowledge and instilled in me the belief that nothing is impossible with trust in God...

To my dear father, the man of great heart and high ambition, and to my precious mother, the woman of sincere prayers and boundless giving,

To my brothers and sisters, you are the light that has accompanied my path.

To my brother's wives and nephews, Yazan and Tawfiq, the spirit of my soul...

To my uncle and maternal uncle, the men of true courage, I would not have arrived without the determination you instilled within me.
In my name and in the name of every dream that has come true, I say: Thank you from the bottom of my heart.

To all members of my esteemed family, from the oldest to the youngest. To those who

have departed this world, but whose spirits remain present in my heart...

To myself, SIHAM: who believed in reaching my goal despite the fatigue,
who continued on the path despite the stumbling blocks,
who fell but did not surrender, who grew tired but did not weaken, who waited and is now reaping the fruits.

To the self who learned how to dream, how to be patient, and how to rise stronger each time...

I dedicate this work as a token of pride, love, and appreciation for what I have achieved, and for what I will continue to achieve,
God willing. I hope that this will be the beginning of good things to come, and that it will be recorded in the scale of good deeds of those who taught me, guided me, and helped me.

Contents

List of tables

List of figures

List OF abbreviation

ADS:	Automated Driving System
AV:	Autonoumous vehicles
C2PSA:	Gross-Stage Partial Spatial Attention
CNN:	Convolutional Neural Networks
FPN:	Feature Pyramid Network
GELAN:	Generalized Efficient Layer Aggregation Network
GNSS:	Global Navigation Satellite System
HOTA:	Higher Order Tracking Accuracy
IDF:	Identification
IMU:	Inertial Measurement Units
KF:	Kalman Filter
ML:	Mostly Lost
MOT:	Multi Object Tracking
MOTA:	Multiple Object Tracking Accuracy
MT:	Mostly Tracked
OCM:	Observation-Centric Momentum
OCR:	Observation based Candidate Recovery
ODD:	Operational Design Domain
ORU:	Observation-Centric Re-Update
PGI:	Programmable Gradient Information
SPPF:	Spatial Pyramid Pooling Fast
SSD:	Signale Shot MoltiBox Detector
YOLO:	You Only Look Once

Chapter 1

General Introduction

1.1 General Introduction

Artificial Intelligence (AI) has emerged as a transformative force across a wide range of domains, enabling machines to perform tasks that traditionally required human intelligence. In recent years, the integration of AI into the field of autonomous driving has brought about significant advancements, particularly in the development of perception systems that allow vehicles to interpret and interact with complex and dynamic environments [1]. These perception systems form the backbone of autonomous vehicles, allowing them to sense, understand, and respond to their surroundings with minimal or no human intervention[2]. Central to the perception stack in autonomous driving are the tasks of object detection and multi-object tracking (MOT). Object detection involves identifying and localizing relevant road users such as vehicles, pedestrians, and cyclists within sensor data [3]. Once detected, these objects must be consistently tracked across frames—a task known as multi-object tracking—which maintains object identities over time and facilitates motion prediction[4], interaction modeling, and decision-making [5]. These capabilities are crucial for ensuring safe navigation, collision avoidance, and compliance with traffic rules.

Modern MOT systems often operate under the tracking-by-detection paradigm, where detection results serve as inputs to the tracking process. Despite the success of this approach, performance significantly degrades in the presence of occlusion, where an object is partially or fully obscured by another object or by scene structures[6]. Occlusion leads to missed detections, identity switches, and fragmented tracks, posing a serious threat to the reliability of autonomous driving systems.

To address these challenges, recent research has explored occlusion-aware strategies[7], such as motion modeling, appearance-based re-identification, and temporal consistency mechanisms [8]. However, achieving robust performance in highly occluded scenarios remains an open problem. As vehicles operate in increasingly dense urban environments, it becomes critical to enhance tracking algorithms with mechanisms specifically designed to handle occlusion.

This thesis focuses on improving the perception module of autonomous vehicles by developing a multi-object tracking approach that explicitly addresses occlusion. This work contributes to the broader goal of building safer, more reliable autonomous driving systems capable of robust performance under real-world constraints.

The primary contributions of this thesis are summarized as follows:

1. **First integration of OC-SORT with recent YOLO versions for autonomous driving:** This work is the first to explore and benchmark the integration of the OC-SORT tracking algorithm with state-of-the-art YOLO detectors—YOLOv8, YOLOv9, and YOLOv11—in the context of autonomous driving. The synergy between advanced detection and observation-centric tracking improves overall system performance under real-world driving conditions.
2. **Comprehensive evaluation on real-world driving data:** We conduct rigorous experiments on the KITTI benchmark dataset, demonstrating that the proposed system—especially the YOLOv11 + OC-SORT configuration—outperforms other setups across multiple evaluation metrics including HOTA, MOTA, IDF1, MT, and ML.
3. **Robustness to challenging visual conditions:** Our system maintains strong performance even under adverse conditions such as variable lighting, shadows, crowded traffic scenes, and especially occlusions, showcasing its real-world applicability in complex urban driving environments.

Chapter 2

Chapter 1:From Sensing to Occlusion Handling: Enabling Perception in Autonomous Vehicles

2.1 Introduction

Autonomous driving technology is revolutionizing modern transportation by enabling vehicles to perceive, interpret, and navigate their environments with minimal human intervention. This chapter establishes the foundational concepts behind these intelligent systems, beginning with the standardized levels of autonomous driving—from partial driver assistance to fully autonomous operation, where no human input is required. At the heart of self-driving vehicles are perception systems powered by advanced sensors such as cameras, radar, and LiDAR, which serve as the vehicle's ”eyes.” Yet, raw sensor data alone is not enough; autonomous systems must also track and predict the behavior of multiple objects, even under challenging conditions like occlusions or complex interactions. This chapter examines these key components and the difficulties they present in ensuring safe and reliable decision-making. By providing an overview of autonomous driving architectures and perception mechanisms, this chapter prepares readers for the in-depth exploration of detection, tracking, and Occlusion handling techniques that follow in subsequent sections.

2.2 Autonomous Vehicles Definition

Autonomous vehicles (AVs) emerged as an innovative concept in the 20th century, representing a paradigm shift in modern transportation. These vehicles are characterized by their ability to drive and make decisions without direct human intervention, utilizing an integrated system of advanced technologies. AVs operate using a combination of sensors and cameras, along with global positioning systems (GPS), radar, and LiDAR (light detection and ranging) technology, which uses lasers to accurately map the surrounding environment. These devices continuously collect data from the environment, giving the vehicle an accurate perception of its surroundings. This data is processed using artificial intelligence algorithms, enabling the vehicle to recognize obstacles, understand traffic signals, and make autonomous decisions while driving in various conditions [9].

2.2.1 Levels of Autonomous vehicles

The Society of Automotive Engineers (SAE) defines six levels of automation for vehicles, ranging form no automation (Level 0) to full automation (Level 5). Fig.2.1. illustrates these various levels.

2.2.1.1 Level 0 (No Automation)

The driver is entirely responsible for the driving task although active safety systems may offer some assistance.

2.2.1.2 Level 1 (Driver Assistance)

The Automated Driving System (ADS) can control either longitudinal or lateral movements, but not both simultaneously. The driver must always be ready to regain control. Typical examples include electronic stability control, anti-lock braking systems, and adaptive cruise control.

2.2.1.3 Level 2 (Partial Automation)

Here, the ADS manages both longitudinal and lateral controls, yet it cannot handle all aspects of the driving task autonomously. The driver must remain present and prepared to take over if necessary. Additional features, such as automatic emergency braking and collision avoidance, are included.

2.2.1.4 Level 3 (Conditional Automation)

Under specific conditions, the ADS can perform all driving tasks independently. However, if the operational parameters are exceeded or a critical failure occurs, the system alerts the driver to intervene.

2.2.1.5 Level 4 (High Automation)

In defined environments with detailed mapping and infrastructure, the vehicle can operate without human intervention. Although a driver can intervene if desired, the ADS is designed to manage fallback scenarios safely.

2.2.1.6 Level 5 (Full Automation)

At this level, the ADS controls all aspects of the driving task across any Operational Design Domain (ODD). Vehicles are designed to function autonomously under all conditions, with the driver only taking over by choice [10].

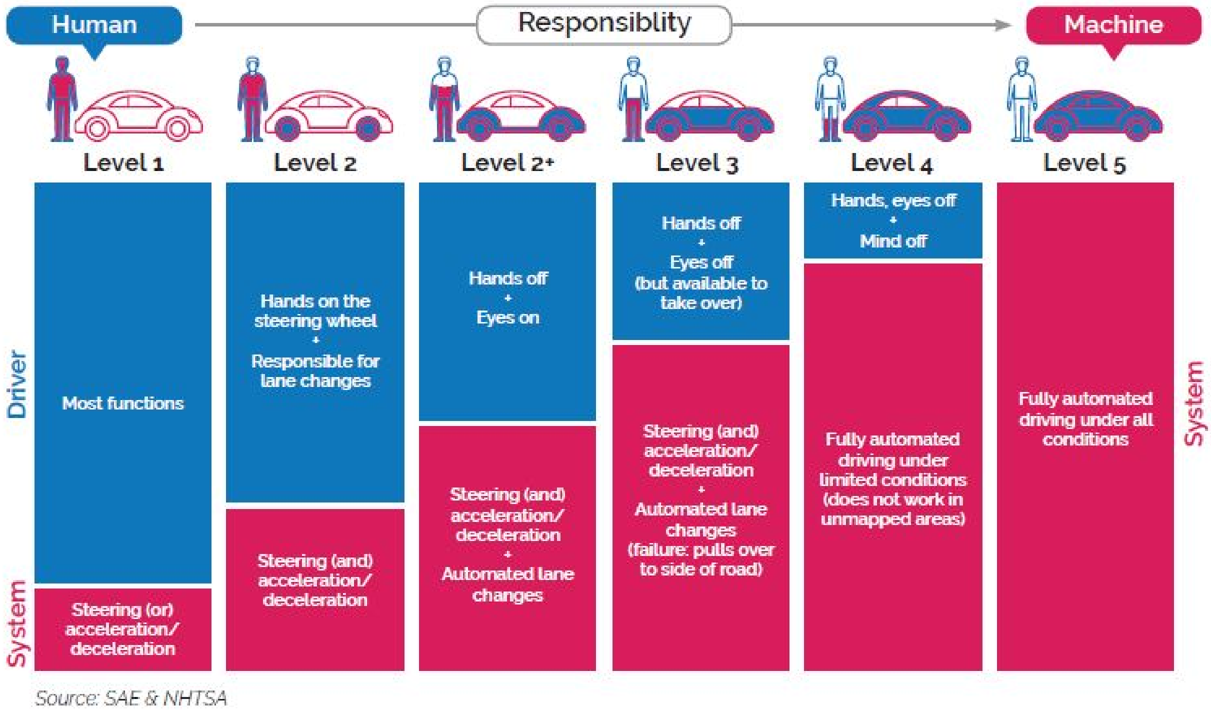


Figure 2.1: Summary of levels of driving automation.

2.3 Perception

An Autonomous Vehicle (AV) must be able to perceive its surroundings independently in order to gather critical information and make accurate driving decisions. By combining data from these sensors, the AV can create a reliable model of the world around it and navigate safely and intelligently [11].

2.3.1 Sensors

Sensors are devices that transform environmental changes or events into measurable data for analysis. They can be categorized based on their operating principles and modes of operation:

- Proprioceptive Sensors (Internal State Sensors): These sensors monitor the internal state of a system by measuring parameters such as force, angular rate, wheel load, or battery voltage. Examples include Inertial Measurement Units (IMUs), encoders, gyroscopes, magnetometers, and Global Navigation Satellite System (GNSS) receivers.
- Exteroceptive Sensors (External State Sensors): These sensors gather information directly from the environment, such as distance or light intensity. Common examples are cameras, radar, LiDAR, and ultrasonic sensors.

Sensors are also classified by their mode of operation:

- Passive Sensors: These rely on ambient energy from the environment, such as vision cameras.
- Active Sensors: These emit energy into the environment and measure the response, as seen in LiDAR and radar systems. In autonomous vehicles (AVs), sensors are critical for environmental perception and vehicle localization. They detect obstacles, interpret visual cues, and determine the vehicle's position. Localization is divided into:
- Relative Localization: Positioning the vehicle relative to nearby landmarks.
- Absolute Localization: Determining the vehicle's position in a global reference frame.

AVs typically integrate cameras, radar, LiDAR, ultrasonic sensors, GNSS, IMUs, and odometry systems to achieve a comprehensive understanding of their surroundings. The strategic placement of these sensors ensures full environmental coverage, which is essential for accurate object detection and safe driving maneuvers. Relying on a single sensor type presents significant challenges in autonomous driving. This section concludes with an evaluation of the strengths and limitations of the three primary sensors—cameras, LiDAR, and radar—for environmental perception in AV applications [12].

2.3.1.1 Camera

Autonomous vehicles (AVs) are equipped with visible-light cameras to provide a 360-degree view of their surroundings. These cameras excel at object detection and recognition, transmitting the collected data to AI-based algorithms for further processing. However, their accuracy declines in low-light conditions, and they generate a substantial

Figure 2.2: Waymo self-driving car.

amount of data that requires extensive processing. To enhance performance in poor visibility, AVs also utilize infrared cameras [11].

2.3.1.2 LiDAR: Light Detection and Ranging

LiDAR sensors use light to measure distances by calculating the time it takes for emitted laser beams to reflect off objects and return to a receiver. These laser beams bounce off the environment and form a point cloud, generating a 3D representation of the surroundings. While LiDAR is a highly effective and precise sensing technology, it is also costly [11].

2.3.1.3 RADAR: Radio Detection and Ranging

This sensor employs radio waves to determine key factors such as distance, velocity, and angle. AVs use radar transmitters to emit radio waves, which are then reflected back and received by radar receivers. Radar functions effectively in most weather conditions and over long distances, though it may occasionally misidentify objects [11].

2.4 Object Tracking

Object tracking is the process of identifying and following the movement of objects over time using a camera in video sequences. It aims to establish a connection between target objects across consecutive frames, relying on their location, shape, or distinctive features within the video frames [13]. Object tracking is classified into two main models: 2D object tracking and 3D object tracking. Selecting the appropriate tracking model is crucial for developing safer and more efficient tracking systems.

2.4.1 2D Object Tracking

This type of tracking relies on processing images captured by monocular cameras, which provide a flat, two-dimensional view of the scene. 2D tracking algorithms use features such as edges, corners, and textures to detect and track objects within an image. While this approach is effective, it has some limitations, including the inability to measure depth or accurately understand the three-dimensional structure of a scene. These limitations can affect the accuracy of motion prediction and decision-making processes.

2.4.2 3D Object Tracking

To overcome the limitations of 2D tracking, more advanced technologies are used, such as stereo vision and LIDAR sensors, which enable the collection of three-dimensional data about objects and their surrounding environment. 3D tracking provides precise information about an object's position, height, and depth, allowing for more accurate motion prediction. For instance, stereo cameras estimate depth by analyzing differences between images captured from two slightly different angles, while LIDAR generates a 3D point cloud, offering a detailed and accurate representation of the environment. By utilizing 3D object tracking, perception systems can gain a deeper understanding of the surroundings, improving object analysis and enabling more precise decision-making. This makes 3D tracking particularly valuable for applications requiring high accuracy in location and distance estimation, such as autonomous driving and mobile robotics [14].

2.5 Occlusion Handling Methods in Object Tracking

Occlusion is an inevitable challenge in object tracking, occurring both during and after an occlusion event. Two primary issues arise during occlusion. First, when two foreground objects overlap, their visual features merge, making it difficult to accurately distinguish and assign pixels to the respective objects. Second, occlusion limits or completely obstructs the visibility of a tracked object, making it challenging to determine its precise location.

Post-occlusion, especially after full occlusion, the primary difficulty lies in re-associating the reappearing object. When an object reappears, it is often unclear whether it is a new object entering the scene or the same object emerging from occlusion. This problem is further complicated when tracking multiple objects with similar appearances. Solutions to these challenges are collectively known as **occlusion recovery methods**. Various strategies have been proposed to handle occlusion, and they can be categorized based on the specific problems they address.

2.5.1 Depth Analysis

A key method for resolving merged foreground blob issues involves estimating the depth of tracked objects. Depth estimation enables the separation of overlapping objects, as the object closer to the camera will have a shorter depth value than objects positioned further away. This facilitates segmentation of occluded objects.

Several techniques have been explored for estimating object depth and scene structure. Some researchers have used stereo cameras to compute depth from video sequences. Another approach involves generating probability density functions for scene depth at each pixel using a training set of detected blobs. This method, which projects a 3D scene into a 2D representation, explicitly models the occlusion landscape to improve tracking accuracy.

2.5.2 Fusion Methods

A common strategy for handling complete occlusion in tracking is fusing object motion and appearance models. This fusion is typically achieved using linear dynamic models or nonlinear dynamics, which continue predicting an object's location during occlusion until it reappears.

For example, Kalman filters have been employed to estimate object position and motion, helping maintain object tracking even when visibility is lost. More advanced non-linear dynamic models, such as particle filters, have also been used for state estimation, offering improved robustness in handling occlusions.

2.5.3 Optimal Camera Placement

Occlusion occurrences can be minimized through strategic camera placement. For instance, installing a 360-degree ultra-wide-angle camera on the ceiling provides a bird's-eye view, preventing occlusions between objects on the ground. However, wide-angle cameras introduce image distortions, which can make objects appear visually similar and harder to distinguish.

An alternative solution is increasing the number of cameras in a surveillance system. When multiple cameras cover the same area, an object occluded in one camera's view may still be visible in another, thereby reducing the likelihood of tracking loss. Several studies have explored multi-camera setups to effectively mitigate occlusion challenges.

By leveraging depth analysis, sensor fusion, and optimized camera placement, occlusion handling in object tracking can be significantly improved, leading to more accurate and reliable tracking performance [15].

2.6 Conclusion

In this chapter, we learned the basic concepts that help us understand how self-driving cars work. We discussed the levels of autonomy, the types of sensors that help a car "see" its surroundings, the importance of tracking objects on the road, and various occlusion management techniques. This information serves as a necessary introduction to help us in the following chapters, where we will delve deeper into the methods and algorithms these cars use to detect and track objects. Understanding these basics makes us better able to comprehend how these advanced technologies are built and why they are so important.

Chapter 3

Chapter 2: From YOLO to OC-SORT: Enhancing Object Tracking with Occlusion Handling

3.1 Introduction

This chapter presented a progressive overview of object detection and tracking advancements, starting from YOLOv8 through YOLOv11, highlighting architectural innovations that enhance detection speed and accuracy. It also introduced OC-SORT as an advanced occlusion handling method to improve robustness in challenging tracking scenarios. Together, these developments form a powerful foundation for occlusion-aware object tracking systems.

3.2 Object Detection

In the field of multi-object tracking (MOT), various types of detectors are used to determine the locations of objects in each video frame. The quality of these detectors is essential for accurate and efficient tracking. With the development of deep convolutional neural networks (CNNs) [16]. Modern object detection systems can be classified into two main categories:

3.2.1 Two-Stage Detectors

These are based on the R-CNN architecture, first generating proposals for regions of interest (ROIs), and then classifying each region. This method is used in most advanced systems such as Faster R-CNN and Mask R-CNN, and is known for its high accuracy despite its relative slowness [17].

3.2.2 One-Stage Detectors

These have a simple architecture and high speed, such as SSD and YOLO. They strike a good balance between speed and accuracy, but they often do not outperform two-stage detectors in accuracy [17].

3.3 Object Tracking In Nutshell

In the field of Multi-Object Tracking (MOT), tracking algorithms are generally divided into two main categories, based on the type of information they rely on: motion or appearance.

3.3.1 Appearance-Based MOT

These methods incorporate visual features of the objects (such as color, texture, and shape) to distinguish and associate them across frames. This is especially useful when objects move similarly or become occluded.

- **BoT-SORT:** Combines both motion and appearance cues. It compensates for camera motion and includes a refined Kalman Filter model, leading to more accurate tracking in dynamic environments.

- **TransTrack:** Introduces a Transformer-based architecture that unifies object detection and tracking into a single step. This not only simplifies the pipeline but also enhances performance in complex scenes[18].

3.3.2 Motion-Based MOT

These algorithms track objects based purely on their movement across frames, without using visual appearance. They often use predictive models to estimate where objects will move next.

- **Kalman Filter:** The Kalman Filter is a mathematical method used to estimate the state of a moving object over time, even when the measurements are noisy or partially missing. It works in two main steps:

-Prediction Step: The filter predicts the next position or state of the object based on its previous state.

-Update Step: If a new measurement (like a detection from YOLO) is available, the filter updates the prediction using this new information to improve accuracy.

The Kalman Filter maintains: An estimate of the current object state (like position and size) , and a confidence level (covariance) for that estimate [19].

- **SORT (Simple Online and Realtime Tracking:** Builds on the Kalman Filter and uses Intersection-over-Union (IoU) to associate detections between frames. It's simple, fast, and effective in many real-time scenarios.

- **ByteTrack:** Improves on SORT by incorporating low-confidence detections during association, which helps recover missed objects and maintain more stable tracks.

3.4 Multiple Object Tracking Pipeline

3.4.1 YOLO Detector

YOLO (You Only Look Once) is an object detection algorithm that merges all detection steps into a single Convolutional Neural Network (CNN). Thanks to its single-stage design, YOLO achieves extremely high speed while maintaining excellent accuracy, reducing background errors and false detections compared to other approaches.

Being an open-source model has significantly accelerated YOLO's development, as developers from all over the world have continually contributed to its improvement. Since its release, YOLO has become the preferred choice for many applications that require real-time object detection without compromising precision. YOLO is inspired by the GoogleNet architecture, combining convolutional layers with fully connected layers [17].

3.4.2 YOLOv8

YOLOv8, developed by Ultralytics, represents a major advancement in the YOLO (You Only Look Once) series of object detection models. It features significant improvements in performance, accuracy, and inference speed, making it ideal for real-time applications such as surveillance systems and autonomous vehicles.

Trained on diverse datasets, YOLOv8 is capable of detecting a wide range of objects with high precision. It also offers strong scalability and flexibility, allowing users to customize the architecture for different tasks. Support for GPU acceleration further enhances training efficiency and model performance. Beyond object detection, YOLOv8 supports multiple computer vision tasks, including image classification, localization, segmentation, object tracking, and pose estimation. These capabilities make YOLOv8 a versatile and powerful tool in the fields of computer vision and image processing. With its high speed and accuracy, it stands out as a core technology in modern AI and visual computing applications [20].

3.4.2.1 YOLOv8 Architecture

- **Backbone and Feature Extraction:** YOLOv8 begins by passing the input image through a refined backbone network. This backbone is responsible for extracting multi-level features using convolutional layers, including components like CBS blocks (Convolution, Batch Normalization, and SiLU activation) to ensure robust feature learning and efficient information flow [21] See the figure3.1.

• **Modified Stem and Bottleneck Modules:** To improve performance and reduce complexity, the standard 6×6 convolution used in earlier versions is replaced with a more efficient 3×3 convolution in the stem. Additionally, the conventional C3 module is replaced by the new ”C2f module”, which aggregates outputs from all bottleneck blocks. Each bottleneck consists of dual 3×3 convolutions with residual connections. Unlike C3, which uses only the final bottleneck's output, C2f captures a richer feature representation, enhancing gradient flow and learning depth.

• **Feature Pyramid Network (FPN):** To support 'multi-scale detection', YOLOv8 incorporates a feature pyramid network. This allows the model to process and detect objects of varying sizes within a single image by combining low- and high-level features across different scales. This is especially useful in complex scenes where both small and large objects are present.

• **Detection Head:** The detection head of YOLOv8 is composed of a sequence of convolutional layers followed by fully connected layers. This module is responsible for generating [21]:

-Bounding box coordinates.

-Objectness scores (likelihood of an object being present).

-Class probabilities (object category).

3.4.3 YOLOv9

YOLOv9 is a state-of-the-art object detection model that excels at balancing inference speed and detection accuracy. Developed by Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao [22], the model introduced innovative techniques aimed at enhancing efficiency and accuracy in their study .

One of the most notable innovations in this version is the implementation of the concept of Programmable Gradient Information (PGI). This technique aims to address the problem of information loss as it travels through successive layers in deep neural networks. By incorporating PGI into the YOLOv9 architecture, learning efficiency is enhanced, helping to preserve important information throughout the detection process. This leads to

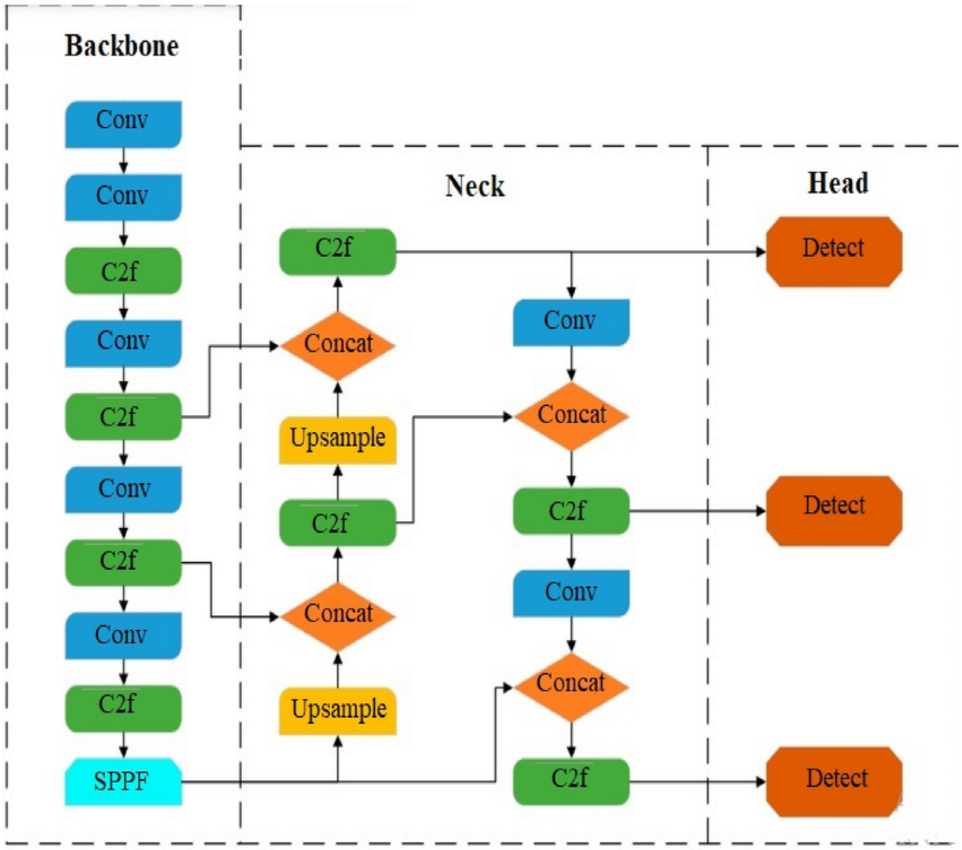


Figure 3.1: The YOLOv8 architecture .

improved gradient accuracy, faster model convergence during training, and higher overall performance.

Another key element of the YOLOv9 architecture, as illustrated in Figure 3.2, is the use of reversible functions. These functions are defined as invertible without losing any information, which is critical in the design of deep neural networks. This approach ensures a continuous flow of information within the model, enabling accurate parameter updates without compromising the quality of the data transferred between layers. In addition, YOLOv9 also introduces the Generalized Efficient Layer Aggregation Network (GELAN), a new architecture that enhances the model's efficiency, accuracy, and scalability. This architecture has demonstrated leading performance on the MS COCO dataset, making YOLOv9 a new standard in the field of object detection [23].

3.4.3.1 YOLOv9 Architecture

YOLOv9 also integrates a lightweight backbone called the Generalized Efficient Layer Aggregation Network (GELAN), as shown in Figure 3.2. GELAN is an extension

of the ELAN module originally introduced in YOLOv7, with improvements targeted at:
-Parameter efficiency.
-Reduced computational complexity.
-Enhanced inference speed.
The GELAN architecture optimizes the gradient flow through carefully designed aggregation paths, resulting in more efficient parameter utilization compared to existing state-of-the-art models. By combining PGI and GELAN, YOLOv9 achieves a high-performing, resource-efficient architecture that maintains a lightweight profile without sacrificing accuracy or speed. This makes YOLOv9 an excellent candidate for real-time applications and deployment in environments with limited computational capabilities [24].

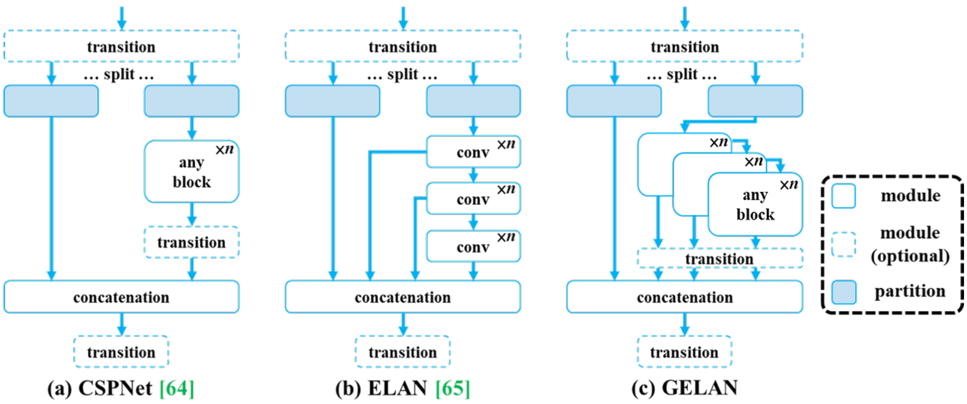


Figure 3.2: The YOLOv9 architecture .

3.4.4 YOLOv11

YOLOv11, developed by the Ultralytics team, represents the latest advancement in the YOLO object detection series, marking a significant leap in computer vision algorithms. While building on the successes of its predecessors, YOLOv11 introduces innovative enhancements that broaden its applications across various fields [25].

Designed to be an efficient, accurate, and flexible detection system, YOLOv11 is a powerful tool for object detection. What sets it apart is its increased versatility, as it now supports additional tasks beyond object detection, such as pose estimation and instance segmentation. This expansion makes it suitable for a much wider range of applications, including autonomous driving and advanced video analysis.

YOLOv11 is not only focused on raw power but also emphasizes achieving higher accuracy and efficiency in addressing real-world challenges across industries. These improvements reflect the continuous evolution of computer vision, opening the door to new, practical applications across multiple sectors [26].

YOLO11 Architecture

The architecture of YOLOv11 introduces several enhancements over previous versions, particularly YOLOv8, with a focus on improving computational efficiency and detection accuracy, making it suitable for real-time tasks such as vehicle detection figure 3.5.

1-Backbone: The backbone of YOLOv11 is responsible for feature extraction from the input image at multiple scales. It involves convolutional layers and custom blocks designed to generate feature maps at different resolutions.

- **Convolutional Layers:** YOLOv11 starts with a series of convolutional layers for downsampling the input image:

$$Conv1 = Conv(I, 64, 3, 2), Conv2 = Conv(Conv1, 128, 3, 2).$$

Conv1 extracts raw features such as edges and contrasts from the image, and then creates a convolutional layer using 64 filters with a kernel size of 3×3 and a stride of 2, which reduces the image dimensionality and increases the representational depth. Conv2 extracts more complex features from the output of Conv1. The convolutional layer uses 128 filters with a kernel size of 3×3 and a stride of 2, which further reduces the dimensionality while increasing the representational depth. These layers are used to gradually reduce the image dimensionality and transform it into an information-rich numerical representation, enabling the network to recognize objects with high efficiency.

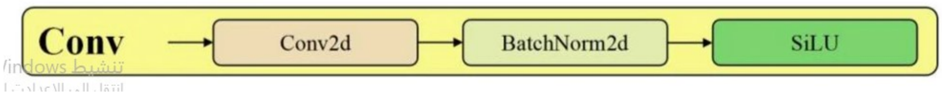


Figure 3.3: Conv module.

- **C3k2 Block:** YOLOv11 introduces the C3k2 block, replacing the C2f block in YOLOv8. This block is more computationally efficient, based on the Cross-Stage Partial (CSP) network. It involves two smaller convolutions (kernel size = 2) to reduce computational cost while maintaining performance:

$$C3k2(X) = Conv(Split(X)) + Conv(Merge(Split(X)))$$

SPPF and C2PSA Blocks: YOLOv11 retains the SPPF (Spatial Pyramid Pooling Fast) block from earlier versions, performing spatial pooling across multiple scales. Additionally, YOLOv11 incorporates the C2PSA (Cross-Stage Partial Spatial Attention) block, which enhances spatial attention, focusing on the most important parts of the image, especially in cluttered scenes with overlapping objects:

$$SPPF(X) = Concat(MaxPool(X, 5), MaxPool(X, 3), MaxPool(X, 1))$$

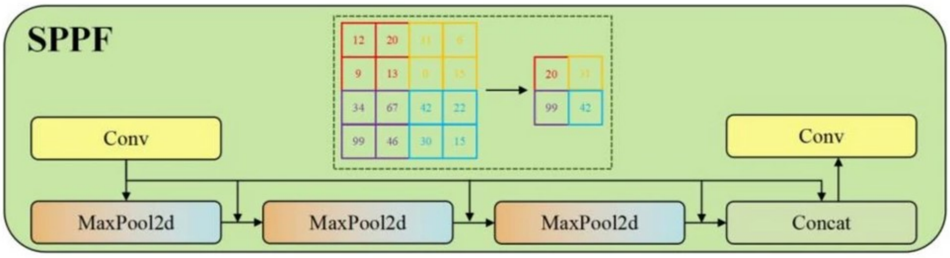


Figure 3.4: SPPF module.

$$C2PSA(X) = Attention(Concat(Xpath1, Xpath2))$$

2-Neck: The neck of YOLOv11 aggregates feature maps from different resolutions and passes them to the detection head.

- **Feature Aggregation:**The neck uses upsampling and concatenation layers to combine feature maps from various scales.

The C3k2 block is used after concatenation to ensure efficient feature aggregation.

3-Head:

YOLOv11's architecture introduces several optimizations, such as the C3k2 and C2PSA blocks, to improve computational efficiency, spatial attention, and overall performance, especially for detecting small or occluded objects in complex scenes.

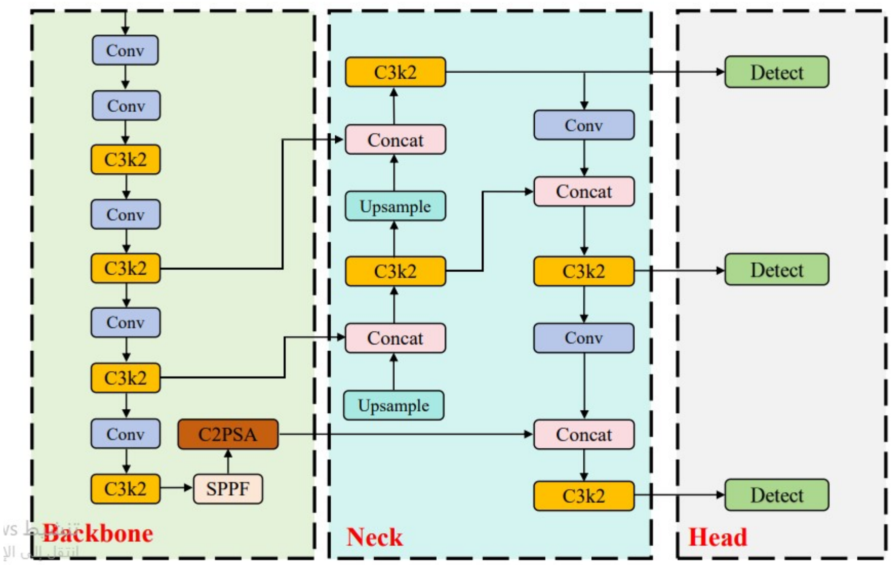


Figure 3.5: YOLOv11 model architecture.

3.4.5 OC-SORT Tracker as a Fusion-Based Occlusion Handling Method

Our approach integrates the recently proposed **OC-SORT** tracker as an occlusion handling module within our fusion framework. Unlike traditional trackers such as SORT, which rely exclusively on a Kalman Filter and IoU-based association under the assumption of linear motion, OC-SORT introduces a robust observation-centric strategy that significantly improves performance in real-world conditions involving occlusions and missed detections [18].

Observation-Centric Tracking: OC-SORT replaces estimation-driven state updates with an observation-centric design that directly fuses motion predictions with actual detections. This approach reduces the impact of accumulated errors during occlusions and enables more accurate recovery and association of lost tracks.

Key Components:

- **Observation-Centric Re-Update (ORU):** When an object reappears after occlusion, accumulated errors in the Kalman filter are corrected by constructing a virtual trajectory between the last observation (z_{t_1}) and the reactivating detection (z_{t_2}). Predict-update steps are executed along this trajectory using interpolated observations to recalibrate the Kalman filter. This update occurs outside the conventional predict-update loop and is triggered only when a lost track is re-associated.

- **Observation-Centric Momentum (OCM):** Direction estimation over short time intervals (Δt) is often noisy, while longer intervals violate the linear motion assumption. To address this, OC-SORT computes direction based on a sequence of past observations rather than predicted states.
- **Observation-Centric Recovery (OCR):** When objects stop or become occluded, traditional models may fail to maintain accurate tracks. OCR is a heuristic strategy that attempts to recover lost tracks by associating their most recent observation with currently unmatched detections, based on IoU, in a secondary matching stage following the main association step [27].

Kalman Filter Use and Its Limitations: The Kalman Filter used in OC-SORT performs:

- *Prediction Step:* Projects object state forward using constant velocity motion.
- *Correction Step:* Refines prediction using newly observed detections (e.g., from YOLO).

However, it assumes linear motion and uninterrupted observations, making it prone to drift during occlusion. OC-SORT addresses this by fusing these predictions with observation-driven updates such as ORU and OCM.

In this work, we extend SORT by integrating OC-SORT as a fusion-based occlusion handling technique. By leveraging observation-centric strategies such as ORU, OCM, and OCR, OC-SORT mitigates the limitations of prediction-only systems and achieves robust tracking in the presence of occlusions and missed detections [19, 28].

3.5 Conclusion

This chapter explored the evolution of object detection models from YOLOv8 to YOLOv11, emphasizing their architectural enhancements, improved computational efficiency, and expanded capabilities across vision tasks. It also detailed the integration of OC-SORT as a fusion-based occlusion handling module, showcasing its observation-centric strategies—ORU, OCM, and OCR—for robust tracking in complex, real-world environments. Together, these advancements contribute to a more accurate and resilient object tracking framework.

Chapter 4

Chapter 3: Implementation and Evaluation

4.1 Introduction

This chapter presents the implementation details and experimental evaluation of the proposed multi-object tracking system designed for autonomous vehicle perception. The framework is built upon the OC-SORT tracking algorithm, which emphasizes observation-centric strategies to enhance tracking robustness, especially in dynamic and occlusion-prone environments. The system integrates various modules, including IoU-based cost calculation, motion modeling with Observation-Centric Momentum (OCM), and advanced track management mechanisms such as Observation-Centric Re-Update (ORU) and Recovery (OCR), to ensure consistent identity assignment over time.

To assess system performance, we conduct extensive evaluations using real-world driving scenarios from the KITTI dataset. These experiments aim to quantify the system's tracking accuracy, identity consistency, and robustness through established metrics such as HOTA, MOTA, IDF1, MT, and ML. A comparative analysis is also performed to benchmark our method against existing tracking approaches. Additionally, qualitative visualizations are provided to offer deeper insight into the system's behavior in challenging conditions, including occlusions and cluttered scenes. The results collectively demonstrate the effectiveness and reliability of the proposed solution for real-time vehicle tracking in autonomous driving contexts.

4.2 Proposed Method

Our tracking framework builds upon the OC-SORT pipeline, incorporating an observation-centric design to improve robustness under challenging scenarios, particularly occlusions and missed detections. As illustrated in Figure 4.1, the method integrates a standard 2D object detector with an enhanced tracking module based on Kalman filtering and novel observation-centric strategies.

4.2.1 Detection Stage

The tracking pipeline begins with object detection using YOLOv11, YOLOv8, or YOLOv9. These detectors generate 2D bounding boxes for each frame, which serve as observations for the subsequent tracking module.

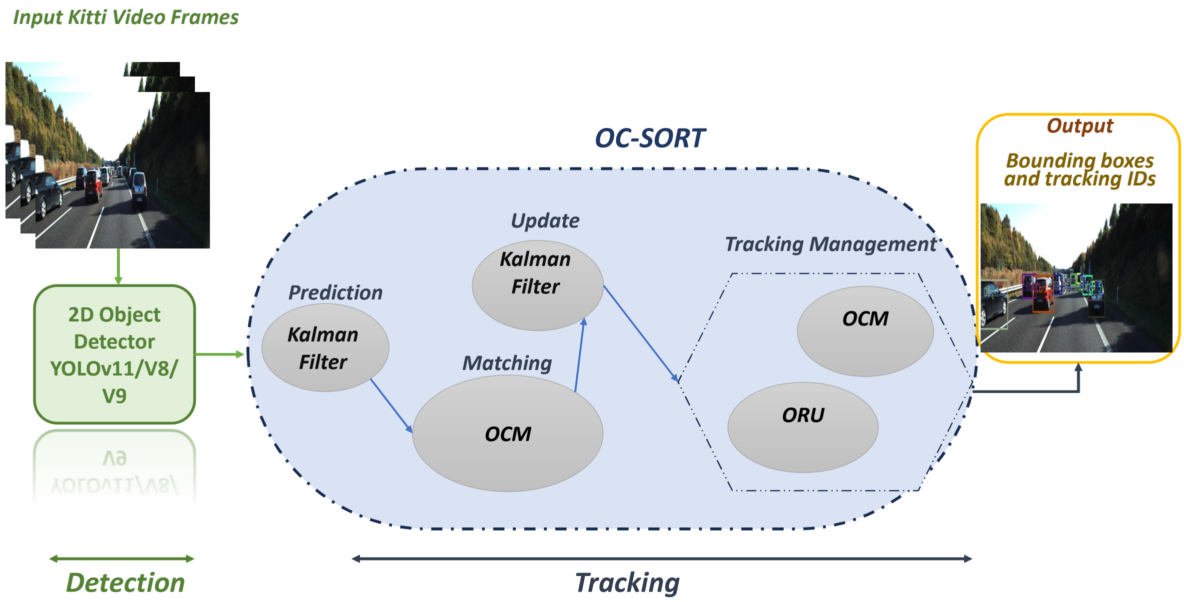


Figure 4.1: Overview of the proposed OC-SORT-based tracking pipeline.

4.2.2 Tracking Stage

4.2.2.1 Prediction

For each tracked object, the Kalman Filter predicts its future position based on a constant velocity motion model. This provides a temporary state estimate used for association with new detections.

4.2.2.2 Matching

The predicted states are associated with current detections using a composite cost function. This cost includes:

- **IoU cost:** Measures overlap between predicted and detected bounding boxes.
- **Observation-Centric Momentum (OCM):** Introduces directional consistency between consecutive observations to stabilize associations. This reduces noise and improves matching accuracy when linear assumptions break down.

4.2.2.3 Update

Once associations are established, the Kalman Filter state is corrected using the matched detection. This standard predict-update loop ensures temporal consistency of object states.

4.2.3 Tracking Management and Occlusion Handling

The tracking management component introduces two critical modules to handle occlusions and track recovery:

- **Observation-Centric Re-Update (ORU):** When an object reappears after occlusion, ORU is invoked to correct the accumulated prediction error. A virtual trajectory is interpolated between the last known and re-detected positions, and intermediate predict-update steps are performed to recalibrate the filter state.
- **Observation-Centric Recovery (OCR):** For unmatched tracks, OCR attempts to recover identity by matching their last known observations with currently unmatched detections using IoU. This secondary matching phase improves re-identification under partial occlusion or temporary object disappearance.

4.2.4 Output

Finally, the framework outputs bounding boxes and tracking IDs for each object in the frame. These results are refined through the above steps to ensure reliable ID assignment, even in the presence of occlusion and motion discontinuities.

4.3 Performance evaluation

We tested the performance of the vehicle identification and tracking system we developed through a series of experiments using real-world data from driving environments such as the KITTI datasets. The goal was to determine the accuracy and efficiency of the system in continuously tracking vehicles without losing or confusing them. We based the evaluation on several important indicators, including:

- **MOTA(Multiple Object Tracking Accuracy):** MOTA is a widely used metric that quantifies overall tracking accuracy by combining three sources of error: false positives (FP), false negatives (FN), and identity switches (IDs). It evaluates how well the tracker can correctly detect and follow multiple objects

$$MOTA = \frac{(FN + FP + ID_s)\sum}{(GT)\sum} - 1$$

- **IDF1(Identification F1 Score):** The IDF1 metric measures the accuracy of identity preservation across the entire tracking sequence. It is defined as the harmonic mean of Identification Precision (IDP) and Identification Recall (IDR). This metric provides a balanced assessment of the system's capability to correctly assign and maintain unique identities to tracked objects over time.

$$IDF1 = \frac{IDP.IDR.2}{IDP + IDR}$$

- **HOTA (Higher Order Tracking Accuracy):** HOTA is an advanced and more comprehensive metric that evaluates not only detection and association accuracy but also the quality of object localization and tracking consistency. It is particularly effective in distinguishing overlapping objects and identifying incorrect transitions between targets [29].

$$HOTA = \sqrt{DetA.AssA}$$

- **MT and ML:** Show the system's ability to track vehicles for a long or short period.

4.4 KITTI Dataset

Developed by the Karlsruhe Institute of Technology and the Toyota Technological Institute at Chicago, the KITTI Vision Benchmark was created to serve as a standard benchmark for research in autonomous driving and computer vision. It enables researchers to evaluate the real-world performance of their algorithms (Geiger, Lenz, Stiller, and Urtasun, 2013). The KITTI dataset includes multiple types of sensor data collected from a vehicle-mounted setup. This setup comprises stereo cameras, a Velodyne 3D lidar scanner, a Global Positioning System (GPS), and an Inertial Measurement Unit (IMU). The dataset contains images, point clouds, calibration files, and ground truth

annotations gathered from urban roads and highways [30].
KITTI provides a standardized framework for evaluating and comparing algorithms across a range of tasks such as object detection and tracking. Due to its comprehensive and well-structured nature, it has become a widely adopted benchmark in autonomous driving research, enabling fair comparisons with state-of-the-art approaches.

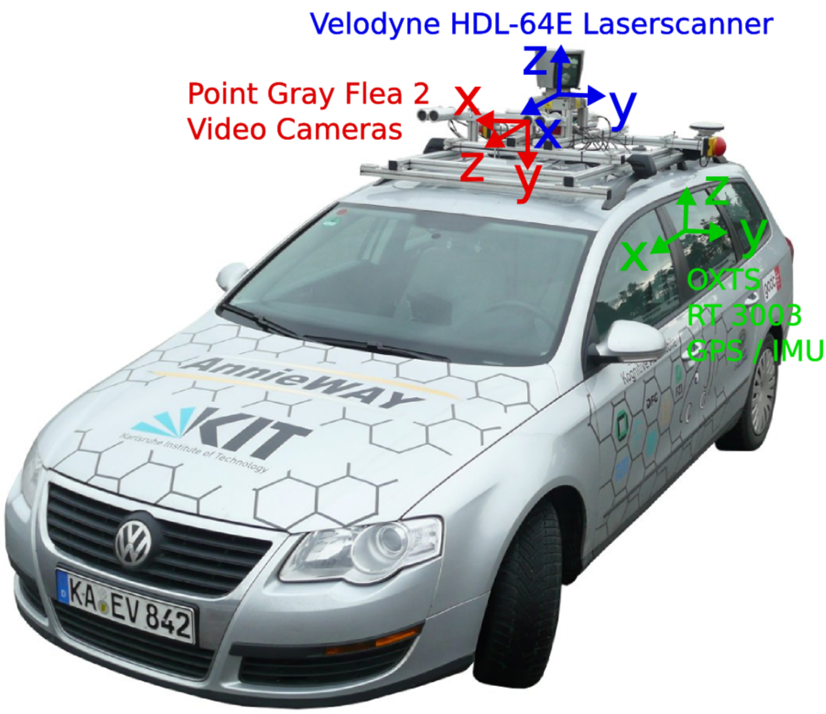


Figure 4.2: An external view of the KITTI data collection vehicle.

The dataset was collected on rural roads and highways around Karlsruhe, Germany, and includes a variety of static objects (e.g., buildings, traffic signs, vegetation) as well as dynamic elements (e.g., vehicles, pedestrians, bicycles).
This project specifically uses 21 Labeled videos from the KITTI dataset, which include corresponding realistic poses.

4.5 System Requirement

Model process was performed on a system powered by an NVIDIA RTX A5000 GPU, featuring 24 GB of video memory to support high-performance deep learning workloads.

4.6 Experimental Results

To evaluate the effectiveness of the proposed model in improving autonomous vehicle perception through multi-object tracking, we conducted multiple experiments using multiple detectors with the OC-SORT tracking algorithm.

4.6.1 Detector-Level Performance Comparison in OC-SORT

Model	HOTA	MOTA	IDF1	MT	ML
YOLOv8	59.98%	61.94%	74.78%	206	74
YOLOv9	57.12 %	57.77%	71.73%	163	93
YOLOv11	**61.07%**	**62.51%**	**75.67%**	**205**	**75**

Table 4.1: Comparison of tracking performance using YOLOv8, YOLOv9, and YOLOv11 as object detectors within the OC-SORT tracking framework.

Table 4.1 presents a comparative evaluation of three YOLO detectors—YOLOv8, YOLOv9, and YOLOv11—integrated with the OC-SORT tracking framework using standard multi-object tracking metrics: HOTA, MOTA, IDF1, MT, and ML. Among the models, YOLOv11 consistently achieves the best performance across all metrics, with the highest HOTA (61.07%), MOTA (62.51%), and IDF1 (75.67%), indicating superior detection quality, association accuracy, and identity preservation. YOLOv8 performs slightly lower, while YOLOv9 shows the weakest results, particularly in MT (163%) and ML (93%), reflecting less robust long-term tracking. YOLOv11 also maintains a strong MT score (205%) and a relatively low ML (75%), confirming its capability to track most objects consistently while minimizing tracking loss. These results demonstrate that the latest YOLOv11 detector, when combined with OC-SORT, yields the most reliable tracking performance among the evaluated variants.

Based on the results of our experiments, it was clear that YOLOv11 was the best choice for object detection in our system. When compared to earlier versions like YOLOv8 and YOLOv9, YOLOv11 consistently performed better across all major evaluation metrics.

This strong performance is due to the architectural improvements introduced in YOLOv11, such as the C3k2 and C2PSA modules, which help the model focus better on important regions and improve detection in challenging environments.

4.6.2 Comparison with State-of-the-Art Trackers

Method	HOTA	MOTA	IDF1	MT	ML
YOLOv5-Small [31]	53.32%	44.99%	71.41%	/	/
YOLOR-CSP[31]	51.81%	47.38%	72.53%	/	/
YOLOv9 + DeepSORT + Optical Flow [29]	47.58%	56.33%	72.32%	/	/
Ours (YOLOv11+OC SORT)	**61.07%**	**62.51%**	**75.67%**	**205**	**75**

Table 4.2: Comparison of tracking performance between the proposed method (YOLOv11 + OC-SORT) and state-of-the-art methods.

Table 4.2 presents a comparative evaluation of our proposed system (YOLOv11 + OC-SORT) against three existing tracking pipelines: YOLOv5-Small, YOLOR-CSP, and YOLOv9 combined with DeepSORT and Optical Flow. The results show that our method outperforms the baselines across all reported metrics. Specifically, our system achieves the highest HOTA (61.07%), MOTA (62.51%), and IDF1 (75.67%) scores, demonstrating improved association accuracy, tracking robustness, and identity preservation. While the other methods lack MT and ML values, our approach achieves a strong MT of 205 and an ML of 75, further highlighting the effectiveness of integrating a powerful detector (YOLOv11) with a modern tracking algorithm (OC-SORT). This consistent improvement confirms the synergy between advanced detection and robust tracking mechanisms.

These results validate the effectiveness of integrating the high-precision YOLOv11 detector with the observation-centric OC-SORT tracker. Together, they form a powerful and efficient perception framework for autonomous vehicles.

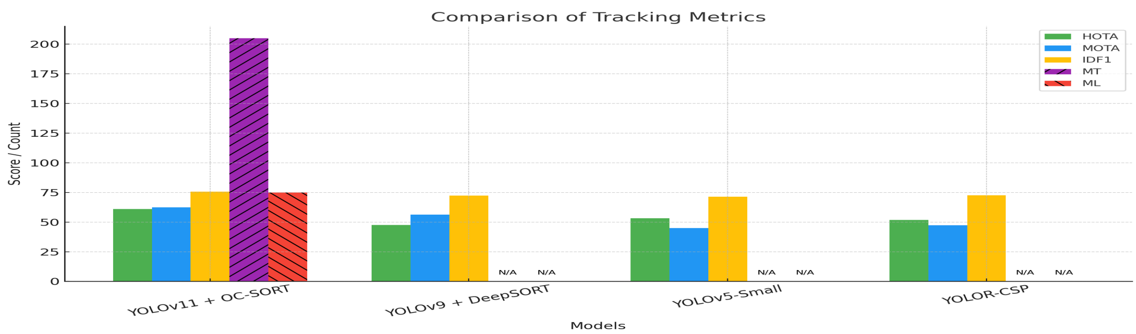


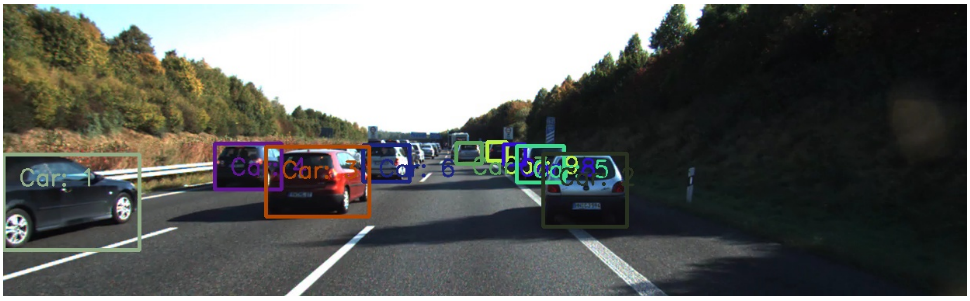
Figure 4.3: Performance chart of different systems (YOLOv11 + OC-SORT, YOLOv9 + DeepSORT, YOLOv5-Small, YOLOR-CSP).

Figure 4.3 presents a clear visual comparison of tracking performance using three key evaluation metrics: HOTA, MOTA, and IDF1, all tested on the KITTI dataset. From the graph, it's clear that the proposed system—YOLOv11 combined with OC-SORT—consistently outperformed the other models, including YOLOv9 with DeepSORT and Optical Flow, YOLOv5-Small, and YOLOR-CSP, across all measured criteria.

These results clearly validate the effectiveness of integrating YOLOv11 with OC-SORT for real-time, accurate, and reliable perception in autonomous driving systems. The improvements observed across all metrics highlight the system's robustness in managing multiple object tracking under real-world driving conditions.

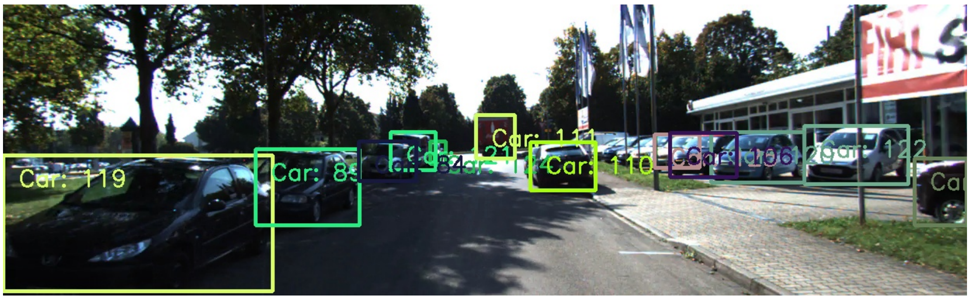
4.6.3 Visualization Insights

The qualitative results illustrated in Figure 4.4 provide compelling evidence of the robustness and reliability of the proposed tracking framework (YOLOv11 + OC-SORT) in real-world autonomous driving scenarios. Each detected object is annotated with a colored bounding box and a unique tracking ID, enabling visual confirmation of identity consistency across frames. The visualization includes object trajectories, which reflect smooth and continuous motion paths—even in sequences involving heavy traffic and complex interactions—demonstrating the tracker's stability. A key strength of OC-SORT is its observation-centric design, which excels at handling occlusions. This is clearly observed in the figure, where the model successfully maintains object identities during full and partial occlusion events without introducing identity switches. The use of the Observation-Centric Re-Update (ORU) mechanism enables the tracker to accurately restore object states after temporary disappearances, while the Observation-Centric Momentum (OCM) term aids in maintaining consistent motion direction under abrupt movements. These features collectively make OC-SORT highly effective in scenarios where conventional trackers tend to fail due to accumulated estimation error during object loss. Furthermore, the tracking system demonstrates remarkable resilience under adverse visual conditions such as varying illumination, strong shadows, and dense crowds. Even when lighting conditions fluctuate rapidly or multiple objects overlap in close proximity, the tracker continues to associate detections accurately and maintain long-term identity consistency. These visualization confirm that OC-SORT is a powerful occlusion handling technique, making it well-suited for safety-critical applications like autonomous driving.



(a) OC-SORT and YOLOv11 in a complex highway environment, demonstrating handling of occluded vehicles

(b) Object tracking in an urban environment.



(c) Object tracking result, showing multiple labeled and ID-tracked cars in a parking setting

Figure 4.4: Visualization of tracking results using YOLOv11 and OC-SORT for occlusion handling on the KITTI dataset.

4.7 Conclusion

This chapter detailed the implementation and evaluation of our object tracking system, highlighting the integration of the YOLOv11 detector with the observation-centric OC-SORT tracking framework. We outlined the key components of the tracking pipeline, including association strategies, motion updates, and occlusion handling mechanisms such as ORU and OCR. Through extensive experiments on the KITTI dataset, the system demonstrated superior tracking accuracy, identity preservation, and robustness under real-world conditions. Quantitative results showed that YOLOv11 combined with OC-SORT outperformed earlier detectors and other state-of-the-art trackers across all major metrics—HOTA, MOTA, and IDF1—while qualitative visualizations confirmed its stability in handling occlusions, motion discontinuities, and complex environments. These findings validate the proposed framework as a reliable and effective solution for multi-object tracking in autonomous driving applications.

Chapter 5

General Conclusion and Future Work

In this thesis, we proposed and evaluated a robust multi-object tracking framework based on the OC-SORT pipeline, integrated with advanced YOLO-based detectors (YOLOv8, YOLOv9, and YOLOv11) to enhance perception for autonomous driving. By incorporating observation-centric strategies—namely Observation-Centric Momentum (OCM) and Observation-Centric Re-Update (ORU)—the model demonstrated substantial improvements in handling occlusion, maintaining object identity, and recovering from temporary detection failures. Among the detectors tested, YOLOv11 consistently outperformed earlier versions across key metrics such as HOTA, MOTA, and IDF1, validating its superior detection and association performance. The system also showed resilience under challenging visual conditions including varying lighting, shadows, and crowded scenes, which are often problematic for conventional tracking models.

Despite its promising results, the proposed framework is not without limitations. The computational complexity of high-resolution detectors and the re-update strategy may hinder real-time performance on low-resource platforms. Additionally, the model relies heavily on the quality of initial detections—any degradation in detection performance directly affects tracking accuracy. Another challenge lies in generalizing the model to unseen domains or adapting to 3D tracking contexts, which require additional depth-aware modeling and sensor fusion.

Future work could explore integrating depth information from stereo cameras or LiDAR to extend the pipeline to 3D multi-object tracking. Finally, incorporating semantic scene understanding and temporal attention mechanisms may further improve identity preservation, particularly in dense or highly dynamic traffic environments. Overall, the results of this study provide a strong foundation for deploying robust and occlusion-aware tracking systems in next-generation autonomous vehicles.

Bibliography

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of field robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14.* Springer, 2016, pp. 21–37.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP).* Ieee, 2016, pp. 3464–3468.

[5] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP).* IEEE, 2017, pp. 3645–3649.

[6] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International journal of computer vision*, vol. 129, pp. 3069–3087, 2021.

[7] Y. Zheng, H. Qi, L. Li, S. Li, Y. Huang, C. He, and D. Wang, "Motion-guided and occlusion-aware multi-object tracking with hierarchical matching," *Pattern Recognition*, vol. 151, p. 110369, 2024.

[8] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6247–6257.

[9] A. Giannaros, A. Karras, L. Theodorakopoulos, C. Karras, P. Kranias, N. Schizas,

G. Kalogeratos, and D. Tsolis, “Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions,” *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 493–543, 2023.

[10] S. Hakak, T. R. Gadekallu, P. K. R. Maddikunta, S. P. Ramu, M. Parimala, C. De Alwis, and M. Liyanage, “Autonomous vehicles in 5g and beyond: A survey,” *Vehicular Communications*, vol. 39, p. 100551, 2023.

[11] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, and W. Cho, “A review on autonomous vehicles: Progress, methods and challenges,” *Electronics*, vol. 11, no. 14, p. 2162, 2022.

[12] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, “Sensor and sensor fusion technology in autonomous vehicles: A review,” *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[13] S. V. Kothiya and K. B. Mistree, “A review on real time object tracking in video sequences,” in *Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on*, 2015, pp. 1–4.

[14] P. Girão, A. Asvadi, P. Peixoto, and U. Nunes, “3d object tracking in driving environment: A short review and a benchmark dataset,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 7–12.

[15] B. Y. Lee, L. H. Liew, W. S. Cheah, and Y. C. Wang, “Occlusion handling in videos object tracking: A survey,” in *IOP conference series: earth and environmental science*, vol. 18, no. 1. IOP Publishing, 2014, p. 012020.

[16] S. Bouraya Jr and A. Belangour, “Multi object tracking: A survey,” in *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*, vol. 11878. SPIE, 2021, pp. 142–152.

[17] B. Li, Y. Liu, and X. Wang, “Gradient harmonized single-stage detector,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8577–8584.

[18] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, “Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification,” in *2023 IEEE International conference on image processing (ICIP)*. IEEE, 2023, pp. 3025–3029.

[19] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, “Observation-centric sort: Rethinking sort for robust multi-object tracking,” in *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, 2023, pp. 9686–9696.

[20] D. Hindarto, "Exploring yolov8 pretrain for real-time detection of indonesian native fish species," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 7, no. 4, pp. 2776–2785, 2023.

[21] A. Elhanashi, P. Dini, S. Saponara, and Q. Zheng, "Telestroke: real-time stroke detection with federated learning and yolov8 on edge devices," *Journal of Real-Time Image Processing*, vol. 21, no. 4, p. 121, 2024.

[22] H.-T. Vo, K. C. Mui, N. N. Thien, and P. P. Tien, "Automating tomato ripeness classification and counting with yolov9," *Int. J. Adv. Comput. Sci. Appl*, vol. 15, no. 4, 2024.

[23] S. A. Fahim, "Finetuning yolov9 for vehicle detection: Deep learning for intelligent transportation systems in dhaka, bangladesh," *arXiv preprint arXiv:2410.08230*, 2024.

[24] X. Wang, H. Qian, L. Xie, X. Wang, and B. Li, "Recognition and classification of typical building shapes based on yolo object detection models," *ISPRS International Journal of Geo-Information*, vol. 13, no. 12, p. 433, 2024.

[25] L.-h. He, Y.-z. Zhou, L. Liu, W. Cao, and J.-h. Ma, "Research on object detection and recognition in remote sensing images based on yolov11," *Scientific Reports*, vol. 15, no. 1, p. 14032, 2025.

[26] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[27] N. D. Binh, "A robust framework for visual object tracking," in *2009 IEEE-RIVF International Conference on Computing and Communication Technologies*. IEEE, 2009, pp. 1–8.

[28] M. Adžemović, P. Tadić, A. Petrović, and M. Nikolić, "Beyond kalman filters: deep learning-based filters for improved object tracking," *Machine Vision and Applications*, vol. 36, no. 1, pp. 1–22, 2025.

[29] D. Djarah, A. Benmakhlouf, G. Zidani, and L. Khettache, "Online multi-object tracking with yolov9 and deepsort optimized by optical flow," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17 922–17 930, 2024.

[30] E. Spackman, "Performance evaluation of icp variants for lidar data registration: A comparative study on kitti visual benchmark suite," 2023.

[31] P. Azevedo and V. Santos, “Yolo-based object detection and tracking for autonomous vehicles using edge devices,” in *Iberian Robotics conference*. Springer, 2022, pp. 297–308.