TU WIEN Informatics

Learning 3D pose and target ID for accurate multi-target tracking

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Alexander Odvar Peter Sing, BSc.
Matrikelnummer 01619928

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dr.techn. Sebastian Zambanini
Mitwirkung: Dr. DI Csaba Beleznai

Wien, 16. April 2022

Alexander Odvar Peter Sing

Sebastian Zambanini

TU WIEN Informatics

Learning 3D pose and target ID for accurate multi-target tracking

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Alexander Odvar Peter Sing, BSc.
Registration Number 01619928

to the Faculty of Informatics

at the TU Wien

Advisor: Dr.techn. Sebastian Zambanini
Assistance: Dr. DI Csaba Beleznai

Vienna, 16th April, 2022

Alexander Odvar Peter Sing

Sebastian Zambanini

Erklärung zur Verfassung der Arbeit

Alexander Odvar Peter Sing, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 16. April 2022

Alexander Odvar Peter Sing

Danksagung

Diese Arbeit entstand in Zusammenarbeit zwischen dem Computer Vision Lab (TU Wien) und dem Institut für Assistive und Autonomous Systems des Austrian Institute of Technology (AIT). Mein besonderer Dank gilt Dr. DI Csaba Beleznai, der mich während des gesamten Projekts intensiv betreut und unterstützt hat. Ich konnte viel von seiner jahrelangen Erfahrung im Computer Vision Bereich lernen und er hat mein Interesse für das Fachgebiet weiter ausgebaut. Des Weiteren möchte ich Dr. techn. Sebastian Zambanini für seine aktive Betreuung und sein sorgfältiges Feedback danken.

Außerdem bedanke ich mich bei meiner Freundin Lisa, die immer für mich da ist und mich in jeder Situation unterstützt.

Schließlich möchte ich meiner Familie und insbesondere meinen Eltern Angela und Andreas, sowie meinem Stiefvater Jörg für ihre kontinuierliche Unterstützung in meinem Leben und meinem Bildungsweg danken.

Acknowledgements

This thesis was a collaboration between the Computer Vision Lab (TU Wien) and the Institute for Assistive and Autonomous Systems from the Austrian Institute of Technology (AIT). I would like to express my sincere gratitude to Dr. DI Csaba Beleznai, who supervised and supported me throughout the whole project. I could learn from his years of experience in the computer vision area, and he continued to grow my interest in the subject. Furthermore, I would like to thank Dr. techn. Sebastian Zambanini for his supervision and the thorough feedback he provided.

Additionally, I would like to thank my girlfriend Lisa, who is always there for me and supportive in every situation.

Finally, I would like to thank my family, especially my parents, Angela and Andreas, as well as my stepfather Jörg for their continuous support in my life and my education.

Kurzfassung

Objekte in Zeit und Raum zu erkennen und zu verfolgen stellt eine zentrale, wissenschaftliche Frage für viele Szenarien bildbasierter Wahrnehmung dar. Jüngste Entwicklungen in Deep Learning ermöglichen verbesserte Repräsentationen von Objekten in Bezug auf ihre Position, Form, Erscheinung und Bewegung. Durch lernbasierte Methoden können klassen- oder objektspezifische Merkmale erfasst und sogar spezifische Korrelationen innerhalb von Bildern einer 3D Szene entdeckt werden, da ein perspektivisches Bild viele Hinweise über die 3D Position, Orientierung, Größe und Identität eines Objektes enthält. Fehlende Erkennungen, Verdeckungen und die Gegenwart mehrerer interagierender Objekte machen diese Aufgabe jedoch komplex und weiterhin ungelöst. In dieser Arbeit wird die Integration mehrerer lernbasierter Erweiterungen der Objektrepräsentationen vorgeschlagen, um diese Probleme zu verringern und die 3D Multi-Target Objekterkennung und -verfolgung präziser zu machen. Eine aufmerksamkeitsbasierte Erweiterung der Repräsentationen wird formuliert, um die Nutzung von Merkmalen, die den räumlichen Kontext beachten, im Rückgrat einer Neuronalen Netzwerk Architektur und dem Wiedererkennungsmodul zu fördern. Als zweiter Beitrag wird eine Repräsentation eingeführt, die ein Objekterkennungsmodul erweitert, um inkrementell neue Klassen von wenigen (1-10) Bildern zu lernen, ohne vorherige Klassen zu vergessen. Die vorgeschlagenen wissenschaftlichen Konzepte berücksichtigen existierende Datensätze und Forschungs- und Evaluierungsmethoden. Zusätzlich wurde im Rahmen der Evaluierungsaufgabe ein Schema zur synthetischen Generierung mehrerer Ziele und ihrer Trajektorien entwickelt, um Szenen mit einer variablen Anzahl an interagierenden Objekten mit Annotationen erstellen zu können. Die vorgeschlagene Methode wird auf dem KITTI Multi-Target Tracking Benchmark Datensatz evaluiert. Sie weist vergleichbare Resultate gegenüber einem Referenzansatz auf, der nur auf einer kinematischen Assoziation mithilfe eines Kalman Filters beruht. Außerdem wurden die ausgearbeiteten Konzepte in einem angewandten Szenario (Bike2CAV Projekt) validiert, bei dem die zeitlich variierende Konfiguration von Verkehrsteilnehmern aus Sicht eines bewegten Fahrzeuges geschätzt wird. Die Forschungsergebnisse deuten darauf hin, dass, trotz der Mehrdeutigkeit der monokularen Sicht, die eingeführten Erweiterungen der Repräsentation zu einer präziseren räumlichen Lokalisierung führen. Des Weiteren demonstrieren die Resultate, dass eine Wiedererkennung mittels Merkmalen Vorteile gegenüber einer einfachen, kinematischen Modellierung hat, da es zu zeitlich stabileren Tracking Ergebnissen führt. Dieser Vorteil könnte bei größeren Datensätzen mit 3D Posen und Tracking Annotationen ausgeprägter sein, was Raum für weitere Forschung lässt.

Abstract

Detecting and pursuing various targets in space and time represents a key scientific question for many vision-based perception scenarios. Recent developments in Deep Learning offer enhanced ways to represent targets in terms of their location, shape, appearance and motion. Learning can capture the significant variations seen in the training data while retaining class- or target-specific cues. Learning even allows for discovering specific correlations within an image of a 3D scene, as a perspective image contains many hints about an object's 3D location, orientation, size and identity. This single-image based spatial reasoning task is the subject of ongoing research. However, detection failures, occlusion, and the presence of multiple interacting targets render this task complex and still unsolved. In this thesis, the integration of multiple learning-based representational enhancements is proposed to mitigate these problems and perform the 3D multi-target detection and tracking task more accurately. In these tasks, an attention mechanism can facilitate discovering the correlation between image features and spatial attributes. An attention-based representational enhancement is formulated to guide learning towards using spatially-aware features in the backbone network within a neural-network architecture and the reidentification branch. As a second contribution, a representation for extending multi-task learning to incrementally learn new classes from a few (1-10) image samples without forgetting is introduced. As monocular 3D estimation is an evolving field, the proposed scientific concepts take existing datasets, research methodologies and evaluation concepts into account. Additionally, a synthetic multi-target trajectory generation scheme was developed to complement the evaluation task, offering a variable number of moving and interacting targets with computed ground truth. The proposed method is evaluated on the KITTI multi-target tracking benchmark dataset. It demonstrates competitive results against a baseline relying solely on a Kalman Filter based kinematic association step. The elaborated research concept has also been validated in an applied scenario (Bike2CAV project), where the time-varying spatial configuration of traffic participants is estimated from the viewpoint of a moving vehicle. The main findings of this research indicate that despite the monocular view ambiguity, the introduced representational enhancements lead to a more accurate spatial localisation. Results also demonstrate that target reidentification is advantageous beyond simple kinematic modelling, leading to a temporally more stable multi-target tracking performance. This advantage might be more pronounced by using larger datasets with extensive 3D poses and tracking annotations, indicating future research opportunities.

Contents

CHAPTER 1

Introduction and Motivation

Spatial awareness and reasoning are fundamental traits of modern vision-based robotic systems [LSML14].However, monocular (single view) vision-based perception is associated with ambiguities such as depth-scale ambiguity or viewpoint invariance that causes ambiguities [AAJD+19, MAR+19]. The former ambiguities arise from projecting the 3D world onto a 2D imaging plane, where multiple 3D scene configurations can result in the same projected image [MBU+19]. This thesis addresses specific vision tasks, namely detection, association and tracking. Object detection is the task of extracting an object's 2D or 3D location from sensor data, the focus in this thesis being RGB camera images. Target association and tracking is the task of associating detections within a given sequence with a consistent identifier, thus generating trajectories of these objects. In these tasks, ambiguities in estimating depth, orientation and association are strongly present, and prior knowledge via learned representations can be introduced to mitigate them. This ambiguity reduction can be achieved by combining prior knowledge from data with the representational power of Deep Learning [GBC16, KSH12] to infer entities such as objects, segmentations, similarities and trajectories. When sufficient data is available, even complex correlations between 2D (image) and 3D (world) spaces can be estimated and learned [CKZ+16]. This work demonstrates several methodology enhancements to learn such complex functions, which estimate the 3D pose directly from a single view and generate additional representations to support object association between time-consecutive image frames.

Deep Learning has progressed not only in terms of representational power but also in its capacity to accommodate multiple classification and regression tasks. Although the first step toward neural networks was already taken in 1943 with the introduction of the artificial neuron [MP43], it took until 2012 and the inception of AlexNet [KSH12] to spark the Deep Learning revolution we have today. Instead of manually designing feature extractors for images like in traditional computer vision, one could now utilise the power of a convolutional neural network (CNN) to learn features from data [Sze22]. Historically,

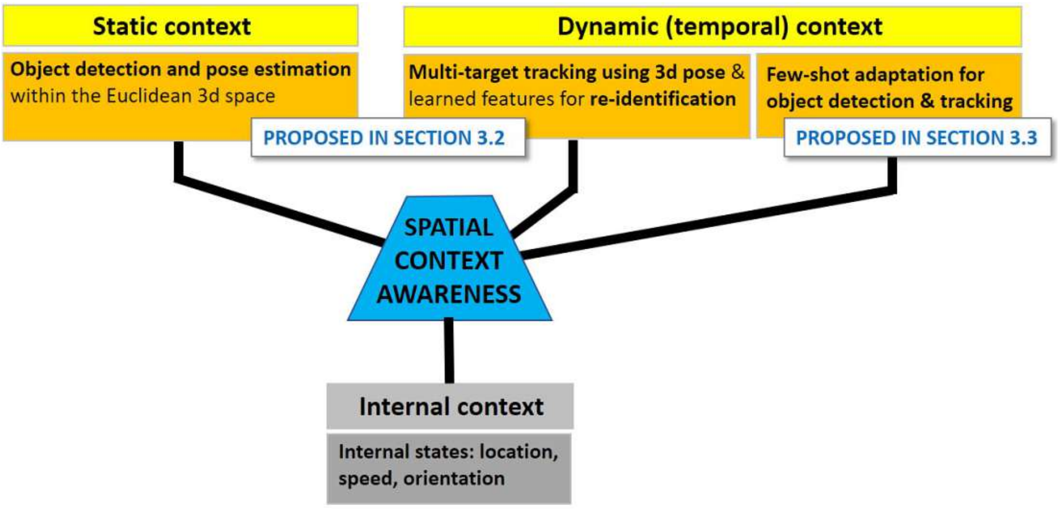

Static context
Object detection and pose estimation within the Euclidean 3d space
Dynamic (temporal) context
Multi-target tracking using 3d pose & learned features for re-identification
Few-shot adaptation for object detection & tracking
PROPOSED IN SECTION 3.2
PROPOSED IN SECTION 3.3
SPATIAL CONTEXT AWARENESS
Internal context
Internal states: location, speed, orientation

Figure 1.1: Targeted research focus: learned representations achieving spatial and temporal context awareness.

this trend has evolved from simple image classification [KSH12], through segmentation [NHH15] and 2D object detection [GDDM14], towards spatially more characterising tasks, such as instance [HGDG17] and panoptic [KHG+19] segmentation, monocular 3D object detection [CKZ+16] and even explicit 3D object geometry estimation [ZWT+21]. Only recently, the additional benefits of combining multiple vision tasks into one end-to-end formulation have been proposed, leading to simplified training, harmonised representations and accuracy improvements [ZWK19]. End-to-end multi-task learning performs multiple tasks simultaneously while sharing parameters between them. Such an approach also improves the generalisation performance of related tasks by utilising shared training information [Rud17].

Image-based 3D object detection and pose parameter regression are typical multi-task learning problems as they require classifying image content into possible classes while also regressing their 3D bounding boxes. Association of detected objects to consistent motion trajectories can be facilitated by including a reidentification task (reID), which distils each object's appearance into a compact and discriminative feature set [WZL+20]. Extending learned representations via few-shot learning of new object categories from previously unseen instances can also encompass a key learning task. These learning tasks, as they capture the spatial and temporal relationship of multiple object classes with respect to an observer (camera), can place the observer into an external context. Figure 1.1 illustrates the three different spatial contexts in robotics applications. At the top, the external context, consisting of static and dynamic context, and the tasks involved are depicted, while the bottom shows the internal context and the corresponding attributes. The presented research focuses on these spatial and temporal estimation tasks. Its motivation is twofold: (i) coping with ambiguities via learning still represents an open research question, and (ii) estimating spatiotemporal context is an essential asset

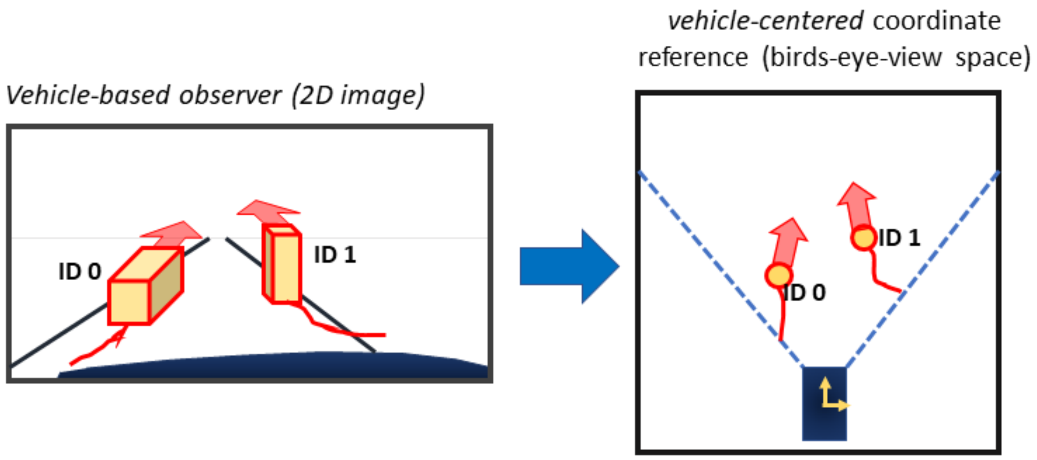

Vehicle-based observer (2D image)
ID 0
ID 1
vehicle-centered coordinate reference (birds-eye-view space)
ID 1
ID 0

Figure 1.2: Targeted spatial and temporal representational enhancements lifting and associating 2D image-based observations into a metric 3D space around a moving observer.

in enabling autonomous vehicles (AV) and robots.

In particular, interpreting the surrounding environment is crucial since recognising other traffic participants and possible obstacles is essential. However, lifting image-based 2D observations into a 3D metric space involves depth information so that an AV can avoid collisions. Figure 1.2 illustrates the transformation from the 2D image space to the Birds-Eye-View (BEV) space, a possible representation of 3D space in object detection. Intuitively, one would think this requires additional sensors or stereo vision [AAJD$^+$19]. However, recent developments have shown that under certain conditions and with enough data, neural networks can infer 3D depth information from training data, even from a single RGB image [CKZ$^+$16], while also reasoning about the identity of the detected object [WZL$^+$20]. Although many representational concepts have been proposed to tackle these core tasks, the inherent ambiguity of classifying objects into the same class while also distinguishing between objects of the same class calls for advanced representations exhibiting robustness during occlusions, clutter and object size variation [CZBO21]. Therefore, this work investigates and proposes using improved spatial attention that supports monocular view geometry and multi-target tracking reasoning.

The elaborated research methodology has been validated within an actual use case of the Bike2CAV[1] Project, funded by the Austrian Research Promotion Agency (FFG). This project used monocular vision-based 3D pose estimation to ensure safe interaction between cyclists and other vehicles.

[1] https://www.bike2cav.at/

1.1 Problem Statement & Challenges

The problem of monocular multi-target object detection and tracking can be defined as follows [FZH+21]:

Given a sequence of single, monocular RGB images depicting a dynamic scene with a variable number of objects, extract each object's 3D location, pose, and motion path with a consistent identity.

This thesis focuses on dynamic street scenes in autonomous driving, observed in time-consecutive street-level views. Therefore, cars, cyclists and pedestrians are the primary classes to detect and track.

The first challenge of this problem is to robustly estimate an object's relative 3D position and pose directly from its 2D appearance. Assuming that the observed object's movement is constrained to a ground plane, this task shall estimate its distance, 3D dimensions, and orientation in terms of an azimuthal angle with respect to an observer. This scenario is depicted in Figure 1.3. The pitch and roll orientations can be assumed to be zero, as these angles typically vary only within a limited range. The lack of depth information in monocular RGB images renders this task ambiguous, where learned priors on object attributes (dimension, orientation) and view-specific constraints must be exploited [KH21].

The second challenge concerns the association of target identities. Occlusions, clutter and detection failures make an association of detection responses between consecutive frames a challenging problem [ZWW+21]. Finally, there is an inherent ambiguity between object classification and reID. Conversely, the model tries to learn a wide range of possible appearances for the same class to classify them correctly, corresponding to the maximisation of inter-class separability. On the other hand, objects should be distinguishable even within a class for the reID task, which corresponds to maximising

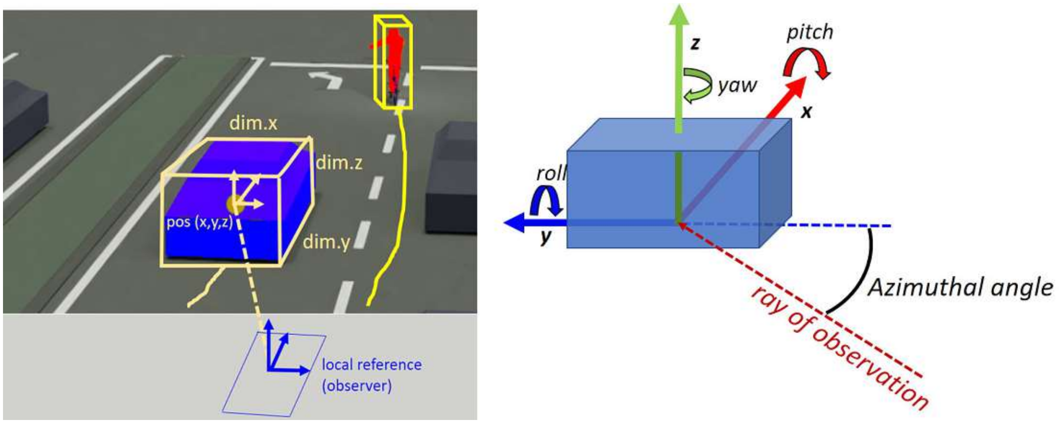


Figure 1.3: Illustration depicting the key pose parameters to be regressed within the monocular 3D pose estimation task.

intra-class compactness and separability. Thus, these learning objectives need to strike a balance, and their incorporation into the learning process requires thorough consideration [YLHW69].

The third challenge is that, besides the temporal and spatial variability of objects, our scenarios depict dynamic scenes where new categories can emerge. Current object detection concepts rely on a closed-world assumption, implying that only a fixed set of categories is learned. This limitation calls for learning frameworks that can introduce novel categories while retaining previously learned models.

In the next section, a monocular 3D object detection and tracking framework is proposed to tackle the first two challenges and a simple few-shot detection framework to tackle the third challenge.

1.2 Contributions

This work proposes a representation-enhanced end-to-end Deep Learning approach for 3D pose-aware multiple-object detection and tracking, using only monocular RGB images as input. Its representational concept is based on an encoder-decoder type multi-task learning scheme while also integrating recent representational breakthroughs devised explicitly for coping with spatial ambiguities and association uncertainties. The tracking integrates a reID approach that utilises a Transformer Encoder [VSP+17] with deformable attention [ZSL+20] to obtain target-specific appearance features using a spatially-delocalised exploration and correlation scheme.

The four key contributions of this thesis are:

- **Enhancing the backbone feature computation network by allowing for long-range correlated feature discovery** via an additional attention layer. This is a key capability to associate local 2D/3D percepts (object centre and keypoints) to 3D-object-level estimates, resulting in a spatial stabilisation along the viewing ray between the camera and object.

- **Tackling the increasing gradient problem of uncertainty guided depth estimation** by utilising the Robust Kullback-Leibler loss term [CHT+21]. This step allows for better convergence during training.

- **Improved reidentification by spatial attention from the Transformer Encoder for appearance-specific target representation**. This addition contributes to an improved 3D multi-target tracking in the presence of apparent abrupt motion.

- **Exploring and integrating a few-shot learning framework without forgetting**, enabling easy extension of learnt base classes by new classes using only a few (typically 3 - 10) manually annotated samples. The feasibility of this scheme has been validated in a different framework, including image acquisition and annotation, providing an interactive learning process with a graphical user interface.

1.3 Thesis Overview

Chapter 2 introduces concepts and state-of-the-art approaches used in object detection and tracking. First, the developments and approaches used in 2D object detection are presented. Then, the main concepts used for monocular 3D object detection and the representations used in state-of-the-art object detection frameworks are described. After that, general concepts used for multi-target tracking are discussed. Joint multi-target object detection and tracking frameworks are illustrated in the following section. The chapter concludes with an introduction to datasets and metrics used for evaluating object detection and multi-target tracking performance.

Chapter 3 focuses on the proposed monocular multi-target object detection and tracking methodology. After describing a synthetic data generation pipeline, the object detection and tracking framework is explained. The network architecture and representation for object detection are first illustrated, followed by the reID feature extraction method and target association strategy. Finally, the few-shot object detection methodology is presented.

Chapter 4 contains quantitative and qualitative evaluation results of the proposed framework on a dataset presented in Chapter 2, the synthetic dataset created using the methodology introduced in Chapter 3 and a private dataset from the Bike2CAV project. Furthermore, an ablation study is conducted to demonstrate the effects of the proposed contributions.

Finally, Chapter 5 comprises the conclusions drawn from this thesis, the limitations of the proposed methodology and possible future works.

CHAPTER 2

Concepts and State-of-the-Art

This chapter introduces the main concepts used in object detection, focusing on monocular 3D object detection and the geometry used. Additionally, the prominent representations used in state-of-the-art monocular object detectors are presented.

Afterwards, the different approaches used in multi-target tracking are discussed. Finally, the combination of the two into one end-to-end multi-task learning concept is described.

2.1 2D Object Detection

Object detection is the task of localising objects in a given image and classifying them into a category. Traditional methods for solving this task usually involve selecting hand-crafted features. However, selecting robust features that cover a large variety of appearances, lighting conditions, and backgrounds is a challenging task [XTY$^+$20]. With the advent of CNNs, neural networks started to outperform these traditional methods [Sze22]. The development of neural networks for object detection started with two-stage approaches, and later on, single-stage approaches were developed [QLL21]. Recently, multi-task frameworks have become popular, combining object detection with tasks such as segmentation [LKSC19] or tracking [ZWW$^+$21].

2.1.1 Two-Stage Detectors

The first neural networks for object detection mainly used a two-stage approach. The task was split into generating regions of interest and then classifying and refining them into appropriate bounding boxes. One of the first models to use this approach was the Region-based convolutional neural network [GDDM14], which showed that deep CNNs perform better in object detection than conventional methods. It uses a selective search method to generate region proposals in various sizes and aspect ratios. These are then run through a CNN to extract features and finally classified by a linear support vector machine model. This method wastes computation power on classifying each region proposal. Fast-R-CNN [Gir15] improved upon this by creating a feature map from the whole image, dividing the regions of interest into cells, and applying max pooling. The number of cells is fixed so that each region proposal generates a feature vector of the same size. This vector is then used as input for several fully connected layers whose output is split into two branches, one for the class scores and one for the bounding box regression. Faster R-CNN [RHGS17] and Mask R-CNN [HGDG17] brought further improvements, but the idea remained the same.

2.1.2 Single-Stage Detectors

The first significant novelty was the introduction of the one-stage detector YOLO (*You Only Look Once*) [RDGF16]. Instead of first generating region proposals and then classifying them, it views object detection as a regression problem. The image is divided into an $S \times S$ grid for which the network predicts B bounding boxes and the respective confidence and class probabilities for that cell. This approach led to significantly lower runtimes while providing a competitive accuracy. Another noteworthy development was the introduction of focal loss as a classification loss function with RetinaNet [LGG+17]. It tackles the issue that there are more background detections than actual foreground object detections.

2.1.3 Anchorless Detectors

All previously mentioned detectors have in common that they use anchor boxes for detections, meaning they have a set of preconfigured bounding boxes, which they adapt depending on the detection input. These have the drawbacks of introducing a significant computational burden due to the necessity of anchor boxes of various shapes and sizes. They also introduce additional hyperparameters regarding the design of these boxes. Therefore, anchorless object detectors have become more popular recently, the first noteworthy ones being CornerNet [LD18] and CenterNet [ZWK19]. The idea is to detect objects as keypoints instead of boxes: In CornerNet, the corners of the 2D bounding box, in CenterNet, the object centres. In CornerNet, the detected corners are then matched via an embedding. CenterNet regresses the bounding boxes in an additional regression branch that shares the same backbone as the classification branch. Especially CenterNet established state-of-the-art results while reducing runtime significantly and thus enabling accurate real-time object detection.

2.1.4 Transformer-Based Detectors

Transformers have already revolutionised the natural language processing domain [VSP+17]. However, their potential for computer vision due to their ability to discover long-range relationships has been uncovered in recent years. With DETR [CMS+20], a simple transformer-based approach for object detection was proposed. It uses a CNN as the feature extractor and then has a transformer encoder-decoder on top. It creates a fixed number of predictions of bounding boxes and their respective class for the given image and can achieve state-of-the-art performance. Vision [DBK$^+$20] and Swin Transformers [LLC$^+$21] are new backbones, replacing the CNN feature extractors, yielding state-of-the-art results in many computer vision tasks, including object detection. The top 7 object detectors on the COCO test dev benchmark dataset [LMB$^+$14] published in a paper use these Transformer backbones [ZLL$^+$22].

2.2 Monocular 3D Object Detection

While 2D object detection tries to estimate the bounding box of an object within the image plane, in monocular 3D object detection, the task is to estimate an object's location, size and pose in 3D space from a single RGB image. This section focuses on geometry, how 3D and image space correlate, and possible representations for the learning task.

2.2.1 Concepts

When taking a picture, points from 3D space are projected and transformed into the 2D image plane (see Figure 2.1). This operation can be written as:

$$p = K[R \quad t]p_w \tag{2.1}$$

where p is the image pixel coordinate, p_w is the 3D world coordinate. $K[R \quad t]$ represents the camera matrix, which can be split into two parts. K is the calibration or intrinsic camera matrix, and $[R \quad t]$ is the extrinsic camera matrix. The latter transforms the 3D world coordinates into camera-centred 3D world coordinates, while the former projects the 3D camera centred points into 2D image coordinates. In 3D object detection, the datasets are usually annotated with the object centre in 3D camera-centred coordinates and thus, only the camera intrinsic camera matrix is required during training. To obtain the pixel coordinates as a final step, one must divide the resulting vector p by its third element,

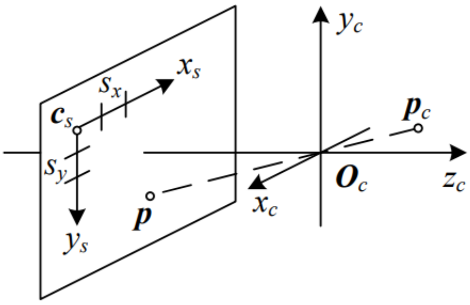


Figure 2.1: Projection of a 3D camera-centred point p_c onto the sensor plane at p. O_c is the optical centre and c_s is the 3D origin of the sensor coordinate system. Taken from [Sze22]

the depth. This operation illustrates the key problem in monocular object detection. To invert this transformation and get the location of an object in the image in 3D space, one has to know the depth. This information, however, is not contained in the image and therefore has to be extracted by other means[Sze22]. Other 3D object detection methods use sensor modalities like LiDAR [LVC$^+$19] or stereo camera setups [LCS19] to measure depth, but these sensors add costs. Therefore, in monocular 3D object detection, the detector estimates the depth based on geometric cues or other assumptions. Various representations can be used to obtain the final prediction, introduced in the following section.

2.2.2 Representations

Keypoint Estimation and Template Matching

The first neural networks used for monocular 3D object detection mainly used template matching methods, where the actual 3D size was chosen from pre-determined templates. These templates were either learned during training or used CAD models. The former method was used in Mono3D [CKZ$^+$16], which was the pioneering work in the field. It first generates dense proposals on a range of proposed ground planes, applies a scoring function and finally classifies and regresses the most promising candidates using Fast R-CNN. Bounding boxes are represented as (x, y, z, θ, c, t). Here, (x, y, z) is the 3D location of the bounding box centre. θ represents the azimuth angle on the ground plane, which is not regressed but classified as either $0°$ or $90°$. c is the object class, and t is a representative size template learnt during training. DeepMANTA [CCR$^+$17] used the latter method, utilising CAD models to recover 3D position and orientation during inference. It also introduces the concept of keypoint estimation, where certain vehicle parts' position is also predicted. This information is then used in a second step to match the detection to one of the template CAD models. This keypoint estimation, which can also be seen as a shape estimation, was picked up by others like Mono3D++ [HS19] and MonoGRNet [QWL63]. The template matching approach has the disadvantages of either a minimal number of templates, leading to a worse representation of the variety of objects or exponentially increasing computational requirements for finding the best match among a large number of templates. Additionally, keypoint and shape estimation has the downside that dense annotations for object shapes such as cars are complex and time-consuming. Hence, only limited data is available as ground truth. [KH21].

Representation Transformation and Dense Depth Prediction

Kim and Kum [KK19] have a different approach to overcoming the ambiguity between size and location in monocular object detection. Instead of estimating bounding boxes on the front view image, they first transform it into a BEV version. In the BEV, the scale of the objects is the same, so this ambiguity is eliminated. The transformation is achieved by first correcting the pitch and roll motion of the camera using the inertial measurement unit of the vehicle used for creating the data and then using inverse perspective mapping to obtain the BEV image. Afterwards, the problem is reduced to detecting an oriented 2D bounding box on the BEV image. [RKC18] follow the idea of the BEV transformation but use deep learning instead of the classical approach to achieve the transformation. Similarly, BirdGAN [SJS19] uses a generative adversarial network (GAN) to create a BEV image by image-to-image translation.

Considerable advancements in monocular depth estimation in recent years, which predicts the depth of each input image pixel, have enabled another transformation approach. [XC18] propose the multi-level fusion network, which uses the monocular depth estimation network MonoDepth [GMB17] to create a point cloud that, combined with the image features, is used to estimate 3D bounding boxes. [WK19] picked up this idea and proposed Pseudo-LiDAR, which directly uses state-of-the-art LiDAR detectors on the generated point cloud.

Direct Regression

With the introduction of anchorless object detectors, especially CenterNet, another, more straightforward representation gained popularity. Because CenterNet predicts object properties at a predicted object centre, it is easy to extend the 2D bounding box prediction to 3D bounding boxes by extending the predicted properties. [JZK19] follow that approach with their SS3D and regress the 2D bounding box, object distance, observation angle, dimensions and projected 3D bounding box corners for each object in their central region. This approach simplifies the network architecture while showing the best state-of-the-art results.

One of the currently best performing monocular 3D object detectors that does not require additional training data, MonoFLEX [ZLZ21], is based on the same approach but introduces additional parameters. In addition to the direct depth estimation in SS3D, the depth is estimated from the predicted corners of the projected 3D bounding box via geometric constraints [BABM19]. A weighted sum of the values, utilising learnt weights, achieves the final depth estimate.

2.3 Multi-Target Tracking

Tracking an object means associating detections of the same object over multiple frames with the same track, for example, via a unique identifier. When doing this for multiple objects in the same images, this is called multi-target tracking in contrast to single-target tracking, where only one object is tracked. There is also a differentiation between online and offline tracking. The former only has access to information from the past and the current sequence frame, while the latter has access to the whole sequence at once. Because autonomous vehicles require online tracking, this is the focus of this thesis. The main challenge in multi-target tracking is being able to reidentify objects even if they are occluded or wholly hidden for some frames and in the presence of an abundance of clutter. There are two main approaches to tackle this problem.

2.3.1 Location and Motion-Based Methods

The goal of location and motion-based methods is to associate detections by predicting the motion of an object based on previous detections. The simplest way of doing this is using a Kalman Filter [K+60] for the motion prediction and then the Hungarian algorithm [Kuh55] for the association. SORT [BGO+16] uses this approach for the 2D case and AB3DMOT [WWHK20] for the 3D case. More complex approaches often include considering camera motion [HHW+22] or using LSTMs for motion prediction [CZBO21].

All these approaches have in common that they are challenged by crowded scenes and fast motion, especially in combination with occlusions or wholly hidden passages. Especially the simpler ones, however, have the advantage that they are fast, with inference speeds of 100ks frames per second.

2.3.2 Appearance-Based Methods

The other idea is to identify objects based on their appearance, mainly achieved by extracting features from the images representing the object's appearance. This can be achieved in a separate detection and embedding approach, where an object detector is used to extract bounding boxes first, which are then used to crop out the detected objects. The cropped images are then fed into an embedding model to generate a target-specific embedding. This approach is illustrated in Figure 2.2 (a). A step towards jointly learning object detection and tracking was the idea of using a two-stage Faster R-CNN approach and introducing an additional embedding branch in the second stage that operates on the extracted regions of interest [VKO+19]. This is the approach shown in Figure 2.2 (b). Based on the extracted features, a matching is achieved by calculating the feature similarities and then using the Hungarian matching algorithm. The main advantage of this approach is its robustness to occlusions and even whole detection gaps, as appearances tend to be stable within sequences [ZWW+21]. However, they require training compared to the simple location- and motion-based methods.

2.3.3 Hybrid Methods

An evident approach to combat the issues of the individual methods is to combine them. This combination can be done hierarchically, e.g. by first trying association via one type and if the confidence there is too low, associate via the other one [CAZS18]. All cues can also be combined directly into one similarity metric for the association [SAS17, XCZH19, SWD+20].

2.4 Joint Monocular Multi-Target Object Detection and Tracking

Because a tracker needs detections to track the object, these tasks are closely tied. With the idea of multi-task learning recently becoming more popular, combining both tasks

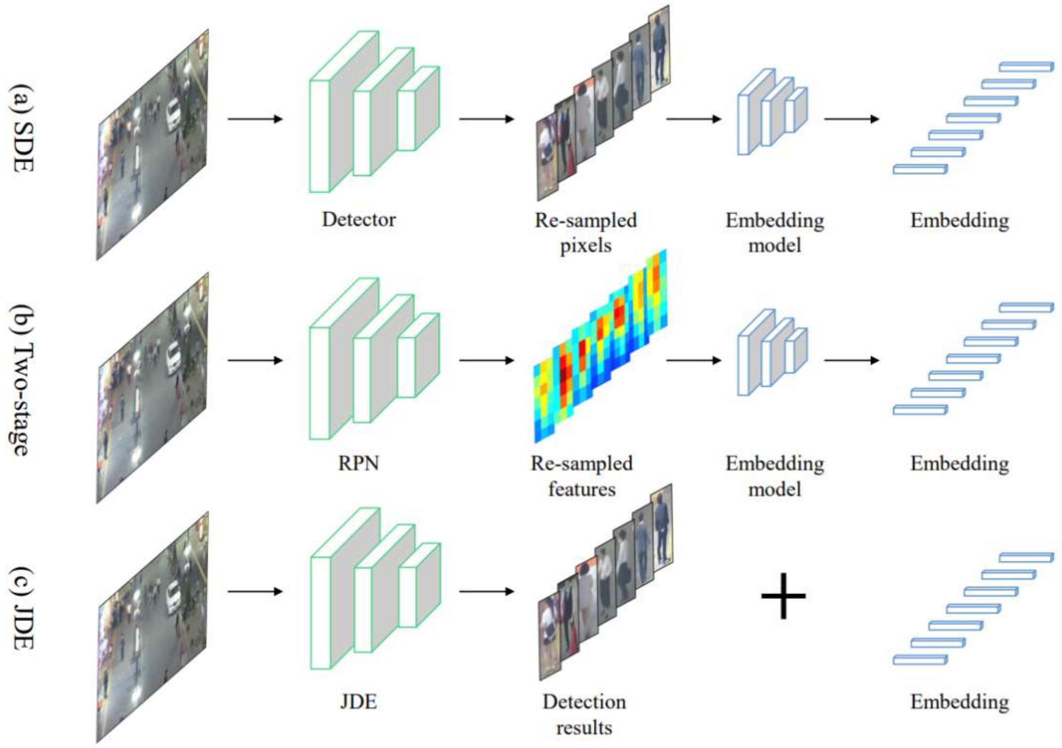


Figure 2.2: Illustration of different tracking approaches: (a) Seperate Detection and Embedding (SDE), (b) two-stage model and (c) Joint Detection and Embedding (JDE). Taken from [WZL$^+$20]

in a multi-task framework is reasonable. The most common approach to do so is by using a shared backbone as a feature extractor and then having one object detection branch and one tracking branch, all of which are trained together. This approach is illustrated and compared to its separate detection and tracking approaches in Figure 2.2. Each task has its loss function, and the complete network is updated with a weighted linear combination. This joint learning has two main advantages, the first one being the reduced model size and therefore reduced computational requirements. The second one is the more subtle idea that training multiple similar tasks at once can improve the performance in the individual tasks [KGC18, LJD19]. Concerning multi-target object detection and tracking, there are again two different tracking cues that can be used, reID and motion prediction. CenterTrack [ZKK20] is an example of the latter. It predicts the displacements of the object centres from pairwise inputs and then associates the predicted displaced centres with the newly detected ones. Trackers that jointly train reID features with the object detector include JDE [WZL$^+$20] and FairMOT [ZWW$^+$21]. The former uses YOLOv3 [RF18] as the detector, adds an embedding head and uses an uncertainty guided, fused loss function for training. FairMOT has a similar approach but uses CenterNet instead of YOLO. The authors argue that an anchor-based object detector introduces an unfairness between the reID and the detection task due to its cascaded architecture, favouring the detection. RelationTrack [YLHW69] addresses the issue that classification and reID have somewhat opposite goals. Classification wants to recognise all class variations as the same class, while reID tries to emphasise differences even within a class. Therefore, they introduce a disentanglement layer that reweights the individual tasks' features to decouple the classification and the reID task further. Additionally, they leverage the powerful spatial context of a Transformer Encoder to achieve an improved object embedding.

2.5 Datasets and Metrics

To train and evaluate object detection and tracking methods, especially data-driven ones, one needs suitable datasets with annotations and generally recognised metrics that quantify performance. Fortunately, large annotated datasets have been made freely available for everyone to use, which helped grow a large research community.

2.5.1 Datasets

Among the first ones regarding 2D object detection were the Microsoft Common Objects in Context (COCO) [LMB$^+$14] dataset and the PASCAL Visuals Object Classes [EZW$^+$05] dataset. Popular datasets for 3D object detection and tracking are the KITTI Vision Benchmark Suite [GLU12] and the nuScenes [HVA$^+$19] dataset.

COCO

The COCO dataset released in 2014 contains 2.5 million labelled instances across 91 common object categories in 328,000 images. It depicts these objects in everyday scenes to put them into their natural context, hence the name. It includes 2D bounding boxes, class labels, and instance segmentation, whereby the latter is not available for all otherwise labelled instances.

KITTI

The KITTI Benchmark Suite was published in 2012 as a challenging benchmark for visual recognition systems. The 3D object detection dataset consists of 80,256 labelled objects from 8 classes, including cars, pedestrians and cyclists, in 7481 training and 7518 test images depicting dynamic street scenes captured from atop a moving car in the Karlsruhe region in Germany. On top of the RGB images, it also contains corresponding image pairs for stereo vision, LiDAR point clouds, and GPS telemetry. In addition to the 3D object detection task, datasets for odometry, multi-object tracking, and segmentation were also released.

nuScenes

In 2020, Motional released the nuScenes dataset, containing 1.4 million 3D object bounding boxes in 40k keyframes, depicting dynamic street scenes in Boston and Singapore. It includes images from 6 different cameras and data from 5 RADARs and one LiDAR sensor, allowing a full 360-degree view of every scene. It also has a more granular classification scheme and more classes in general, with a total of 23. Compared to KITTI, it also contains data with challenging visibility conditions, including scenes at nighttime or during rain. Each object is also assigned a unique identifier allowing tracking evaluation in addition to the object detection task.

2.5.2 Metrics

Quantifying the performance of object detection and tracking methods requires metrics that enable a comparison of the different methods.The metrics used for the official evaluation of the datasets mentioned above are introduced in this section.

Average Precision and Intersection over Union

A metric for evaluating the quality of an object detector is the average precision (AP). It is the interpolation of the area under the precision-recall curve. Precision is given as:

$$P = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.2}$$

To determine whether a detection is counted as a true positive, the intersection over union (IoU) is used. The 2D case is defined as the area of the intersection divided by the area of overlap between the estimated box and the ground truth, hence the name. For the 3D case, the intersection between the polygons given by the oriented bounding boxes multiplied by their y-axis overlap is used as the intersection volume. This intersection volume is then divided by the sum of the two volumes of the bounding boxes minus the intersection volume to give the 3D IoU. The IoU of a detection and a ground truth label have to pass a set threshold to classify a detection as a true positive. This threshold varies between different datasets and classes. In KITTI, the threshold for the car class is set to 0.7, while the threshold for cyclists and pedestrians is 0.5. However, one cannot draw any conclusions about the performance from precision alone. A detector that only outputs boxes with a high confidence value and thus avoids false positives may have a decent accuracy but still not be considered functional. That is why one usually combines precision with recall. Recall is defined as:

$$R = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.3}$$

It, therefore, is an indicator of how many of the ground truth objects were detected. Again, one can not draw any conclusions from recall alone because a detector could predict boxes everywhere to achieve a recall of 1. Average precision now combines the two metrics to create a number more indicative of the actual performance. An object detector has to put out a confidence in a detection with a value between 0 and 1. The average precision as utilised in the KITTI benchmark uses the precision and recall at different thresholds for this confidence as follows:

$$AP_{40} = \frac{1}{40} \sum_{R \in \{0,1/39...,1\}} P_{\text{interp}}(R) \tag{2.4}$$

$$P_{\text{interp}}(R) = \max_{\tilde{R}:\tilde{R}>=R} P(\tilde{R}) \tag{2.5}$$

Equation 2.4 means that the precision is interpolated at 40 equally spaced recall levels and then averaged. Equation 2.5 defines that the interpolated precision is the maximum precision, where the recall value is greater than R, the recall level currently examined [PND20]. This metric is computed classwise and can be combined with the mean average precision by averaging the values across the different classes.

KITTI uses the average precision for cars considered moderately challenging to detect as the primary evaluation metric, while nuScenes uses the mean average precision over all classes.

MOTA

In multi-target tracking, more metrics can be used to compare different algorithms. One of the first ones was MOTA [BS08] which stands for multiple object tracking accuracy. A matching between object detections and ground truths must be established for every frame. These matchings are based on a distance metric between them and are only considered valid if this distance does not exceed a specified threshold. The formula for MOTA is then given as:

$$\text{MOTA} = 1 - \frac{\sum_{t=0}^{n}(M_t + FP_t + MME_t)}{\sum_t G_t} \tag{2.6}$$

The metric is calculated for an entire sequence of N images and, t represents the timestep in the sequence. M_t represents ground truth objects not assigned to a tracking hypothesis. FP_t corresponds to false positives, meaning detections that could not be assigned to a detection ground truth. MME_t are mismatch errors, meaning that an assignment for the current frame contradicts an assignment of the previous frame, and hence an identity switch occurs. G_t is the number of ground truths for the frame in question.

HOTA

Although being used as the evaluation metric for multi-target tracking on datasets such as KITTI, the MOTA metric has a lot of issues, such as an overemphasis on detection vs. association. Therefore, [LOD$^+$21] proposed a new metric called higher order tracking accuracy (HOTA). The basic definitions for matchings, such as true positive and false negative, are the same as in MOTA. While these terms are already used for detection, with HOTA, the concept of true positive associations, false negative associations and false positive associations are introduced for each true positive detection. True positive associations (TPA) are defined for a given true positive detection as true positives with the same ground truth ID and the same predicted ID as the original detection. False negative associations (FNA) are ground truth detections with the same ID as the original true positive that were falsely associated or missed by the tracker. Finally, false positive associations (FPA) were assigned the same ID as the original true positive by the tracker, but either have a different or no ground truth ID. These newly introduced concepts are used to construct the HOTA metric for a given localisation threshold as follows:

$$\mathrm{HOTA}_\alpha = \sqrt{\frac{\sum_{c\in\{\mathrm{TP}\}} A(c)}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}} \tag{2.7}$$

$$A(c) = \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(\mathrm{c})|} \tag{2.8}$$

Here, c are the elements of the set of true positive detections, TP are true positive detections, FN are false negative detections and FP are false positive detections.

To differentiate between the contributions of detection and association performance, HOTA can be decomposed into two separate scores as follows:

$$\mathrm{DetA}_\alpha = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|} \tag{2.9}$$

$$\mathrm{AssA}_\alpha = \frac{1}{|\mathrm{TP}|} \sum_{c\in\{TP\}} A(c) \tag{2.10}$$

The final HOTA metric is obtained by averaging over the HOTAs for 19 localisation thresholds from 0.05 to 0.95 in intervals of 0.05.

$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\alpha d\alpha \approx \frac{1}{19} \sum_{\alpha\in\{0.05,0.1,\ldots,0.9,0.95\}} \mathrm{HOTA}_\alpha \tag{2.11}$$

The overall DetA and AssA values are calculated in the same way.

2.6 Summary

In this chapter, the various approaches for object detection and tracking, in general, were presented, with a particular focus on the monocular 3D multi-target case. All state-of-the-art approaches utilise Deep Learning and have either a convolutional encoder-decoder network or a Vision Transformer network as a feature extractor. There are a variety of possible representations for the 3D object detection task that all have different tradeoffs. The currently best-performing frameworks use either a representation transformation or some direct regression. The currently most successful multi-target tracking approaches are jointly trained with object detector frameworks to exploit the advantages of multi-task learning. Public datasets with annotations exist to compare different approaches to both detection and tracking, and there is still active development, with nuScenes only being released in 2019. There are also meaningful metrics that enable comparison on these datasets, mainly mean average precision for object detection and the recently proposed HOTA for tracking, which tackles many of its predecessor's issues, MOTA.

CHAPTER 3

Proposed Methodology

This chapter describes the proposed methodology for the monocular 3D multi-target object detection and tracking framework. Section 3.1 illustrates the proposed synthetic data generation method and the generated toy dataset. Section 3.2 focuses on the framework, including the backbone network, the representations used, and the proposed improvements. The reID feature extraction mechanism and the target association process are explained before concluding with the loss functions used. Section 3.3 describes the few-shot learning approach used and illustrates the usage of the proposed GUI.

3.1 Synthetic Data Generation and Toy Dataset

The 3D datasets available mainly cover complex street scenes, which are challenging for 3D object detection and tracking in general, but especially for the monocular case. Therefore it may be desirable to test concepts on a simpler dataset. On the other hand, annotating data with 3D bounding boxes takes an enormous effort and would not be sensible for creating only simple training cases. Therefore, an approach that requires less work is to generate synthetic training datasets because one can generate the annotations simultaneously. Therefore, a data generation framework that creates video sequences of simple objects with corresponding 3D tracking annotations was created in the 3D animation software Blender (`https://www.blender.org/`). The annotations include:

- the position of the object in 3D space and its metric dimensions
- the rotation angles of the object
- the camera matrix used to obtain the image

The objects move along Lissajous curves [CR81] during the sequences, whose parameters can be varied between different sequences. Lissajous curves are parametrised curves defined by the following two equations:

$$x = a \sin(\omega_1 t) \tag{3.1}$$

$$y = b \sin(\omega_2 t + \phi) \tag{3.2}$$

where a and b are parameters that scale the curve along the x- and y-axis, respectively, and ϕ denotes the phase shift between the two oscillations. If the frequency ratio is a rational number, the equations form closed curves [MM15].

The generated objects include a cube, a cylinder and a monkey head, all available in Blender natively. The appearance can be adjusted by utilising different colours or materials. Two example images with different colours and positions are shown in Figure 3.1. The overall dataset created consists of 16 different scenes, with 100 images each. Per scene, the three objects depicted in Figure 3.1 - cube, cylinder and monkey head - move along a different Lissajous curve with randomised parameters. Per object, there are three different identities in the training set, being characterised by different colours and each appearing in three scenes. The training set consists of 15 sequences, while the validation set consists of one sequence with unseen identities.

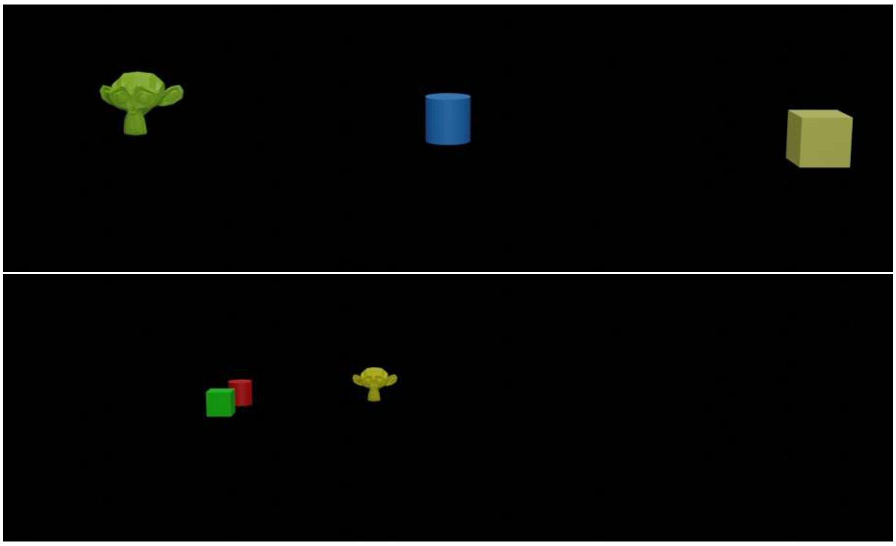


Figure 3.1: Example images from the generated synthetic dataset.

3.2 Monocular 3D Multi-Target Detection and Tracking

The overall proposed architecture is illustrated in Figure 3.2. The individual components, including the feature extractor backbone, the representation used for detection and the reID branch, are further described in the following section.

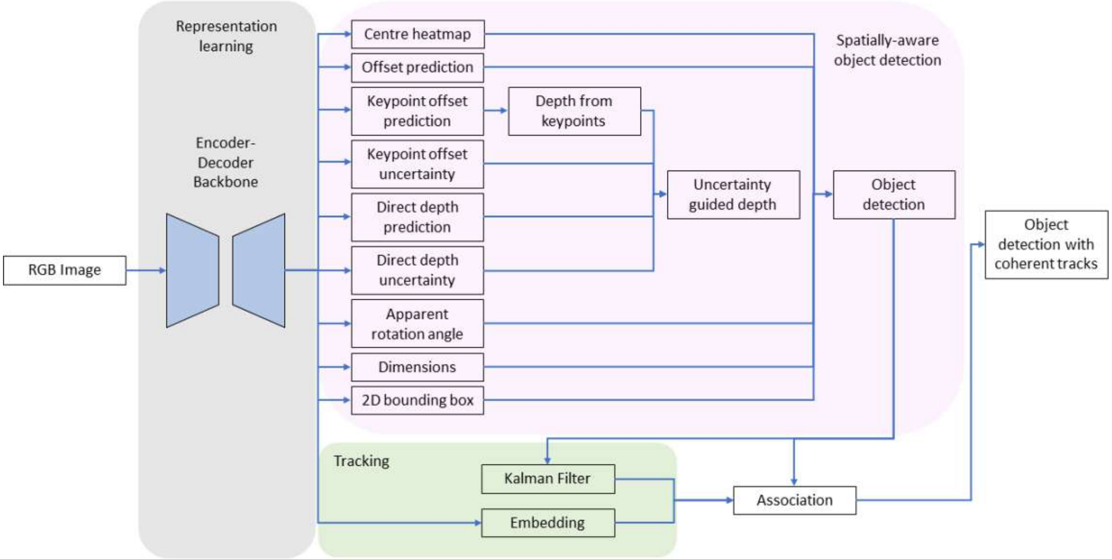


Figure 3.2: Illustration of the overall network architecture.

3.2.1 Computational Backbone

Like many state-of-the-art object detection frameworks, the backbone network utilises the hierarchical layer fusion network DLA-34 [YWSD18] as a basis. It strikes a good balance between runtime, complexity and accuracy and therefore was the logical choice. Like in CenterNet [ZWK19], all hierarchical aggregation connections are replaced by a Deformable Convolution Network [ZHLD19]. These deformable convolutions should allow the network to consider more long-range connections, which has been shown to improve detection performance in various object detectors. In [ZCZ^{+}19], the authors formulate deformable convolutions as a form of spatial attention and find that they consider the query content and the relative position. The authors also showed that utilising the key content improved object detection accuracy in their experiments while only slightly increasing the complexity. This observation led to the idea to include additional transformer attention in the DLA-34 network, as shown in Figure 3.3. However, because the deformable convolutions already cover the query content and the relative position, only the term focussing on the key content is non-zero. The output feature map is downsampled to a quarter of the original image size like in previous works.

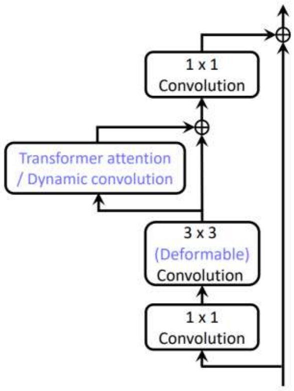


Figure 3.3: Illustration of attended residual block used in the backbone. Taken from [ZCZ+19]

The proposed method uses a keypoint based representation approach with the projected 3D object centre as the keypoint like in [ZWK19, LWT20]. The object detection task is split into two branches or heads, a keypoint branch that predicts said centre and the corresponding class on a heatmap and a regression branch that predicts the complete 3D information of the object. The output of the two branches is a grid that represents a downsampled version of the image. The keypoint branch outputs probabilities that there is an object at the location of the value in this grid. After non-maximum suppression, the most confident estimates are used for the regression. The regression information is then contained at that location in the output grid of the regression branches, as depicted in Figure 3.4.

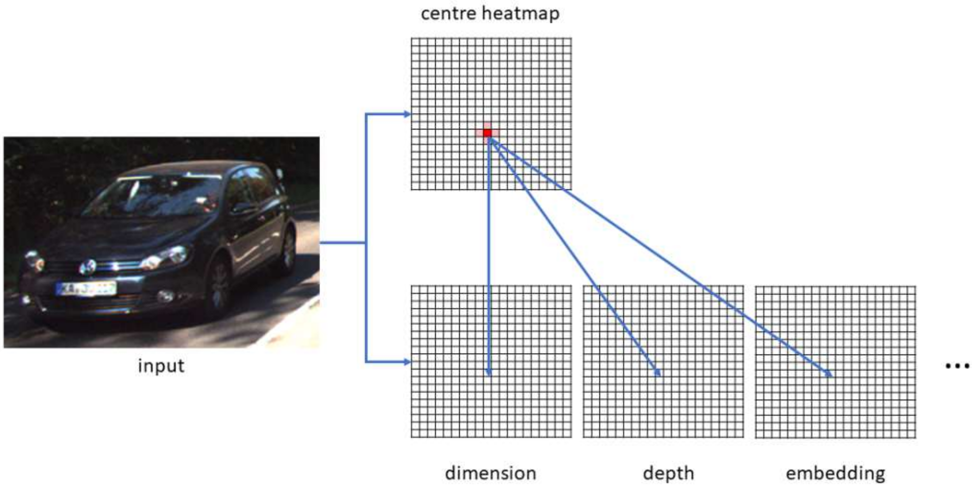


Figure 3.4: Illustration of the way the predicted centres are used to obtain the regression predictions.

The regressed 3D information is illustrated in Figure 3.5 and includes:

- The offset of the predicted centre due to the discretisation (c)
- The offset of the keypoints (3D bounding box corners and top and bottom centre points) from the discretised object centre (f)
- The uncertainty in that keypoint offset prediction
- The directly regressed depth (h)
- The uncertainty in that depth
- The apparent rotation angle (g)
- The objects' dimensions (height, width and length) (e)
- The objects' 2D bounding box dimensions (d)

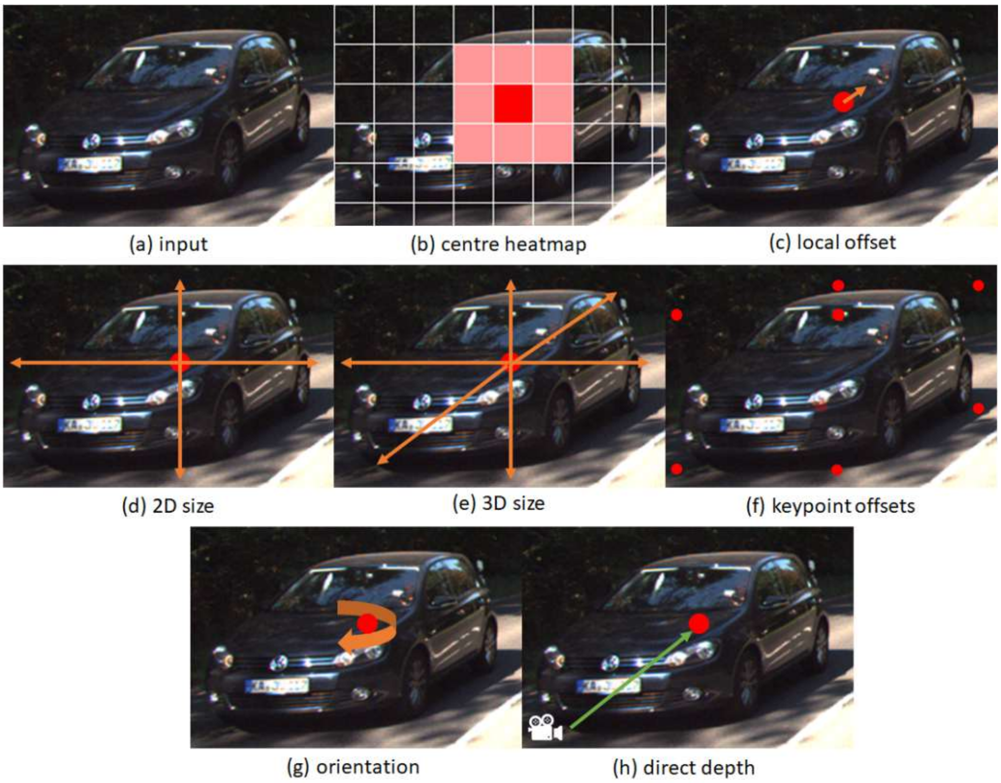


Figure 3.5: Overview of output of prediction heads. Image from the KITTI dataset [GLU12], overall graphic self made.

3.2.2 Object Detection and Representation

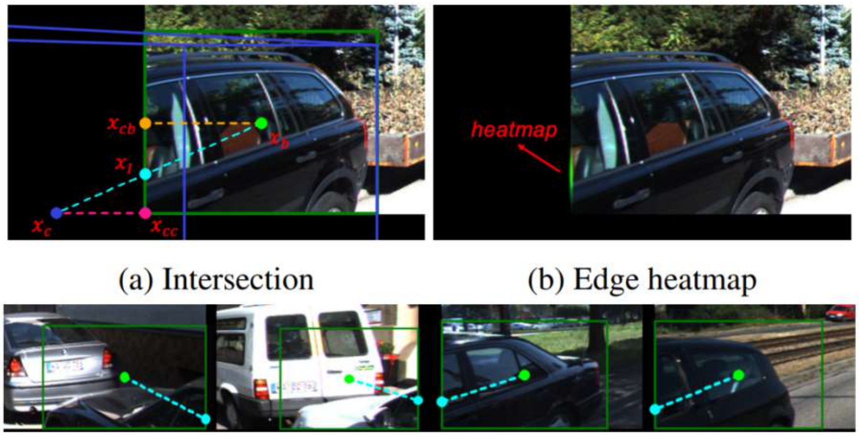
The keypoint head consists of a convolution layer, followed by batch normalisation and a leaky-ReLU activation, leading to another convolution layer. Each regression task gets its head with the same structure as the keypoint branch.

The projected 3D object centre (x_c, y_c) is obtained during training from the given 3D location (x, y, z) via the intrinsic camera matrix K as follows:

$$\begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} = \frac{K \begin{pmatrix} x \\ y \\ z \end{pmatrix}}{z} \tag{3.3}$$

A 2D Gaussian kernel then creates a ground truth heatmap from these projected centres following [ZWK19], with the standard deviation being derived from the dimensions of the projected 3D bounding boxes (see Figure 3.5 (b)). The larger the bounding box is on the downsampled heatmap, the larger the radius of the Gaussian kernel.

For objects whose 3D centre is outside the image, the intersection point between the line connecting the projected 3D centre and the 2D centre and the edge of the image is used as input to a 1D Gaussian kernel. This representation makes the used centre more physically meaningful, as shown in Figure 3.6. Additionally, an edge fusion module, which processes the edge of the image concatenated to a 1D vector through 1D convolutional layers, further decouples inside and outside objects. The result of the 1D convolutions is finally added to the edges of the input feature map.



(a) Intersection

(b) Edge heatmap

Figure 3.6: Illustration of the intersection point and the edge heatmap. Taken from [ZLZ21].

Seven parameters encode the 3D information of an object: $[x, y, z, h, w, l, \alpha]$. x, y and z describe the 3D location of the object centre, while h, w and l represent the dimensions of the object. α denotes the apparent yaw angle of the object on the ground plane.

However, as mentioned before, the regression branch does not directly regress all of these parameters.

First of all, instead of predicting the x and y location directly, it only calculates the discretisation offset from the centre that is predicted using the heatmap (see Figure 3.5 (b) and (c)). The additional feature information from the edge fusion module is utilised in the offset prediction for objects whose centre is outside the image.

Then, instead of directly regressing the object's dimensions, the offset to the class-specific average is predicted for each dimension.

The apparent yaw angle is directly regressed.

One of the most challenging parts of monocular object detection is depth prediction, and the most basic solution is to let the network directly predict the estimated depth. However, due to the inherent ambiguity between dimension and depth, it is hard for the network to make accurate predictions, especially from a single point. In order to assist the network in this depth prediction task, [ZLZ21] proposed the use of an adaptive depth ensemble. In addition to the direct depth regression, they propose regressing the depth from other keypoints of the object as well (see Figure 3.5 (f)). The keypoints chosen are the eight corners of the 3D bounding box and the points above and below the centre point on the corresponding face. However, the depth is not directly regressed for each keypoint but through five vertical supporting lines as depicted in Figure 3.7.

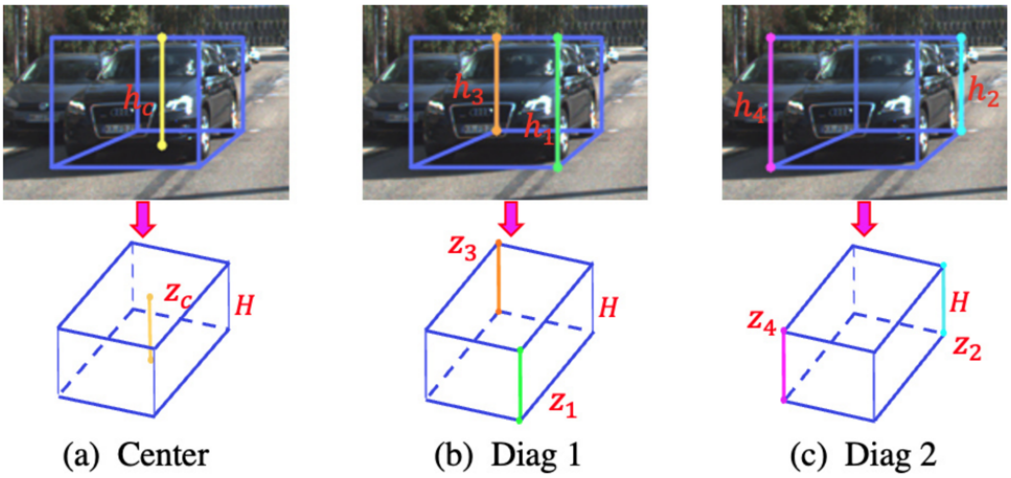


Figure 3.7: Illustration of the three supporting lines used for depth estimation. Taken from [ZLZ21].

The depth of each of these vertical lines is computed via its pixel height h_l and the proposed object height H as:

$$z_l = \frac{fH}{h_l} \tag{3.4}$$

To get the final depth prediction, the model also predicts an uncertainty value for each of the predicted depths and combines the predictions based on the inverse of the uncertainty:

$$z_{\text{soft}} = (\sum_{i=0}^{M} \frac{z_i}{\sigma_i} / \sum_{i=0}^{M} \frac{1}{\sigma_i}) \tag{3.5}$$

Here, z_i is the depth predicted and σ_i the predicted uncertainty in that depth using the ith supporting line or the direct depth prediction. M is the total number of supporting lines used.

3.2.3 ReID and Tracking

Using two sub-models, one for the detection task and one for the tracking task, has the disadvantage of an increased computational cost and not utilising shared features of the two. Therefore, the proposed architecture uses Joint Detection nd Embedding. In the proposed methods, objects are tracked through a combination of a Kalman Filter with reID from an embedding.

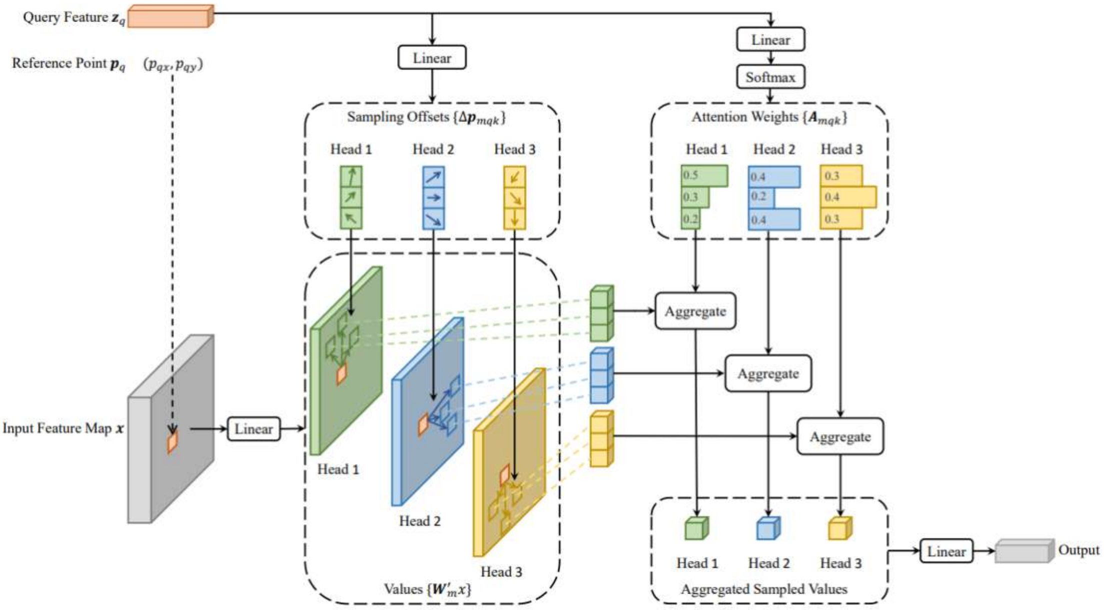


Figure 3.8: Overview of deformable attention module introduced by [ZSL+20]. Taken from [ZSL+20]

The Kalman Filter [K$^+$60] is a means of iteratively estimating parameters based on potentially erroneous observations, which in this application are the bounding box estimates of the detector network. The embedding is obtained by feeding the disentangled features for reID into a Transformer Encoder [VSP$^+$17] with deformable instead of multi-head attention as introduced by [ZSL$^+$20] (see Figure 3.8).

First, a position embedding is added to the extracted features to preserve spatial relations. These feature maps are then forwarded through three separate linear layers. The output of the first one is used to generate sampling offsets for each query node. These offsets are then used together with the output of the second linear layer for each node to obtain key sample vectors for each query node. The third linear layer is followed by a softmax and generates attention weights for each feature. These attention weights are multiplied for each sample vector, and the result is aggregated. Finally, the result is the input to a final linear layer whose output is the final embedding. During inference, this embedding generates a cost matrix between existing tracks and detections in the new image, then used to match them via the Hungarian algorithm [Kuh55]. Additionally, a Kalman Filter [K$^+$60] excludes physically impossible tracks by keeping track of the objects' approximate expected position and velocity.

3.2.4 Loss Functions

The multi-task learning framework is trained using several losses, whose values are combined using a weighted sum to form the total loss. The individual loss functions are described in this section.

Keypoint Classification Loss

The penalty-reduced pixel-wise logistic regression with focal loss [ZWK19, LGG$^+$17] is applied to the centre estimation on the downsampled heatmaps, which is defined as follows:

$$L_{\text{centre}} = \frac{-1}{N} \sum \begin{cases} (1 - \hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \log(1 - \hat{Y}_{xyc}) & \text{otherwise} \end{cases} \tag{3.6}$$

α and β are tunable hyperparameters that, in the proposed method, are set to their default parameters of $\alpha = 2$ and $\beta = 4$. The indices xy denote the location on the heatmap, while c denotes the class. Y represents the value on the ground truth heatmap, while $\hat{Y}$ is the value on the predicted heatmap. Compared to the standard cross-entropy loss, focal loss tackles the issue of the substantial imbalance between foreground and background points on object detection heatmaps while also enabling differentiation between easy and challenging examples. The imbalance can lead to a significant loss even for effortlessly classified examples with a high estimated probability. Introducing the term $(1 - p)^{\alpha}$ helps combat both issues at once by firstly weighting the loss term in order to combat the imbalance and secondly including the predicted probability to down weight easy examples while up weighing challenging examples. The parameter α determines the magnitude of this focus. The term $(1 - y)^{\beta}$ represents the penalty reduction part of the loss function. Because points close to the actual projected centre are acceptable too, especially when the object is close, the area around the centre is less penalised compared to faraway points.

Regression Losses

The regression of the discretisation offset is optimised using the L1 loss for objects whose centre is inside the image and the log-scale L1 loss for those whose centre is outside the image:

$$L_{\text{offset}} = \begin{cases} |\hat{\delta}_{inside} - \delta_{inside}| & \text{if inside} \\ \log\left(1 + |\hat{\delta}_{outside} - \delta_{outside}|\right) & \text{otherwise} \end{cases} \tag{3.7}$$

The apparent angle is directly regressed and clamped to $[-\pi, \pi]$ for estimating the orientation. The loss is calculated as an L2 loss:

$$L_{\text{angle}} = (\sin(\hat{\alpha}) - \sin(\alpha))^2 + (\cos(\hat{\alpha}) - \cos(\alpha))^2 \tag{3.8}$$

Here, α denotes the ground truth of the apparent angle and $\hat{\alpha}$ the predicted value.

The dimensions are regressed using the L1 loss as follows:

$$L_{\text{dimension}} = \sum_{k \in \{h,w,l\}} |\bar{k}_c e^{\hat{\delta_k}} - k| \tag{3.9}$$

where $\hat{\delta_k}$ represents the predicted offset from the statistical mean for that class c. The keypoints are regressed as offsets from the projected centre and optimised using L1 loss:

$$L_{\text{keypoints}} = \frac{\sum_{i=0}^{N_k} I_{\text{inside}}(k_i)|\hat{\delta}_{ki} - \delta_{ki}|}{\sum_{i=0}^{N_k} I_{inside}(k_i)} \tag{3.10}$$

Note that the term $I_{\text{inside}}(k_i)$ means that only keypoints visible inside the image are penalised.

Although the task is 3D object detection, experiments by [ZLZ21] have shown that incorporating a loss for 2D bounding boxes improves accuracy for the 3D task. Therefore, the network predicts the 2D bounding boxes too by regressing the distances of the top left and bottom right corners to the centre point. The GIoU loss [RTG+19] is then applied for optimisation:

$$GIoU = IoU - \frac{A^c - U}{A^c} L_{\text{2D}} \qquad = 1 - GIoU \tag{3.11}$$

Instead of only using the IoU, as described in Section 2.5.2, there is an additional term subtracting the portion of the smallest enclosing bounding box of the two compared boxes, denoted as A^c, that is not covered by the union.

Instead of the Laplacian Kullback-Leibler loss used in [ZLZ21], the robust Kullback-Leibler loss proposed in [CHT+21] is used to optimise the direct depth. The Laplacian Kullback-Leibler loss has the issue that its gradient w.r.t to μ usually increases during training as the uncertainty σ decreases. This gradient increase can lead to loss imbalance in multi-task learning scenarios as L1 and L2 losses either have a decreasing or constant loss. During experiments, this sometimes led to an explosion of the depth loss during validation. Additionally, the function is not differentiable at $\hat{y} = y$, so if the prediction $\hat{y}$ exactly matches the ground truth y. To overcome these issues, [CHT+21] propose a robust Kullback-Leibler loss that uses an exponential moving average to normalise the uncertainty and a mix of a Gaussian and a Laplacian Kullback-Leibler loss:

$$L_{\text{Robust KL}} = \frac{1}{\hat{w}} \begin{cases} \frac{1}{2}e^2 + \log \sigma & |e| <= \sqrt{2} \\ \sqrt{2}|e| - 1 + \log \sigma & |e| > \sqrt{2} \end{cases} \tag{3.12}$$

$$\hat{w} \leftarrow \alpha \hat{w} + (1 - \alpha) \frac{1}{N} \sum_{i=0}^{N} \frac{1}{\sigma_i} \tag{3.13}$$

where $e = |\hat{y} - y|/\sigma$ is the L1 error of the prediction, σ denotes the predicted uncertainty in the estimation, and N is the number of predictions made. α is a hyperparameter that determines the impact of new observations on the exponential moving average of the inverse of the uncertainties $\hat{w}$.

The depths estimated using the keypoints are also optimised with the same loss function. However, only visible keypoints contribute to the added log term, allowing the model to disregard non-visible points actively. Additionally, the final depth prediction obtained via Equation 3.5 is optimised via an L1 loss function.

An L1 loss is used on the 3D boxes being generated using the predicted orientation, dimension, location and uncertainty guided depth to guide the combination of the different subtasks:

$$L_{\text{bounding box}} = \sum_{i=0}^{7} |\hat{v}_i - v_i| \tag{3.14}$$

where v_i denotes the corner points of the 3D bounding boxes.

ReID loss

During training, the reID task is implemented as a classification task. The embeddings are extracted at the ground truth location of the object centre (see Figure 3.4) and then fed through a linear layer with the number of unique objects in the training data K as its output dimension. The cross-entropy loss is then used as the loss function:

$$L_{\text{reID}} = -\sum_{i=0}^{N} \sum_{k=0}^{K} L^i(k) \log\left(p(k)\right) \tag{3.15}$$

Here, $p(k)$ denotes the predicted probability that the detection belongs to object k of the K unique identities in total in the training set. $L^i(k)$ is the one-hot encoded representation of the ground truth identity label, and N is the total number of detections in the image.

3.3 Few-Shot Object Detection and Tracking

3.3.1 Backbone

Here, the intention was to use the same backbone as mentioned earlier, along with the proposed added attention changes. However, the goal was also to implement the network on an embedded device (NVIDIA Jetson Xavier NX) and that it is real-time capable. These requirements meant that directly using PyTorch was not enough since running times were too high, as further evaluated in the results section. Instead, the network is run using TensorRT-Torch, the highly optimised framework for PyTorch on NVIDIA Hardware. However, this framework does not support deformable convolutions directly, and hence the DLA-34 is replaced by a standard ResNet-101. This change negatively impacts detection performance but is a necessary sacrifice.

3.3.2 Object Detection and Representation

The requirements were only to have a 2D object detector in this setup because the depth information is available via a stereo camera configuration. Therefore, the detection task is simplified, and only the object centres and the 2D bounding box sizes need to be predicted.

Again, the object centres, this time the 2D ones, are predicted using a heatmap, where the ground truth heatmap is created using a 2D Gaussian kernel. Because only the 2D centres are of interest here, there are no outside centres, and thus no edge heatmap is used. However, because the network should adapt to new, unseen classes with very little training data (1 - 10 images), the convolution block used as the classification head is replaced by an adaptive cosine head introduced by [ZCW+21]. It transforms the input feature map into a similarity heatmap that indicates how similar the features are to a learned class prototype. Additionally, an adaptive, class-specific scale factor τ_c normalises different intra-category variances. The final similarity function is as follows:

$$S_{xyc} = \sigma(\tau \cdot \tau_c \frac{W_c^T \cdot F_{xy}}{|W_c| \cdot |F_{xy}|}) \tag{3.16}$$

where τ is a general scaling factor and W_c are the learnable prototypes.

The 2D bounding box is parametrised via its dimensions and centre $[c_x, c_y, w, h]$.

3.3.3 Losses

Again, the penalty reduced cross-entropy loss is used for the heatmap, and the offsets and the dimensions are optimised using L1 loss

$$L_{\text{centre}} = \frac{-1}{N} \sum \begin{cases} (1 - \hat{Y}_{xyc})^{\alpha} \log{(\hat{Y}_{xyc})} & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \log{(1 - \hat{Y}_{xyc})} & \text{otherwise} \end{cases} \tag{3.17}$$

$$L_{\text{offset}} = |\widehat{\delta_{inside}} - \delta_{inside}| \tag{3.18}$$

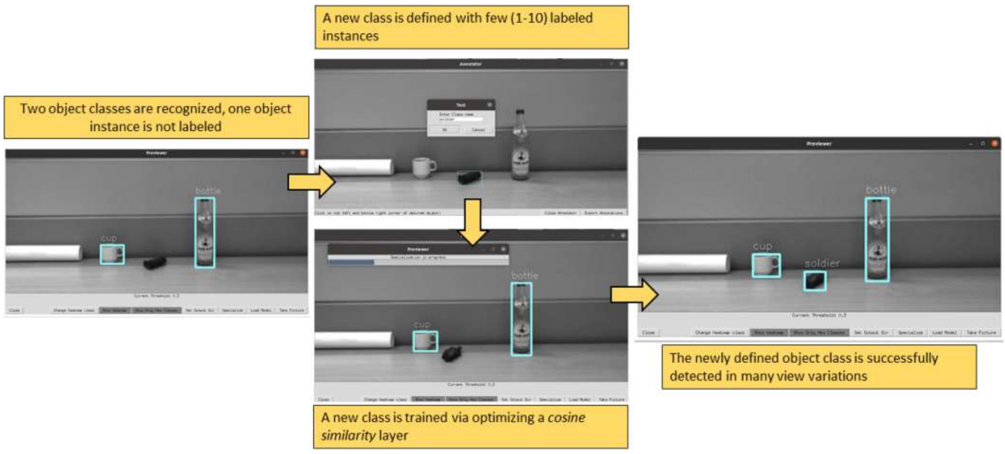
$$L_{\text{dimension}} = \sum_{k \in \{h,w\}} |\hat{k} - k| \tag{3.19}$$

3.3.4 Training and Inference

The large-scale COCO [LMB+14] dataset is used for the initial training to obtain prototypes for the base categories and learn a class-agnostic bounding-box regressor. In the few-shot learning phase, a new head is added parallel to the previous ones, generating the similarity heatmaps for the new classes. It is initialised using weights from the original classification head to preserve previously learned high-level knowledge. Then all other network parameters are fixed, and only the new head is optimised using the same penalty reduced cross-entropy loss as during base training. For the inference part, the outputs of the different classification heads are stacked to obtain the heatmap predictions for all categories. The bounding boxes are extracted at the found points of interest.

3.3.5 Graphical User Interface

A graphical user interface (GUI) was designed to ease model loading and few-shot learning in practical applications, and it is showcased in Figure 3.9.



Two object classes are recognized, one object instance is not labeled

A new class is defined with few (1-10) labeled instances

A new class is trained via optimizing a *cosine similarity* layer

The newly defined object class is successfully detected in many view variations

Figure 3.9: Graphical User Interface that enables life capturing, annotating and specialising on new classes.

The GUI allows a user to look at a live video feed of an attached camera. The user can then load a model pre-trained on a large dataset or a previously specialised one. The user can then capture an image of a new object that the network should detect from a live camera feed. The image can then be annotated by simply clicking on the estimated top left and bottom right corner of the object's bounding box and then specifying the name of the class. One can also annotate multiple new classes at once. After annotating the desired number of images, the user can specialise the model with a simple button push. After the specialisation is finished, the newly trained model is directly loaded, and the new objects should be detected.

3.4 Summary

In summary, the proposed monocular 3D object detection and tracking framework uses a feature extractor shared between separate branches for each task. This backbone has been modified with additional attention layers to enhance its ability to capture long-range relationships.

The object detection branch predicts object centres via a probability heatmap and regresses the 3D bounding box information at that location. An uncertainty guided ensemble of direct and keypoint based depth prediction is used for the depth estimation. This depth estimation is optimised using the robust Kullback-Leibler loss, mitigating an increasing gradient during training.

For the reID task, a transformer encoder with deformable attention is used instead of vanilla CNNs capturing long-range spatial relationships to extract meaningful object embeddings. During inference, these embeddings are combined with a Kalman Filter, which excludes physically impossible matches, to generate a cost matrix used for the association between detections and tracks using the Hungarian algorithm.

Finally, a few-shot detection framework that uses an adaptive cosine similarity was combined with a GUI to enable simple live model adaption to new classes for 2D tracking.

CHAPTER 4

Results and Discussion

In the previous section, the proposed algorithmic concepts were introduced. As shown in Figure 4.1, the proposed contributions affect all stages of object attribute learning and tracking. Spatial awareness and target-specificity were vital guiding principles to enhance the proposed object detection and tracking scheme. Accordingly, experiments have been devised to demonstrate the representational strength of these concepts through several experiments. Following validation experiments are designed and addressed:

- Enhancing long-range representation within the backbone feature computation step using additional attention layers
- Disentangling features for the object detection and the reID task using a Global Context Disentanglement layer
- Stabilising the gradient during training using a robust version of the Kullback-Leibler loss
- Extracting meaningful long-range features for reID using a Transformer Encoder and comparing it to a standard convolutional layer, and comparing the association results to a simple Kalman Filter

These concepts are evaluated in the context of the monocular 3D multi-target object detection and tracking tasks. This experimental section describes the employed training and validation methodology and the used datasets and validation metrics. Next, qualitative and quantitative analysis for the four experiments is presented.

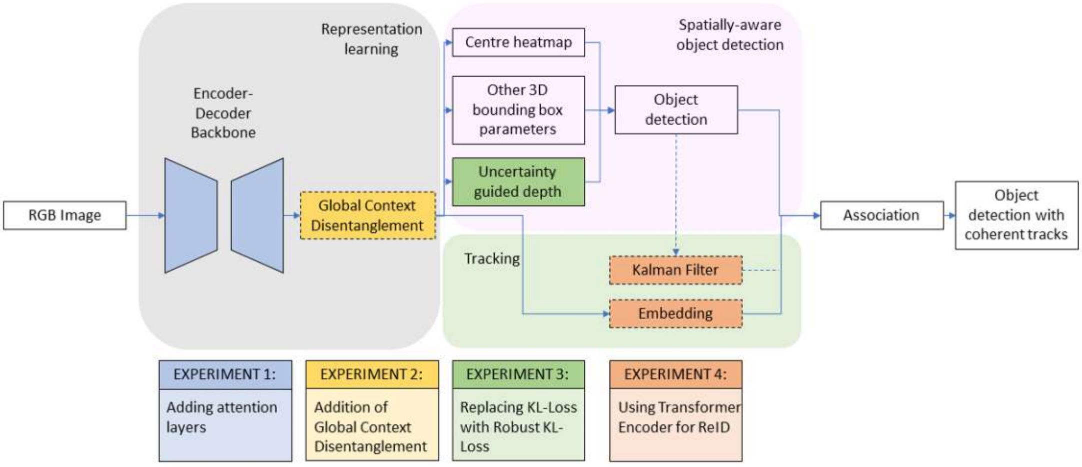

Representation learning
Encoder-Decoder Backbone
RGB Image
Global Context Disentanglement
Spatially-aware object detection
Centre heatmap
Other 3D bounding box parameters
Uncertainty guided depth
Object detection
Tracking
Kalman Filter
Embedding
Association
Object detection with coherent tracks
EXPERIMENT 1:
Adding attention layers
EXPERIMENT 2:
Addition of Global Context Disentanglement
EXPERIMENT 3:
Replacing KL-Loss with Robust KL-Loss
EXPERIMENT 4:
Using Transformer Encoder for ReID

Figure 4.1: Overview of different stages and conducted experiments

4.1 Methodology

This section describes the methodology used for evaluating the proposed framework. First, the dataset and the metrics used are described, followed by an explanation of how the framework is compared to other approaches. Finally, the training parameters used are given.

4.1.1 Datasets

The tracking results are evaluated on the KITTI tracking benchmark dataset using the training/validation split proposed by [VKO$^+$19]. The details about the split are shown in Table 4.1, which also includes a different split proposed by [ZKK20]. This latter split has the issue that by splitting the sequences in half, there are objects that are both in the training and the validation data, denoted in Table 4.1 as "Obj. ID overlap". These shared object identities are problematic for appearance-based tracking schemes, as this means there is an information leak from the training to the validation set. Hence, the performance may appear higher than it actually is, as further discussed in Section 4.1.3.

The dataset depicts dynamic street scenes focusing on cars, pedestrians and cyclists. An attempt has also been made to evaluate the framework on the much larger nuScenes dataset. Cleaning the data from impossible to see object annotations proved to be a challenging task, and hence it was deemed out of scope for this thesis, and without this pre-processing step, there were no sensible results. However, evaluation was performed on a synthetic dataset depicting simple objects, including a monkey head, a cylinder and

Split	Training		Validation		Obj. ID overlap
	No. seq.	No. img.	No. seq.	No. img.	
Full training	21	8,008	0	0	No
CenterTrack split [ZKK20]	21	3,999	21	4,009	Yes
MOTS split [VKO^{+}19]	**12**	**5,027**	**9**	**2,981**	**No**

Table 4.1: Three splits used for evaluation on the KITTI dataset. All experiments conducted in this thesis use the MOTS split.

a cube on a black background moving on tracks defined by Lissajous figures. Object detection was also evaluated on the KITTI dataset and its 3D object detection benchmark using the split suggested by Mono 3D [CKZ^{+}16]. This work contains no results on the KITTI test set because the upload to the evaluation server is restricted to published papers.

4.1.2 Metrics

HOTA [LOD^{+}21] is the primary metric for the tracking benchmark, with detection accuracy and association accuracy also being reported to break down the individual contributions of the two. Object detection is evaluated using the AP at an IoU of 0.7 for the car class across three difficulty levels - easy, moderate and hard - defined on the official website of the benchmark and shown here in Table 4.2 [1].

Difficulty Level	Min. 2D bounding box height	Max. occlusion level	Max. truncation
Easy	40 Pixel	Fully visible	15 %
Moderate	25 Pixel	Partly occluded	30 %
Hard	25 Pixel	Difficult to see	50 %

Table 4.2: The definitions of the diffculty levels of the KITTI 3D object detection benchmark as stated on the official website.

4.1.3 Comparison to State-of-the-Art

A comparison with other 3D detection frameworks is difficult because there are currently no other published results for monocular multi-target 3D object detection on the KITTI validation dataset. Most other works tackle 2D object detection and tracking or utilise LiDAR detections for training. Therefore, a baseline only using a Kalman Filter with a Hungarian Matching for the association is used as a comparison. This baseline utilises the same object detection approach. However, it does not utilise reID features for target

[1] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

association, only relying on the IoU of detections in a new frame with bounding boxes that the Kalman Filter predicts based on the previous frame instead.

Additionally, comparisons in 2D object detection to other state-of-the-art tracking frameworks are given. However, this comparison is not fair in several regards. Motion-based trackers such as CenterTrack [ZKK20] use all 21 available sequences for training, only splitting each sequence in half as the training/validation split. This split leads to better object detection performance, as the model has more variety in the training data, and therefore it is sensible for these types of tracking frameworks.

However, this split should not be used with appearance-based trackers because the validation set most likely contains objects used to train the appearance-based model. This overlap probably leads to better performance on these objects, while there is no guarantee for a good generalisation. Therefore, the validation set results of DEFT [CZBO21], for example, are likely higher than they would be if the two sets were wholly separated. Additionally, DEFT trains only for the 2D task on that benchmark, while this work trains for both 3D and 2D.

4.1.4 Training

The network was jointly trained on both 3D object detection and reID simultaneously. The training parameters used are shown in Table 4.3.

Parameter	**Value**
Input image size	384 × 1280
Resizing method	Padding
Optimiser	AdamW
Learning rate	3×10^{-4}, with decay at 80 and 90 epochs by 10
Training epochs	100
Augmentation	Random horizontal flip
Total loss weights	reID: 1 , detection: 1
Detection loss weights	L_{centre}: 1, $L_{\text{offset inside}}$: 0.5, $L_{\text{offset outside}}$: 0.1, L_{angle}: 1, $L_{\text{dimension}}$: 1, $L_{\text{keypoints}}$ 0.2, $L_{\text{direct depth}}$: 1, $L_{\text{keypoint depth}}$: 0.2, $L_{\text{soft depth}}$: 0.2, $L_{\text{bounding box}}$: 0.2, L_{2D}: 1

Table 4.3: The parameters used for training across all experiments.

4.2 Qualitative Results

Figure 4.2 shows qualitative results on the KITTI tracking validation set. The same colour of the boxes means the same track ID has been assigned. As one can see, most of the ids are consistent across the sequence. The pedestrian with the red box at the red light disappears for some frames but is recognised several frames later as the same object. The car's identity is teal in the first two frames and changes to purple in the last one. This identity loss is likely to the truncation introduced in the last frame. In Figure 4.3, one can see a situation where the reID approach shows its advantage: While turning into a street, the sudden movement exceeds the capabilities of the Kalman Filter and leads to a new identity in every frame for the four cars on the right, as can be seen at bottom three images. On the top three images, one can see that the reID approach manages to track three of the four cars accurately in the last two frames and one of them even for all three frames.

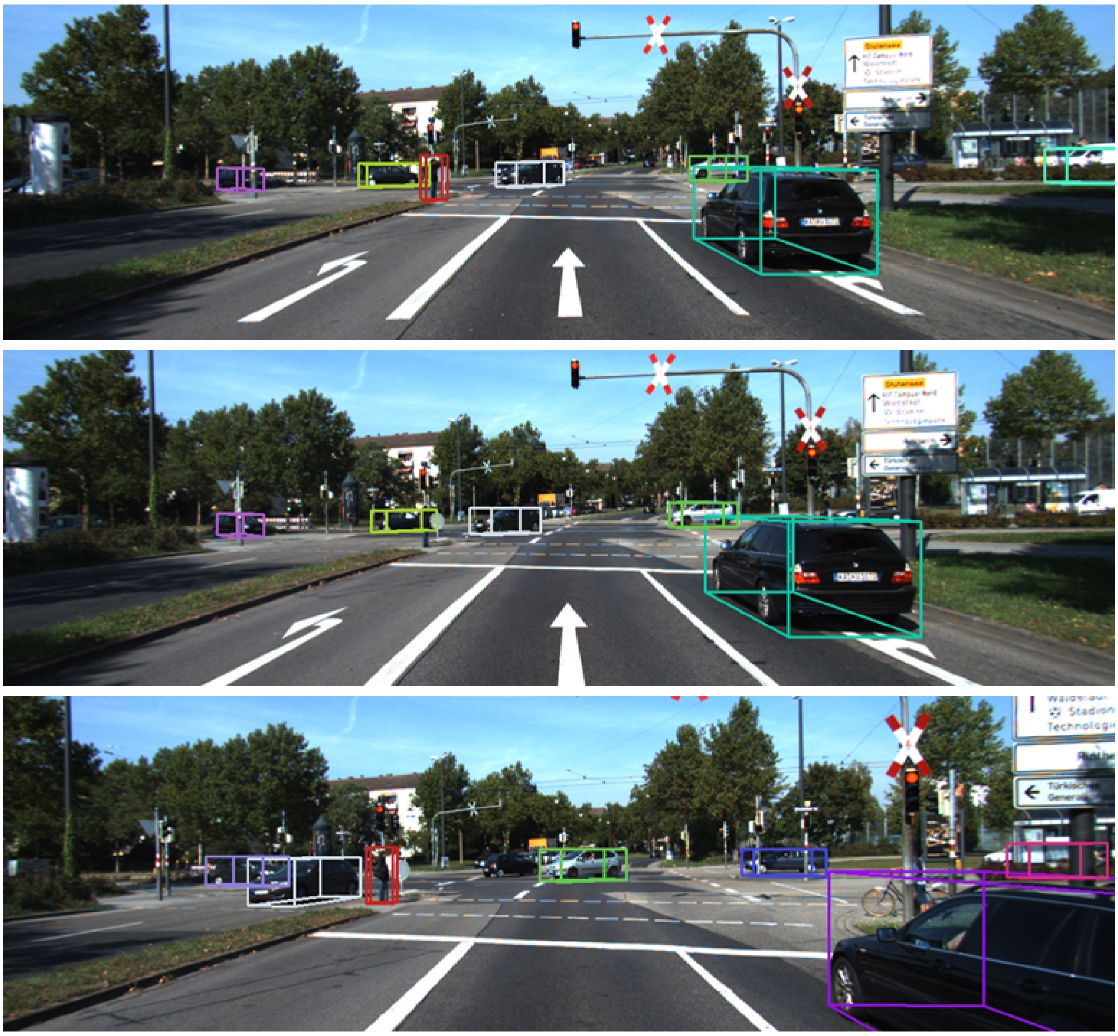


Figure 4.2: Example tracking results of proposed method on the KITTI validation set. Same coloured bounding boxes denote the same track identity assigned.

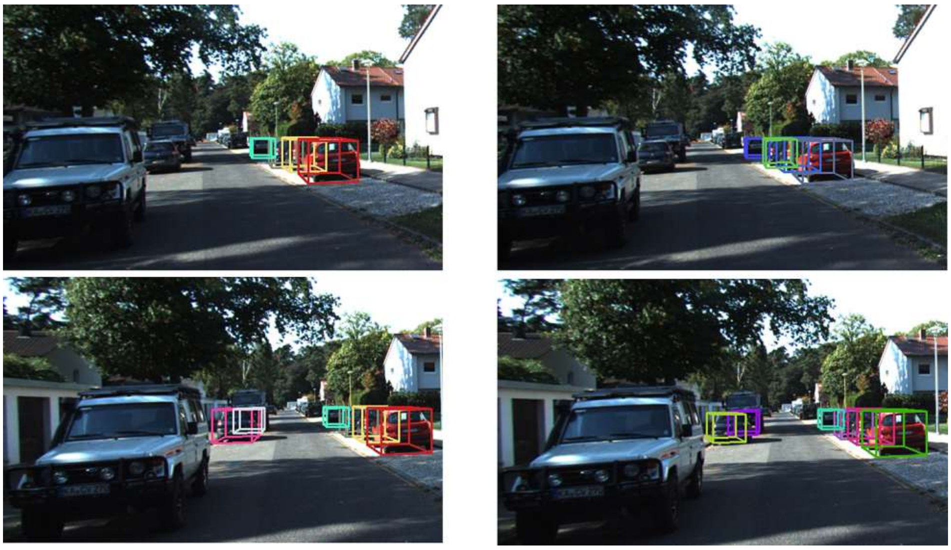


Figure 4.3: Comparison between proposed reID approach (left) and simple Kalman Filter (right) on the KITTI validation set.

4.3 Quantitative Results

The monocular 3D object detection and tracking results on the KITTI tracking validation set can be found in Table 4.4. The results are averaged over three runs with different random seeds due to a non-negligible run to run variation. One can see an improvement in HOTA compared to the baseline using the Kalman Filter only. The 3D object detection results on the KITTI 3D object detection validation set are shown in Table 4.5, showing the competitiveness with state of the art. The 2D multi-target tracking results compared to DEFT are shown in Table 4.6. There is a massive performance difference between all three entries shown. DEFT trained on the entire training dataset and evaluated on the KITTI test split has the best results. The proposed method follows in second, with decent results considering that 2D detection is not the focus. To compare the proposed framework more directly to DEFT, the published code has been used with the same parameters as stated in the paper but using the same training and validation split introduced by [VKO+19] that was used for training the proposed method. The results are underwhelming compared to both other results. Note, however, that no adaptations were made to the parameters in DEFT. Therefore, the comparison is still not fair, this time favouring the proposed method.

Method	**HOTA**	**DetA**	**AssA**
Baseline (Kalman Filter)	30.86	22.79	42.68
Proposed Method	**30.96**	**22.89**	**42.81**

Table 4.4: The monocular 3D multi-target tracking results of the proposed method compared to the Kalman Baseline.

Method	AP_{3D}		
	Easy	**Moderate**	**Hard**
SMOKE [LWT20]	14.76	12.85	11.50
MonoGeo [ZMY+21]	18.45	14.48	12.87
Ground-aware Monocular 3D Obj. Det. [LYL21]	23.63	16.16	12.06
MonoFlex [ZLZ21]	23.64	17.51	14.83
Proposed Method	**20.56**	**15.00**	**11.79**

Table 4.5: The monocular 3D object detection results of the proposed method compared to state of the art methods.

Method	**HOTA**	**DetA**	**AssA**
DEFT on test split	74.23	75.33	73.79
DEFT trained on train/val split	37.66	28.55	50.33
Proposed Method	**56.64**	**54.68**	**59.26**

Table 4.6: The 2D multi-target tracking results of the proposed method compared to DEFT [CZBO21].

4.4 Ablation Studies

After demonstrating the overall results of the proposed framework, this section focuses on the effect of the proposed changes. In Table 4.7, the quantitative results of these changes are shown. It demonstrates that all the proposed changes positively impact the model performance. The detailed effects of the individual components are discussed in the following section.

Method	**HOTA**	**DetA**	**AssA**
Without attention in backbone	30.04	21.88	42.34
With GCD module	28.63	19.98	42.02
Without Robust KL loss	29.57	21.23	42.57
With vanilla CNNs instead of Transformer Encoder	30.25	21.77	43.5
Full model	**30.96**	**22.89**	**42.81**

Table 4.7: The impact of the proposed changes to the 3D multi-target tracking results.

4.4.1 Attention-Enriched Backbone

The first entry in Table 4.7 shows the model performance with a vanilla backbone network. Compared to the full model, there is a slight improvement in HOTA of 0.92%. The AssA only increased by 0.44%, while the DetA increased by 1.01%. The former shows that the ability to extract representative appearance features only minimally benefits from the additional attention layer. The latter demonstrates that the additional information

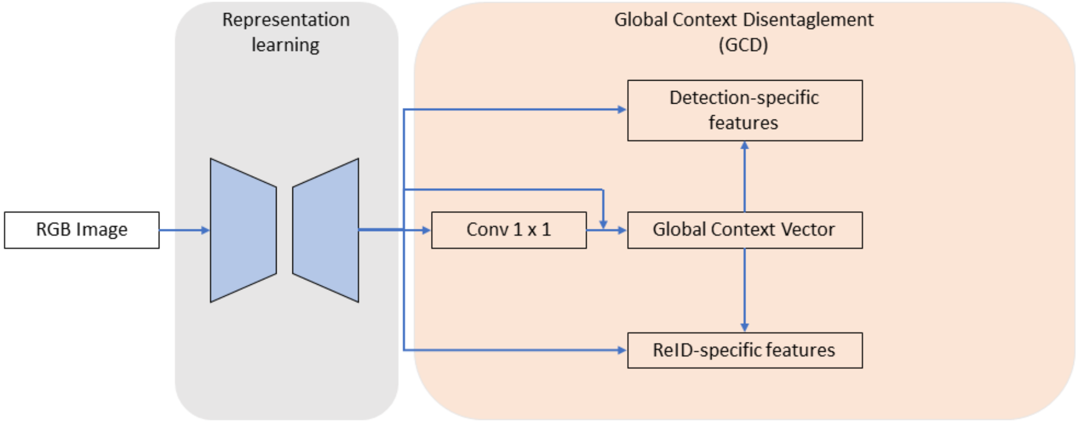


Figure 4.4: Global Context Disentanglement approach by [YLHW69]

extracted leads to an increased ability to extract meaningful spatial representations from the images, thus enabling better detection results. This observation is in line with the results of [ZCZ+19], who saw similar improvements in their studies regarding 2D object detection.

4.4.2 Global Context Disentanglement (GCD)

While multi-task learning has been proven to be beneficial in object detection and tracking tasks, there is still an optimisation contradiction, as shown in [LZL+20]. While in object detection, the network wants to represent objects of the same class similarly, in reID, two object embeddings, even ones of the same class, should be as far apart as possible. To overcome this issue, [YLHW69] proposed a Global Context Disentangling module to decouple feature representations for reID and 2D detection. Here, this approach was applied to 3D detection. The feature decoupling works by first calculating a global context vector as follows:

$$z = \sum_{j=0}^{N_p} \frac{\exp\left(W_k x_k\right)}{\sum_{p_m=0}^{N} \exp\left(W_k x_m\right)} x_j \tag{4.1}$$

where the W_k are learnable weights that stem from a 1×1 convolution layer. $\{x_i\}_{i=1}^{N_p}$ are the input feature maps with $N_p = H \times W$, where H and W are the height and width of the feature maps, respectively. Two transforms then decouple this global vector into two task-specific weight maps, which are finally added to the original feature maps to generate the decoupled feature maps, which are the input to the two different branches:

$$d_i = x_i + W_{d2} ReLU(\Psi_{ln}(W_{d1} z)) \tag{4.2}$$

$$r_i = x_i + W_{r2} ReLU(\Psi_{ln}(W_{r1} z)) \tag{4.3}$$

where the W_x are learnable weight matrices, $ReLU$ is the rectified linear unit activation function, and Ψ_{ln} is the batch normalisation operator. d_i are the feature maps that are used for object detection, and r_i those that are used for the reID branch.

The results of introducing this additional layer are shown in Table 4.7. It shows that while the approach led to improvements in the 2D case in RelationTrack, it yields worse results when applied to the proposed framework. Therefore, it was not utilised in the final model.

4.4.3 Robust Kullback-Leibler Loss

Replacing the Kullback-Leibler loss with the robust version proposed in [CHT+21] increased the HOTA, as illustrated in Table 4.7. This difference indicates that mitigating the increased gradient caused by the decreasing uncertainty during training helps with optimisation.

4.4.4 Transformer Encoder for ReID Feature Extraction

The performance increase due to the additional use of reID features for track association is shown in Table 4.8. This difference indicates that the extracted embeddings provide helpful information for the reID process and that certain situations are handled better via appearance features than simple motion features. However, when only relying on the appearance-based approach, the performance decays and becomes worse than relying solely on the Kalman Filter. Additionally, Table 4.8 also compares the Transformer Encoder based embeddings against the simple convolutional embeddings proposed by FairMOT. These embeddings yielded a better AssA score, but the HOTA suffered slight losses. This result may indicate that, while the CNN approach is competitive regarding the association, it interferes more with the object detection task and thus leads to worse overall performance.

Method	**HOTA**	**DetA**	**AssA**
Proposed Method only reID	29.16	22.40	38.89
Proposed Method vanilla CNN	30.25	21.77	43.50
Proposed Method	**30.96**	**22.89**	**42.81**

Table 4.8: The monocular 3D multi-target tracking results of the proposed method compared the proposed method without the Kalman Filter to exclude impossible tracks and to using vanilla CNNs for reID as in [ZWW+21].

4.5 Evaluation on other Datasets

Evaluation has also been conducted on synthetically generated data using the proposed methodology from section 3.1. The qualitative results are shown in Figure 4.5. Here, one can see that the monkey head is accurately detected and identified across the three frames. The cube is accurately detected in all three images, but it is not reidentified correctly. The cylinder is not detected accurately in the first frame and also receives a new identity for each frame. The quantitative results are shown in Table 4.9. Only the 2D tracking results are shown because the model was not able to make any accurate 3D predictions. This poor performance is probably due to the lack of surroundings that enable recognising a ground plane and utilising other hidden cues. Additionally, the reID part struggles to extract meaningful features that distinguish the individual objects, which may be due to the simplicity of the objects.

Task	**HOTA**	**DetA**	**AssA**
Cube	7.44	52.21	1.08
Cylinder	7.05	49.93	1.03
Monkey head	9.08	50.21	1.67

Table 4.9: The monocular 2D multi-target tracking results of the proposed method on the synthetically generated dataset.

The framework was also qualitatively evaluated in the scope of the FFG Project Bike2CAV, illustrated in Figure 4.6, but due to the lack of annotations, no quantitative results can be shown. As one can see in Figure 4.6, the framework manages to detect and track the cyclist accurately but fails to detect the car. This may be caused by the different camera setups of the training dataset (KITTI) and the evaluation dataset since they vary in position and field of view. In Figure 4.7, one can see an example where the method was not able to reidentify the car after a missed detection.

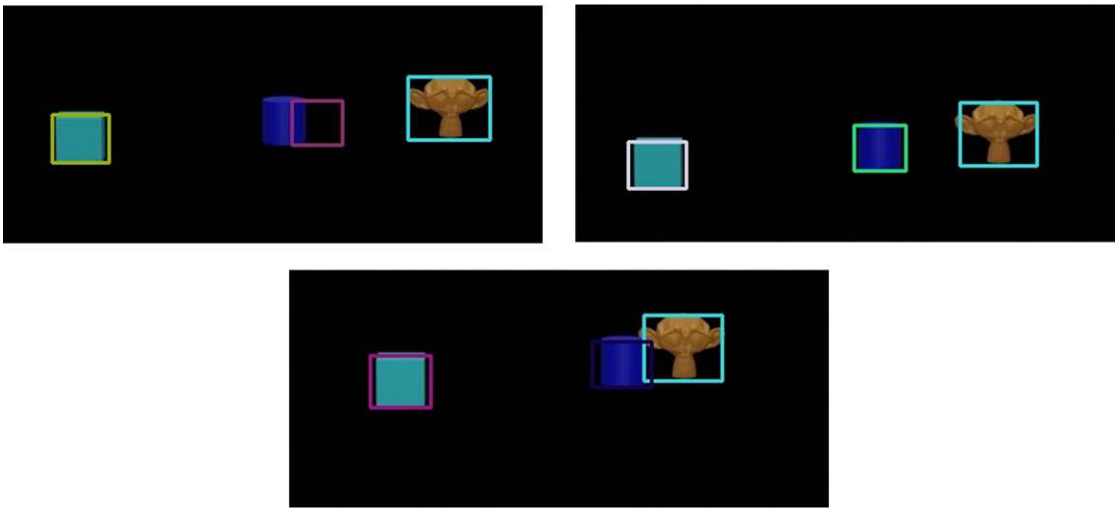


Figure 4.5: Qualitative results on the synthetically generated data-

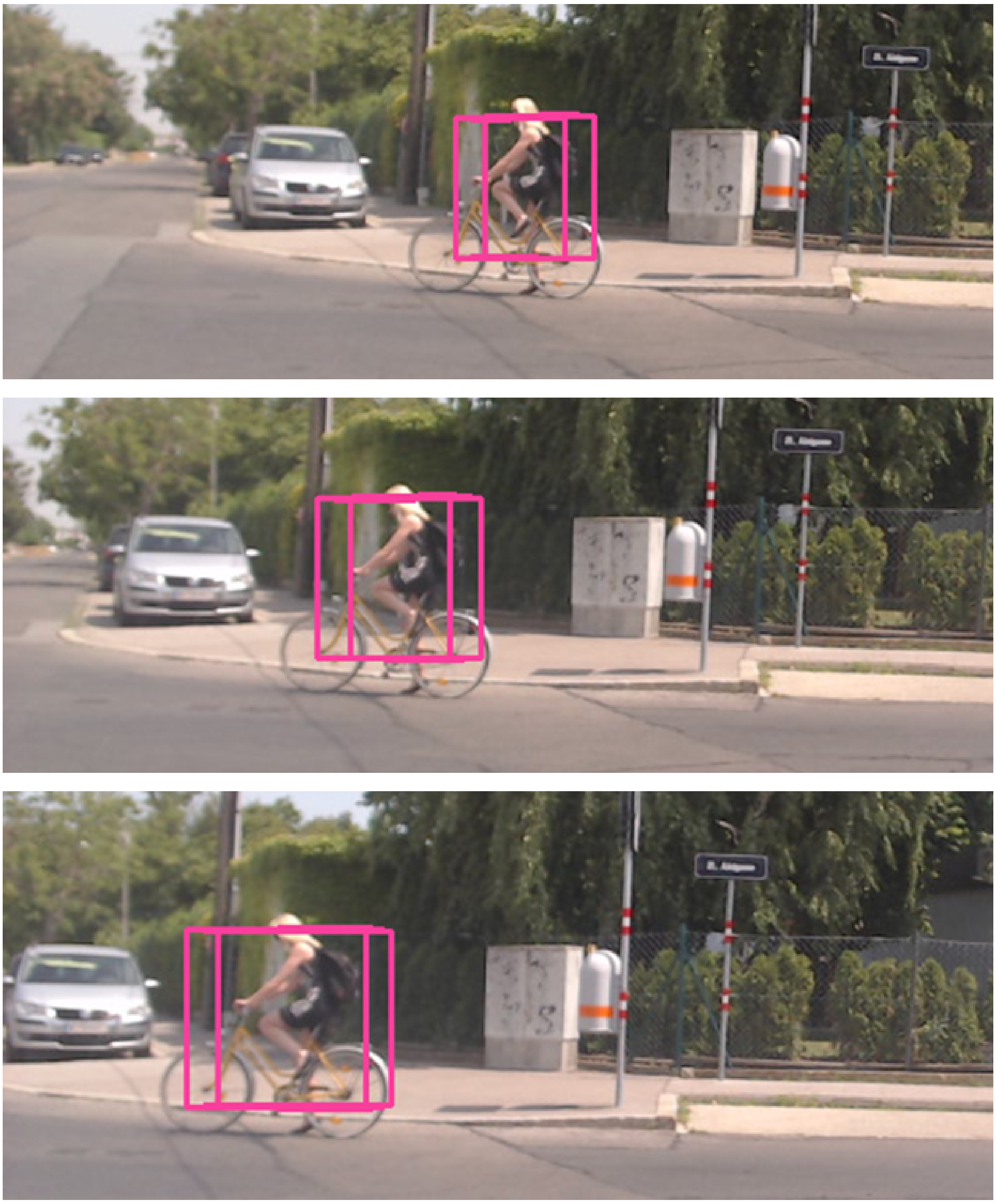


Figure 4.6: Qualitative results on the Bike2CAV dataset where the cyclist is accurately detected and tracked but the car is not detected.

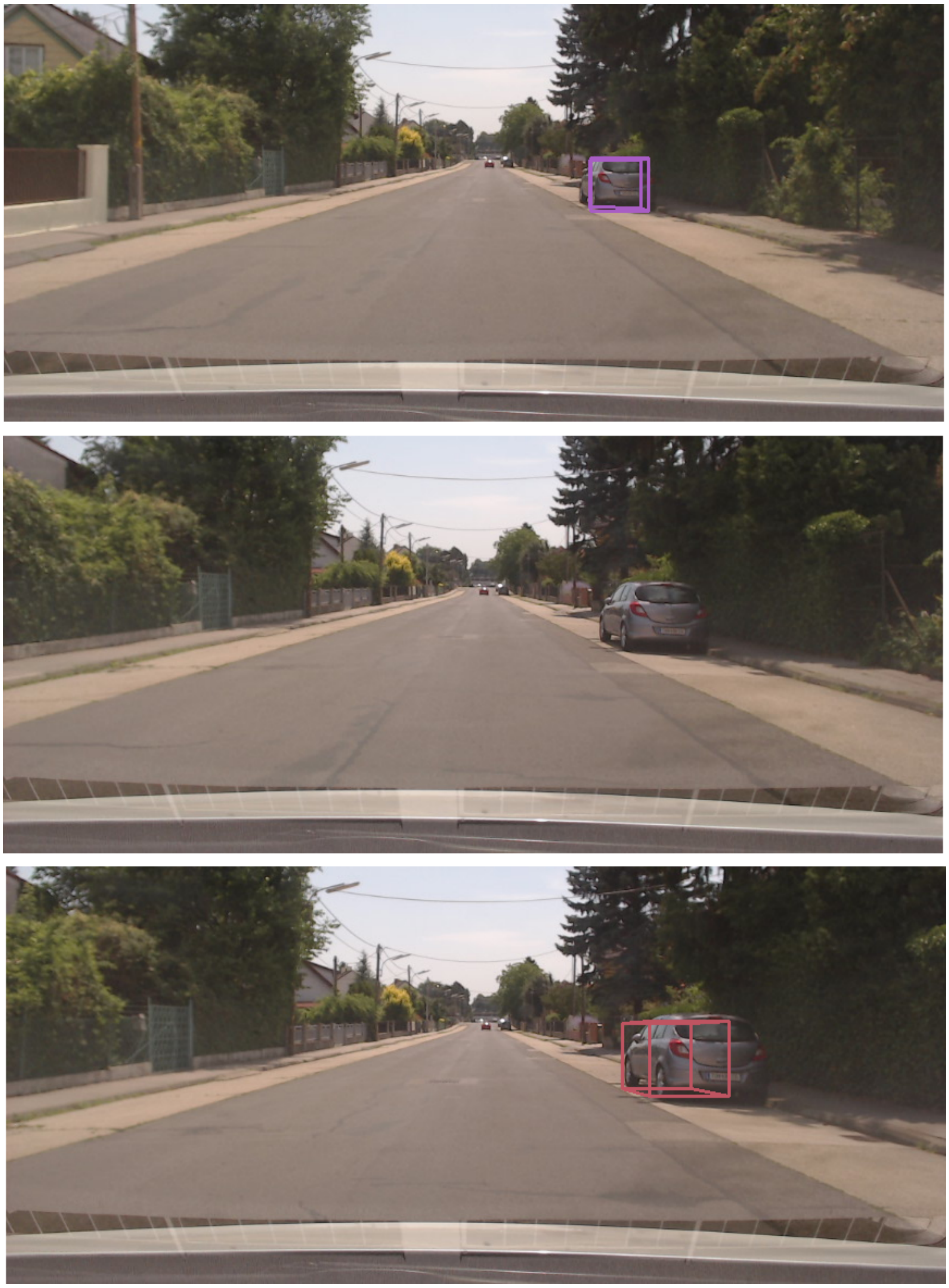


Figure 4.7: Qualitative results on the Bike2CAV dataset where a car is detected in the first frame, missed in the second frame and then detected correctly again but falsely identified in the third frame.

4.6 Evaluation of the Few-Shot Detection Framework

This section presents a quantitative and qualitative evaluation of the few-shot detection framework. First, the dataset and the training procedure are described, followed by qualitative results on a public dataset and a private live demonstration. Finally, quantitative results on the public dataset are shown. The motivation for this framework was to be able to add classes after training with an easy-to-use GUI. This was demonstrated as a proof-of-concept with a 2D framework only because a 3D few-shot detection framework is out of scope for this thesis.

4.6.1 Dataset

The training and evaluation are conducted on the COCO dataset using the official training and validation splits. For the qualitative results of the live demonstrator, the whole training set, including all classes, was used for the base class training. The specialisation then used five images of the novel class acquired and annotated using the presented GUI. Five randomly selected images from the COCO dataset were used as negative examples, ensuring that the novel class is neither contained in the original training data nor the randomly selected images. For the qualitative and quantitative evaluation on the COCO dataset, two classes, namely the "cow" and the "sandwich" class, were excluded during base class training by simply skipping the label. Afterwards, the model was specialised using one, five and ten images for each class plus the same amount of randomly selected negative examples. However, results above 0.0 mAP were achieved only when using ten images in the COCO benchmark.

4.6.2 Training

The training procedure consists of two stages. In the first stage, the network is trained on the large scale COCO dataset for learning the base classes. Here, the training parameters are shown in Table 4.10. The second stage is the few-shot training stage, where the network learns novel categories on one to ten images per class. This training stage uses an initial learning rate of 1×10^{-5} and a cosine annealing decay strategy [LH16] with a

Parameter	Value
Input image size	512×512
Resizing method	Affine transform
Optimiser	AdamW
Learning rate	1×10^{-4}, with decay at 180 epochs by 10
Training epochs	206 (early stopping)
Augmentation	Random horizontal flip, scaling (0.6 - 1.3) and cropping

Table 4.10: The training parameters for the first training stage of the few-shot detection framework.

minimal learning rate of 1×10^{-6}. The model is finetuned for 200 epochs, with the other parameters remaining the same as shown in Table 4.10.

4.6.3 Qualitative Results

The results in Figure 4.8 show mixed results on the COCO validation set. The network manages to detect the newly learned classes in the shown examples partly. The cow is detected in the top left image, and the sandwiches are detected correctly in the bottom right image. In the top right image, one of the cows depicted is detected, while the others are not. Additionally, false positives, where an object is detected both as a base class and a new class object, happen, as shown in the bottom left image. Here the banana is detected and classified as a banana and a sandwich. Figure 4.9 shows the detection results in the developed GUI before and after few-shot training. The clay soldier is not detected in the upper image as that class was not part of the original training data. The clay soldier is detected in the bottom image after training on five images, manually annotated using the included annotation tool.

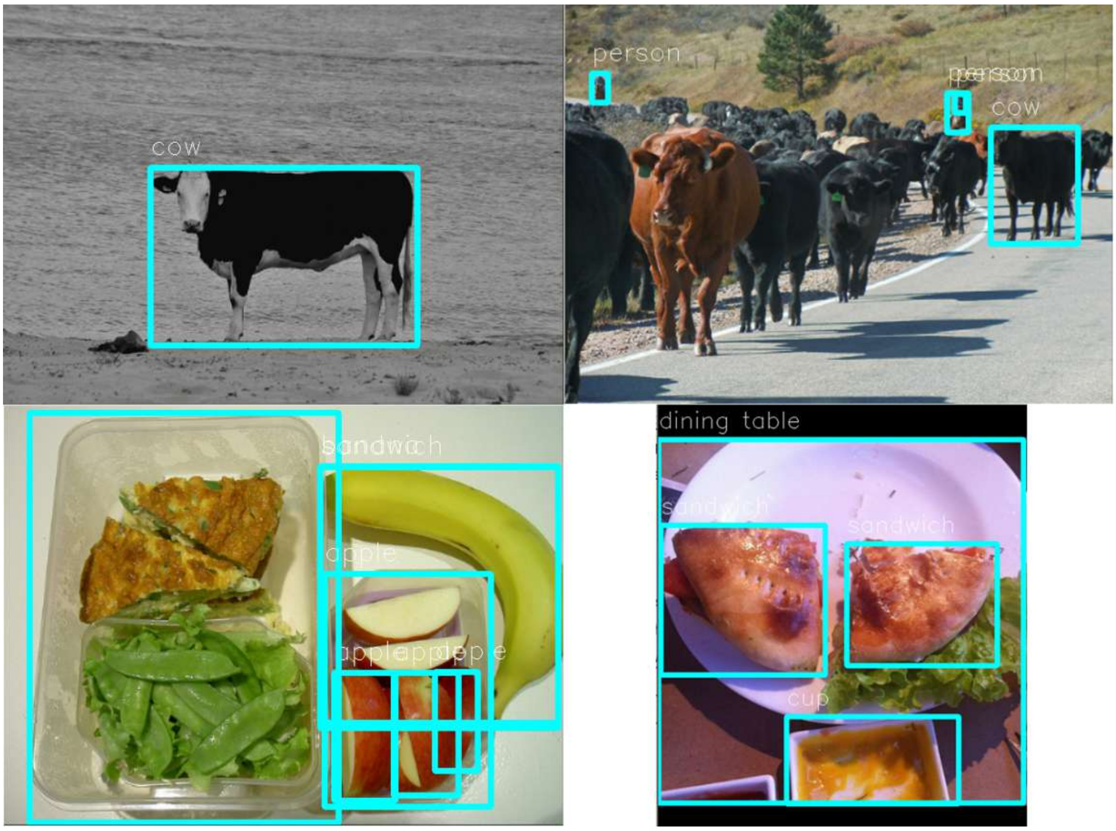


Figure 4.8: Qualitative results of the few-shot detection framework on the COCO validation set.

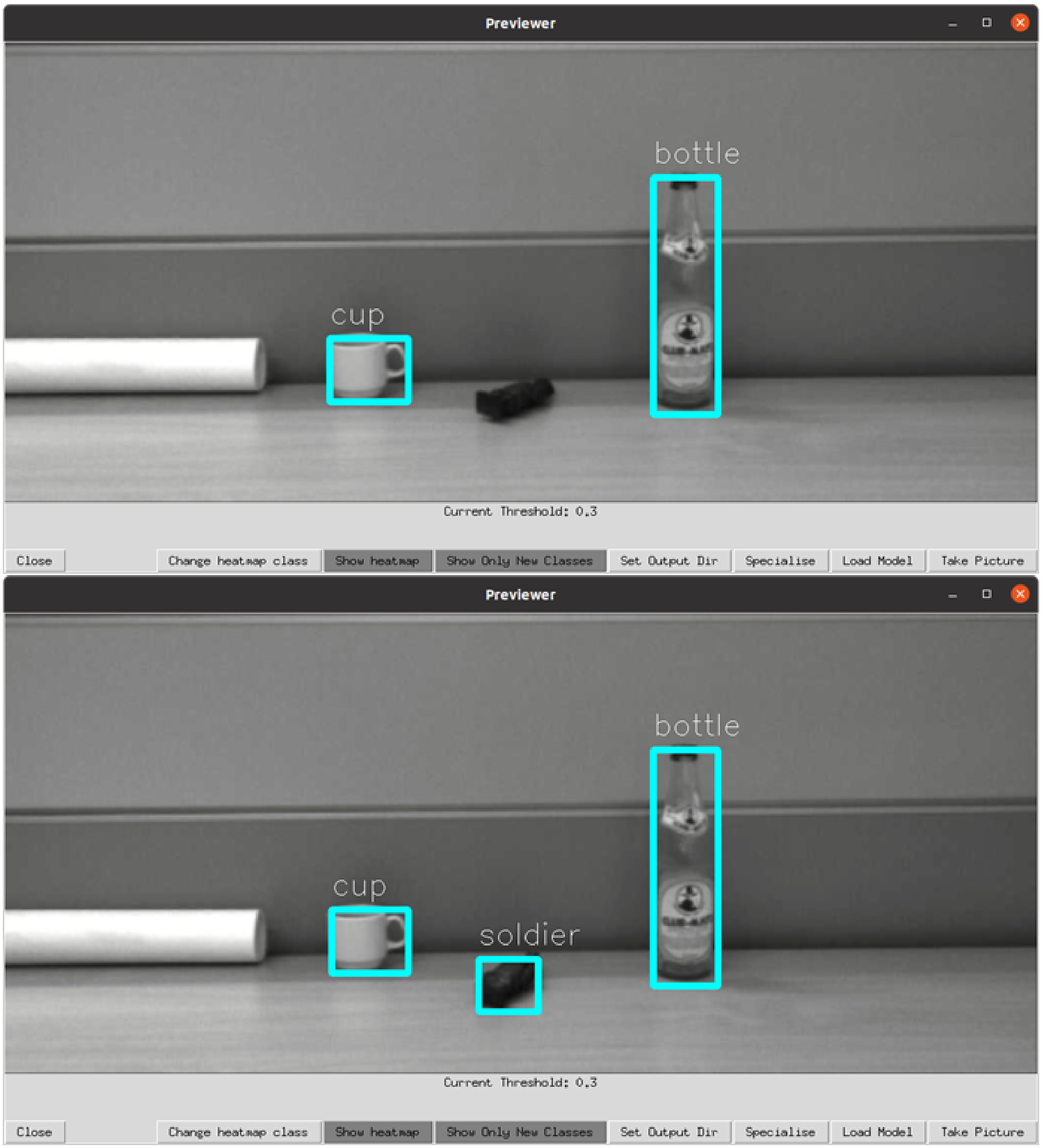


Figure 4.9: Qualitative results of the few-shot detection framework using the developed GUI for live detection and finetuning. The top image shows the results before finetuning, the bottom image after.

4.6.4 Quantitative Results

The quantitative results on the COCO validation set are shown in Table 4.11. As one can see, based on only ten training images per class, the network manages an mAP on the novel classes of about 2%, while the mAP on the base classes stays the same at 35.5%.

While these results are underwhelming compared to state-of-the-art methods such as Meta-DETR or FSOD, they demonstrate the existence of basic functionality.

Method	**mAP**	
	Base	**10-shot**
FSOD	-	22.4
Meta-DETR	-	30.5
Proposed Method	**35.5**	**2.0**

Table 4.11: The few-shot detection results of the proposed framework on the MS COCO val2017 split compared with state-of-the-art methods. Note that the state-of-the-art methods use 60 base and 20 novel classes, while the proposed framework uses 78 base and 2 novel categories.

4.7 Summary

Overall, the proposed reID-based tracking framework performs slightly better than the simple Kalman Filter approach, although training data is limited. However, motion features also need to be considered to achieve the performance by excluding impossible tracks using a Kalman Filter in parallel. The evaluation on the KITTI dataset shows competitiveness with state-of-the-art monocular 3D object detection methods and hints at competitive performance in the 3D tracking benchmark.

The proposed spatial-aware appearance feature extraction scheme using a Transformer Encoder with deformable attention yields improvements compared to the vanilla CNN approach. The same holds for the proposed enhancement using additional attention layers in the backbone and utilising the robust instead of the Laplacian Kullback-Leibler loss.

In the synthesised dataset, the lack of surrounding environments proves to be too challenging for 3D detection, and the appearance variety during training is not enough to achieve satisfying reID results. For 2D detection, the only issue was differentiating between cylinders and cubes, while the monkey head was detected accurately.

The qualitative results on the Bike2CAV data demonstrate the real-world capabilities of the framework.

Finally, the few-shot detection approach was evaluated qualitatively and quantitatively on the COCO dataset, showing worse results compared to state-of-the-art methods but still proving the feasibility of the proposed framework. Additionally, qualitative results on the live demonstrator were shown, showcasing the GUI and practical applicability of the method.

CHAPTER 5

Conclusions and Future Work

This section provides concluding remarks that highlight the main findings and results from the presented experiments. These results are related to the original scientific concepts and endeavours toward enhancing reasoning within a spatial context. Establishing an accurate spatial context is crucial for many applications, as correctly estimated depth (distance to the camera), object dimension, and orientation parameters significantly contribute to performing robust path planning, avoiding collisions or interacting with the environment. Temporal reasoning for an arbitrary number of moving targets also benefits from improved spatial accuracy, as target motion paths become less noisy or spurious. Learning-based target representation enhancements can well complement these spatial and temporal aspects, leading to improvements with better discernible targets moving along more accurately estimated spatial locations. Furthermore, our scenarios are dynamic, where previously unseen object categories might appear. To cope with this phenomenon, a simple learning paradigm was introduced, demonstrating the feasibility of adding newly learned categories to a previously existing pool of category models. This concept was implemented as a real-time demonstrator with the ability for the user to add new categories via a graphical interface.

The introduced learning concepts, which target a reduction of spatial and tracking ambiguities, seem to accomplish these task-specific objectives across all experiments for the targeted critical tasks. These improvements are reflected in the HOTA scores and its components, DetA and AssA. From an algorithmic point of view, these improvements are triggered by the following representational enhancements: The attention-enriched backbone and the robust KL loss contribute to an increase in object detection performance of more than one per cent. Using appearance features to associate detections to tracks is especially helpful in scenarios where a sudden movement change occurs, as demonstrated in Figure 4.3. These sudden changes often occur in safety-critical situations, e.g. when an AV turns into a street, where a correct association is crucial for a correct assessment of the situation. Based on the experimental results, the advantages and drawbacks of the presented representational enhancements are analysed in terms of a machine learning workflow and practical use. Since the application-driven need for monocular depth-aware detection and tracking methodologies is increasing, an outlook to elaborate on possible methodology extensions and future developments is also presented.

5.1 Conclusions

The topic of this thesis focuses on challenging learning and regression tasks. Their challenging nature stems from the fact that the presented image-based recognition involves estimating 3D spatial parameters, which are associated with ambiguities as the projection of the 3D world onto a 2D image plane involves loss of information. Motivated by the enormous representational capacity of deep neural networks, the thesis investigates how to formulate learning and regression tasks such that different spatial estimates jointly enforce valid proposals. Moreover, enlarging the spatial range of learned representations within the image is a second research target. The perspective view (as represented by converging parallel lines, textures, shading, variable blur and haze) implicitly hints at scene depth and object distance. However, conventional CNNs capture only a limited range of spatial correlations, leading to a localised analysis that limits the discovery of cues establishing a broader spatial context.

This thesis shows that formulating an end-to-end learning scheme for joint monocular 3D object detection and tracking using appearance-based reID features is possible, outperforming simple motion-based tracking. Experiments also demonstrate that using a Transformer Encoder for spatially-aware appearance feature extraction is superior to a simple convolutional embedding. As increasing the number of parameters to be estimated directly calls for the need for larger amounts of training data, integrating many tasks into an end-to-end optimised monocular 3D pipeline is not straightforward. However, using a common backbone representation, mutually supporting parametric representations to be regressed and re-used representations across different tasks render the overall learning task tractable. Nevertheless, it is evident from the experiments that the training dataset of limited size does not permit the full exploitation of the devised representational enhancements. Accordingly, it is shown that simply adding additional attention layers (described in Section 3.2.1) does not directly lead to significantly improved performance, and further data and investigation are necessary. Also, the Global Context Disentanglement approach (described in Section 4.4.2) does not directly translate from the 2D to the 3D case.

In light of the original research motivation and postulated applied task, the obtained results accomplish measurable detection and tracking improvements within the metric 3D space compared to the current state-of-the-art. LiDAR-based systems still represent the spatially most accurate detection/tracking schemes in this domain. Given the improved spatial accuracy and tracking obtained, this accuracy gap to LiDAR has become smaller.

The proposed monocular 3D learning and inference framework demonstrates the potential for practical applications. It represents a real-time analysis solution for capturing the spatial and temporal attributes of an unknown number of targets around a moving vehicle. Due to these capabilities, the developed analysis pipeline has been deployed as a runtime optimised environment perception module in the Bike2CAV project. Current large-scale testing is ongoing, targeting the enhanced safety of cyclists in city environments.

5.2 Future Work

The proposed methodology simultaneously estimates multiple unknowns, such as object 3D pose, dimensions and motion trajectory. Due to the multitude of postulated learning tasks, such a learning step requires a large dataset labelled in terms of multiple categories, distinct trajectories and within a 3D metric space. Discovering complex non-linearities and correlations between the 2D-3D domains while balancing individual learning tasks' accuracy depends on the quality and quantity of training data. A significant drawback was the limited availability of multi-attribute (category, track, 3D) annotated datasets, which hindered reaching the maximum representational potential of proposed algorithmic concepts (attention, robust loss, reID).

Therefore, experiments on more diverse and challenging datasets (e.g., nuScenes) could be conducted to further elaborate on the effect of the enhanced spatial reasoning capabilities of the framework. Another strategy to further enhance the amount and diversity of training data could be to use a self-supervised enrichment scheme that augments the training data by moving objects within an image while maintaining the background.

Furthermore, a sensor fusion approach with LiDAR data could be explored to combine the strong objectness and appearance information contained in images with the 3D accuracy of LiDAR sensors.

List of Figures

List of Tables

Bibliography

[AAJD+19] Eduardo Arnold, Omar Y. Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[BABM19] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019.

[BGO+16] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference in Image Processing (ICIP)*, pages 3464–3468, 2016.

[BS08] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[CAZS18] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2018.

[CCR+17] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2040–2049, 2017.

[CHT+21] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10379–10388, 2021.

[CKZ+16] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2147–2156, 2016.

[CMS+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

[CR81] H. Martyn Cundy and A. P. Rollett. *Mathematical models.* Tarquin Publications, Norfolk, 3rd ed. edition, 1981.

[CZBO21] Mohamed Chaabane, Peter Zhang, J. Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[EZW+05] Mark Everingham, Andrew Zisserman, Christopher K. I. Williams, Luc van Gool, Moray Allan, Christopher M. Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges Workshop*, pages 117–176, 2005.

[FZH+21] Zhaoxin Fan, Yazhi Zhu, Yulin He, Qi Sun, Hongyan Liu, and Jun He. Deep learning on monocular object pose detection and tracking: A comprehensive overview. *arXiv preprint arXiv:2105.14291*, 2021.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[GMB17] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[HHW$^+$22] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. Mat: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022.

[HS19] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019.

[HVA$^+$19] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[JZK19] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. *arXiv preprint arXiv:1906.08070*, 2019.

[K$^+$60] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.

[KH21] Seong-heum Kim and Youngbae Hwang. A survey on deep learning based methods and datasets for monocular 3d object detection. *Electronics*, 10(4):517, 2021.

[KHG$^+$19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019.

[KK19] Youngseok Kim and Dongsuk Kum. Deep learning based vehicle position and orientation estimation via inverse perspective mapping image. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 317–323, 2019.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[Kuh55] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[LCS19] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7636–7644, 2019.

[LD18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[LGG+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[LH16] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[LJD19] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[LKSC19] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly supervised object detection with segmentation collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9735–9744, 2019.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[LOD+21] Jonathon Luiten, Aljos A. Os Ep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, 2021.

[LSML14] Martin Lochner, Charlotte Sennersten, Ahsan Morshed, and Craig Lindley. Modelling spatial understanding: Using knowledge representation to enable spatial awareness in a robotics platform. In *The 6th International Conference on Advanced Cognitive Technologies and Applications*, pages 689 – 699, 2014.

[LVC+19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019.

[LWT20] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4289–4298, 2020.

[LYL21] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2):919–926, 2021.

[LZL+20] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv preprint arXiv:2010.12138*, 2020.

[MAR+19] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6840–6849, 2019.

[MBU+19] Matjaž Mihelj, Tadej Bajd, Aleš Ude, Jadran Lenarčič, Aleš Stanovnik, Marko Munih, Jure Rejc, and Sebastjan Šlajpah. Robot vision. In *Robotics*, pages 107–122. Springer, 2019.

[MM15] Franz Josef Mehr and María Teresa Mehr. Parametrische Kurven und Oberflächen. In *Excel und VBA*, pages 195–213. Springer, 2015.

[MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[NHH15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1520–1528, 2015.

[PND20] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. Da Silva. A survey on performance metrics for object-detection algorithms. In *International*

Conference on Systems, Signals and Image Processing (IWSSIP), pages 237–242, 2020.

[QLL21] Rui Qian, Xin Lai, and Xirong Li. 3d object detection for autonomous driving: a survey. *arXiv preprint arXiv:2106.10823*, 2021.

[QWL63] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A general framework for monocular 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 doi: 10.1109/TPAMI.2021.3074363.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[RF18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[RHGS17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[RKC18] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[RTG$^+$19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.

[Rud17] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[SAS17] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 300–311, 2017.

[SJS19] Siddharth Srivastava, Frederic Jurie, and Gaurav Sharma. Learning 2d to 3d lifting for object detection in 3d for autonomous vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4504–4511, 2019.

[SWD+20] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Fgagt: Flow-guided adaptive graph tracking. *arXiv preprint arXiv:2010.09015*, 2020.

[Sze22] Richard Szeliski. *Computer Vision*. Springer International Publishing, Cham, 2022.

[VKO+19] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7942–7951, 2019.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[WK19] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 857–866, 2019.

[WWHK20] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063*, 2020.

[WZL+20] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision (ECCV)*, pages 107–122, 2020.

[XC18] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2345–2353, 2018.

[XCZH19] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3988–3998, 2019.

[XTY+20] Youzi Xiao, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79(33):23729–23791, 2020.

[YLHW69] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *IEEE Transactions on Multimedia*, 2022 doi: 10.1109/TMM.2022.3150169.

[YWSD18] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2018.

[ZCW$^+$21] Gongjie Zhang, Kaiwen Cui, Rongliang Wu, Shijian Lu, and Yonghong Tian. Pnpdet: Efficient few-shot detection without forgetting via plug-and-play sub-networks. In *EEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3822–3831, 2021.

[ZCZ$^+$19] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6688–6697, 2019.

[ZHLD19] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9308–9316, 2019.

[ZKK20] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision (ECCV)*, pages 474–490, 2020.

[ZLL$^+$22] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[ZLZ21] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3298, 2021.

[ZMY$^+$21] Yinmin Zhang, Xinzhu Ma, Shuai Yi, Jun Hou, Zhihui Wang, Wanli Ouyang, and Dan Xu. Learning geometry-guided depth via projective modeling for monocular 3d object detection. *arXiv preprint arXiv:2107.13931*, 2021.

[ZSL$^+$20] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[ZWK19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[ZWT$^+$21] Zou Wenbin, Wu Di, Tian Shishun, Xiang Canqun, Li Xia, and Zhang Lu. End-to-end 6dof pose estimation from monocular rgb images. *IEEE Transactions on Consumer Electronics*, 67(1):87–96, 2021.

[ZWW^{+}21] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.