KIDELTA LEARNING
Scalable AI for Automated Driving

Deliverable 13

Synthetische Daten zur Erweiterung des Datensatzes für weiterführende Untersuchungen (Corner Cases, etc.) sind vorhanden

Version	1.0
Editor	Dr.-Ing. Amin Hosseini und Marius Bachhofer
Project coordination	Mercedes-Benz AG und ZF Friedrichshafen AG

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

Contents

List of Figures

List of Tables

1 Introduction

The document at hand provides information about the developements of the "Synthetische Daten"-Stream of KI Delta Learning (see chapter 3). It contains one result of TP1 - "Datengewinnung".

The chapters 1 to 4 of this document give a general introduction into the project and its deliverables. They show how the 18 deliverables contribute to the project aim and how they complement to deliver the project outcome. Chapter 5 gives the detailed content which is produced in TP1:

Results TP1

- E3.4.1.3 - Synthetic data to extend the data set for further investigation

1.1 Project Description

The goal of the KI Delta Learning project is the development of methods and tools for the the efficient extension and transformation of existing AI modules of autonomous vehicles to the challenges of new domains or more complex scenarios. AI modules are the core of the cognitive intelligence of automated vehicles and thus a key technology for ever higher levels of automation of assistance systems up fully to autonomous driving. Therefore, AI modules are of central importance to the future value creation of the German automotive industry. The market launch strategy of the German automotive industry for these assistance systems is proceeding step by step toward ever higher levels of automation and larger areas of application for automation. The focus of the project is the gradual expansion of the domains of application of assistance systems and the associated AI modules, which will be developed in parallel in six directions according to the most relevant use cases.

Thus, within the project, various deltas - the gaps between known domains of applications and new domain areas - are considered including deltas due to improved sensor technology, due to different traffic areas such as highways or construction sites, due to changes in country and corresponding traffic rules and signage, due to new forms of traffic and road users such as e-scooters, due to changing environmental conditions such as day, night, sun or rain, as well as deltas due to the advancement of neural network designs. A stepwise, structured extension of AI modules towards the six mentioned deltas is called "Delta Learning". This will not necessarily involve all six extensions simultaneously. Rather, the automation of assistance systems will gradually increase through efficient Delta Learning. KI Delta Learning aims to incrementally extend AI modules that have already been trained for limited areas and locations of use without completely re-executing the otherwise usual training and optimization process at a very high cost. So far, no sufficiently efficient and stable methods and tools exist for such Delta Learning.

Hence, the focus of the project is on method development. In two orthogonally operating but interlocked subprojects (TP2 and TP3), these methods are developed on the one hand starting from overcoming the deltas under the focus of transfer learning and on the other hand from the didactic approach. Furthermore, questions of the automotive suitability of the developed methods are investigated (TP4) as well as necessary data generated, recorded and processed (TP1).

The fundamental extension of current generations of AI algorithms expected to be achieved within the project enables a decisive leap towards the large-scale realization of autonomous vehicles. Thus, KI Delta Learning represents an important innovation building block for the competitiveness of the German automotive and supplier industry in an increasingly competitive economy.

1.2 General Deliverables Overview

The project will work on the set goals over a period of three years. It pursues the step-by-step improvement of the models and methods to be developed in four successive project steps, the project increments (PI). At the end of each project increment, which goes hand in hand with the defined project milestones, the (interim) results achieved in the work packages are documented in the form of deliverables. A total of 18 deliverables were defined in the VHB, which serve to communicate the results both internally and externally to the funding agency (BMWi, project sponsor). These 18 deliverables were distributed among the four project increments and can be grouped into following topics.

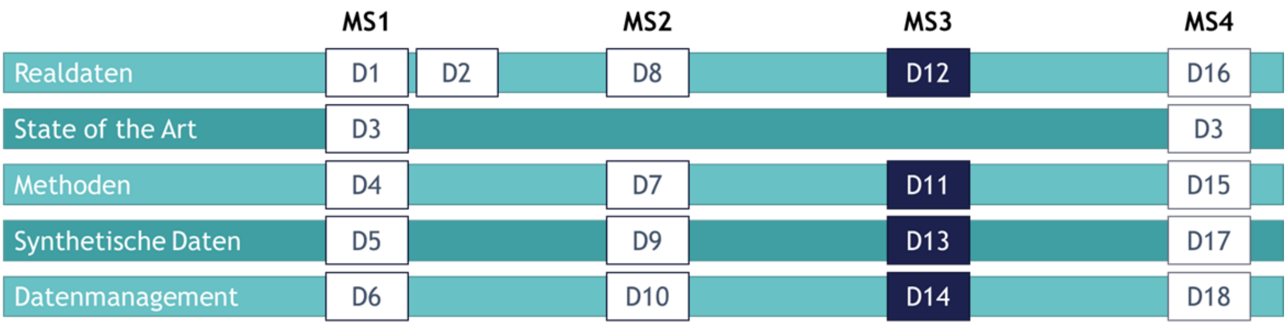


Figure 1.1: Timeline of the Deliverables and grouping to Milestones

The third project increment started and ended with a delay of three months compared the the initial planning as given in the VHB: The end of the third project incerment was shifted from month 27 (03/2022) to month 30 (06/2022). The reason are still the effects of the CoVid-19 pandemic. The results of third project increment are provided in the follwing deliverables, they represent Milestone 3:

1. D11 Eine Defizitanalyse sowie ein Optimierungsplan für die ausgewählten Delta Learning-Methoden mit vorhandenem Datensatz sind vorhanden

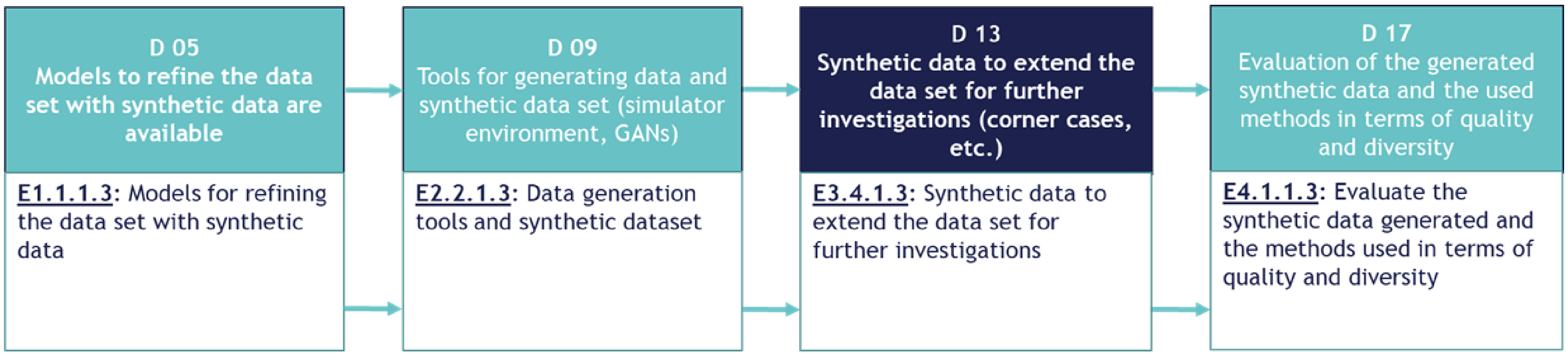
2. D12 Datensatz für die Realdaten aus PI3 mit definierten Annotierungen für unterschiedliche Domäne ist vorhanden & für Partner zugreifbar

3. D13 Synthetische Daten zur Erweiterung des Datensatzes für weiterführende Untersuchungen (Corner Cases, etc.) sind vorhanden

4. D14 Ein optimiertes Produktivsystem auf Basis eines kontinuierlichen Performance-Monitorings und Auswertens der Benutzerakzeptanz ist vorhanden

1.3 Deliverable Context

Deliverable D13 "Synthetische Daten zur Erweiterung des Datensatzes für weiterführende Untersuchungen (Corner Cases, etc.) sind vorhanden" is based on the content of D05 and D09. D13 reports the extension of the synthetic data set that is produced and used in the project.


D 05
Models to refine the data set with synthetic data are available
E1.1.1.3: Models for refining the data set with synthetic data
D 09
Tools for generating data and synthetic data set (simulator environment, GANs)
E2.2.1.3: Data generation tools and synthetic dataset
D 13
Synthetic data to extend the data set for further investigations (corner cases, etc.)
E3.4.1.3: Synthetic data to extend the data set for further investigations
D 17
Evaluation of the generated synthetic data and the used methods in terms of quality and diversity
E4.1.1.3: Evaluate the synthetic data generated and the methods used in terms of quality and diversity

2 Results TP1

2.1 E3.4.1.3: Synthetic data to extend the data set for further investigation

2.1.1 Synthetic data generation in CARLA (DLR)

Introduction

Synthetic data samples are a promising approach to reduce the effort and costs that are associated with the elicitation of annotated data samples. Collecting data in the real world requires (1) the equipment of a vehicle with partially expensive sensors, (2) a difficult sensor calibration, (3) driving around and collecting data as well as (4) manually annotating the acquired data by humans. Simulations obviate the need for buying and assembling the test vehicle as well as manually annotating the data, since the annotations are directly generated.

In the context of generating data for Delta Learning methods, the CARLA simulator is a suitable choice [2]. It allows the variation of lighting and weather conditions, supports different maps and sensor models can be parameterized to an extent. These features enable the creation of a quite diverse synthetic datasets.

In this work, we describe the implemented processes to acquire synthetic data in CARLA together with high-quality annotations.

Methods

The synthetic generation of a dataset for the perception in automated driving requires multiple steps: (1) A suitable sensor suite has to be defined/implemented, which is suitable for the intended use case. (2) Besides of the raw sensor data, corresponding annotations/labels have to be extracted. (3) Metadata of the simulated scenarios can be logged, in order to later search through the database based on specific attributes. (4) Optionally, only key-frames can be stored, if time-sequences are not of interest for the use case.

In the following chapters, the different steps are described in more detail, especially with respect to their implementation in CARLA.

Definition of the sensor suite Often, we are interested in the generation of synthetic data samples, which shall help in improving the performance of a ML model in a real-world task, e.g. object detection in images. To do so, it is the best to reduce the gap between real and virtual samples as good as possible. One aspect here is the used sensor suite in real and virtual world, which should be aligned as good as possible.

If the real vehicle is already equipped with sensors, the extrinsic and intrinsic calibrations of the sensor can be replicated in the simulation. We performed this replication step on the basis of the sensor suite of the *nuScenes* dataset [3] and describe this procedure in the following.

General information about the sensor suite of nuScenes can be retrieved from the publication [3] and the webpage [4]. The recording campaign was done using a Renault ZOE, which was equipped with six cameras, one spinning LIDAR, five long range RADAR sensors and one IMU & GPS. Concrete extrinsic calibrations of the sensors can be extracted from the downloaded dataset itself, where intrinsic and extrinsic calibrations are provided for each recorded scenario. The extrinsic calibration of the sensors is given relative to the midpoint of the rear axle of the Renault ZOE.

With this knowledge available, we can now replicate this sensor setup in CARLA. The current version of CARLA (0.9.13) does not contain a car model of the Renault ZOE, therefore we searched for the most similar car that is available by comparing the dimensions of the Renault ZOE with the dimensions of all available cars. We found the Citroën C3 to be the most similar and selected it as the recording car for the simulations. An alternative to this solution would be to add a new car model for the Renault ZOE to CARLA, but this requires an appropriate 3D model and quite some effort. Therefore, we decided in favor for the similar vehicle.

Now, the simulated car has to be equipped with the sensors as in the nuScenes dataset. We extracted the extrinsic calibration from the dataset and had to transform them, so they are usable in CARLA:

1. CARLA uses a left-hand coordinate system, while nuScenes uses a right-hand coordinate system,
2. CARLA specifies rotations as (yaw, pitch, roll) tuples, while nuScenes specifies them as quaternions,
3. CARLA uses the midpoint of the vehicle as relative center, while nuScenes uses the midpoint of the rear axle as relative center.

While the first two points can easily be addressed by simple spatial transformations, addressing the third point requires additional knowledge about the Renault ZOE. In order to determine the offset between the midpoint of the vehicle to the midpoint of the rear-axle, we used the description of the vehicle' s dimensions from [5]. Having the positions and orientations determined, the sensors can be added to the Citroën C3 in CARLA, while using the intrinsic parameters that were also extracted from the nuScenes dataset.

Acquisition of annotations In order to get annotations for a raw camera sensor in CARLA, one has to spawn the corresponding sensors with the exact same transformation. Currently, we use the the Depth sensor and also the Instance Segmentation sensor, which is available in CARLA since version 0.9.13.

We do not use the original Semantic Segmentation sensor from CARLA, because the resulting segmentation is not detailed enough for some classes. An example is that all vehicles are tagged as class "vehicle" instead of more fine-granular classes like "car", "truck", "bus", etc. Since these fine-granular classes are default in almost any real dataset, we derive these by exploiting the instance segmentation. In order to do so, it is currently necessary to compile CARLA from source with some modifications, which are provided in [6]. With this modified CARLA version, it is possible to retrieve IDs in the instance segmentation, that correspond to the actor IDs used by CARLA. With these connections, we can now inspect the actor, e.g. by looking up the concrete vehicle model. Based on this information, a more fine-granular class mapping can be created and a corresponding semantic segmentation can be calculated.

Besides of that, we also extract 3D Bounding Boxes for the objects visible in an image. By default, CARLA does not provide a simple way to decide for visible bounding boxes, but again, by exploiting the instance segmentation, we can filter all bounding boxes of all actors by those IDs, that are visible in the instance segmentation. This again requires the above-mentioned modification of CARLA.

Logging of metadata Besides of the sensor annotations, metadata might also be interesting to be logged for a simulated scenario. By doing so, it will be easily possible to filter a dataset for specific attributes, e.g. weather conditions. Besides of that, one might also be interested in extrinsic or intrinsic calibrations in use, when taking an image.

To account for these different use cases, we currently store the following metadata in a JSON file:

- Weather conditions (e.g. precipitation or cloudiness)
- Location (i.e. the used CARLA map)
- Intrinsic calibrations
- Sensor positions and orientations over the course of a scenario

Discussion & Conclusion

We showed, how based on a real sensor suite, a virtual replication can be derived that will then be used to simulate synthetic data. We also described, how the newly available instance segmentation can be exploited to derive further information and how metadata can be extracted. Together, this builds the foundation to generate a large-scale synthetic dataset, which can then be used for training and testing of ML perception models.

Literature

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, CARLA: An Open Urban Driving Simulator. arXiv, 2017. doi: 10.48550/ARXIV.1711.03938.
Referenced at: *1*

[3] H. Caesar et al., nuScenes: A multimodal dataset for autonomous driving. arXiv, 2019. doi: 10.48550/ARXIV.1903.11027.
Referenced at: *1 2*

[4] Motional, “nuScenes.” https://www.nuscenes.org/nuscenes (accessed Jun. 08, 2022).
Referenced at: *1*

[5] Renault, “Will Renault ZOE fit in your garage? Length, height, width, etc.” https://dubai.renault.ae/cars/zoe/dimensions.html (accessed Jun. 08, 2022). Referenced at: *1*

[6] suniique, “Add support for retrieving CarlaActor’s ID from instance segmentation camera.” https://github.com/carla-simulator/carla/pull/5452 (accessed Jun. 08, 2022). Referenced at: *1*