VALERIE22 - A photorealistic, richly metadata annotated dataset of urban environments

Oliver Grau
Korbinian Hagn
oliver.grau@intel.com
korbinian.hagn@intel.com
Intel Deutschland GmbH
Neubiberg, Bayern, Germany

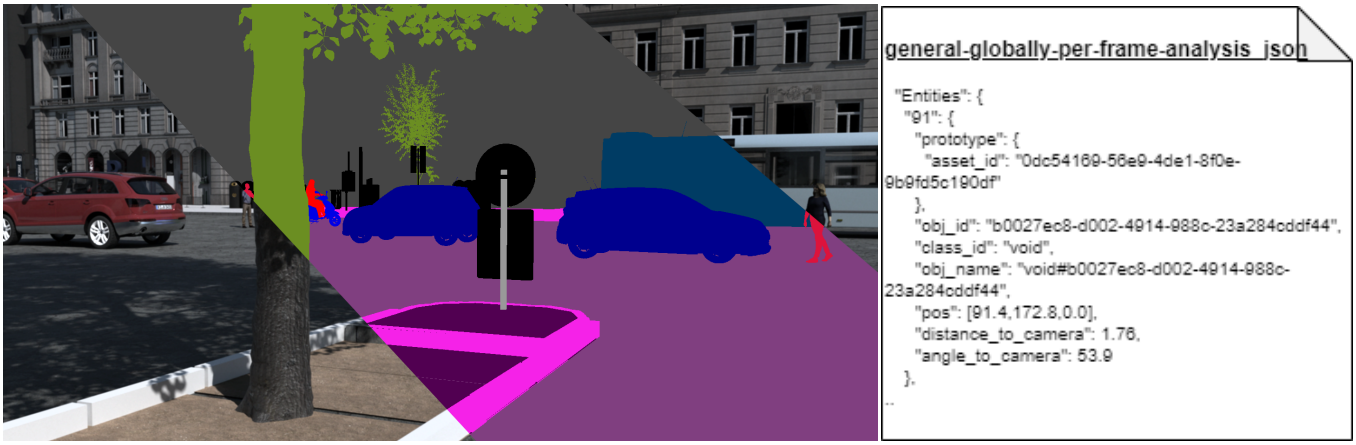

Figure 1: VALERIE22 - A photorealistic, richly metadata annotated dataset of urban environments

ABSTRACT

The VALERIE tool pipeline is a synthetic data generator [14] developed with the goal to contribute to the understanding of domain-specific factors that influence perception performance of DNNs (deep neural networks). This work was carried out under the German research project *KI Absicherung* in order to develop a methodology for the validation of DNNs in the context of pedestrian detection in urban environments for automated driving.

The *VALERIE22* dataset was generated with the VALERIE procedural tools pipeline providing a photorealistic sensor simulation rendered from automatically synthesized scenes. The dataset provides a uniquely rich set of metadata, allowing extraction of specific scene and semantic features (like pixel-accurate occlusion rates, positions in the scene and distance + angle to the camera). This enables a multitude of possible tests on the data and we hope to stimulate research on understanding performance of DNNs.

Based on cross-domain semantic segmentation experiments, i.e. training on synthetic data and evaluation on target real world data, a comparison with several other publicly available datasets is provided, demonstrating that *VALERIE22* is one of best performing synthetic datasets currently available in the open domain. [1]

CSCS '23, December 05, 2023, Darmstadt, Germany

ACM ISBN 979-8-4007-0454-3/23/12.
https://doi.org/10.1145/3631204.3631866

CCS CONCEPTS

• **Computing methodologies** → **Image and video acquisition**; *Object detection*; **Image segmentation**.

KEYWORDS

Synthetic Data, AI Validation, Autonomous Driving, Pedestrian Detection, Object Detection, 2D-Bounding Box Detection

ACM Reference Format:
Oliver Grau and Korbinian Hagn. 2023. VALERIE22 - A photorealistic, richly metadata annotated dataset of urban environments. In *Computer Science in Cars Symposium (CSCS '23), December 05, 2023, Darmstadt, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3631204.3631866

1 INTRODUCTION

Recently, great progress has been made in applying machine learning techniques to deep neural networks to solve perceptional problems. Automated vehicles (AV) are a recent focus as an important application of perception from cameras and other sensors, such as LIDAR and Radar [36]. Although the current main effort is on developing the hardware and software to implement the functionality of AVs, it will be equally important to demonstrate that this technology is safe.

[1] Available here: https://huggingface.co/datasets/Intel/VALERIE22

The German collaborative research project *KI Absicherung* [1] was a cross industry and academia effort to develop a methodology for the validation of DNNs in the context of pedestrian detection in urban environments for automated driving. Specifically, one important goal of that project was to make the safety aspects of ML-based perception functions predicable. As one important research stream of this project synthetic data generation was used as a base, as this allows full control over domain-specific scene parameters and the ability to generate parameter variations of these. Further, additional metadata annotations were specified and automated computation of these were added to the synthesis pipeline.

The VALERIE tools pipeline was developed as a research tool to improve quality of data synthesis and to get an understanding of factors that determine the domain gap between synthetic and real datasets. For that a powerful synthesis pipeline has been developed, which allows the fully automated creation of complex urban scenes. In this paper we only summarize some of the functionalities of the VALERIE synthesis pipeline and focus on a description of the (meta-)data formats of the *VALERIE22* dataset that was generated with the tool chain. More details on the synthesis tools can be found in [14].

Additionally, we present evaluation results to assess the quality of our synthetic data compared to other synthetic datasets in the autonomous driving domain.

1.1 Related work

In [14] we suggest a computational data synthesis approach for deep validation of perception functions based on parameterized synthetic data generation. We introduce a multi-stage strategy to sample the input domain and to reduce the required vast amount of computational effort. This concept is an extension and generalization of our previous work on parameterization of the scene parameters of concrete scenarios. We extended this parameterization by a probabilistic scene generator to widen the coverage of scenario spaces and a more realistic sensor simulation. These approaches were used to generate the scenes and data in the *VALERIE22* dataset.

Techniques to capture and render models of the real world have been matured significantly over the last decades. Computer generated imagery (CGI) is increasingly popular for training and validation of deep neural networks (DNNs) as synthetic data can avoid privacy issues found with recordings of members of the public and can automatically produce ground truth data at higher quality and reliability than costly manually labeled data. Moreover, simulations allow synthesis of rare scene constellations helping validation of products targeting safety critical applications, specifically automated driving. Because of the progress in visual and multi-sensor synthesis, now building systems for validation of these complex systems in the data center becomes not only feasible but also offers more possibilities for the integration of intelligent techniques in the engineering process of complex applications.

The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g. [3]). Several methodologies for verification and validation of AVs have been developed [8, 21, 22] and commercial options exist.[2] These tools were originally designed for virtual testing of automotive functions, like braking systems and then extended to provide simulation and management tools for virtual test drives in virtual environments. They provide real-time capable models for vehicles, roads, drivers, and traffic which are then being used to generate test (sensor) data as well as APIs for users to integrate the virtual simulation into their own validation systems.

Recently, specifically in the domain of driving scenarios, game engines have been adapted [27, 34]. Another virtual simulator system, which gained popularity in the research community is CARLA [11], also based on a commercial game engine (Unreal4 [12]). Although game engines provide a good starting point to simulate environments, they usually only offer a closed rendering set-up with many trade-offs balancing between real-time constraints and a subjectively good visual appearance to human observers. Specifically, the lighting computation in this rendering pipelines is limited and does not produce physically correct imagery. Instead, game engines only deliver fixed rendering quality typically with 8bit per RGB color channel and only basic shadow computation.

In contrast, physical-based rendering techniques have been applied to the generation of data for training and validation, like in the *Synscapes* dataset[33]. For our experimental work we use the physical-based open source Blender Cycles renderer[3] in high dynamic range (HDR) resolution.

The effect of sensor and lens effects on perception performance has only been limited studied. In [4, 24] the authors are modeling camera effects to improve synthetic data for the task of bounding box detection. Metrics and parameter estimation of the effects from real camera images are suggested by [23] and [5]. A sensor model including sensor noise, lens blur, and chromatic aberration was developed based on real data sets [15] and integrated into our validation framework.

Looking at virtual scene content, most recent simulation systems for validation of complete AD system include simulation and testing of the ego-motion of a virtual vehicle and its behavior. The used test content or scenarios are therefore aiming to simulate environments with a large extension and are virtually driving a high number of test miles (or km) in the virtual world provided [8, 25, 32]. This might be a good strategy to validate full AD stacks, one problem for validation of perception systems is the limited coverage of data testing critical and performance limiting factors.

A more suitable approach is to use probabilistic grammar systems [10, 33] to generate 3D scenarios which include a catalog of different object classes and places them relative to each other to cover the complexity of the input domain. The *VALERIE22* dataset demonstrates the effectiveness of our probabilistic grammar system together with our previous scene parameter variation [30] with a novel multi-stage strategy. This approach allows to systematically test conditions and relevant parameters for validation of the perceptional function under consideration in a structured way.

The remainder of this contribution is structured as the following: The next section will give an outline of our synthesis approach and a description of the generated meta-data. In section 3 a brief

[2]For example Carmaker from IPG or PreScan from TASS International.
[3]https://www.blender.org/

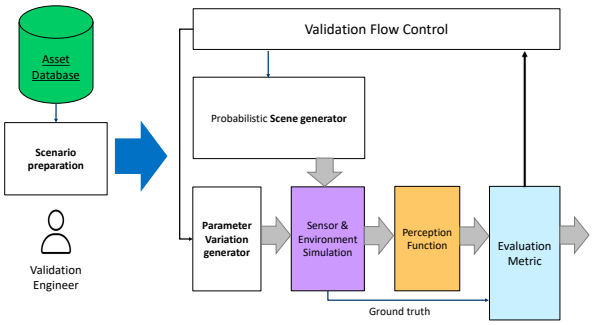

Figure 2: Overview of VALERIE pipeline flow as defined in [14].

overview of the *VALERIE22* dataset and its characteristics can be found.

In section 4 we will give a comparison of *VALERIE22* with a number of publicly available real and synthetic datasets.

2 VALERIE DATA SYNTHESIS PIPELINE

VALERIE is composed of several modules, as depicted in fig. 2. The validation flow control is in principle designed to run automated validation strategies in a data center, with the help of an orchestration based on slurm[4]. A description of the concept of these modules is outside the scope of this paper, see [14] for more details. The aim in here is to only give an overview over some of the modules in the data synthesis part, so that the reader is able to understand the features of the dataset and how to identify objects in the rendered frames.

2.1 Computation of synthetic data

Synthetic data is generated with graphics methods. Specifically for color (RGB) images, there are many software systems available, both commercially and as open source. For the generation of the dataset described in this paper Blender was used as a base to import, edit, and rendering of 3D content.

The generation of highly varied synthetic data involves the following steps:

(1) A 3D scene model with a city model is generated using a terrain/street generator. Parameters like width of a street and pavement, type of segment (e.g. tall houses, sub-urban residential, green/park, place, etc.) and materials for roads, sidewalks, segments are generated based on a scene description. Alongside this process the semantic information about the types and geometry of the segments is passed as input to the next step.

(2) A placement step is inserting 3D assets, like cars, vegetation, road elements and pedestrians into the scene. This placement is inserting objects based on a density declaration (per segment) and a list of assets for this type of segment (e.g. road, sidewalk, etc.). The result is a complete scene. Fig. 4 shows examples of scenes with a variation of person densities.

[4]https://slurm.schedmd.com/documentation.html

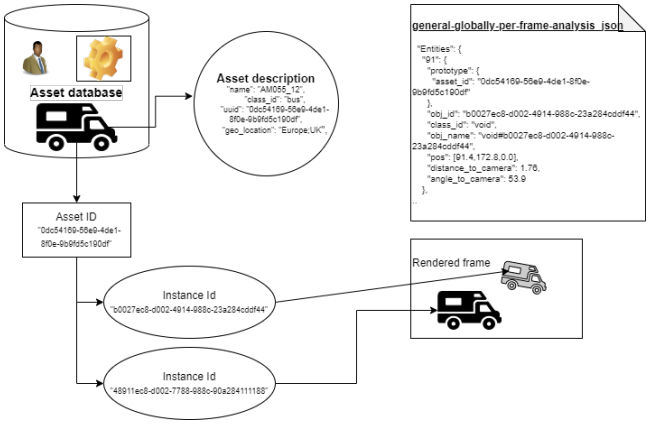

Figure 3: Object identifiers allow tracking of object instances through the rendered frames and metadata.

(3) (optionally) a set of scene parameters can be varied before each rendering pass. This includes position of objects, cameras and time-of-the-day (to vary the sun position) and many more.

The steps (1) and (2) are computed in the **Probabilistic scene generator** in fig. 2 and step (3), the variation of scene parameters is executed in the module **parameter variation generator**.

The dataset contains a multitude of additional metadata. For example all objects in the scenes are tagged with an identifier (see next section) and semantic and scene information, like position in the scene and distance + angle to the camera is documented in form of json files. This enables a multitude of possibilities to analyze the data, and we hope to stimulate research on understanding performance of DNNs with our dataset.

2.2 Assets and object instances

The assets[5] in the asset database (left side in fig. 2) have a unique identifier in form of a UUID (Universally Unique IDentifier). This identifier is used in the scene description either explicitly (for static objects) or in selection lists used by the probabilistic scene generator.

The asset id[6] is also used to identify objects in the rendered frames. The dataset contains metadata files (json format) with a list of objects and their asset ids. Objects are also identified with a specific UUID. This is depicted in fig. 3. In the appendix, section on *Metadata* an example json file is listed. The "entities" key, in this example "91" is an integer and corresponds to the instance label (see below) of the instance ground truth. With the help of the scene metadata files and the unique UUIDs of the assets it is possible to identify assets in the rendered scene. This can be used for statistical purposes or to retrieve more information from the asset database (not included in the dataset).

The scene composition and also the used assets in *VALERIE22* are European, e.g. the traffic signs and road markings are German. The types of houses are also mainly European style.

[5]An asset here means a 3D model or 2D texture.
[6]id == identifier for brevity.

Figure 4: Variation of density of pedestrians in the street and on side walk (top) low, to high (bottom).

2.3 Ground truth and metadata

The *VALERIE22* dataset provides a very rich set of metadata annotations and ground truth:

- pixel-aligned class groups (semantic label image)
- pixel-aligned object instances (label image)
- object 2D bounding box
- object 3D bounding box
- object position and orientation, angle and distance to camera
- object occlusion (only for person class)
- scene parameters, specifically time-of-the-day and sun (illumination)
- camera parameter, including pose in scene

The labels for object classes will be mapped to a convention used in annotation formats and follows the Cityscapes convention [7] for training and evaluation of the perception function. The 2D image of a scene is computed along with the ground truth extracted from the modeling software rendering engine.

2.4 Sensor Simulation

We implemented a sensor model to simulate real sensor behavior. The module works on HDR images in linear RGB space and floating point resolution as provided by the Blender Cycles renderer. The resulting image resolution is 1920x1200.

We simulate a camera error model by applying **sensor noise**, as added Gaussian Noise (mean=0, variance: free parameter) and an automatic, histogram-based exposure control (linear tone-mapping), followed by non-linear **Gamma correction**. Further, we simulate the following lens artifacts **chromatic aberration**, and **blur**. Fig. 5 shows a comparison of the standard tone-mapped 8bit RGB output of Blender (left) with our sensor simulation (right). The parameters were adapted to approximate the camera characteristic of Cityscape images. The images do not only look more realistic for the human eye, they also are further closing the domain gap between the synthetic and real data (for details see [15]).

3 DATASET DESCRIPTION

The *VALERIE22* dataset is a product of our deep variational data synthesis pipeline method, as sketched out in the previous sections. The dataset is structured in sequence groups. Each group has certain characteristics, like the amount of different assets used, the scene complexity and composition.

There is a history of experiments and development of the synthesis pipeline. Each sequence therefore corresponds to a stage in the development of the data synthesis pipeline and exhibits distinctive features differentiating them from one another is aspects, like number of 3D model assets used or variations of camera positioning (ego-vehicle) or time-of-the day and linked to that the sun position.

The most distinctive features of each sequence are described in Table 1. The following gives a brief description of the main characteristics of the sequence groups.

Table 1: Characteristics of each sequence generated at 48.18° N, 11.58° E in the *VALERIE22* dataset.

Seq.	Characteristics	Frames	Cameras per Scene	Scenes
0050	Fixed street layout 2 crossings Night scenes	1000	1	200
0052	Similar to 0050 Time 5:30 to 21:00 (GMT+1)	1800	1	300
0054	2 crossings Few traffic signs Time 10:05 (GMT+1)	480	1	480
0057	2 crossings 7 facades Varying street width Time 6:00-20:00 (GMT+1)	1000	1	1000
0058	Similar to 0057 Time 6:30-20:00 (GMT+1)	1395	1	1395
0059	2 crossings Varying street width Time 10:30 (GMT +1)	1306	2	653
0060	T-junction Varying street width Random ego-vehicle looking direction	1430	2	715
0062	Similar to 0058 Time 7:00-10:12 (GMT+1) South looking direction	10855	2	700

Figure 5: Realistic sensor effect simulation, (left) standard Blender tone-mapped output, (right) the sensor simulation output.

The early sequences *50* and *52* are based on a simple grammar with a fixed street and one crossing. They were kept in for variety, specifically since sequence *50* contains some night scenes with artificial light sources.

The sequences from *54* to *62* were generated with our procedural scene generator, as briefly introduced in section 2.1. The general layout is described in Table 1, column *Characteristics*.

Most variations in the dataset were created by linear stepping through a parameter interval or random sampling of these. Examples are time-of-the-day to control the sun settings or position and orientation of the camera. The parameters used in variation runs are documented in a json file with the actual parameter variations. However, the sun camera parameters are also documented in the 'per-frame-analysis' file.

The scene generator allows to specify statistical variations, e.g. on Gaussian distribution of object and pedestrian densities in certain zones, e.g. on the road or side walk. Further, some high-level features map, for example the width of the street onto an automatic layout of the street (autolane feature): Depending on the street width it includes park lanes and separate lanes in each direction. Table 2 shows an excerpt from the scene generator configuration file, as used in sequence *54*.

Table 2: Scene parameters for sequence *54*.

```
"facade_dist": 4,
"static_stuff_distance": 1,
"density_road_persons": 0.005,
"stdvar_road_persons": 0.001,
"density_side_persons": 0.01,
"stdvar_side_persons": 0.005,
"density_lane_cars": 0.005,
"stdvar_lane_cars": 0.002,
"density_parking_cars": 0.003,
"stdvar_parking_cars": 0.01,
"parking_separator_spacing": 25,
"tree_density": 0.01,
"tree_density_sidewalk": 0.01,
"parking_separator_var": 2.0
```

The use of the procedural scene generation and statistical placement generate scenes with a very balanced distribution. The 3D assets were inserted from our asset database, as depicted in fig. 2.

We could demonstrate that our data sequences produced less bias problems than early sequences on the manual placements in the *SynPeDS* dataset. The next section gives more details of the influence of various parameters and compares it to other available datasets.

4 EVALUATION

To evaluate the quality of our dataset we conducted several experiments using the semantic segmentation task. We compare the segmentation performance of a `DeeplabV3+` model trained on our synthetic data and compare the performance with models trained on several synthetic datasets. The performance of these models is then evaluated on five different real world automotive segmentation datasets. Use cases of our metadata include improved training and identification of impairing factors (for more details see [16, 18]).

Next, we investigated the segmentation performance on the person class of the *CityPersons* dataset if we train the model on subsets of our dataset. We additionally evaluated the person class performance with models trained on subsets of the *SynPeDS* dataset [29] provided by the KI Absicherung project[7]. Finally, we investigated how the performance of the models differs for the number of unique person assets used to create the datasets and their subsets.

Lastly, we investigated how the number of training images influences the segmentation performance. Again we trained on subsets of our dataset and the *SynPeDS* dataset and evaluated the segmentation performance on all classes with the *DeeplabV3+* segmentation model.

4.1 Computation and evaluation of perceptional functions

State-of-the-art perception functions consists of a multitude of different approaches considering the wide range of different tasks. For

[7]Currently a publication of the SynPeDS dataset is under preparation, see https://www.ki-absicherung-projekt.de/

experiments presented in this chapter, we are considering the task of semantic segmentation. In this task, the perception function segments an input image into different objects by assigning a semantic label to each of the input image pixels. One of the main advantages of semantic segmentation is the visual representation of the task which can be easily understood and analyzed for flaws by a human.

In this work, we considered the `DeeplabV3+` CNN-based model which originated from [6] and utilizes a `ResNet101` [19] backbone. The backbone has been pre-trained on the *ImageNet* dataset [9].

We compare our dataset to three different synthetic datasets. The first dataset is the synthetic dataset *SynPeDS* [29] consisting of urban street scenes inspired by the preceding two real-world datasets. The second dataset is the *GTAV* dataset [27], created by sampling data from the 3D game of the same name. Last, the *Synscapes* dataset [33] which is intended to synthetically re-create charateristics of the *Cityscapes* dataset is considered.

To compare our dataset we train segmentation models on each of these datasets with their respective training split and evaluate the segmentation performance on five real-world datasets. Usual data augmentation strategies are applied, e.g., random image cropping and flipping, saturation and contrast distortions as well as application of additive Gaussian noise. The training was performed on 2 Nvidia Quadro RTX 6000 graphics cards and a batch size of 4 images per GPU. The learning rate was set to $lr = 0.005$, with a weight decay of 0.0001 and momentum of 0.9 for the stochastic gradient descent (SGD) optimizer. Each model is trained for 50 epochs and the applied loss function is cross entropy.

An overview of all the used datasets in this work is given in tab. 3. The first dataset we considered is the *Cityscapes* dataset [7], a collection of European urban street scenes in the daytime with good to medium weather conditions. The second dataset is the A2D2 by [13], similar to the *Cityscapes* dataset it is a collection of German urban street scenes and additionally it has sequences from driving on a freeway. The third dataset is the *BDD100K* dataset [35] a diverse dataset recorded in North-America at diverse weather conditions. Next, the *India Driving Dataset* dataset [31], which was recorded in India and contains entirely different street scenes compared to the European or American datasets. Last, the *Mapillary Vistas* dataset [26], a world wide dataset with emphasis on northern America. All of these datasets are labeled on a subset of 11 classes which are alike in these datasets to provide comparability between the results of the different trained and evaluated models.

To measure the performance of the task of semantic segmentation the mean Intersection over Union (mIoU) from the COCO semantic segmentation benchmark task is used [28]. The mIoU is denoted as the intersections between predicted semantic label classes and their corresponding ground truth divided by the union of the same, averaged over all classes. We showed in our previous work how to use the extensive metadata accompanied to our dataset to detect data biases in person detectors due to the underlying training data used to train the bounding box detectors [18].

Another work investigated the usage of the metadata to calculate visual impairing factors, i.e., factors that lead to detrimental detection performance of a person detector such as increased occlusion or decreased contrast. Re-training a person detector with a focus on harder to detect samples, according to these factors, improves the overall detection performance [16].

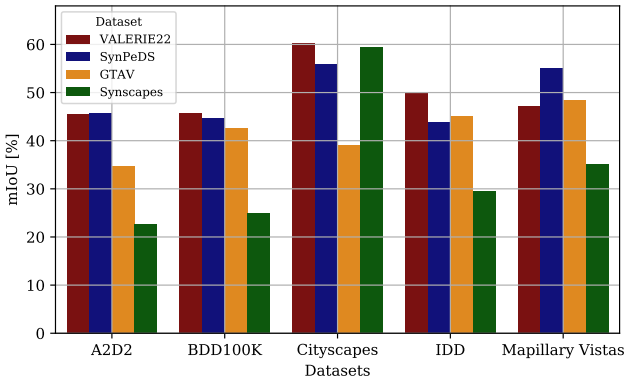

Figure 6: Cross-domain 11-class segmentation performance of synthetic datasets *VALERIE22*, *SynPeDS*, *GTAV* and *Synscapes* evaluated on real world datasets *A2D2*, *BDD100K*, *Cityscapes*, *IDD* and *Mapillary Vistas*.

4.1.1 Cross domain evaluation. To demonstrate the quality of our synthetic dataset we conducted several cross-domain performance experiments with other real-world automotive and synthetic datasets. This cross-domain performance analysis is also commonly referred to as generalization distance. We trained a DeeplabV3+ model on our *VALERIE22* dataset, as well as for the *SynPeDS*, the *GTAV* and the *Synscapes* dataset. Next, we evaluated the segmentation performance on the validation data split of real-world datasets *A2D2*, *BDD100K*, *Cityscapes*, *IDD* and *Mapillary Vistas*.

As the real-world and synthetic datasets do not have exactly the same semantic annotation format, the segmentation models were trained on a subset of 11 labels per dataset to ensure consistency of classes across. The labels are defined as follows: Road and sidewalk incorporate the road-markings and the curb respectively. Further, the building, sky, car and truck classes are used, which are consistent across these datasets. Pole, traffic light and traffic sign classes are mapped from similar sub-classes in the used datasets, e.g., utility pole in *Mapillary Vistas*. The vegetation class consists of the *Cityscapes* sub-classes terrain, i.e., plants covering the ground, and the original vegetation class, i.e., trees and bushes. Last, the person class is defined as all humans in the dataset, e.g., pedestrians and riders.

The mIoU generalization performance has been found to be a good predictor of the domain discrepancy compared to other well known domain distance measures, e.g. Fréchet Inception distance (FID) [20] or Kernel Inception distance (KID) [2], as was shown in [17]. Additionally, the segmentation performance captures the understanding of the whole scene of an image compared to just single objects as would for example the 2D bounding-box detection cross domain performance. The mIoU cross-domain generalization performance results over all 11 classes are depicted in fig. 6. Our *VALERIE22* dataset performs best on three datasets (BDD100K, Cityscapes, IDD) and just marginally worse than the *SynPeDS* trained model on A2D2. Compared to the mainly North-American based *Mapillary Vistas* dataset our dataset shows a significant domain shift. Although, still the cross-domain evaluation of *VALERIE22* is significantly better than *Synscapes* and close to *GTAV*.

Table 3: Description of real-world and synthetic datasets used in our work with number of annotated frames for semantic segmentation task. The metadata Sky Model describes the time of day, weather and location of the scene. The sensor metadata describes the used sensor and additional parameter such as focal length and placement in the world coordinate system. The Object Level metadata provides information about placement of an object in the world coordinate system and additional metrics such as contrast, occlusion etc. as described in [18].

	Dataset	Annotated frames	Tasks				Metadata		
			Sem.-Seg.	Inst-Seg.	2D Det.	3D Det.	Sky Model	Sensor	Object Level
Real-world	*A2D2*[13]	41K	✓	✓	-	✓	-	✓	-
	BDD100K[35]	10K	✓	-	✓	-	-	-	-
	Cityscapes[7]	5K	✓	-	-	-	-	-	-
	IDD[31]	10K	✓	✓	-	-	-	-	-
	Mapillary Vistas[26]	25K	✓	✓	-	-	-	-	-
Synthetic	*GTAV*[27]	250K	✓	✓	-	✓	-	-	-
	Synscapes[33]	25K	✓	✓	✓	✓	-	-	-
	SynPeDS[29]	150K	✓	✓	✓	✓	✓	✓	-
	VALERIE22 (ours)	13,5K	✓	✓	✓	✓	✓	✓	✓

Most notably our dataset outperforms the *SynPeDS* dataset on the *Cityscapes* dataset. This comes as a surprise as the *SynPeDS* dataset was created to synthetically resemble the *Cityscapes* dataset.

Even though our dataset consists of significantly fewer frames for training the segmentation model, the cross-domain performance is on-par or better compared to the other synthetic datasets. We attribute this to the high diversity of assets and sequences which was a central focus in the development of our dataset. To further underline this hypothesis we analyzed the influence of asset diversity on the generalization performance.

4.1.2 Number of Assets. We conducted experiments to understand the influence of diversity of the training data. Therefore, cross-domain performance is evaluated by comparing the number of unique training assets and the resulting cross-domain segmentation performance.

While comparing automotive real-world and synthetic images it becomes obvious that most images and scenes in real-world images are unique, whereas in synthetic images the scenes are often composed of repetitive content, i.e., a limited amount of unique assets, which are continuously differently arranged. In synthetic datasets the 3D assets, i.e., the 3D meshes and textures of objects in a scene, are expensive to create at a high fidelity and should therefore be used as much as possible. Training a pedestrian detector on a dataset consists of too few unique person assets will lead to a strongly biased detector which is able to detect solely the few trained person assets, but will fail to generalize on other persons. Overfitting will therefore occur if the training data is of low diversity and the model will fail to generalize, but it is non-obvious on how much diversity is actually needed to generalize well.

To understand the required diversity we investigated the semantic segmentation performance on the *person* class of a `DeeplabV3+` model trained with different subsets of the *VALERIE22* and the *SynPeDs* datasets. The subsets, i.e., sequences, of our dataset are described in the Appendix whereas the subsets of the *SynPeDS* dataset, i.e., tranches, are described in [29]. To track the number of unique person assets per subset in our dataset we just have to count the occurrences of unique asset IDs in the scene metadata files of a sequence.

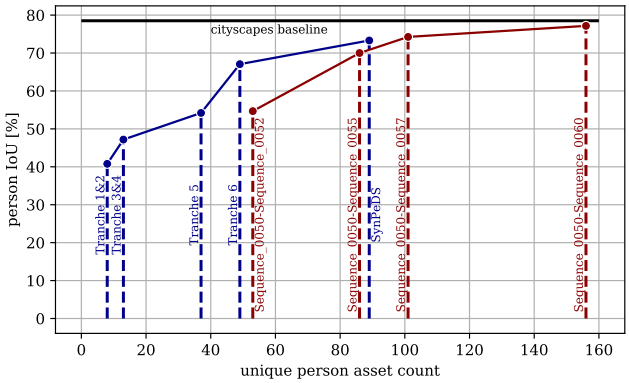

Figure 7: Unique person assets per *SynPeDS* (blue) tranche or *VALERIE22* (red) sequence and person class generalization performance on the *Cityscapes* dataset.

Each subset of both datasets represents a stage in the process of its development and therefore these dataset subsets consist of an increasing number of pedestrian assets the further the development progressed. The trained models are cross-validated on the *Cityscapes* validation dataset to investigate the cross-domain generalization performance. Fig. 7 shows the resulting number of unique person assets in the dataset subsets compared to the cross-domain person class performance measured as IoU on the *Cityscapes* dataset.

The *VALERIE22* subset for higher unique person counts clearly outperforms the *SynPeDS* subset in the cross-domain performance. While a low number of unique assets will lead to overfitting on these assets, a higher number clearly benefits the generalization capabilities of the model. Both, the *VALERIE22* trained models and the *SynPeDS* trained model benefit from an increasing number of person assets on the cross-domain performance. The model trained

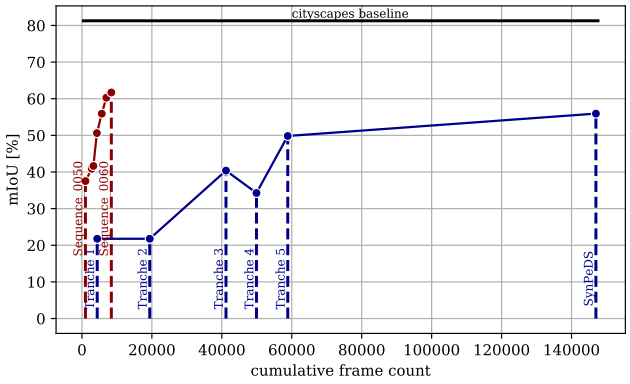

Figure 8: Number of training frames per *SynPeDS* (blue) tranche or *VALERIE22* (red) sequence and overall generalization performance on the *Cityscapes* dataset.

on our full *VALERIE22* dataset is just < 1% absolute worse in performance than the baseline *Cityscapes* trained model. The results clearly indicate the more diverse a dataset with regard to person assets, the better the generalization capabilities of a segmentation model on this class.

4.1.3 Number of Training Images. Training with a diversified dataset shows significant improvement on the cross-domain performance. This might also raise the question on the performance difference if we have a huge number of training images with lower asset diversity compared to a smaller count of images but with a higher number of assets. A very low number of images should obviously lead to overfitting, but training with a huge dataset with only marginal differences between images could lead to overfitting as well. From our previous experiment we found that the person asset diversity in the overall *VALERIE22* dataset is higher compared to the *SynPeDS* dataset and this leads to a better segmentation performance. However, the number of training images is vastly different between these datasets. To understand the influence of the number of training images we compared the cross-domain performance on all 11 classes on the *Cityscapes* dataset again trained on subsets of the *VALERIE22* and *SynPeDS* datasets. Fig. 8 shows the generalization results with the respective cumulative frame counts that were used to train each segmentation model.

While no model reaches the baseline performance of 82.34%, the cross-domain performance with sequences of our *VALERIE22* dataset reach higher mIoU performance values with far fewer image frames than the *SynPeDS* dataset. As previously shown, the diversity in the *VALERIE22* dataset continuously improved, which is evident by the increasing cross-domain performance, whereas the performance of the *SynPeDS* model even deceased for tranche 4. In tranche 4 a significant pedestrian object distribution bias was introduced into the dataset as was found in [14]. Here we additionally showed how to utilize the exact positioning metadata of the person assets in the images to identify the pedestrian distributions and understand if data biases were introduced. Overall, it is clearly visible in this result that only increasing the frame count by reiterating the same assets in the scenes is no viable strategy to increase the cross-domain generalization performance.

5 SUMMARY

This paper describes the *VALERIE22* dataset. The dataset and its underlying scene models are generated completely automated with a parametric scene generation and rendering pipeline. The results of cross-evaluation semantic segmentation experiments with real and other synthetic datasets demonstrates the performance of this approach. Compared to European datasets, VALERIE22 is performing best (or equal) to the synthetic *SynPeDS*, *GTAV* and *Synscapes* datasets.

VALERIE22 comes with a rich set of metadata annotations making it a valuable asset for research on understanding performance and domain aspects of DNNs.

ACKNOWLEDGMENTS

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project Methoden und Maßnahmen zur Absicherung von KI basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung). The authors would like to thank the consortium for the successful cooperation.

REFERENCES

[1] KI Absicherung. [n. d.]. Project home page, https://www.ki-absicherung-projekt.de/en. https://www.ki-absicherung-projekt.de/en/ Accessed: 2023-07-20.
[2] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2021. Demystifying MMD GANs. arXiv:1801.01401 [stat.ML]
[3] Wilhelm Burger and Matthew J. Barth. 1995. *Virtual Reality for Enhanced Computer Vision.* Springer US, Boston, MA, 247–257. https://doi.org/10.1007/978-0-387-34904-6_19
[4] Alexandra Carlson, Katherine A. Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. 2018. Modeling Camera Effects to Improve Visual Learning from Synthetic Data. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops.*
[5] Alexandra Carlson, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. 2019. Sensor transfer: Learning optimal sensor effect image augmentation for Sim-to-Real domain adaptation. *IEEE Robotics and Automation Letters* 4, 3 (2019), 2431–2438.
[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.
[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Las Vegas, NV, USA, 3213–3223.
[8] Werner Damm and Roland Galbas. 2018. Exploiting learning and scenario-based specification languages for the verification and validation of highly automated driving. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS).* IEEE, 39–46.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09.*
[10] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. 2020. Meta-Sim2: Learning to Generate Synthetic Datasets. In *ECCV.* virtual conference.
[11] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. *CoRR* abs/1711.03938 (2017). arXiv:1711.03938 http://arxiv.org/abs/1711.03938
[12] Epic Games. [n. d.]. Unreal Engine 4. https://www.unrealengine.com.
[13] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. 2019. A2D2: AEV Autonomous Driving Dataset. http://www.a2d2.audi.

[14] Oliver Grau, Korbinian Hagn, and Qutub Syed Sha. 2022. *A Variational Deep Synthesis Approach for Perception Validation.* Springer International Publishing, Cham, 359–381. https://doi.org/10.1007/978-3-031-01233-4_13

[15] Korbinian Hagn and Oliver Grau. 2021. Improved Sensor Model for Realistic Synthetic Data Generation. In *Computer Science in Cars Symposium* (Ingolstadt, Germany) *(CSCS '21)*. Association for Computing Machinery, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3488904.3493383

[16] Korbinian Hagn and Oliver Grau. 2022. Increasing Pedestrian Detection Performance through Weighting of Detection Impairing Factors. In *Proceedings of the 6th ACM Computer Science in Cars Symposium* (Ingolstadt, Germany) *(CSCS '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/3568160.3570225

[17] Korbinian Hagn and Oliver Grau. 2022. *Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation.* Springer International Publishing, Cham, 127–147. https://doi.org/10.1007/978-3-031-01233-4_4

[18] Korbinian Hagn and Oliver Grau. 2023. Validation of Pedestrian Detectors by Classification of Visual Detection Impairing Factors. In *Computer Vision – ECCV 2022 Workshops*, Leonid Karlinsky, Tomer Michaeli, and Ko Nishino (Eds.). Springer Nature Switzerland, Cham, 476–491.

[19] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6629–6640.

[21] Philipp Junietz, Walther Wachenfeld, Kamil Klonecki, and Hermann Winner. 2018. Evaluation of different approaches to address safety validation of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 491–496.

[22] Nidhi Kalra and Susan M. Paddock. 2016. Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? https://www.rand.org/pubs/research_reports/RR1478.html Santa Monica, CA; RAND Corporation.

[23] Zhenyi Liu, Trisha Lian, J. Farrell, and B. Wandell. 2020. Neural Network Generalization: The Impact of Camera Parameters. *IEEE Access* 8 (2020), 10443–10454.

[24] Zhenyi Liu, Trisha Lian, Joyce E. Farrell, and Brian A. Wandell. 2019. Neural Network Generalization: The Impact of Camera Parameters. *IEEE Access* 8 (2019), 10443–10454. https://api.semanticscholar.org/CorpusID:208909946

[25] T. Menzel, G. Bagschik, and M. Maurer. 2018. Scenarios for Development, Test and Validation of Automated Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 1821–1827. https://doi.org/10.1109/IVS.2018.8500406

[26] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *International Conference on Computer Vision (ICCV)*. https://www.mapillary.com/dataset/vistas

[27] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for Data: Ground Truth from Computer Games. In *Computer Vision – ECCV 2016*. Springer International Publishing, Cham, 102–118.

[28] Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2016. Fully Convolutional Networks for Semantic Segmentation. arXiv:1605.06211 [cs.CV]

[29] Thomas Stauner, Frederik Blank, Michael Fürst, Johannes Günther, Korbinian Hagn, Philipp Heidenreich, Markus Huber, Bastian Knerr, Thomas Schulik, and Karl-Ferdinand Leiß. 2022. SynPeDS: A Synthetic Dataset for Pedestrian Detection in Urban Traffic Scenes. In *Proceedings of the 6th ACM Computer Science in Cars Symposium* (Ingolstadt, Germany) *(CSCS '22)*. Association for Computing Machinery, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/3568160.3570230

[30] Qutub Syed Sha, Oliver Grau, and Korbinian Hagn. 2020. DNN Analysis through Synthetic Data Variation. In *Computer Science in Cars Symposium* (Feldkirchen, Germany) *(CSCS '20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3385958.3430479

[31] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C V Jawahar. 2018. IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments. arXiv:1811.10200 [cs.CV]

[32] Mingyun Wen, Jisun Park, and Kyungeun Cho. 2020. A scenario generation pipeline for autonomous vehicle simulators. *Human-centric Computing and Information Sciences* 10 (12 2020). https://doi.org/10.1186/s13673-020-00231-z

[33] Magnus Wrenninge and Jonas Unger. 2018. Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing. arXiv:1810.08705 [cs.CV]

[34] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. 2000. Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net* 4, 6 (2000), 2.

[35] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proc. of CVPR*. Seattle, WA, USA, 1–14.

[36] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. 2020. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* 8 (2020), 58443–58469.