Training and Validation of Visual Perception Functions for Autonomous Driving with Synthetic Data

M.Sc. Korbinian Hagn
geb. in Bad Tölz

Dissertation
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel
eingereicht im Jahr 2023

Kiel Computer Science Series (KCSS) 2024/2 dated 2024-01-18

ISSN 2193-6781 (print version)
ISSN 2194-6639 (electronic version)

Electronic version, updates, errata available via https://www.informatik.uni-kiel.de/kcss

The author can be contacted via korbinian.hagn1@gmail.com

Published by the Department of Computer Science, Kiel University

Multimedia Information Processing Group

Please cite as:

▷ Hagn, K. *Training and Validation of Visual Perception Functions for Autonomous Driving with Synthetic Data* Number 2024/2 in Kiel Computer Science Series. Department of Computer Science, 2024. Dissertation, Faculty of Engineering, Kiel University.

```
@book{Hagn2024,
  author    = {Korbinian Hagn},
  title     = {Training and Validation of Visual Perception Functions
               for Autonomous Driving with Synthetic Data},
  publisher = {Department of Computer Science, Kiel University},
  year      = {2024},
  number    = {2024/2},
  doi       = {10.21941/kcss/2024/2},
  series    = {Kiel Computer Science Series},
  note      = {Dissertation, Faculty of Engineering,
               Kiel University.}
}
```

About this Series

The Kiel Computer Science Series (KCSS) covers dissertations, habilitation theses, lecture notes, textbooks, surveys, collections, handbooks, etc. written at the Department of Computer Science at Kiel University. It was initiated in 2011 to support authors in the dissemination of their work in electronic and printed form, without restricting their rights to their work. The series provides a unified appearance and aims at high-quality typography. The KCSS is an open access series; all series titles are electronically available free of charge at the department's website. In addition, authors are encouraged to make printed copies available at a reasonable price, typically with a print-on-demand service.

Please visit `http://www.informatik.uni-kiel.de/kcss` for more information, for instructions how to publish in the KCSS, and for access to all existing publications.

1. Gutachter:	Prof. Dr.-Ing. Reinhard Koch Christian-Albrechts-Universität zu Kiel Kiel, Germany
2. Gutachter:	Prof. Dr. Hanno Gottschalk Technische Universität Berlin Berlin, Germany

Datum der mündlichen Prüfung: 11.01.2024

Zusammenfassung

Diese Arbeit befasst sich mit der Nutzbarkeit synthetisch erzeugter Daten für das Training und die Validierung visueller Wahrnehmungsfunktionen beim autonomen Fahren. Synthetisch erzeugte Bilder ermöglichen die Erstellung sicherheitskritischer Szenarien, die in der realen Welt potenziell gefährlich zu erfassen sind, und liefern zusätzlich pixelgenaue Ground-Truth Annotationen. Bei der Anwendung synthetischer Bilder auf Wahrnehmungsfunktionen, die anhand von realen Daten trainiert wurden, stellt sich jedoch das Problem der Überbrückung der Domänenlücke. Dies gilt sowohl für das Training als auch für die Validierung mit synthetischen Bildern. Daher muss die Domänenlücke hinreichend verstanden werden, um synthetische Bilder zu erzeugen, die für das Training und die Validierung verwendet werden können.

Es wird eine neue Diskrepanzmetrik eingeführt und angewendet, um die Parameter einer realistischen Sensorsimulation zu optimieren und die Domänenlücke effektiv zu reduzieren. Mehrere Einflussfaktoren auf die Domänenlücke werden hierzu untersucht. Die Faktoren, welche die visuelle Erkennung beeinträchtigen, werden vorgestellt und es wird gezeigt, dass sie einen großen Einfluss auf die Erkennbarkeit von Fußgängern haben. Aus den Erkenntnissen dieser Einflussfaktorenanalyse konnte ein Kalibrierungsverfahren einer gewichteten Verlustfunktion entwickelt werden, um die Wahrnehmungsleistung bei realen Fußgänger zu erhöhen.

Neue Methoden zur Validierung werden vorgestellt. Die tiefe Variationsdatensynthese und die Klassifizierung von Faktoren, welche die visuelle Erkennung beeinträchtigen. Während erstere Methode durch parametrisierte probabilistische Bilderzeugung nach Wahrnehmungsfehlern sucht, erkennt die letztere Methode Wahrnehmungsfehler durch die Nichtübereinstimmung eines Klassifikators und der tatsächlichen Erkennung.

Die Erkenntnisse aus Training und Validierung flossen ein in die Erstellung von zwei synthetischen Validierungsdatensätzen, *VALERIE* und *SynPeDS*.

Abstract

This work deals with the usability of synthetically generated data for training and validation of visual perception functions applied in autonomous driving. Synthetically generated images allow the creation of safety critical scenarios which are potentially dangerous to capture in the real-world and additionally deliver pixel perfect ground truth annotations. However, applying synthetic images to perception functions trained on real-world data poses the problem of bridging the domain gap. This is true for both training and validating with synthetic images. Therefore, the domain gap has to be sufficiently understood to generate synthetically images viable to be used for training and validation.

A new domain discrepancy metric is introduced and applied to optimize the parameters of a realistic sensor simulation effectively reducing the domain gap. Several influence factors on the domain gap are disentangled. The visual detection impairing factors are introduced and shown to have a high influence on the detectability of pedestrians. Additionally, these factors are used to calibrate a weighting loss function to increase the perception performance on real-world pedestrians.

New methods for perception validation are introduced. The deep variational data synthesis and the classification of visual detection impairing factors. While the former method searches for perception faults by parameterized probabilistic image creation, the latter method detects perception faults by the disagreement of a detectability classifier and the actual detection result.

The findings of both training and validating were influencing the creation of two synthetic validation datasets, *VALERIE* and *SynPeDS*.

Acknowledgements

First and foremost I want to express my sincere gratitude to my supervisor and mentor Dr.-Ing. Oliver Grau with whom I worked and researched at Intel Labs. I could learn many things and was able to grow as a researcher under his guidance and support.

I want to thank Prof. Dr.-Ing. Reinhard Koch for his academic supervision of this thesis and Prof. Dr. Hanno Gottschalk for reviewing this thesis.

I want to thank my colleagues Qutub Syed Sha and Peter Nöst for their help and support throughout this work. I want to thank Intel Labs for giving me the opportunity to work and research on this thesis.

I want to thank my brother Josef Hagn and my friend Max Büttner for proofreading this thesis and giving valuable feedback. Additionally, I want to thank Roman Raschke, Eduard Engel, Max Bierling, Benjamin Schramm and Rudolf Lehner for their support and distraction when I was in need for one.

I want to sincerely thank my mother Silvia Hagn for giving me the chance to always pursue what I set my mind to and always believing in my success. I want to thank Sabina Materak for her relentless support through all ups and downs this thesis has brought. Last, I want to thank Samuel for giving me the greatest joy with his bright smile.

Contents

List of Figures

List of Tables

List of Acronyms

Definition of acronyms used in the thesis:

CNN	convolutional neural network
CDF	cumulative distribution function
DIW	detection impairment weighting
DNN	deep neural network
EMD	earth movers distance
FID	Frechét Inception distance
FN	false negative
FP	false positive
FPPI	false positive per image
FPN	feature pyramid network
GAN	generative adversarial network
IoU	intersection over union
IS	Inception score
KID	kernel Inception distance
laMR	log-average miss rate
mIoU	mean Intersection over Union
MR	miss rate
PCA	principal component analysis
RPN	region proposal network
SSD	single shot multibox detector
SWD	sliced Wasserstein distance
TP	true positive
TPR	true positive rate

Chapter 1

Introduction

Autonomous driving is on the verge of becoming an integral part of our every day life. Powered by the ongoing development leaps of artificial intelligence and machine learning methods the dream of fully automated driving comes within reach. These advancements are highly driven by powerful sensor based perception functions. These functions utilize a range of sensors, such as monocular or stereo cameras, LiDAR and RADAR sensors to perceive the real-world and plan their driving accordingly.

As part of these achievements the question of safety emerges more and more urgent. Safety is paramount as part of the development in the automotive industry but was not yet the center of attention in the development process of perception functions. But the recent shift towards safe AI brought up a whole field of research with many nuances, laying more focus on the perceptive part of the automated driving stack.

While the datasets used for training the perception functions often stem from real-world sensor recorded data, also the trend to utilize synthetically generated sensor data is fueled by the shift of focus on safe AI.

1.1 Motivation

Validation of a fully autonomous driving stack in the real-world is a tedious task involving many hundreds of thousands of kilometers to be driven on the street to assure that every aspect, including safety relevant scenarios, is captured [KP16]. Re-evaluation of autonomous driving functions for every new software iteration is therefore not only time-consuming but also expensive. Simulation on the other hand has been used to validate software or hardware in the loop and is an inexpensive and fast alternative.

Focusing on the perception task with monocular camera sensors the simulation part is then mainly the generation, i.e., rendering, of synthetic street scenes. One of the main advantages of synthetically rendered imagery is the ability to provide rich meta annotations such as ground truth information and scene descriptions. Additionally, the long-tailed distribution of automotive street scenes can be sampled and validated without any risk to human life. This long-tail distribution samples are essentially the rare accidents or near-accidents that cannot easily and safely be captured in real-world test scenarios. Using this synthetic data for re-training the perception function then helps to reduce these gaps in the perception function.

However, there is one major factor hindering the sole usage of synthetic data: The domain gap. This gap is described as the difference of training and validation datasets distribution. This difference can be as subtle as changes in the color distribution of the images, or as prominent as changes of the buildings or persons due to different geolocations. Understanding and overcoming the domain gap is the key to successfully apply synthetic data for training perception functions but also to improve its applicability for validation.

1.2 Research Question

The usage of synthetic data is steadily increasing due to the ease of data generation, i.e., image rendering and annotation, and the capabilities of risk-free synthesis of spurious or dangerous events which is especially useful in testing autonomous driving algorithms.

The main research question this thesis tries to answer is twofold: First, how can we utilize solely synthetic data for training visual perception functions which are thereafter applied to real-world datasets. This directly leads to the consequence of understanding the differences between synthetic source and real-world target domain, also named as the domain gap. Measuring this domain gap with the right distance or discrepancy measures has a significant influence in understanding the factors defining the domain gap. With the right measures in place one can find and disentangle the factors influencing the domain gap and continuously improve

the synthetic image generation process to bridge the remaining domain gap.

Second, how can this improved synthetic data be used for validation of real-world perception functions. With the improved synthetic data the domain gap does not significantly influence the validation of perception functions anymore. Understanding the shortcomings of current state-of-the-art semantic segmentation and 2D bounding box pedestrian detectors is essential to improve the safety of such algorithms in the real-world.

1.3 Publications

This thesis main contributions are presented in academic publications which are attached in Chapter 7. All publications are listed in their chronological appearance with a short summary of their contents in the following:

Publication 1: DNN Analysis through Synthetic Data Variation

Qutub Syed Sha, Oliver Grau and Korbinian Hagn, Published in *2020 Proceedings ACM Computer Science in Cars Symposium.* [SGH20], Chapter 7.1. This contribution introduces the concept of variational data synthesis to analyze and validate visual perception functions. Through parameterization of a generative content rendering system this approach shows how to create validation data given a previously defined validation goal. Results of this paper explain the influence of pedestrian object occlusion rates and visible pixels towards the detection by two state-of-the-art semantic segmentation models.

Publication 2: Improved Sensor Model for Realistic Synthetic Data Generation

Korbinian Hagn and Oliver Grau, Published in *2021 Proceedings ACM Computer Science in Cars Symposium.* [HG21], Chapter 7.2. This paper proposes a method to improve data synthesis methods for automotive datasets by introducing a realistic sensor simulation model. Additionally, the earth movers distance (EMD) based domain divergence measure is introduced and utilized as a parameter optimization criteria to adapt the sensor

simulation from synthetic to real-world images overall increasing the cross-domain performance of a semantic segmentation model by over 7%.

Publication 3: Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation

Korbinian Hagn and Oliver Grau, *Published in: Fingscheidt, T., Gottschalk, H., Houben, S. (eds) Deep Neural Networks and Data for Automated Driving. Springer, Cham.* [HG22b], Chapter 7.3. This book chapter is a more in-depth continuation of the previous publication about improving the quality of synthetic data generation methods by realistic sensor simulation. Here, additional focus is laid on the extraction of sensor parameters from real-world datasets and application of the extracted parameters on the sensor simulation. Moreover, the EMD domain divergence criteria is thoroughly compared to the well-established Frechét Inception distance (FID) domain distance. It was found that the EMD better projects the generalization performance on the target dataset than the FID.

Publication 4: A Variational Deep Synthesis Approach for Perception Validation

Oliver Grau, Korbinian Hagn and Qutub Syed Sha, *Published in: Fingscheidt, T., Gottschalk, H., Houben, S. (eds) Deep Neural Networks and Data for Automated Driving. Springer, Cham.* [GHS22], Chapter 7.4. This book chapter first applies the concept of variational *deep* data synthesis. It introduces the module of probabilistic scene generation, variation of scene parameters and application of the realistic sensor simulation introduced in previous publications. It is shown by the creation of synthetically generated images with high numbers of diverse objects at various illumination settings that we can effectively validate pedestrian detection algorithms on the influence of factors such as, different occlusion objects, additive Gaussian noise, and pedestrian training data distributions.

Publication 5: Validation of pedestrian detectors by classification of visual detection impairing factors

Korbinian Hagn and Oliver Grau, *Proceedings of the 17th European Conference on Computer Vision Workshops (ECCVW 2022), 2022*, Chapter 7.5. In this publication the concept of visual detection impairment factors are introduced. These factors severely influence the detectability of objects, here pedestrians, for object detectors. We showed how these factors can actually influence the detectability. Additionally, we introduced a classification method of pedestrian objects into *detectable* and *non-detectable* according to these factors and applied this classification to validate a real-world pedestrian detector finding training data biases, such as ethnicity or age biases.

Publication 6: Increasing pedestrian detection performance through weighting of detection impairing factors

Korbinian Hagn and Oliver Grau, *2022 Proceedings ACM Computer Science in Cars Symposium.*, Chapter 7.6. This paper applies the visual detection impairment factors to calibrate an empirical weighting loss of a real-world pedestrian detector. Training pedestrian samples are here weighted according to their extracted impairment factors. We show that this empirical detection impairment weighting loss (DIW loss) improves the state-of-the-art on a real-world pedestrian detection benchmark.

Publication 7: SynPeDS – A Synthetic Dataset for Pedestrian Detection in Urban Traffic Scenes

Thomas Stauner, Frédérik Blank, Michael Fürst, Johannes Günther, Korbinian Hagn, Philipp Heidenreich, Markus Huber, Bastian Knerr, Thomas Schulik, Karl Leiss, *2022 Proceedings ACM Computer Science in Cars Symposium.*, Chapter 7.7. This publication is the official paper alongside the publication of the KI-Absicherung project's synthetic dataset, *SynPeDS*. It contains ground truth for a plethora of visual perception tasks in the automotive domain, such as semantic segmentation, instance segmentation, 2D and 3D bounding boxes, and pose information. We demonstrate the quality of the dataset by semantic segmentation cross-domain generalization

experiments on the person class and on all classes. Additionally, we investigate the influence of pedestrian assets on the cross-domain generalization performance on real-world data.

1.4 Thesis Overview

The thesis is structured as follows: Following the introduction, the state-of-the-art is outlined. Next, our advancements on training with synthetic data for visual perception functions are described. In Chapter 4 the variational data synthesis method and further validation methods are explained. Chapter 5 shows how the preceding two chapter's findings were used to help the creation and improve the two synthetic validation datasets, *VALERIE* and *SynPeDS*.

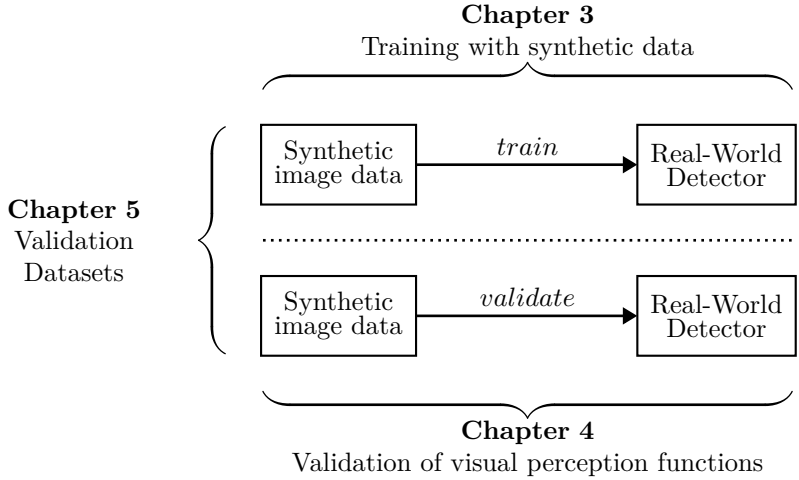


Figure 1.1. Contributions of the chapters 3 to 5.

Figure 1.1 summarizes the Chapter 3 to 5 in a figurative way. The subsequent Chapter 6, conclusion, summarizes the key findings of this thesis. Last, the publications that were produced as a part of this thesis are listed in chronological appearance order in Chapter 7.

Chapter 2

Background

In this chapter we give an overview of current state-of-the-art and recent developments of the visual perception tasks of semantic segmentation and object detection for autonomous driving. Furthermore, the recent work on synthetic data generation for training and validation of these tasks is outlined. SotA methods for training and validation of these visual perception functions with an overview on current validation datasets are described.

2.1 Visual Perception Tasks

Throughout this thesis we consider two visual perception tasks that are trained and validated with a set of different methods, namely semantic segmentation and object detection.

2.1.1 Semantic Segmentation

Semantic segmentation is the task of assigning a label class to each pixel of an input image. This task can be seen as a development of a multi-class classification problem.

This work considers two well-established convolutional neural network (CNN) segmentation architectures: the `DeeplabV3+` [CZP+18], an extension of the original `Deeplab` [CPK+17] model, and the `Detectron2` [WKM+19] model. The `DeeplabV3+` utilizes the atrous convolution, a spatial dilation convolutional kernel, to extract rich feature maps from the backbone. `Detectron2` on the other hand utilizes a feature pyramid network (FPN), i.e., a union of feature maps with different scales extracted from the backbone. In this work we applied the `ResNet101` [HZR+16] as default backbone

for low level feature extraction which was pre-trained on the *ImageNet* [DDS+09] dataset.

The default performance evaluation metric is the mean Intersection over Union (mIoU) which is widely applied in semantic segmentation benchmarks [COR+16; VSN+18]. The mIoU in percent is calculated by the following equation:

$$mIoU = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_s}{TP_s + FP_s + FN_s} \times 100\%. \tag{2.1.1}$$

Here, $\mathcal{S}$ is the set of all segmentation classes with S being the cardinality of this set. The true positive (TP) are the number of pixels correctly classified to the class s, false positive (FP) are the number of pixels incorrectly classified to the class s and false negative (FN) are the number of pixels incorrectly not classified for the class s.

There have been adaptations proposed to mitigate some shortcomings of the mIoU by focusing more on the segmentation contour [RTG+19; FMW+18] or on a per-image basis [CLP+13], but most detection benchmarks use the original mIoU definition.

2.1.2 Object Detection

The second perception task this work focuses on is object detection. Object detection is the task of localization and classification of objects in an image. Due to the automotive setting the perception task in our work is pedestrian detection. Again, CNN-based architectures are mainly used for this task. These architectures split into two groups: one-stage and two-stage detectors. One-stage detectors such as the `YOLOv3` [RF18], `YOLOv4` [BWL20] and single shot multibox detector (SSD) [LAE+16] apply the localization and classification in a single step by a fixed number of predictions. In our work we used the SSD model with a `ResNet50` [HZR+16] backbone. The two-stage detectors such as `R-CNN` [GDD+14], `Faster R-CNN` [Gir15; RHG+15], and `Mask R-CNN` [HGD+17] apply a region proposal network (RPN) to pre-filter potential regions of interest before sending these proposals to the detection head which performs the actual localization and classification task. In our work we used a development of the `R-CNN` [GDD+14] and `Faster R-CNN` [Gir15; RHG+15] models, named `Cascade R-CNN` [CV19]. `Cascade R-CNN` ap-

plies cascaded bounding box regression, i.e., multiple detection heads are in sequence with every iterative detection head having a higher intersection over union (IoU) threshold of predicted bounding box and ground truth. With the `Cascade R-CNN` we applied the high quality `HRNet` [WSC+20] and the efficient `MobileNet` [HZC+17] backbone for feature extraction. Both model's backbones have been pre-trained on the *ImageNet* [DDS+09] dataset as is common practice. For evaluation of the performance of an object detector in particular a pedestrian detector the miss rate (MR) metric is used. The MR for a given confidence threshold c is defined as follows:

$$\mathrm{MR}(c) = \frac{\mathrm{FN}(c)}{\mathrm{TP}(c) + \mathrm{FN}(c)}. \tag{2.1.2}$$

The MR is the ratio of miss-detected, i.e., FN, to all relevant objects. For the automotive pedestrian detection task the MR is especially important because missing a pedestrian detection can be potentially fatal. However, for an autonomous vehicle the number of FP detections is important as well, as too many FP detections would lead to a deterioration of the car's driving function. Therefore, many pedestrian detection benchmarks [ZBS17; BKF+19; DWS+09; ZXW+19] apply the log-average miss rate (laMR) metric. The laMR defines the notion of false positive per image (FPPI) as follows:

$$\mathrm{FPPI}(c) = \frac{\mathrm{FP}(c)}{N}. \tag{2.1.3}$$

The FPPI measures the number of FP over the number of images N for the confidence threshold c. Combining equation 2.1.2 and equation 2.1.3 to form the laMR:

$$\mathrm{laMR} = exp(\frac{1}{9} \sum_{f \in \mathcal{F}} log(\mathrm{MR}(\underset{\mathrm{FPPI}(c) \leqslant f}{\operatorname{argmax}} (\mathrm{FPPI}(c))))) \tag{2.1.4}$$

With f being an equally spaced interval $f \in \mathcal{F} = [10^{-2}, 10^{0}]$. Lower laMR values in benchmarks indicate higher detection performance. A recent increase in popularity of models based on the transformer architecture [DBK+21], originally developed from the natural language processing

domain, the pedestrian detection benchmarks are still dominated by CNN-based detection models.

2.2 Generation of Synthetic Data

The main focus of this work is the utilization of synthetic data for training and validation of visual perception functions which is an accepted technique for computer vision applications [BB95]. For automotive applications there are already several methods exploiting synthetic data for verification [**kalra16driving**; JWK+18; DG18]. The scenarios are aiming to simulate environments spanning a huge virtual space and are simulating a high amount of virtual driving routes in the virtual world [MBM18; WPC20; DG18]. While these methods focus on the full autonomous driving stack, the training and validation methods of the perception function can be decoupled. This allows to create tailored validation strategies focusing on the synthesis of the sensor impressions without the need of a full physical vehicle simulation. Approaches exploiting these factors and using grammar systems to generate 3D scenarios have been proposed by [DKF20; WU18]. A mixture of a grammar system, i.e., the explicit description of a scene to render and validate, and probabilistic scene generation is used to create the *VALERIE* dataset. Especially in the automotive domain game engines have been adopted to create synthetic data by extraction of images and labels from the rendering pipeline [WEG+; RVR+16; RHK17]. One of the most popular simulator systems is the *CARLA* [DRC+17] system based on the game engine `Unreal4` [Gam]. Another approach to render synthetic images is the physical-based rendering technique. This technique was used for the creation of the *Synscapes* [WU18], the *VALERIE* dataset as well as one part of the *SynPeDS* dataset (with the other part created in `Unreal4` [Gam]). These datasets build upon the physical-based open source `Blender Cycles` [Fou] renderer.

2.3 Training with Synthetic Data

Training a perception function with synthetic data is limited by the domain distance, i.e., the difference of source and target data distributions. Finding

the right metrics to measure this domain distance is the key to understand and disentangle the factors decreasing the remaining gap from synthetic to real data.

2.3.1 Domain Gap Measurement and Metrics

A popular method to measure the domain distance is based on the classification output of a `InceptionV3` [SVI+16] model trained on the *ImageNet* [DDS+09] dataset, named Inception score (IS) [SGZ+16]. The work of [HRU+17; BSA+18] instead of relying on the classification output of the `InceptionV3` model, propose to use the features from intermediate layers to form the FID and kernel Inception distance (KID) respectively. These metrics have been successfully applied to train domain adaptation methods. However, it was shown that these metrics cannot reliably predict if the classification performance increases when domain adapted data is applied as training data [RV19]. For instance, a decreased FID score after domain adaptation does not necessarily correlate with an increased segmentation performance when this data is used for training.

Another approach is to measure the performance directly by training solely on the synthetic dataset and validate the model on the real-world target data. This approach is referred to as cross-evaluation or cross-domain generalization and is applied by several previous works [WU18; SAS+18; RSM+16a]. It should be noted that this measure is inherent asymmetric [LLF+20b] which has to be considered when comparing it to symmetric distance measures. Our work in Chapter 3 builds upon the cross-domain generalization performance and introduces a new measure on a per-image basis mitigating weaknesses of these averaging metrics.

2.3.2 Influence Factors on the Domain Gap

Reducing the domain distance is an open research field and especially in the area of domain adaptation several methods to diminish the domain gap by application of generative adversarial networks have been proposed [THS+17; LCW+15; GL15; THS+18]. While these techniques can actually reduce the domain distance measured by FID or KID, the disentanglement of the underlying factors are not well understood.

One influence factor is the additive noise in the training data which is a common technique for image augmentation in training to prevent overfitting [Bis95]. Work by [CCN+16; NCC+18] applied different sensor effects which are not adapted to the target dataset to the training set and reported a degradation of the cross-domain performance. However, modeling camera effects to improve the learning with synthetic data for 2D bounding box detection has been proposed by [CSV+18; LLF+20a]. Learning the camera sensor parameters as a style-loss from a real-world dataset extracted from a `VGG-16` [SZ15] model's feature vector and applying these parameters for training with synthetic data for 2D bounding box detection was shown by [CSV+19] to improve the cross-domain generalization performance. However, using a `VGG-16` [SZ15] model trained on the *ImageNet* [DDS+09] dataset poses the problem of optimizing features unrelated with the actual parameters of the target dataset. We propose in Chapter 3 a method to directly optimize sensor artifact simulation parameters on the target data distribution.

2.3.3 Visual Detection Impairing Factors

Visual detection impairing factors are defined to be influential on the capability of a detector to reliably detect an object. Some of these factors, such as occlusion rate or contrast, have been previously studied. For example, the authors of [ZBO+16] conclude in their work that the contrast measure of an object ought to have no influence on the object detection capability, however, we are able to show in Chapters 3 and 4 that this factor has indeed a significant influence. The occlusion rate, i.e., the ratio of visible to occluded and visible pixels, as well as the distance of an object to the observer are well acknowledged to have a significant influence on the detection capability [DWS+11; DWS+09]. In Chapter 3 we introduce several additional influence factors and show in section 4.2.2 the actual influence on the task of pedestrian detection.

2.3.4 Training and Sampling Methods

In Chapter 3 a novel sample weighting loss for pedestrian detection based on the notion of visual detection impairing factors is introduced. Sampling

methods are a well-researched topic with the most popular approaches being importance sampling [KM53] and hard example mining [MGE11; SGG16]. These methods do not assign higher weights but specifically oversample harder training samples. Harder samples in the sense of hard example mining [MGE11; SGG16] are determined by the gradient loss of a sample. Boosting algorithms such as AdaBoost [FS97] iteratively weight miss-classified or harder samples higher.

For single-stage detectors the focal loss [LGG+17] gained high popularity by putting higher weight to harder samples steered by the cross-entropy loss of each sample. The focal loss, as well as Re-sampling [CBH+02; DGZ17] or cost-sensitive weighting [Tin00; KHB+17] methods all tackle the class-imbalance problem of single-stage detectors. The class-imbalance occurs due to the relevant objects, e.g., pedestrians, only making up for a small part in the image whereas most of the objects are part of the background. Training a detection head with uniform weights for all objects would actually exaggerate the loss of background objects. Two-stage detectors eliminate this problem by applying a RPN, extracting and balancing only relevant objects from the image before sending these proposals to train the detection head with.

Specifically targeting the problem of pedestrian detection are the repulsion loss [WXJ+18] and the aggregation loss [ZWB+18] which enforce the bounding box proposal to be very tight on the pedestrian because in automotive scenarios pedestrians are often in crowds and partially occluded by one another.

A somewhat different approach is self-paced learning [KPK10]. Here, the learning of easier examples first is encouraged. The training of easier examples first should prevent the model from being stuck in a bad local optimum.

Actually, reaching local minima or even the global minimum is discouraged as this would result in overfitting to the training data which leads to bad cross-domain performance on the validation data. Therefore, regularization techniques have been proposed tackling this problem [KPK10; MMX+17; JMZ+15], improving the robustness against the training set bias [RZY+18].

Chapter 3.2 focuses on offline pre-weighting training samples according to visual detection impairment factors ajar to the human visual system

without additional hyperparameter tuning. This approach is in contrast to most of the mentioned online weighting methods including meta-learning approaches [RZY+18; TP12; ADG+16; LUT+17].

2.4 Validation of Visual Perception Functions

The validation of visual perception functions is of high relevance to guarantee a safe autonomous driving functions. Therefore, current automotive safety standards as the ISO 26262 [ISO18] and ISO DIS 21448 [ISO21b] have been developed, defining so-called safety cases for safety-related functions, which form an argument to achieve only a residual risk by the collection of evidences supporting this claim. A plethora of work focusing on the connection of AI safety and these mentioned standards has already been conducted [BGH17; SQC17; GHP+20; SS20; ACP21; BKS+21]. Findings of these methods are already developed into a new safety standard ISO/AWI PAS 8800 [ISO21a]. Other approaches try to further tie these safety arguments to AI methods by measuring the relevance of established metrics in relation to the safety [CNH+18; HSR+20; SKR+21; CKL21]. Another example of such measure is shown in [LGH+21a] by tying the visual detection impairment factor of distance to the object with the safety relevance of detecting this object.

These developments brought up the notion of functional insufficiencies [GMB18] in AI models. Our approaches in Chapter 4 specifically target the lack of generalization [SSH20]. A lack of generalization describes the fault that the model could not perform as expected on the target domain.

Validating of a machine learning model for such functional insufficiencies is done by the creation and testing of corner cases [AGG+21; BSS+21]. Corner cases are defined by [BBL+19] as *"non-predictable relevant object in relevant location"*. In other words these corner cases are rare events in the tail of the distribution function of automotive street scenes, such as collisions or near-collisions. In our validation data generation approach we find corner cases by probabilistic sampling of a scene parameter space, i.e., corner case samples are a portion of the data samples generated by this method.

2.5 Visual Perception Datasets

Several datasets for automotive applications, specifically targeting visual perception, have been proposed. Real-world semantic segmentation datasets such as the *Audi Autonomous Driving Dataset* (*A2D2*) [GKM+20], *Berkeley Deep Driving Dataset* (*BDD100K*) [YCW+20], *Cityscapes* [COR+16], *India Driving Dataset* (*IDD*) [VSN+19], and *Mapillary Vistas* [NOR+17] are used throughout this work in cross-domain performance experiments. These datasets form a cross-section of many geolocations including North- and South-America, Europe, and Asia. We constrained our work on Europe and Germany and therefore the *Cityscapes* dataset with the primary German geolocation is of special interest in this thesis. Another Asian automotive dataset to mention is the *ApolloScape* [HCG+18] dataset. But due to restrictive licensing this dataset is not used. For the task of pedestrian detection additional bounding box annotations for *Cityscapes* are provided. This dataset is referred to as *CityPersons* [ZBS17]. Additionally, the *EuroCity Persons* (*ECP*) [BKF+19] is considered for our validation application in Chapter 4. However, several more pedestrian detection dataset are available, such as the *WiderPerson* dataset [ZXW+19], the *Caltech* [DWS+09] dataset or the *ETH* [ELV07] dataset.

These datasets are very valuable for training and as adaptation target for domain adaptation techniques, e.g., our realistic sensor simulation. But, even though real-world datasets adhere to existing standard workflows of crowdsourcing annotations [LWZ+16; SDF12; KRF+16], human annotators inherently introduce inaccuracies or errors into these ground truth annotations. These inaccuracies or errors are defined as label noise and have been studied [NDR+13] with mitigation strategies already being proposed [RLA+14; LYS+17; JZL+18; Vah17; HMW+18]. With regard to the validation of visual perception functions this label noise poses a great challenge. Differentiating insufficiencies of perception models due to actual functional insufficiencies or due to erroneous labels in the validation data is a tedious process and can hardly be automated.

Synthetic data provides a solution to this problem, as ground truth annotations are always accurate. Provided that the implementation does not contain any errors. Several synthetic datasets for automotive perception task have been proposed, for example the *Synthia* [RSM+16b] dataset,

the *Synscapes* [WU18] dataset and the *GTAV* [RVR+16; RHK17] dataset. These datasets are valuable for benchmarking cross-domain adaptation methods but do not allow for additional creation of corner-case data. Here, autonomous driving simulators such as *Carla* [DRC+17] and the *LGSVL* simulator [RST+20] prove to be beneficial [LGH+21b; GHA21]. Procedural methods for road generation [PJX+20] can enhance the capabilities of these methods. Some methods try to reduce the remaining domain gap by synthetization of test images through generative approaches [RBK+21]. But similar to label noise the image inconsistencies introduced by these generative models with regard to the corresponding annotation data makes it unfeasible for validation. In our work we utilize synthetic data from the *VALERIE* and *SynPeDS* datasets whose scenes are created by variational methods without the need for a full-fledged autonomous driving simulation.

Chapter 3

Training with Synthetic Data

In this chapter we present our research on training visual perception functions with synthetic data. If we want to achieve a high performing perception function on the target domain it is necessary to overcome and close the domain gap between synthetic source data and real-world target data.

Beginning with Section 3.1, we define and validate an appropriate metric to measure the domain gap from synthetic to real-world datasets. We investigate the influence of a realistic sensor simulation on the domain gap and additionally design an optimization method to adopt these sensor simulation parameters for a real-world automotive dataset. Hereby a novel cross-domain generalization metric based on the EMD or Wasserstein-2 distance is used. This measure is based on a redefinition of the cross-domain per-image performance as a domain gap proxy. Following the investigations of the sensor simulation influence on the domain gap, we disentangle several additional factors on their respective influence on the domain gap. These factors are the number of graphical assets used to create a synthetic dataset, the number of frames to train a perception function and the structural similarities of real-world and synthetic data based on the segmentation heatmaps.

In Section 3.2 the detection impairing factors are introduced. These factors represent influential factors that are responsible for impairing the visual detection of pedestrians. Visual detection impairing factors are defined to be influential on the detection abilities by the human vision system. In this section we show how to utilize these impairing factors to detect missing training data in the synthetic source domain by comparing the distributions of visual detection impairing factors of a synthetic and real-world dataset. Furthermore, we show how one can

boost the detection performance of a pedestrian detector by weighting training samples according to a visual detection impairing factor calibrated loss function.

In Chapter 4 we utilize these factors to validate pedestrian detectors and detect training data biases.

3.1 Overcoming the Domain Gap

When training deep neural network (DNN)- or CNN-based visual perception functions with a synthetic source dataset it is advantageous that the training samples have the same or at least a similar distribution as the target domain distribution. The difference in distributions of source and target domain is known as domain gap. A domain gap from source and target domain will lead to detection performance degradation. Therefore, when using synthetic training data, the sample distribution of the target domain should be modeled as close as possible. However, the target data distribution is not easily available and moreover a complex combination of individual distributions, such as lighting, textures, and even more subtle factors like the number of unique persons in the dataset. Without the possibility of direct comparison of these sample distributions one can instead measure the domain gap by indirect measures. One of these measures is the domain generalization distance, i.e., the distance of a performance measure when trained on the synthetic domain compared to when trained on the target real-world domain. Such measures and metrics are essential if we want to be able to model the synthetic data distribution to increasingly resemble the real-world distribution. Additional, these metrics allow disentangling of certain influence factors on the domain gap, i.e., understanding their influence on the model performance, as we can show in Subsection 3.1.3.

3.1.1 Measuring the domain gap

Measuring the domain gap for visual perception learning is based to a large extend on the advances of domain adaptation techniques [THS+17; LCW+15; GL15; THS+18]. Some predominantly used measures in this area

of research are the IS [SGZ+16], the FID [HRU+17], and the KID [BSA+18]. All of these measures are based on either extracting feature vectors (FID, KID) or softmax scores from the `InceptionV3` [SVI+16] network trained on the *ImageNet* [DDS+09] dataset. Hereby, the relevant features or scores are extracted from both, the source dataset and the target dataset, and subsequently a distance metric between those is calculated. However, these measures are, while being useful to train generative networks such as generative adversarial network (GAN), non-predictive of the actual performance [HG21]. In this work we showed for the task of semantic segmentation that the target domain mIoU performance of a CNN trained with a dataset of lower FID value, i.e. potentially smaller domain gap, is worse than a model trained on a dataset with a higher FID value.

From these observations we constructed a new performance based domain discrepancy measure with a close link to the actual cross-domain performance. This new measure is based on the EMD of the target to target domain performance compared to the source to target performance. The performance itself is measured as mIoU on a per-image basis. While this measure is related to the overall cross-domain performance measurement of the mIoU, the EMD-based metric can capture the performance of a perception function at harder and easier to classify images as these would be averaged out in a measure on the average of whole target dataset. This measure is explicitly designed to be a discrepancy measure and not a distance measure of the domain gap. A distance is inherently symmetric and would not capture the difference of training on the source domain and evaluating on the target domain compared to training on the target domain and evaluating on the source domain. A discrepancy is therefore necessary if we want to have our measure strongly tied to the actual domain generalization performance of source to target domain.

We prove the hypotheses of the viability that our EMD-based domain discrepancy measure on the per-image performance measure can be used as a proxy for the domain gap. Therefore, we conducted an experiment by training an ensemble of `DeeplabV3+` semantic segmentation models with a `ResNet101` backbone on the *Cityscapes* dataset. The weights of each model were initialized randomly, and each model was evaluated after the training on the *Cityscapes* validation dataset on a per-image basis. The results are mIoU histograms over the *Cityscapes* validation dataset. Next,

we compare the histograms by calculating the cumulative distribution function (CDF)s of each histogram. On the resulting CDFs we apply the 2-sample Kolmogorov-Smirnov test to each possible pair of the ensemble, and we get a minimum $p - \text{value} > 0.95$. Resulting, we cannot reject the null-hypotheses that the per-image based performance histogram is not a good proxy of the training dataset.

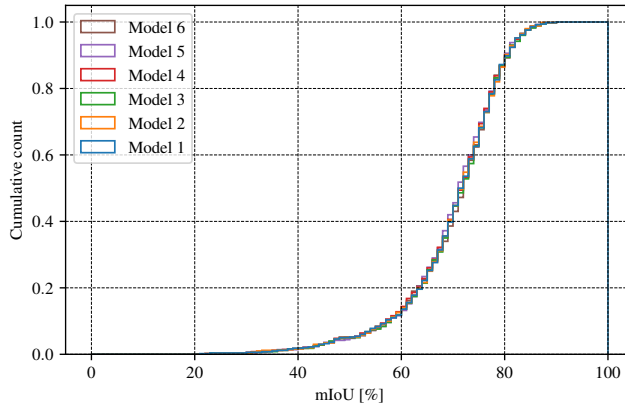


Figure 3.1. CDF of ensembles of `DeeplabV3+` models trained on Cityscapes and evaluated on the corresponding validation set. (Source: Publication 2 in Chapter 7.2 [HG21])

In Figure 3.1 one can see the CDFs of the resulting per-image performance histograms. This experiment proves that only the training dataset is responsible for the shape of the per-image based mIoU performance.

3.1.2 Realistic Sensor Simulation

The domain gap is the main cause of reduced performance on the target dataset when training with synthetic data, therefore it has proven beneficial to adapt the source synthetic dataset to the target dataset reducing this exact gap. The research area which deals with this task specifically is domain adaptation. While there is a plethora of domain adaptation strategies based on generative models showing impressive results on the visual

adaptation of images on a target domain, there is a major drawback if one chooses to use these adapted images for validating a model. This drawback is the non-deterministic behavior of these generative domain adaptation models on the input images which leads to inconsistencies between the visually adapted image and the original ground truth. An often seen example is the visible Mercedes-Star of the ego car in the *Cityscapes* dataset that is hallucinated into the synthetic image when adopted to this dataset [RAK22; HTP+18; PEZ+20]. If we use these adopted images to validate a perception model for detection faults, it is hard to impossible to track an error back to the real source of the problem due to the non-matching input and ground truth. However, for validation with synthetic data it is still important to minimize the domain gap of the training dataset to the target validation dataset, as with a high domain gap between these datasets one cannot safely determine if a perception fault was caused due to the domain difference, e.g. different geolocation, or due to actual errors in the detection model. Therefore, understanding the influential factors of the domain gap is a key component to improve validation with synthetic images as well as improving the cross-domain performance when training with synthetic datasets. If these influential factors are well enough understood we can then model, i.e., simulate, these factors and apply them on our synthetic dataset deterministically.

One of these influential factors we have identified are the sensors used to capture the real-world imagery data. More specifically the sensor lens artifacts of cameras which are inherently observable in real-world datasets as can be seen for example in an image of the *A2D2* dataset in Figure 3.2. Synthetically generated images on the other hand often simulate a pinhole camera model [Stu14] that does not realistically simulate any sensor lens artifacts. In real-world datasets we could observe several sensor lens artifacts such as blur, chromatic aberration, and additive sensor noise. Furthermore, in datasets such as the *Cityscapes* dataset, the images were captured in a high dynamic range format and then subsequently tone-mapped and gamma corrected to an 8-bit integer RGB low dynamic range to be displayable on most modern screens. Tone-mapping and gamma correction have a major influence on the style and even the usability of the image for perception as under exposed areas in an image such as persons in the shadow on the sidewalk can be highlighted. Unfortunately, detailed

Figure 3.2. Real-world images (here A2D2) exhibit sensor lens artifacts which have to be closely modelled by an image synthetization process to decrease the domain distance of synthetic to real-world datasets to make them viable for training and validation. (Source: Publication 2 in Chapter 7.2 [HG21])

specifications about the tone-mapping process on real-world datasets recordings are often not released to the public.

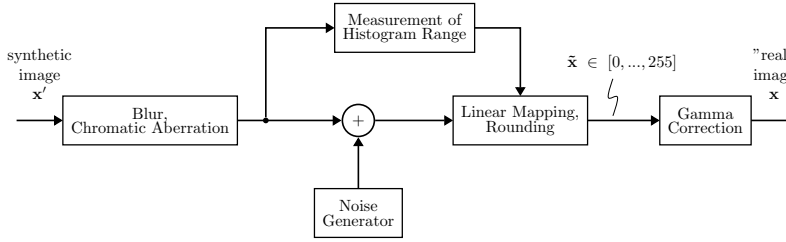


Figure 3.3. Our sensor artifact simulation pipeline. (Source: Publication 3 in Chapter 7.3 [HG22b])

With our observations on the sensor lens artifacts of real-world datasets we build a sensor simulation pipeline applying these artifacts on our synthetic images and evaluate the influence on the cross-domain generalization performance, i.e. the domain gap. Our exemplary sensor simulation pipeline is depicted in Figure 3.3.

(a) (b)

Figure 3.4. Left (a): Synthetic images without lens artifacts. Right (b): Applied sensor lens artifacts, including exposure control. (Source: Publication 3 in Chapter 7.3 [HG22b])

Beginning from our image synthetization process we receive 16-bit OpenEXR images. On these synthetic 16-bit OpenEXR floating point image x' from the rendering pipeline a blur and chromatic aberration is applied, followed by an additive Gaussian noise generator. In parallel, in the *measurement of histogram range* block function, the histogram of the pixel RGB values is calculated by binning these values into 100 equidistant bins in the range of $[0, 10^6]$. These bins are then used in the *Linear Mapping, Rounding* block to saturate, i.e. fix the RGB value to $[255, 255, 255]$, of the pixels in the highest s% of bins. Here, s is a parameter to tune the saturation mapping operation. The remaining bins are equidistantly mapped to the range $[[0, 0, 0], [255, 255, 255]]$. We found that a saturation of about 2% will lead to a saturated sky, which is generally without interesting details for

pedestrian perception, and higher detail on the darker parts of the image, typically where pedestrians are located on the road or sidewalk. Last, we apply a gamma correction to further enhance the detail in darker areas of the image. Both, the saturation and gamma correction are part of the typical tone-mapping process of a real-world camera recording.

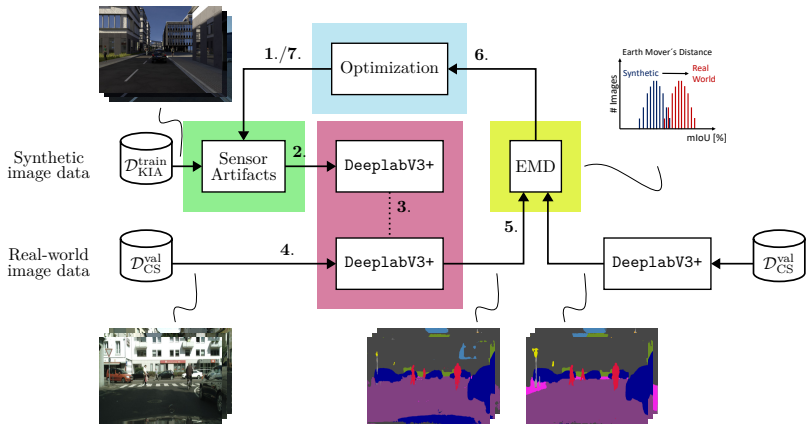


Figure 3.5. Optimization of sensor artifacts to decrease discrepancy between real and synthetic datasets. (Source: Publication 3 in Chapter 7.3 [HG22b])

Figure 3.4 shows the applied sensor simulation on an image from the *SynPeDS* dataset. While unnecessary information on the top of the image is lost due to the saturation process, the detail in the darker areas of the image are enhanced for example on the pedestrian on the right. Furthermore, the blur, Gaussian noise, and chromatic aberration lead to fraying of the edges which were previously unrealistically sharp.

It is very important to know how to set the individual parameters of our sensor artifact simulation to better match the appearance of the target dataset. Therefore, we extracted the relevant parameters to tune for the sensor simulation manually from the target datasets images. As these manually extracted values are inaccurate and therefore non-optimal only

limited improvement over the original baseline can be achieved. However, these extracted parameters can be used as a well suited starting point for a black-box optimization of the sensor parameters onto the target dataset. The previously introduced EMD domain gap discrepancy measure is hereby used as the optimization loss. Figure 3.5 depicts this black-box parameter optimization.

The optimization process starts with the previously manually extracted sensor parameters and applies those with our sensor artifact simulation on the synthetic image data. After training for one iteration the EMD discrepancy between synthetic and real-world data is calculated. The trust region reflective [SLA+15] optimization method continuously changes the parameters until either the local optimum is reached or the step size of parameter changes is below 10^{-6}. In our work, as seen in Publication 2 in Chapter 7.2, we were able to show that after the optimization an increase in cross-domain performance from the *SynPeDS* dataset to the *Cityscapes* dataset of around 7% mIoU is achieved.

3.1.3 Disentangling Domain Gap Influence Factors

A realistic sensor artifact simulation is only one of the many influence factors that makes up the domain gap between synthetic and real-world images. Understanding the influence factors by comparison of domain discrepancy and distance measures while tuning some parameters can be a tedious process with limited success if not done carefully. Most synthetic datasets do not deliver additional meta information on their image synthetization process which would prove useful to understand and disentangle the domain gap influence factors even further. Disentangle in this context means to understand the individual contribution of a factor on the domain gap. In our work, we could resort on the *VALERIE* and the *SynPeDS* datasets which were specifically designed to deliver as much metadata about the synthetization process of the images as possible. Utilizing this rich metadata allows us to better understand and disentangle the domain gap influence factors than with most other synthetic datasets, as we show in the following subsections.

Number of Assets

Comparing automotive real-world and synthetic images it is evident that most images and scenes in real-world images are unique whereas in synthetic images the scenes are composed of repetitive content but continuously differently arranged. This comes to no surprise as 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 comprising only a single unique person asset will lead to a strongly biased detector which is able to detected solely the one trained person asset but will fail to generalize on other persons. It is obvious that overfitting will 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 bridge the domain gap and generalize well. In our experiments we investigate the semantic segmentation performance on the person class of a `DeeplabV3+` model trained with different subsets of the *VALERIE* and the *SynPeDs* datasets. These datasets and their subsets are described in more detail in Chapter 5. Each subset of either dataset represents a stage in the process of its development and therefore do these dataset subsets consist of an increasing number of pedestrian assets the further the development progressed. Each of the trained models is cross validated on the *Cityscapes* validation dataset to investigate the cross-domain generalization performance. Figure 3.6 shows the resulting number of unique person assets in the dataset subsets compared to the cross-domain person class performance measured as mIoU on the *Cityscapes* dataset.

The *VALERIE* 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 *VALERIE* trained models and the *SynPeDS* trained one benefit from an increasing number of person assets on the cross-domain performance. The model trained on the full *VALERIE* dataset is only $<$ %1 worse in performance than the baseline *Cityscapes* trained model. It is evident that more diversity of person assets is beneficial to the generalization capabilities of a segmentation model. But, even though the first tranches up until

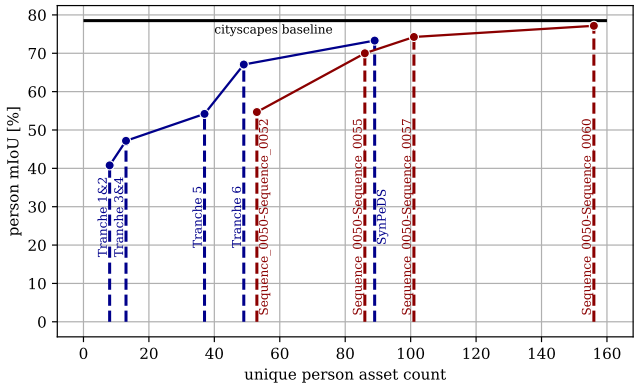


Figure 3.6. Unique person assets per *SynPeDS* (blue) tranche or *VALERIE* (red) sequence and person class generalization performance on the *Cityscapes* dataset.

tranche 6 of the *SynPeDS* dataset have only a few number of unique assets, these are sufficient to achieve the same performance as earlier Sequences of the *VALERIE* dataset. To better understand this behavior, the difference between these datasets has to be investigated and further influential factors on the domain gap have to be disentangled.

Number of Training Images

While training with a diversified dataset shows significant improvement on the cross-domain performance it also raises 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 can lead to overfitting as well. From our previous experiment we found that at least the person asset diversity in the overall *VALERIE* dataset is higher compared to the *SynPeDS* dataset. However, the number of training images is vastly different between these datasets. To understand the influence of the number of training images we compare the overall cross-domain performance measured in mIoU on

the *Cityscapes* dataset with `DeeplabV3+` models trained on subsets of the *VALERIE* and *SynPeDS* datasets. Figure 3.7 shows the generalization results with the respective cumulative frame counts that were used to train each segmentation model.

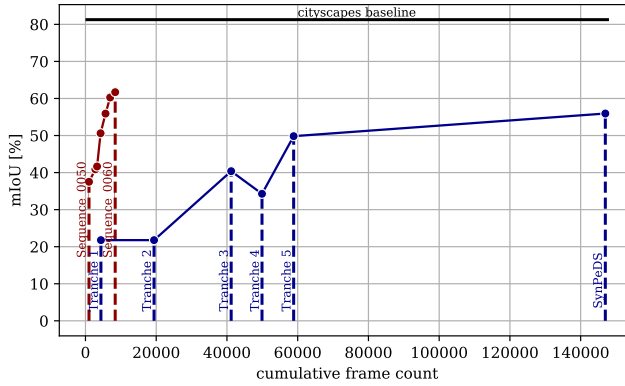
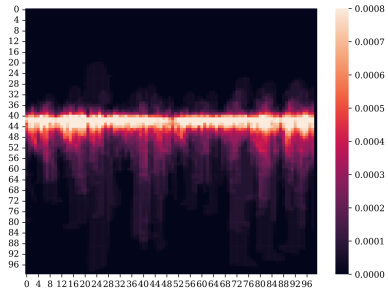


Figure 3.7. Number of training frames per *SynPeDS* (blue) tranche or *VALERIE* (red) sequence and overall generalization performance on the *Cityscapes* dataset.

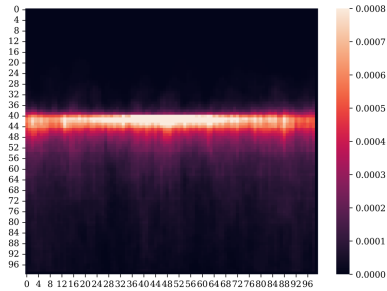
While no model comes close to reaching the baseline performance of 82.34%, the cross-domain performance with Sequences of the *VALERIE* reach higher mIoU values with far fewer image frames than the *SynPeDS* dataset. The diversity in the *VALERIE* dataset continuously improved which is evident by the increasing cross-domain performance, whereas the performance of the *VALERIE* model even deceased for tranche 4. In tranche 4 a significant pedestrian object distribution bias was introduced into the dataset which we detected with our validation methods as can be seen in section 4.1. 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.

Structural Similarities from Segmentation Heatmaps

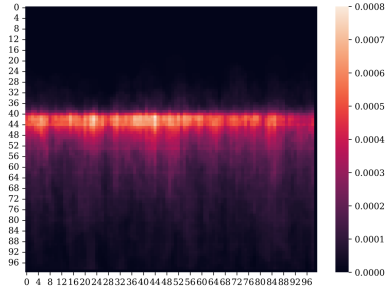
We investigated the influence of the person asset diversity of a synthetic dataset and found it to be influential on the cross-domain generalization.



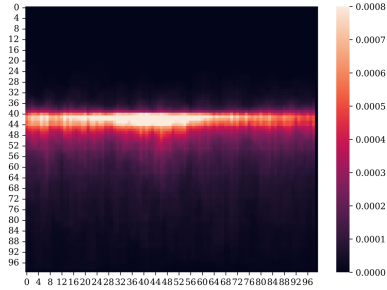
(a) *VALERIE* Sequence 0050



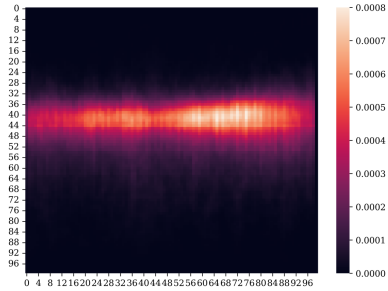
(b) *VALERIE* Sequence 0058



(c) *VALERIE* Sequence 0060



(d) *VALERIE* Sequence 0050-0060



(e) *Cityscapes*

Figure 3.8. Cumulative semantic segmentation heatmaps derived from different *VALERIE* sequences and from the *Cityscapes* dataset.

Understanding the influence of the placement of objects in the scene is the

next logical step to explain the remaining domain gap. This difference of object placements between source and target dataset can be understood as the structural similarities of the respective scenes in the datasets. One method to measure this structural similarities is to compare the semantic ground truth per class of the synthetic and real-world domain. To do this, the number of occurrences for every class individually and per pixel in the ground truth are accumulated across a dataset. Next, the accumulated results are sub-sampled and binned to a 100x100 grid. Thereafter, these results are normalized to the range $[0, 1]$. The overall result is a 2D histogram of normalized class occurrences for every class in the dataset. Figure 3.8 exemplary depicts these histograms on the progression of *VALERIE* Sequences ((a) to (d)) compared to the target *Cityscapes* (e) heatmap for the person class.

By visual analysis of the histograms one can determine first differences in the person distribution between early *VALERIE* sequences and the *Cityscapes* dataset. The sequences 0050 and 0060 show clear person silhouettes around the main horizontal distribution. This occurs when the number of frames is quite low when calculating these histograms. Comparing the result of combined sequences 0050 to 0060 with the *Cityscapes* heatmap it is evident that the latter distribution is vertically more spread out around the main horizontal distribution and there is a focus or center on the right side of the histogram. This distribution can be explained by the way the images were recorded. The *Cityscapes* dataset was recorded from a car in right-hand driving countries where most persons are visible on the nearer right side of the car camera and persons on the opposite side of the road are more often occluded by oncoming cars.

To actually measure a difference between the histograms or heatmaps we are utilizing the sliced Wasserstein distance (SWD) [BRP+15]. The SWD is a sample based form of the EMD for 2- and higher-dimensional data. To compute this distance the source and target 2D histograms are projected by a random sampled vector to a 1-dimensional vector each. Next, the EMD is calculated on the projected vector. This process is repeated for 1000 iterations. The final distance result is the average of all projected distance results. We apply these calculations to the *VALERIE* and *SynPeDS* dataset with the target dataset *Cityscapes*. The SWD results with the cross-domain generalization performance on the person class can be seen in Figure 3.9.

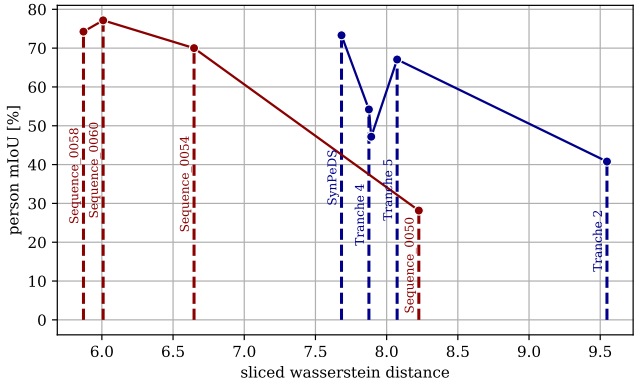


Figure 3.9. Sliced Wasserstein distance per *SynPeDS* (blue) tranche or *VALERIE* (red) sequence and overall generalization performance on the *Cityscapes* dataset.

Lower SWD results indicate a greater similarity between source and target person heatmaps, i.e., person placements.

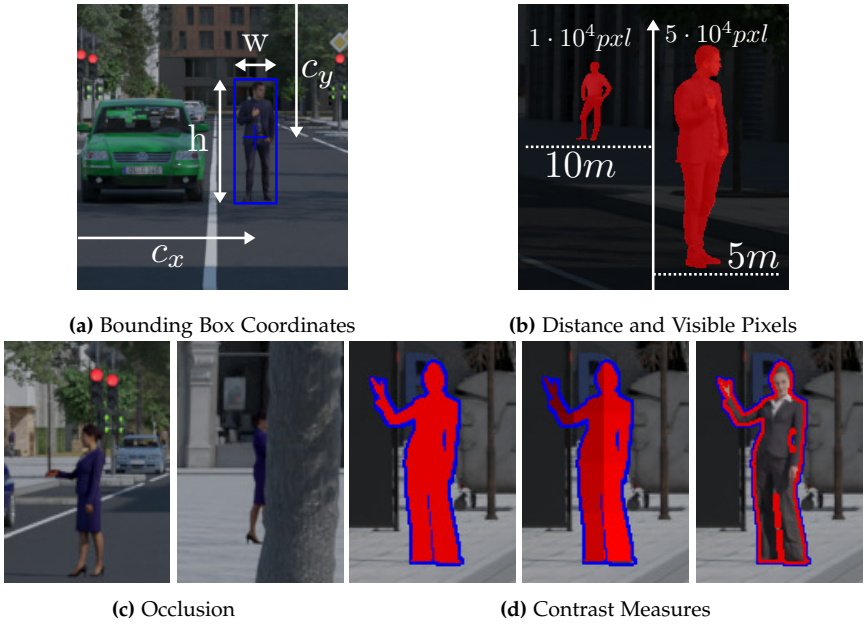
The sequences 0058 and 0060 of the *VALERIE* dataset achieve the highest cross-domain generalization performance and have the lowest SWD to the target dataset. Another interesting observation can be made for these sequences, as the 0058 sequence has a lower SWD as the 0060 sequence. Comparing these sequences again visually in Figure 3.8 one can see that there is a higher person distribution from the middle to right on the main horizontal distribution compared to the distribution of the 0060 sequence. Making the 0058 distribution more similar to the target *Cityscapes* distribution. For the *SynPeDS* dataset the results are differing from earlier tranches to later, more mature tranches. The tranche 2 shows the lowest mIoU and highest SWD values, but while tranche 5 has a higher SWD than tranche 4 the cross-domain performance is higher than the latter. With the result from the number of assets we know that tranche 5 has more pedestrian assets than tranche 4 and achieves therefore a better generalization even though the person distribution is dissimilar to the target dataset. The overall dataset of *SynPeDS* reaches the lowest SWD for this dataset but with higher performance than the *VALERIE* sequence 0054, again due to a higher number of assets in the former dataset. Concluding,

the SWD of the class distribution can be used to measure the similarities of source and target datasets and helps to better understand differences in the datasets.

3.2 Visual Detection Impairing Factors

We identified factors influencing the domain gap when training with synthetic data and applying the trained model to real data. Our work also investigates the reasons and factors that impede a successful detection of an object. We show that by understanding these factors we can leverage these to further measure domain distances, increase the detection performance and implement validation strategies to identify missing training data. The factors are termed visual detection impairing factors as they hinder understanding, i.e., impede the detection of an object. We previously focused on the task of semantic segmentation which gave us the capabilities to investigate on the overall scene structure differences. Following, we focus on the task of object detection, and due to the autonomous driving setting on the task of pedestrian detection. Results from the investigations on visual detection impairing factors are published in Publications 5 in Chapter 7.5 and 6 in Chapter 7.6.

The visual detection impairing factors we are considering in our work are visualized in Figure 3.10. The factors can be grouped in four major categories. The first category is the location of an object in the image. This category is covered by the bounding box coordinates center locations c_x and c_y. The second category is the size of an object in the image. For this category we use the width and height dimensions of the bounding box. Additionally, we use the distance to the camera measured in $[m]$ which strongly correlates to the last factor, the actual number of visible pixel of an object in the image. The third category is the occlusion of an object with the single factor occlusion rate. The occlusion rate is the quotient of visible to visible plus occluded pixels of an object. The fourth and last category is the contrast of an object. The contrast measures the visual difference of an object to its background. Low contrast values, e.g., dark clothed person standing in the shadow, makes it harder to detect a person. In our work we use three different formulations of contrast measures.



(a) Bounding Box Coordinates

(b) Distance and Visible Pixels

(c) Occlusion

(d) Contrast Measures

Figure 3.10. The potential detection performance impairing factors we consider in this work: (a) bounding box coordinates (o_{cx}, o_{cy}, o_h, o_w), (b) distance and number of visible pixels of a pedestrian (o_d, o_{vp}), (c) rate of occlusion (o_{ocl}), (d) contrast of a pedestrian (red) to its background (blue) calculated by the full pedestrian silhouette (o_{cfull}), segment wise (o_{cmean}) and edge wise (o_{cedge}). (Source: Publication 5 in Chapter 7.5 [HG23])

The first is calculated by averaging the RGB values of a person instance and calculating the Euclidean distance to the average of the surrounding background pixels. The second contrast measure segments the person into 12 segments and calculates the Euclidean distance to the adjacent background pixels with the results being averaged. The third and last contrast measure considers only a small pixel border of the person instance and calculates the Euclidean distance to the surrounding background.

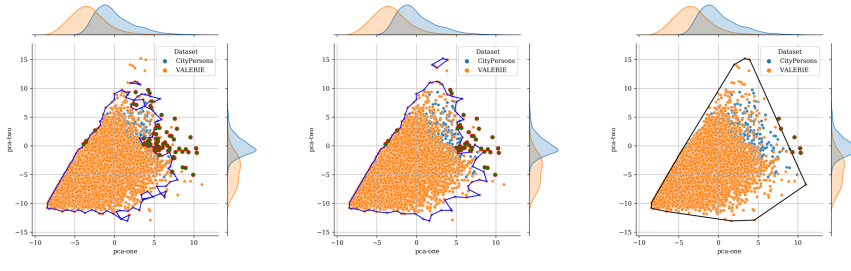
A more in depth explanation and calculation of the individual factors can be found in the Publications 5 in Chapter 7.5 and 6 in Chapter 7.6.

In the following subsection we show how we can utilize these factors to find differences in source and target datasets and reason on missing data that has to be added to the source training dataset to improve cross-domain generalization. Subsequently, we show how these factors allow us to improve a pedestrian detector by steering the training loss to focus on harder to detect samples according to these visual impairing factors.

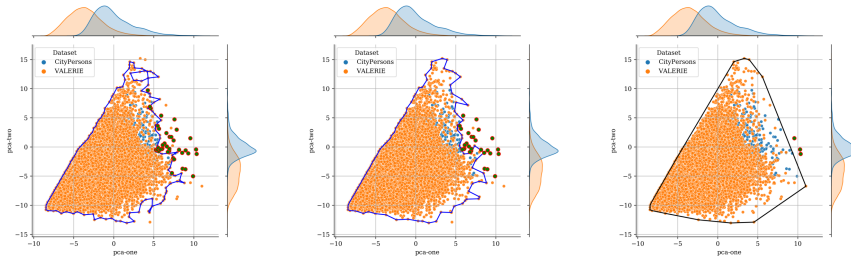
3.2.1 Missing Training Data Detection

The previously defined impairment factors can be used to describe a dataset and then use this dataset description to compare it with the description of another dataset. In our experiment we begin by extracting each individual factor for each person in the synthetic *VALERIE* dataset and the real-world *CityPersons* dataset. The *CityPersons* dataset is an extension of the *CityScapes* dataset and delivers additional bounding box annotations for persons. With the resulting visual impairing factors per person for both datasets one could already calculate differences by just comparing these factors. But due to the factors being closely correlated, e.g., distance and number of visible pixels, it is useful to reduce the redundancy by application of a principal component analysis. After reducing the dimensionality of the impairing factors with the PCA, we can visualize both datasets on a 2D plot by extraction of the first and second major PCA component and plot every pedestrian onto a scatter plot. For subsequent detection of dissimilarities between source and target dataset we can calculate an area around our source synthetic dataset points and find target real-world dataset points that are not enclosed by this area. These non-enclosed data points are persons with specific visual detection impairing factors that are not included in our source dataset. To calculate the surrounding area we use alpha shapes [EM94]. Alpha shapes are a special form of Delaunay triangulation [Del+34] where the α parameter denotes the maximum radius of a circle in the triangulation around each data point. In other words, the distance between points if there will be an edge or not. For low values of α the shape is more rugged and follows the points on the borders more closely, whereas for high values of α the shape will be similar to the complex hull of the points with the actual complex hull being $\alpha = \infty$. We visualized the PCA results for α values $\alpha = 0.7$, $\alpha = 1.0$ and $\alpha = \infty$ with

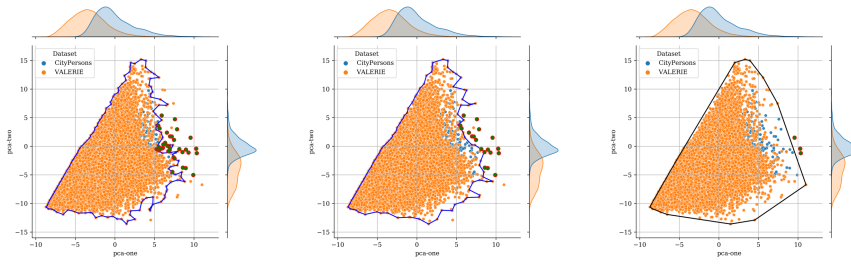
impairing factors extracted for *VALERIE* sequences 0057, 0057-0058 and 0057-0060 and *CityPersons* in Figure 3.11.



(a) Seq. 0057 $\alpha = 0.7$ **(b)** Seq. 0057 $\alpha = 1.0$ **(c)** Seq. 0057 $\alpha = \infty$



(d) Seq. 0057-0058 $\alpha = 0.7$ **(e)** Seq. 0057-0058 $\alpha = 1.0$ **(f)** Seq. 0057-0058 $\alpha = \infty$



(g) Seq. 0057-0060 $\alpha = 0.7$ **(h)** Seq. 0057-0060 $\alpha = 1.0$ **(i)** Seq. 0057-0060 $\alpha = \infty$

Figure 3.11. Alpha-shapes generated by the PCA of visual detection impairing factors from three different *VALERIE* sequence combinations and three different α values. Orange indicate *VALERIE* data points, blue indicate *Cityscapes* data point inside the alpha-shape, and red indicate *Cityscapes* data points outside the shape.

As mentioned for lower values of α the shape is more rugged and

more missing data points, visualized as red points, are detected. For the complex hull shape only few outliers are detected and there is a considerable amount of blue *CityPersons* data points being in the shape but not covered by orange *VALERIE* data points. Further, for later sequences, e.g., 0057-0060, the number of missing data points is fewer even with lower values of α. To highlight the progress from *VALERIE* sequences and better understand the right choice of the α parameter we calculated the number of missing data points for different values of α for each of the three sequences. The results are shown in Table 3.1.

Table 3.1. Number of *Cityscapes* visual detection impairing factor PCA data points not included in the alpha-shape by sequences of the *VALERIE* dataset for different values of α.

	α										
Sequence	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	∞
0057	576	278	172	137	106	92	70	51	44	40	6
0057-0058	365	179	121	100	72	52	38	30	28	28	3
0057-0060	302	143	97	73	49	40	35	30	23	19	3

With values of $\alpha < 0.7$ the alpha shape will omit too many points further off from the main distribution, creating a very dense shape around the points and therefore many outliers will be detected. Whereas for an $\alpha = \infty$, i.e. the complex hull, the shape does not capture the actual shape spanned by the points and too few outliers will be detected. We found reasonable α values are in the interval $0.7 \leqslant \alpha \leqslant 1.0$, but a visual inspection of the actual produced shapes will nonetheless prove useful. In our work, especially in the creation of the *VALERIE* dataset, this tool was used to detect missing data points which were subsequently added for each iteration of the sequences. This is visible in the number of outliers in Table 3.1 which continuously decreases for sequences 0057-0058 and 0057-0060 for every α value being used.

3.2.2 Detection Impairment Weighting Loss

As previously shown, the visual detection impairment factors are useful to detect differences in datasets. In this subsection we introduce another

approach on how to utilize these factors and boost the pedestrian detection performance of an object detector by steering the training loss towards harder to detect samples according to these factors. The detailed approach is defined in Publication 6 in Chapter 7.6.

The general idea behind our approach is to calibrate the training loss of a pedestrian detector to put more attention towards objects which are harder to detected according to the visual detection impairing factor of these samples. The workflow of this approach is depicted in Figure 3.12.

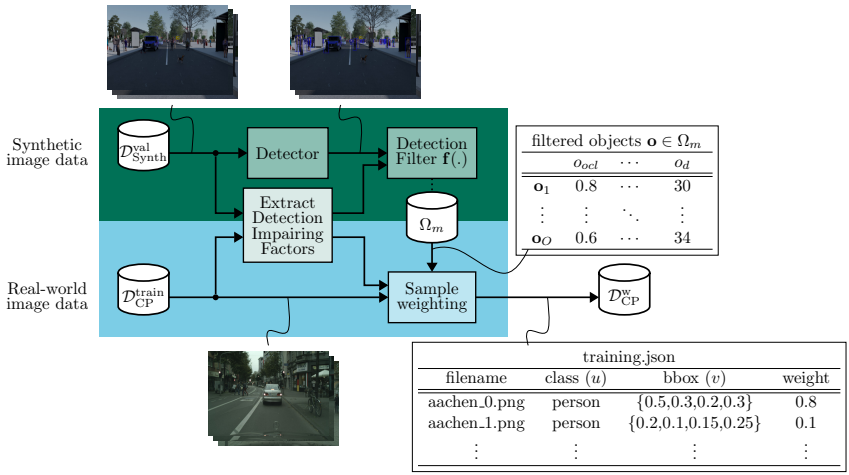


Figure 3.12. Generation of the training weights for pedestrian objects of the real-world *CityPersons* dataset. (Source: Publication 6 in Chapter 7.6 [HG23])

First, we have to define which samples are harder to detect. Therefore, we begin by inferencing on a synthetic dataset with a pedestrian detector which was pre-trained on this synthetic domain. For the synthetic data we use the sequences 00057 and 0058 from the *VALERIE* dataset. In parallel to the detection the impairing factors per pedestrian object are extracted from each inferenced image. The detection results and the corresponding impairing factors are forwarded to the detection filter $f(.)$. This filter evaluates the detections with the ground truth annotations and dismisses all true positive detected pedestrians so that only the missed detections

and their impairing factors remain. The miss detections are subsequently stored in the weighting dataset Ω_m. To now calibrate the training loss on a real-world dataset we extract the visual detection impairing factors from the *CityPersons* dataset. Next, each pedestrian object is weighted in the *Sample Weighting* block on its vicinity to the objects in the Ω_m dataset. To measure the vicinity the Mahalanobis distance [Mah36] per object is calculated. This distance assigns lower weights for samples far from the point cluster of non-detected persons and higher weights for samples closer to the cluster. To calculate this distance we have to first calculate the mean over all objects in the Ω_m dataset:

$$\overline{\mathbf{o}}_m = \frac{1}{|\Omega_m|} \sum_{i \in \Omega_m} \mathbf{o}_i. \tag{3.2.1}$$

Next, the Mahalanobis distance is calculated for a pedestrian sample:

$$d_{mh}(\mathbf{o}, \overline{\mathbf{o}}_m) = \sqrt{(\overline{\mathbf{o}}_m - \mathbf{o})^T V_m^{-1} (\overline{\mathbf{o}}_m - \mathbf{o})}. \tag{3.2.2}$$

In equation 3.2.2, the pedestrian sample is denoted by $\mathbf{o}$ and with V_m^{-1} being the inverse covariance matrix of the dataset Ω_m. The distance is then calculated for each pedestrian object in the *CityPersons* dataset and the weighting results are stored alongside the original ground truth annotations in the $\mathcal{D}_{CP}^{w}$ dataset.

This resulting dataset is then used to re-train the pedestrian detector with our modified loss, termed detection impairment weighting (DIW)-loss, which has been adopted to take the calibrated weights into account:

$$\begin{aligned} J^{total}(p, u, t^s, v) &= \alpha \cdot (J^{cls}(p, u) + \lambda[s = 1] J^{loc}(t^s, v)), \\ where\ \alpha &= \begin{cases} \frac{1}{1 + d_{mh}(\mathbf{o}_s, \overline{\mathbf{o}}_m)}, & \text{if } s = 1 \\ \gamma, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.2.3}$$

Here, the total training loss consists of the sum of the classification loss J^{cls} and the localization loss J_{loc} multiplied with the weighting term α. The classification loss is a categorical cross entropy loss with the inputs p denoting the predicted class probability and u denoting the ground truth target class. The localization loss is the $smooth_{L_1}$ loss from [Gir15] accumulated over all bounding box coordinate predictions t and bounding box

ground truths v. Additionally, λ is a parameter to weigh the contribution of classification and localization loss. Here, $\lambda = 1$ and $[s = 1]$ is the Iverson bracket which evaluates to 1 if the predicted class s is correct. In simpler terms, the localization regression loss is only applied if the classification prediction is a pedestrian. Our actual weighting of the samples is implemented in the α value which evaluates to values in the range $[0, 1]$ for the person class or to the parameter γ otherwise. The parameter γ can be used to tune the influence of the background class on the overall loss. The value of this parameter is usually set to $\gamma = 0.5$.

When training a real-world pedestrian detector with this DIW-loss we are able to improve the state-of-the-art in pedestrian detection on the *CityPersons* benchmark measured by the laMR metric, as defined in Equation 2.1.4. This is mainly achieved by strongly reducing the false positive rate and by slightly improving the true positive detections per image. Furthermore, we can evaluate the influence of each visual detection impairing factor on the weighting loss by an ablation experiment. The detailed results can be found in Publication 6 in Chapter 7.6.

Chapter 4

Validation of Visual Perception Functions

In this chapter we describe the methods we developed to validate visual perception functions for autonomous driving, using synthetic data. With the findings from the previous chapter we are able to generate and render more realistic synthetic data, suitable for validation of algorithms trained on real-world datasets.

The first method in this chapter is the variational data synthesis for perception validation. This approach shows how to generate parameterized realistic synthetic data with rich meta-data and use it to validate perception functions on the influence of object distributions, additive noise, and object occlusions. The method and the validation results are further described in the Publications 1 in Chapter 7.1 and 4 in Chapter 7.4.

The second part of this chapter describes our validation approach for training data bias detection by classification of detection impairment factors. Training data biases are prediction or classification biases that occur if the model insufficiently generalizes, e.g., if a model is trained to classify persons with a dataset consisting of the same person over and over again it will not be able to correctly classify unseen persons. The method utilizes visual detection impairing factors, introduced in Chapter 3.2, to calibrate a classifier that distinguishes if a person in the image is detectable or non-detectable. This classifier is then used to find training data biases in real-world pedestrian detection datasets. Furthermore, the overall influence of the visual detection impairment factors on the detectability of a person are analyzed. This method is further described in Publication 5 in Chapter 7.5.

4.1 Variational Deep Data Synthesis for Perception Validation

To validate visual perception functions we rely heavily on the data to test the correct purpose. For example, in validating a pedestrian detector for autonomous driving it is necessary to generate the safety critical scenarios where persons are very close to the ego-vehicle or children running on the street and not only the common street scenes found in most camera recorded real-world datasets. While the latter scenes are easy to capture and make up the majority of most datasets, the safety critical scenarios are hard to capture in the real-world due to the endangering of the recorded persons and the infrequent occurrences of such scenarios [KP16]. Even if these few events could be recorded, it would give us some fixed validation data without the means to modify the content of the images and further understand possible detection flaws. As an example: We found a miss detection of a child in front of the car at an early sunset. Would this miss detection occur if the child would stand in broad daylight? Recapture such a scene at different conditions would take a great amount of time and with no guarantee that this is even possible. This is the reason for the development of variational deep data synthesis approach for validation of visual perception functions. Here, we synthesize a sufficiently large amount of probabilistically parameterized scenes, including safety critical scenarios, and evaluate a perception function for perceptive flaws, such as miss detections of pedestrians. If found, we can reuse the scene in which the flaw occurred and synthesize the image again under different parameters, e.g., altering the time of day to produce different lighting conditions. This allows us to work out the underlying cause of the perception functions fault.

Figure 4.1 depicts the block diagram of our variational deep data synthesis approach, named *VALERIE*. The validation engineer starts with the scenario preparation. This is a description on how to variate the scene parameters, such as time of day or street width, and which assets from the asset database to use to synthesize the street scenes. The chosen assets are then fed into the data synthesis block, whereas the parameter variation description is fed to the *VALERIE* validation flow control. The validation

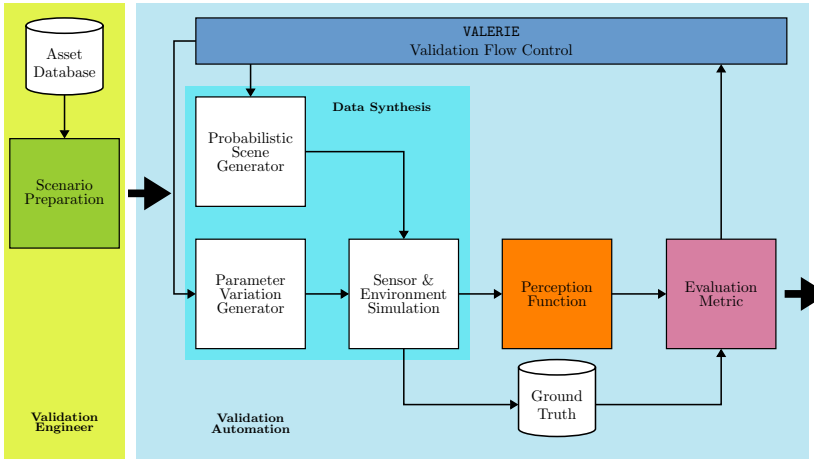


Figure 4.1. Block diagram of the proposed validation approach. (Source: Publication 4 in Chapter 7.4 [GHS22])

flow control then creates the different combinations of scene and parameter variations and sends them to the data synthesis as well. The data synthesis block, which we examine in more detail in the next subsection, now creates an autonomous driving street scene with a realistic sensor simulation. This is done in the first step by probabilistic generation of the street scene, i.e. street and building layouts as well as object and person placements. Next, the chosen parameter variation, such as time of day, according to the validation flow control is generated. Last, the image is synthesized in the render engine and a realistic sensor simulation according to our work in Chapter 3.1.2 is applied. The generated image is then forwarded to the perception function under test which creates an inference result for further evaluation. In the last step the evaluation metric on the detection result is calculated with the help of the ground truth from the render engine. This can be mIoU for the task of semantic segmentation or TP and FN values for pedestrian detection.

The evaluation metric result is stored for further evaluation but also

Figure 4.2. Example of scene parameter variation, in this case the time of the day is varied, causing dramatic changes in the scene illumination and according contrast variations. (Source: Publication 4 in Chapter 7.4 [GHS22])

sent to the validation flow control. The flow control sets the next scene and parameter variations to be generated in the data synthesis block. These variations can now either follow the ranges given by the scenario description or the previous scene is reused and recreated with a different set of parameters. This choice depends on the result of the evaluation metric. For example if a pedestrian detector evaluates for FNs, i.e. miss detections, the flow control chooses to recreate the scene under different lighting conditions to check if the miss detections were caused by difficult lighting conditions. Some examples of such time of day parameter variations are given in Figure 4.2. The process ends if all parameter combinations are exhausted. The evaluation metric results and the corresponding parameter variations per image are then carefully examined by the validation engineer to pinpoint reasons for the perception flaws as is demonstrated in Chapter 4.1.2.

4.1.1 Generation of Synthetic Validation Data

The generation of synthetic data in our variational deep data synthesis approach consists of three major components. The first component is the probabilistic scene generator. It first generates the ground layout of the automotive scene, i.e., the layout for streets, sidewalk and buildings. Next, the generator places three-dimensional objects or assets from the database on the previously defined layout. For example, cars are placed on the street or on parking spots, whereas persons are placed on the sidewalk and on the street. As the name of the component suggests the whole scene, including the object placement, is probabilistically generated, i.e., sampled, from a given range of parameters. These parameters are set by the validation flow control and define for example the min and max width of the street and sidewalks or the probability of a person being placed on the street.

The second component is the parameter variation generation. These parameters include, among others, the time of day, and geolocation settings. By changing the time of day parameter the scene lighting can change drastically. The variation generation is therefore the main instrument of the validation flow control to search for parameter combinations that can cause perception flaws.

The last component is the sensor & environment simulation. By utilizing Blender 6, which allows the importing, editing, and scripting of 3D content, the predefined scenes are then rendered with the physically based renderer *Cycles*. Subsequently, the realistic sensor simulation model derived in Chapter 3.1.2 is applied to the rendered image to recreate the sensor impression of a real-world recording. Additionally, metadata from the rendering process is extracted to generate the necessary data for an accurate ground truth annotation for a range of perception tasks. These task include semantic segmentation, instance segmentation, 2D & 3D bounding box detection and depth estimation from monocular images.

The continuous development of this synthetization method eventually resulted in the generation of the *VALERIE* dataset. This dataset is described in more detail in Chapter 5.1.

4.1.2 Validation Results

As the focus of this validation approach is on visual perception functions, we validated perception function models for the task of semantic segmentation and 2D bounding box pedestrian detection. The validated segmentation model is the `DeeplabV3+` model with a `ResNet101` backbone pre-trained on the *ImageNet* dataset. This model is fine-tuned on a range of real-world and synthetic datasets and validated utilizing our variational deep data synthesis method. For the 2D bounding box pedestrian detection task, the SSD model fine-tuned on tranche 3 of the synthetic *SynPeDS* dataset is validated. The feature extractor of the SSD model is a `ResNet50` backbone pre-trained on the *ImageNet* dataset.

To measure the model performance on the semantic segmentation task, the mIoU is calculated. The performance of the pedestrian detection is measured by calculating the true positive rate (TPR). The TPR, also known as the sensitivity, is calculated as the quotient of TPs over the sum of TPs and FNs. This measure allows filtering out all predictions on images with missed pedestrian objects, i.e., with TPR< 1.

In our work several influence factors on the perception function are validated. Beginning with the influence of the pedestrian distribution in the training data on the perception generalization. Therefore, we use the sequence 0058 of the *VALERIE* dataset, generated by our variational synthesis method, for training the segmentation model and compare the dataset's pedestrian distributions to other synthetic datasets and to the target dataset *Cityscapes*.

Next, the influence on the segmentation performance on images with additive Gaussian noise is evaluated. Here, the realistic sensor simulation as part of the data synthesis component allows us to simulate images with increasing additive Gaussian noise.

Last, we validate the pedestrian detector and the semantic segmentation model on the influence of different person occluder objects. The parameterization of the scene layout allows exchanging objects in front of the pedestrian in a scene and enables us to evaluate the respective influence of each object on the detection performance.

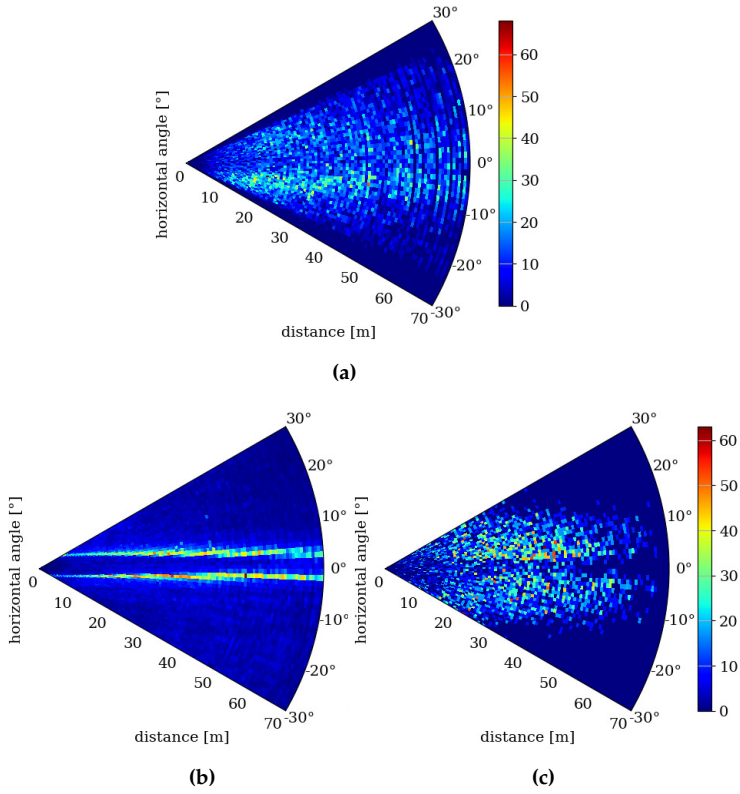


Figure 4.3. Pedestrian distribution over horizontal angle and distance. (a): *Cityscapes*. (b): *SynPeDS* Tranche 3. (c): *VALERIE* synthetic data. (Source: Publication 4 in Chapter 7.4 [GHS22])

Influence of Object Distributions

The object, i.e., person distribution of a dataset, was found in Chapter 3.1.3 to have a considerable influence on the cross-domain generalization quality when training with synthetic data. In Figure 4.3 the spatial distribution of pedestrians in the *Cityscapes* (a), *SynPeDS* tranche 3 (b), and *VALERIE* sequence 0058 (c) are shown. These visualizations represent histograms of persons placed in the datasets images at a distance to the camera in

$[m]$ and at a horizontal viewing angle of -30° to 30°, corresponding to the left and right side of the image. As previously discussed it is of advantage to recreate or match the spatial distribution of objects in the source synthetic domain to the one of the target real-world domain. The *Cityscapes* dataset is considered again as the target dataset. In this dataset the distributions of pedestrians show a uniform distribution with slight tendency to a positive horizontal angle. Looking into the *Cityscapes* dataset one notices the right-handed driving in all of these images with oncoming traffic occasionally occluding pedestrians on the left side, explaining the slight tendency of more visible pedestrians on the right side or positive horizontal angles. The *SynPeDS* tranche 3 distribution shows two sharp parallel lines where most of the datasets pedestrians are located on. This indicates a strong potential bias in the pedestrian distribution. We can strengthen this hypothesis by visual evaluation of perception results as shown in the last subsection of this chapter. The *VALERIE* Sequence 0058 distribution on the other hand shows a more uniform person distribution which better resembles the *Cityscapes* distribution. For both synthetic datasets one further interesting observation can be made. The number of pedestrians at further away distances of around $50m$ is significantly higher than the one of the *Cityscapes* dataset. This can be explained by the manual annotations of the real-world dataset. While human annotators have a hard time drawing bounding boxes for persons above such distances, due to the small size of the person, synthetically generated images deliver pixel perfect ground truth information at any distance and size.

Influence of Noise on Detection Performance

The influence of noise on the detection performance is evaluated by applying Gaussian noise with increasing variance on the input image, followed by subsequent segmentation inference and performance evaluation. In this experiment the `DeeplabV3+` segmentation model is trained on the real-world datasets *A2D2*, *Cityscapes*, and on the sequence 0058 of the synthetic *VALERIE* dataset. The variance σ of the Gaussian noise is continuously increased in steps of 1 in the range $\sigma^2 \in [0, 20]$. At every step the segmentation performance, mIoU, is measured for each of the three trained models. The resulting graph for this experiment is shown in Figure 4.4. The

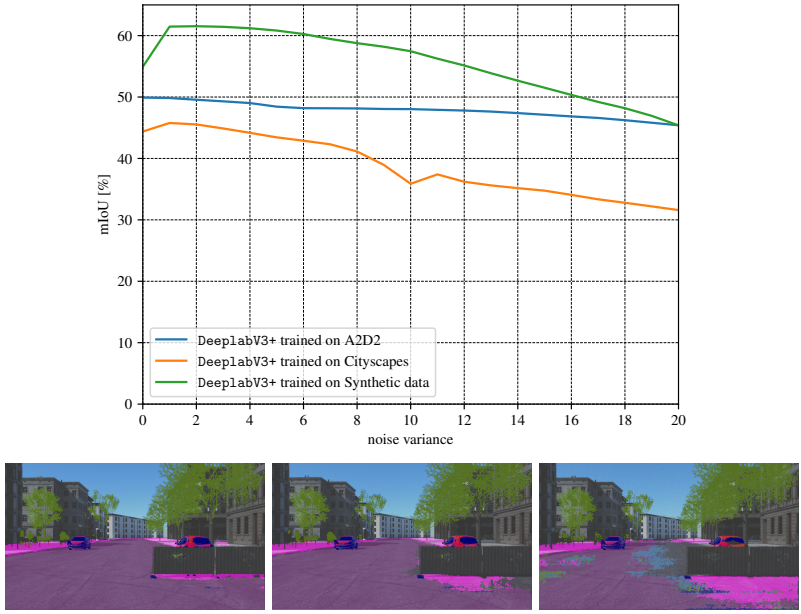


Figure 4.4. Top: mIoU performance decreases with increasing noise variance. Bottom (left to right): segmentation maps with increasing noise variance $\sigma^2 \in \{0, 10, 20\}$, image pixels $x_i \in [0, 255]$. (Source: Publication 4 in Chapter 7.4 [GHS22])

images below the plot are segmentation results of the *Cityscapes* trained model at σ^2 values of $\{0, 10, 20\}$. While the *VALERIE* and *Cityscapes* trained models increase in performance for small values of σ^2 the model trained on *A2D2* continuously declines in performance. The initial increase of the performance can be explained that both *Cityscapes* and the *VALERIE* exhibit noise in their respective training images similar to these values of Gaussian noise. For the *A2D2* dataset these values seem to mismatch with the noise in the training images as no initial increase of performance can be observed. The source of decline in performance is clearly visible in the prediction image with $\sigma^2 = 20$. Here, object boundaries begin to fray and the segmentation prediction starts to smear out accross classes.

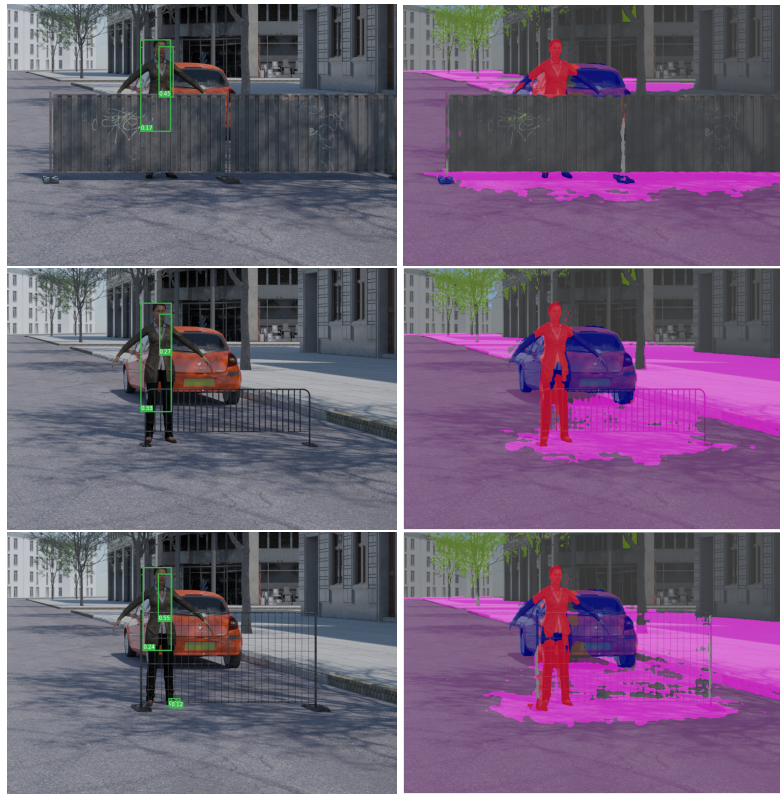


Figure 4.5. Scene with variation of occluding objects. Left: 2D bounding box detection. Right: semantic segmentation (Source: Publication 4 in Chapter 7.4 [GHS22])

Influence of Occluder Objects

To understand the influence of the occluder object on the pedestrian detection performance we utilized our scene generator to create images of a pedestrian on the street with different occluders in front of it. Next, the SSD and the `DeeplabV3+` semantic segmentation model are used to compute the inference on these images. The resulting predictions are shown in Figure 4.5. For all three variations of the occluder object, the SSD predicts two

partial bounding boxes. One of those predictions always covers the upper half of the person with low confidence, whereas the other prediction covers a smaller part of the upper half with higher confidence in two of the three images. Even though the occlusions of the person differ significantly, the visibility of the upper half is enough to correctly determine the presence of a pedestrian. Another observation is that the boxes do not cover the spread out arms of the person, which hints towards missing pedestrian poses in the training data or even missing bounding box anchor scales in the SSD model. In the segmentation prediction the conclusions differ understandably. While the person is correctly determined to be present in the images and correctly classified as person in all images, the occluder in front of the person has only little influence on the prediction. Again the spread out arms of the person are not correctly detected, emphasizing the hypothesis of missing training data because both detection models were trained on the same synthetic dataset with no such poses included. Additionally, another detection fault is visible in the segmentation prediction of the person. The road below the pedestrian is in all images classified as sidewalk even though the remaining road is correctly classified. This suggests another training data bias. Here, the training data did not include enough pedestrians standing on the road, but instead the pedestrians were placed on the sidewalk in most of the training images. The segmentation model was trained on the tranche 3 of the *SynPeDS* dataset. Re-examining the pedestrian distributions in this dataset in Figure 4.3 b), strengthens this hypothesis. The sharp person distribution stems from pedestrians only placed on the sidewalk and none placed on the streets.

4.2 Classification of Visual Detection Impairing Factors

In this section, we describe a validation approach for 2D bounding box detectors based on the previously in Chapter 3.2 introduced visual detection impairing factors. This method allows detecting training data biases of pedestrian detectors, such as missing ethnicity or poses. Furthermore, in this result we present additional evidence for the actual object detection performance influence of the visual detection impairing factors. This

method is described in full detail in Publication 5 in Chapter 7.5.

4.2.1 Data Bias Detection

By definition, the visual detection impairing factors allow separating the set of objects in the dataset, i.e. pedestrians, into *detectable* and *non-detectable* subsets. Because the impairment factors define the border between *detectable* and *non-detectable* pedestrians. We utilize this observation and train a binary classifier to learn this exact border function. Applying this classifier on the impairing factors extracted from a validation dataset and comparing the classification result with the predictions from a pedestrian detector under test enable us to validate the training data bias of the detector. Especially, if the classifier is carefully trained and predicts the pedestrian to be *detectable*, but the detector does not detect this pedestrian, then this is a strong evidence on an existing data bias of the training data.

Starting by training of the binary classifier. Figure 4.6 depicts the training workflow. To create a training dataset for the classifier we first distinguish the persons of a synthetic training dataset into the *detectable* and the *non-detectable* subsets. This is done by predicting the pedestrian bounding boxes in this synthetic training dataset with a pedestrian detector trained on the same synthetic domain. For the synthetic training dataset $\mathcal{D}_{Synth}^{train}$, we use a subset of the *VALERIE* sequence 0060. The pedestrian detector is trained on the *VALERIE* sequence 0058. The pedestrian detector is a Cascade R-CNN model with a `HRNet` backbone. To generate the necessary data to train the binary classifier we extract the visual detection impairment factors from the training dataset. The visual impairment factors used in this method are introduced in Chapter 3.2. In the next step, the prediction results and the extracted impairment factors are forwarded to the *Accumulate Results* component. Here, the split between *detectable* and *non-detectable* class is carried out. Every pedestrian in the ground truth without a successful prediction, i.e. $\text{IoU} < 0.5$, is allocated to the *non-detectable* class ($s = 0$) or to the *detectable* class ($s = 1$) otherwise. The results are stored as ground truth data $\mathcal{S}$ and the extracted impairment factors on the other hand are stored into the training dataset Ω as input to train the classifier.

With the input Ω and ground truth $\mathcal{S}$, the binary classifier is subse-

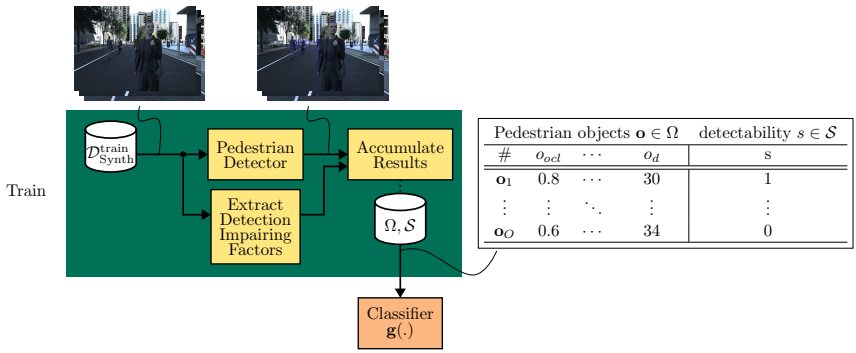


Figure 4.6. Training a classifier to distinguish between detectable and non-detectable pedestrian objects. (Source: Publication 5 in Chapter 7.5 [GHS22])

quently trained. The binary classifier is a multi layer perceptron with 4 hidden layers and 20 nodes per layer. As loss function the binary-cross entropy is used. The training data is split into 80% training and 20% validation data. After training, the resulting classifier reaches a F1 score of 0.93 on the validation data.

With the successful trained classifier, it is then used to validate a real-world pedestrian detector. The validation workflow is shown in Figure 4.7.

Again, we start with a synthetic dataset $\mathcal{D}^{val}_{Synth}$. This dataset is the remaining subset of the *VALERIE* sequence 0060 which is now used for validation. The pedestrian detector under test predicts on this dataset while in parallel the visual detection impairing factors are extracted. The detectors under test are two Cascade R-CNN models with `HRNet` backbones each. The first model is trained on the *CityPersons* and the second one on the *EuroCity Persons* dataset. In the next step, the inference and extraction results are accumulated and forwarded to the previously trained binary classifier. The classification results, the pedestrian prediction results and additional metadata $\mathcal{D}^{val}_{Meta}$ from the input dataset $\mathcal{D}^{val}_{Synth}$ are then stored in the result dataset $\mathcal{D}^{val_cl}_{Synth}$. The additional metadata originates from the asset database used to synthesize the validation dataset and includes additional information per pedestrian object, such as the geolocation, the pose, the

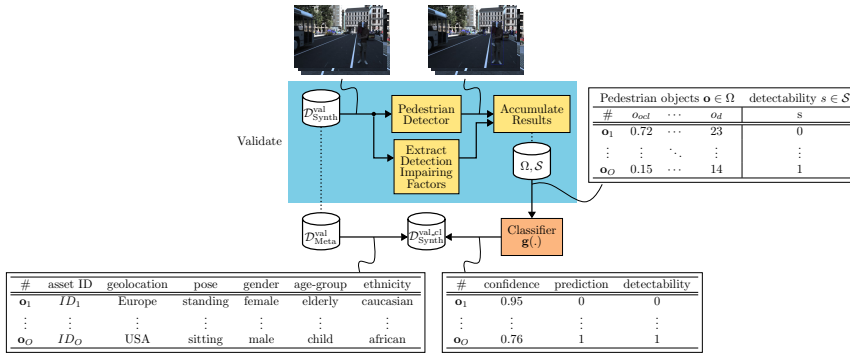


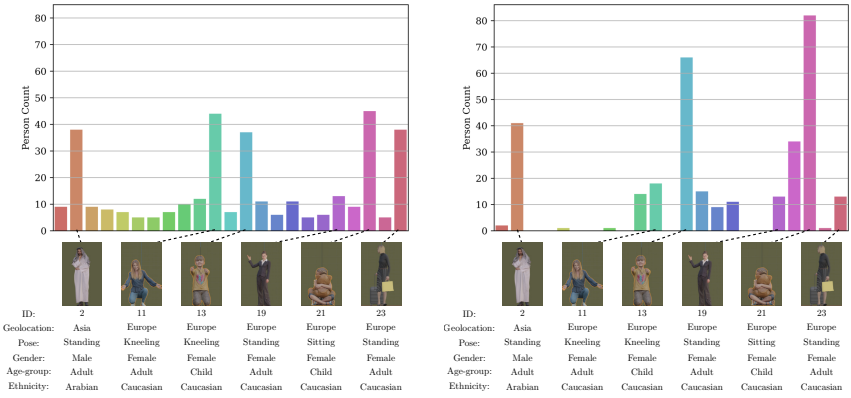
Figure 4.7. Generation of validation data to detect data biases in the pedestrian detector. (Source: Publication 5 in Chapter 7.6 [HG23])

gender, the age-group, and the ethnicity. This metadata is essential to understand the underlying training data biases.

Before we can validate the pedestrian detector under test we first have to filter the result dataset $\mathcal{D}_{Synth}^{val_c^l}$. Only the pedestrians where the classifier predicted the *detectable* class and the pedestrian detector missed the person are relevant for our validation. The filtered dataset now contains only pedestrian objects where miss-detections occurred due to a training data bias, i.e., due to missing training data.

Figure 4.8 shows the histogram of these found training data biases with a pedestrian detector trained on the *CityPersons* (a) dataset and one trained on the *EuroCity Persons* (b) dataset.

Directly comparing the found 23 *CityPersons* to the found 15 *EuroCity Persons* dataset biases the *EuroCity Persons* dataset has an overall lower training data bias. The *CityPersons* trained detector is heavily influenced by the geolocation and ethnicity of the validation data and shows its significant influence of the central European training data. This is especially noticeable if the person wears uncommon non-European clothes as for the person ID 2 which is clothed in traditional Arabian clothing. The *EuroCity Persons* trained detector shows a similar sensitivity to the non-European clothed pedestrians ID 2 which is even more often not detected. Another observation is that both datasets are miss-detecting more often if the



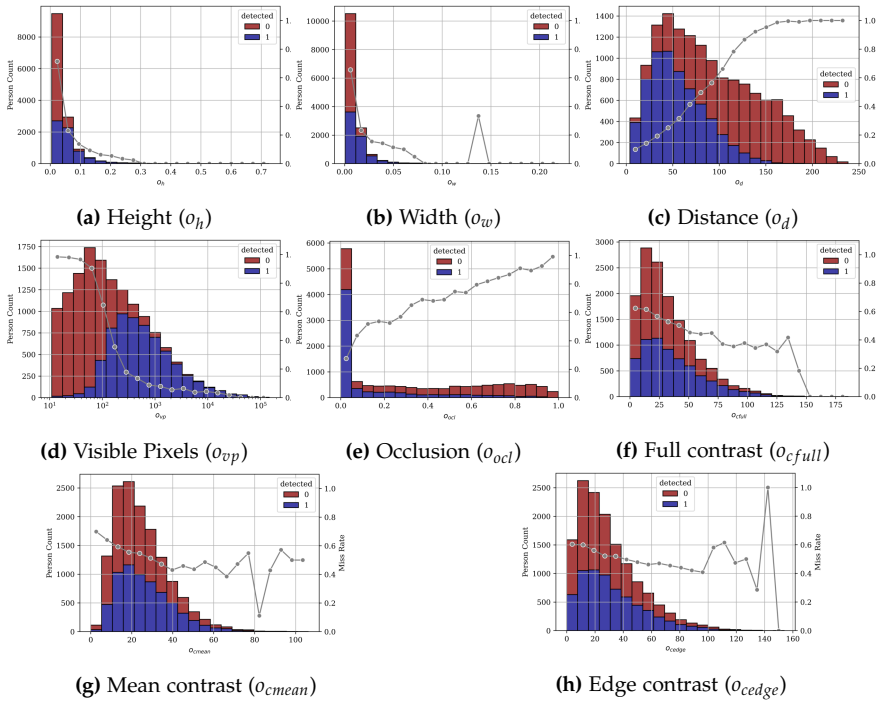
(a) CityPersons **(b)** EuroCity Persons

Figure 4.8. Distribution of found data biases, i.e., miss-detected pedestrians. (Source: Publication 5 in Chapter 7.6 [HG23])

person is sitting or kneeling as can be seen with the IDs 11, 13, and 21. These miss-detections are even worse for the child age-group with sitting pose. Especially this pedestrian group has a very high urgency of detection and even higher if located on the road. IDs 19 and 23 are miss-detected by both detectors although having a European ethnicity and a standing pose, but both these assets have rather uncommon gestures. In Publication 5 in Chapter 7.5 we further discuss these results and additionally enhance the evaluation to automatically detect and extract pedestrian pose biases.

4.2.2 Performance Influence of Visual Detection Impairing Factors

An analysis of the performance influence of visual detection impairing factors was done in Publication 6 in Chapter 7.6. In this work the ablation study results suggested that the highest influence on the detectability of a person are the number of visible pixels. We can now take the accumulated result of the classifier training, i.e., the training dataset Ω and the ground truth dataset $\mathcal{S}$, and visualize each individual impairment factor as a



(a) Height (o_h) **(b)** Width (o_w) **(c)** Distance (o_d)

(d) Visible Pixels (o_{vp}) **(e)** Occlusion (o_{ocl}) **(f)** Full contrast (o_{cfull})

(g) Mean contrast (o_{cmean}) **(h)** Edge contrast (o_{cedge})

Figure 4.9. Influence of visual impairing factors of a pedestrian on the detection performance, i.e. miss rate (gray). (Source: Publication 5 in Chapter 7.6 [HG23])

histogram for *detected* (1) and *non-detected* (0) persons. Additionally, we plot the MR, as defined in Equation 2.1.2, per bin into each histogram plot. Figure 4.9 shows the resulting histograms. The influence of the size category, i.e., height, width, distance and visible pixels is clearly visible on the detection performance. When height, width, and visible pixels values increase, then the MR decreases. For higher distances which relate to smaller pedestrian sizes, the MR decreases. These are clear indicators that the size of the pedestrian object in the image is of high importance for the detectability of a person and again emphasizes the importance of the number of visible pixels count factor. The occlusion factor shows a clear increasing MR tendency for increasing occlusion values, intuitively

explained by lesser visible pixels leading to worse detection probabilities. For the contrast measures the result is not as expressive, but for each defined contrast measure the MR continuously declines with occasional outliers at very high contrast values.

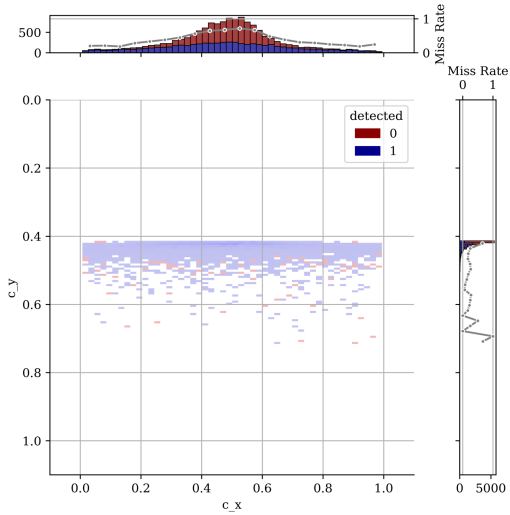


Figure 4.10. Influence of person placement in the image on the detection performance measured as miss rate (gray). (Source: Publication 5 in Chapter 7.6 [HG23])

The remaining impairment factors, i.e., the vertical and horizontal center coordinates of the 2D bounding boxes, are visualized as a 2D heatmap in Figure 4.10. Again the MR is plotted into the marginal histogram distributions for the horizontal (above) and vertical (right) center point position. The MR for the horizontal placement increases for values around the center but decreases to a fixed value at the outer sides. Calculating the Spearman correlation for the MR and the horizontal placement results in a value of 0.017, indicating no clear influence of this factor on the detection performance. Evaluating the influence of the vertical placement by examining the corresponding MR graph is not conclusively possible. As is evident for values above 0.6 there are only few samples present and for values below 0.4 are no samples present. This is a consequence of the fixed viewing

angle of the *VALERIE* sequence 0060 and the planar street scenes with no vertical displacement of the camera. While the result of the influence of this vertical placement factor remains inconclusive a valuable feedback for future synthetic data generation could be gathered, i.e., to consider additional camera viewing angles.

Chapter 5

Validation Datasets

This chapter describes the synthetic validation datasets *VALERIE* and the synthetic pedestrian dataset (*SynPeDS*). Extensive parts of the findings in this work were used to generate, develop and refine these datasets.

The *VALERIE* dataset sequences were produced by continuous development and generation of synthetic validation data from the previously described deep variational data synthesis approach (Chapter 4.1). Several publications in this work are based on the usage of this dataset, i.e., Publication 4 in Chapter 7.4, Publication 5 in Chapter 7.5, and Publication 6 in Chapter 7.6. This dataset is at the time of writing not yet released to the public, but will be released to help the community of automotive computer vision researchers to research new methods or refine existing ones for training and validation of visual perception functions with synthetic data.

The *SynPeDS* dataset is one of the major results of the KI-Absicherung project. This dataset was developed to help the community of validation researchers and engineers in academia and industry to further the development on safe perception functions in autonomous driving. Results of our work for training with synthetic data were used to increase the realism and thus following the usability of the dataset. Additionally, several publications in this work are based on the usage of this dataset, i.e., Publication 2 in Chapter 7.2, Publication 3 in Chapter 7.3, and Publication 7 in Chapter 7.7.

5.1 VALERIE

The *VALERIE* dataset is a product of our deep variational data synthesis validation method. For validating of a visual perception function a signifi-

cant amount of image and ground truth data was generated. This data was gathered and stored into the respective *VALERIE* sequences which are used throughout this work. With increased understanding and disentanglement of the relevant influence factors on the synthetic to real-world domain gap the gained knowledge was used to improve the quality of the data generation process. Each sequence therefore corresponds to a stage in the development of the data synthesis pipeline and exhibits distinctive features differentiating them from one another. The most distinctive features of each sequence are described in Table 5.1.

With increasing development efforts and increased understanding of the intricacies of influence factors on the domain gap, the structural complexity from sequence to sequence was increased. Structural complexity can hereby be understood as the amalgamation of scene layout, object placements, asset diversity, environment, and sensor simulation. As was shown in Chapter 3.1.3, the best domain generalization performance on the *Cityscapes* dataset was achieved with the combined dataset of sequences 0054 to 0060. We found that due to the much higher number of frames and lower diversity of this sequence the cross-domain performance after training is reduced. The lower diversity is owed to the fact that this sequence reuses scenes and re-renders them with different lighting conditions. While these scenes are important to be incorporated in a validation dataset they would add a heavy training bias if they are directly included into the training dataset. A viable option for the usage of the 0062 sequence in training would be to sub-sample the data.

Exemplary images from the sequences 0058 and 0060 of the *VALERIE* dataset are shown in Figure 5.1.

As described in the Chapter 4.1.1, the dataset includes ground truth annotation data for several perception tasks from monocular images. These tasks include semantic segmentation, instance segmentation, 2D bounding box detection, 3D bounding box detection, and depth estimation. Additionally, for every sequence and every image a scene description file includes all meta information from the data synthesis process. These meta information include the time of day, the definition, direction and position of assets, the definition of the camera and its position in the scene, the visual detection impairing factors per pedestrian, and the overall geolocation position. Especially the meta information, as shown with the visual

Table 5.1. Characteristics of each sequence generated at 48.18° N, 11.58° E in the *VALERIE* 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

perception impairing factors, are important for a validation method to better understand and draw conclusions from found perception faults.

The *VALERIE* dataset is planned to be released to the research community to increase the efforts for validating visual perception functions and hopefully increase the safety in autonomous driving as a whole.

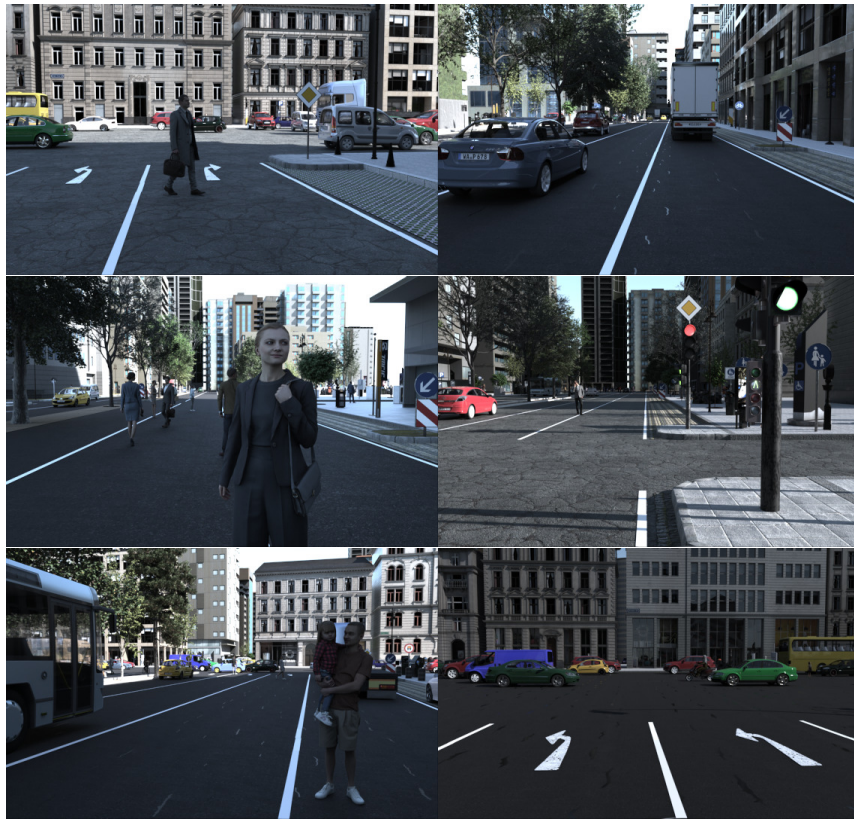


Figure 5.1. Our fully parameterizable generation pipeline allows rendering pedestrians at any size, occlusion, time of day, and distance to the camera. (Source: Publication 5 in Chapter 7.6 [HG23])

5.2 SynPeDS

The *SynPeDS* dataset is one of the key results of the KI-Absicherung project. This project and the resulting dataset were a collaborative effort of 28 partners in technology and academia in Germany to set the foundation of research on safeguarding autonomous driving perception functions. The dataset rendering was done by three different partners: Mackevision,

BIT-TS, and Intel. Whereas the first partner implemented the image rendering process in the real-time game engine Unreal [Gam], the other partners implemented the rendering as physically based rendering pipeline. BIT-TS implemented their physically based rendering with the Blender [Fou] render engine whereas Intel used the OSPray [Int] render engine. Because these pipelines where developed independently by each data producing partner, some implementations of ground truth annotations and meta information differ. The dataset is split into 9 different tranches with distinctive features added for each tranche similar to the sequences in the *VALERIE* dataset. The added features per tranche and for each rendering pipeline are listed in Table 5.2.

Table 5.2. Features added in *SynPeDS* dataset per data tranche and data pipeline (physical-based rendering (PBR), real-time engine (RT)). (Source: Publication 7 in Chapter 7.7 [SBF+22])

Tranche	New Features	PBR	RT
1, 2, 3	Preparation for large-scale data production	×	×
4	Frame-to-frame variations	×	×
	Meta information on AssetIDs	×	×
	Bodypart segmentation	×	
	Procedural sun model		×
5	Sensor noise as post-processing	×	
	Procedural clouds model		×
	Ground truth for pose estimation		×
	Meta information on occlusion		×
6	Environmental effects: wetness and sun glare		×
	Out-of-distribution assets		×
	Variations of camera sensor parameters		×
7	Camera and LiDAR sensor models using PBR with OSPRay and different LiDAR sensor parameters	×	
	Meta information on AnimationID		×
	Environmental effects: fog, vignetting		×
8	Night scenes with artificial light		×
9	Specific user requests for contrast or material		×

As with the *VALERIE* dataset, the *SynPeDS* dataset could take advantage from the findings and disentanglement of domain gap factors. Improvements from tranche 1 to the subsequent tranches are for example the addition of pedestrian assets due to findings of the cross-domain generalization in Chapter 3.1.3 or the application of our sensor simulation on the image data due to findings from Chapter 3.1.2. The analysis of the domain generalization performance and the influence of pedestrian assets on the person class domain generalization capability have been published in conjunction to this dataset in Publication 7 in Chapter 7.7.

The *SynPeDS* dataset includes ground truth annotation data for a range of visual perception tasks. These tasks include instance segmentation, semantic segmentation, bodypart segmentation, 2d bounding box detection, 3d bounding box detection, depth estimation, and pose estimation. Unlike the *VALERIE* dataset these tasks are not limited to monocular camera image perception. Starting at tranche 7 a realistic LiDAR simulation was added and can be used as input for these perception tasks. The meta information per image is similar to the meta information of the *VALERIE* dataset but varies strongly throughout the tranches and is very limited for earlier tranches, i.e. tranche 1 to 3.

The dataset is available through an industry friendly licensing model. This allows not only academic researchers but also the industry to use the dataset for research and especially development of new perception validation methods. This dataset is the first synthetic dataset with such an open licensing model.

Chapter 6

Conclusions

This work focuses on the usage of synthetically generated realistic sensor images for training and validation of visual perception functions for autonomous driving applications. The main contributions are divided into three topics: Training with synthetic images, validation with synthetic images, and the characterization of synthetic datasets for training and validation.

Training a visual perception function solely on synthetically generated imagery and applying this function on real-world data poses the problem of how to overcome the domain gap. We investigated several domain distance and discrepancy measures and found that a major shortcoming is that these measures do not predict the actual target domain, i.e., generalization, performance. Therefore, we introduced a new earth movers distance (EMD) performance based domain discrepancy measure which directly correlates to the expected performance on the target dataset. Furthermore, we showed that we can utilize this EMD measure as optimization loss to optimize the parameters of a realistic sensor simulation on the synthetic data and achieve a reduced domain discrepancy to the target *Cityscapes* dataset. With help of the metadata that comes along the synthetically generated imagery we were able to entangle several influence factors on the remaining domain gap. These factors include, the number of training assets in the synthetic data, the number of training frames, and the scene structure measured through the sliced Wasserstein distance (SWD) of segmentation heatmaps. Additionally, we introduced the visual detection impairment factors. We showed through several experiments that these factors have a significant influence on the detection performance of a pedestrian detector. Extraction of the impairment factors from source and target datasets allows us for example to detect missing training data in

the synthetic images. Last, we utilize the impairment factors to calibrate the training loss of a real-world pedestrian detector on harder to detect samples. With this approach we improve state-of-the-art performance on pedestrian detection.

The main tasks for validation of a perception are semantic segmentation and 2D bounding box detection, especially with a focus on pedestrian detection. In our work we introduce a method named variational deep data synthesis for perception validation. This method introduces a fully parameterizable data generation and validation approach for a perception function under test. Synthetic images are hereby generated with regard to the findings of the domain gap factors. Hereby we can effectively reduce the domain discrepancy and guarantee that found perception flaws stem from the perception model or from the training data but not from mismatched domains. We showed how the pedestrian placement during training, the image noise, and the type of the occluding object during inference influence the performance of segmentation and 2D detection functions. Furthermore, a new method to detect training data biases of pedestrian detectors is derived. This method utilizes the visual detection impairment factors and trains a classifier to distinguish between *detectable* and *non-detectable* pedestrians by their respective impairment factors. If classification and the actual detection result differ then a training data bias is present. Further, we show that the visual detection impairment factors have significant influence on the detection performance except for the pedestrian placement in the image.

The preceding findings of training and validating visual perception functions for autonomous driving influenced the creation process of two synthetic validation datasets. First, the *VALERIE* dataset which is a product of the data generation process of the deep variational data synthesis method. This dataset consists of physically based rendered images in 8 sequences which were continuously improved by findings from the domain distance entanglement results. Second, the *SynPeDS* dataset which resulted from the collaborative KI-Absicherung project. This dataset consists of 9 tranches with images both rendered in a physically based renderer and a real-time engine. Similar to the first dataset, this dataset benefited from the findings of the domain distance entanglement experiments and was continuously improved in quality.

Overall, this thesis improves the understanding in using synthetic data for training as well as for validation of visual perception functions. Yet, there are several domain gap influence factors still to be found and analyzed. For validation methods this work is part of the initial ignition of a branch of research with high relevancy for safe autonomous perception functions and safe AI in general.

Chapter 7

Publications

7.1 Publication 1

DNN Analysis through Synthetic Data Variation

Qutub Syed Sha, Oliver Grau and Korbinian Hagn

Published in

2020 Proceedings ACM Computer Science in Cars Symposium. [SGH20]

Reprinted with permission from Qutub Sayed Sha

DOI: 10.1145/3385958.3430479

DNN Analysis through Synthetic Data Variation

Qutub Syed Sha
syed.qutub@intel.com
Intel Deutschland GmbH
Neubiberg, Germany

Oliver Grau
Oliver.Grau@intel.com
Intel Deutschland GmbH
Neubiberg, Germany

Korbinian Hagn
Korbinian.Hagn@intel.com
Intel Deutschland GmbH
Neubiberg, Germany

Figure 1: Rendered image of an urban scene with pedestrian.

ABSTRACT

This contribution discusses the use of variational data synthesis as a tool to analyze and understand limitations of performance of DNNs (deep neural networks) in perception tasks. To date, no universally accepted methodologies for validating ML (Machine Learning) -based perception exist. Instead of aiming for the randomized acquisition of huge amounts of validation data, either from real world capture or from simulation, we propose a guided concept to analyze perception performance using systematic parameter variations.

The concept is based on parameterized, generative content used for data synthesis in our validation engine. The latter is composed of the actual data synthesis module, automated execution and evaluation of the perception function under test and a control module, which allows specification of parameter variation towards a validation goal. Further we investigate the use of physical parameters, like object occlusion rates and pixel area for the identification of critical cases for perception. We present experiments for semantic segmentation of pedestrians in an urban environment using two different DNN algorithms.

CSCS '20, December 2, 2020, Feldkirchen, Germany

ACM ISBN 978-1-4503-7621-1/20/06…$15.00
https://doi.org/10.1145/3385958.3430479

CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**; *Rendering*; *Model verification and validation*; Shape modeling; • **Software and its engineering** → Software creation and management.

KEYWORDS

datasets, neural networks, validation, rendering

ACM Reference Format:
Qutub Syed Sha, Oliver Grau, and Korbinian Hagn. 2020. DNN Analysis through Synthetic Data Variation. In *Computer Science in Cars Symposium (CSCS '20), December 2, 2020, Feldkirchen, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3385958.3430479

1 INTRODUCTION

Perceptional functions have been a core research topic of computer vision and artificial intelligence over the decades. 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. 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.

To date, no universally accepted methodologies for implementing and validating safety exist. While traditional functional safety standards such as ISO26262 focus on technical safety (i.e. faults do not cause a malfunction that threatens safety), it is clear that the overall behavior of a vehicle needs to be validated for safety beyond just absence of faults. This extended concept is also known as 'safety of intended functionality'. Validation of this will play

an essential building block in the productization and successful marketization of AV technology.

A typical validation strategy is to test the AV system under real conditions; assumptions vary, but estimate that an AV needs to be tested with millions of km (240 Mio km [Kalra and Paddock 2016; Wachenfeld and Winner 2015]) and some models even estimate up to $10^9 km$ [Shalev-Shwartz et al. 2017]. Since the system needs to be validated with every major software and hardware release, validation needs to be implemented in the data center with simulated and captured sensor data. The automotive industry is currently investigating these approaches, but a major bottleneck yet to be solved is to scale up validation concepts into data center and cloud and to come up with more efficient validation methods than just plain 'trying' of scenario catalogs.

Perception has been recognized as one of the hardest problems to solve in any automated system. This paper suggests a computational approach for the validation of perception functions based on synthetic data generation.

This paper describes our underlying parameterized generative approach of the possible scenario space (we call this validation parameter space) and its use in our validation engine. We then present several experiments to quantitatively demonstrate the importance of factors like occlusion rate and object size on the performance of DNNs in the important task of pedestrian segmentation in urban settings.

The remainder of this paper is structured as follows: The next section will give an outline of related work in the field. In section 3 we give an overview of our approach. Section 4 describes our parameterization and introduces the concept of validation parameter space. In section 5 we give an outline of our validation engine, including a realistic sensor simulation and effective computation of the required variations. The paper finishes with a description of experiments and concluding remarks. The experiments show examples of parameter variation and how these influence the performance of the DNNs under test.

2 RELATED WORK

Techniques to capture and render models of the real world have been matured significantly over the last decades [Magnor et al. 2015] . We are now able to synthesize virtual scenes in a visual quality that is hard to distinguish from real photographs for human observers. On the other hand, we have emerging complex technical and particular autonomous systems sensing the real world and aiming at resembling some perceptional tasks formerly only feasible by humans. Because of the complexity of reality and the related increasing complexity of the technical tasks, validation, that makes sure these systems work as intended and are safe are increasingly important. 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. One prominent example in this context is the validation of automated driving.

The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g. [Burger and Barth 1995]). One advantage of synthetic data is that metadata and in the particular ground truth, data can be generated alongside the simulated sensor data; enabling a completely automated evaluation. Further, parameters and experiments can be varied to any extend, allowing systematic validation concepts: We make use of this option in our *computational validation* approach. Finally, in safety-critical applications, like AD, rare and potentially dangerous situations can be simulated without threatening (real) humans. An additional non-technical benefit is, that synthetic data does not raise concerns over the privacy of individuals from the public, as it is the case for captured real data sets.

The virtual simulation systems are designed to test a complete AD system, e.g. there are interfaces to control the ego-motion of a virtual vehicle. The current approach in validation is either to simulate a large number of test miles (or km) in the virtual world provided with the simulator; several commercial options exist [1]. Another virtual simulator system, which gained popularity in the research community is CARLA [Dosovitskiy et al. 2017].

One problem is that the test data or tested routes should include all possible conditions that lead to problems. The space of conditions or parameters which need to be considered for testing is vast. One concept to concentrate or compress this space is to formulate and test catalogs of scenarios [Menzel et al. 2018]. This is currently assembled by experts, like in the Pegasus project [Consortium 2020]. Neither a randomized nor a catalog approach makes currently use of automated methods or more sophisticated mathematical models to meaningful sample and search this space.

The approach presented in this paper to synthesize validation data that is designed to systematically test conditions and parameters that are relevant for validation of the *perceptional function* under consideration in a structured way.

3 OVERVIEW

Validation using data synthesis is generally based on a fixed catalog of test data: the test content is passed to the sensor and environment simulation. That is typically a rendering process and can be achieved with computer graphic methods in the case of visual sensors and extended methods for other sensors. This step is called sensor and environment simulation.

The test content is providing detailed information for this process. That includes a description of the scene, with all static and dynamic objects, their material properties and the light sources or other relevant active sources of energy for the sensors, also included in the description. The rendering process is then simulating the sensor impressions from these parameters. Further, ground truth data is provided through the rendering process. This can be for example pixel-accurate label information that gives the information about what class of object is attached to that pixel. This is useful for training and evaluation of semantic segmentation (see section 5.2).

Fig. 2 shows our approach of computational validation. First, the content is not directly passed to the sensor and environment module, as it is not 'linear content', but contains a generic description of the scenario with parameters that can be varied. The concept of

[1] For example Carmaker from IPG or PreScan from TASS International, now a Siemens company.

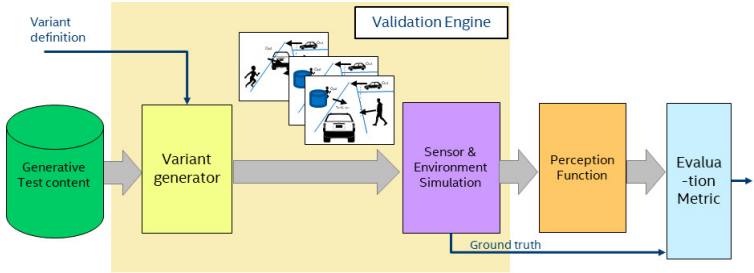


Figure 2: Our computational validation approach.

this generic description is described in the next section. The generic content is used to generate a variation of the scene under the control of our validation engine ('VALERIE'). The specific instantiated variant is then injected into the sensor and environment simulation, followed by perception and evaluation function.

The results of the evaluation will be typically used to inform the next iteration of variations by a V&V (Verification and Validation) engineer in a guided approach. A brief outline of the approach will be discussed in section 5.3.

4 GENERATIVE CONTENT AND VALIDATION PARAMETER SPACE

A central aspect of our validation approach is to parametrize variations of scene, sensor and activity states in an unified validation parameter space (VPS). In computer graphic a scene is considered as a collection of objects O_i and these are usually organized in scene graphs (see e.g. [Wernecke 1994]) and this model is also basis for file format specifications to exchange 3D models and scenes, e.g. VRML[2] or glTF[3].

Each object in this graph can have a position and orientation and scale in a (world) coordinate system. These are usually combined into a transformation matrix T. Several parameterizations for position and orientations are possible, for the position usually a Cartesian 3-vector, orientation includes notations like Euler-angles or quaternions.

Objects O_i are described as geometry, e.g. as a triangular mesh and appearance (material).

Sensors, like a camera, can also be represented in a scene graph and so can be light sources. Both also have a position and orientation and the same transformation matrices like for objects can be applied (except scaling).

Objects in a scene graph can be manipulated by considering their properties or attributes as a list of variable parameters. Table 1 gives a qualitative overview of those parameters. Most attributes are of geometrical nature, but also materials or properties of light sources can be varied.

[2]ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2004 https://www.web3d.org
[3]www.khronos.org/gltf

In addition to static properties, a scene graph can include object properties that vary over time. Table 1 includes already some of them, like the trajectories of objects and sensors, indicated as $T(t)$. Computer graphic systems handle these temporal variations as animations and in principle, any attribute can be varied over time by these systems.

We introduce an important restriction in the current implementation of our validation and simulation engine: Our animations are fixed, i.e. they do not change during run-time of the simulation. This could be different for example when a completely autonomous system is simulated, as the actions of the system might change the way other scene agents react. We will include these aspects in the discussion and outlook and how it could be mitigated.

For the use in our validation engine, as described in the next section, we augment a description of the scene in a scene graph (the asset) as outlined above, with an explicit description of parameters that are variable in a validation run. Currently, our engine considers a list of numerical parameters with the following attributes:

```
parameter_name, scene_graph_ref, type, minimum, maximum
```

A concrete example to describe variations of the position of a person in a 2-D plane in pseudo markup notation is:

```
{ { p1, scene.person-1.pos.x, FLOAT, 0.0, 20.0} ,
  { p2, scene.person-1.pos.y, FLOAT, 0.0, 10.0} }
```

with $p1$ and $p2$ being unique parameter identifiers, $scene.person-1.pos.x + scene.person-1.pos.y$ refer to the variable attributes in the scene graph. *FLOAT* denotes the parameter type to be a floating-point number. The last two arguments specify the parameter range [0.0..20.0].

4.0.1 Specification of variable validation parameter. The variant generator in approach depends on the provision of a generative scene model. This consists of a 3D scene model (also called a 3D asset), consisting of the static and dynamic objects. On top of this, we define variable parameters in this scene as an explicit list, as explained in section 4.

Table 1: Overview of parameters to vary in a scene.

Object class	Variable parameters
Static objects, e.g. Houses	Limited (position + orientation + size)
Streets, roads	Geometry (e.g. position, Size of lanes, etc.), friction (as function of weather conditions)
Vehicles	T_v =(Position, orientation), trajectory $T_v(t)$
Humans (pedestrian)	T_p = (*Position*, *orientation*), trajectory $T_p(t)$
Environment	Light, weather conditions
Sensors	T_s =(Position, orientation), trajectory $T_s(t)$, sensor attributes

For the specification of a validation run all or a subset of these parameters are selected and a range and sampling distribution for that specific parameter is added. For example, to vary the x-position of a person in the scene along a line with the homogeneous distribution and a step-size of 1m we define:

```
{ { p1, HOMOGENEOUS, 1.5, 5.5, 1.0 },
  { p2, HOMOGENEOUS, 4.0, 5.0, 1.0 } }
```

The parameters refer to the following: $p1$ and $p2$ refer to parameter declarations of x and y position of $person - 1$ in the example of section 4. *HOMOGENEOUS* refers to a homogeneous sampling distribution. Other modes include *GAUSSIAN*. The parameters 1.5, 5.5, 1.0 define to the parameter range [1.5..5.5] and the initial step size of 1.0 in m. 4.0, 5.0, 1.0 refer to the parameter range [4.0..5.0] and the initial step size of 1.0 in m.

5 VALERIE: A PARAMETERIZED VALIDATION ENGINE

This section describes some details of our computational validation engine 'VALERIE'. As shown in fig. 2 VALERIE is controlling the computational validation execution. That includes the generation of variants from the generative content and the efficient computation of synthetic data from this, as described in the next section. This is then followed by a description of the computation and evaluation of the perceptional function on this data. Finally, this chapter gives a brief description of the flow control implemented in VALERIE.

5.1 Computation of synthetic data

Synthetic data is generated with graphics methods. Specifically, for color (RGB) images, there are software systems available, both commercially and as open source. For our experiments in this paper, we are using Blender[4], as this tool allows import, editing, and rendering of 3D content, including scripting.

The generation of synthetic data involves the following steps: First, a 3D scene model (called asset here) with a city model and pedestrians is prepared, with naming conventions and is stored in one or more files. Scene graph representations allow an object-oriented decomposition of the scene. The top or root node contains a list of objects, which can be further divided into components. The nodes of the graph can be assigned with a name string.

[4]www.blender.org

Any object in the scene graph can be addressed by the following naming convention:

```
rootobject_name.{subcomponent_name}.attribute
```

For the example used in the section 4 *scene.person-1.pos.x* refers to a path from the root object *scene* to the object *person-1* and addresses the attribute *pos.x* of *person-1*. The object names are composed of: *ObjectClass-ObjectInstanceID*. These conventions are used to assign a class or instance labels during ground-truth generation.

The labels for object classes will be mapped to a convention used in annotation formats (like Cityscape [Cordts et al. 2016]) 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 tool's compositor.

Figure 3: A) Urban scene with instance of pedestrian at different distances (left). B) Semantic segmentation with detected pedestrian (on the left).

Fig. 3 shows an example scene of a street used in our experiments. The instance of a person is inserted in different positions, which is described by the attribute *pos.x*. This parameter relative to the camera position determines the 'distance' of the person.

Using a second parameter *pos.y*, as included in the example in section 4 would allow the positioning of the person in a plane, spanned by x+y axis of the coordinate system defined by the scene graph.

5.2 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 experiments presented in this paper, 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 analysed for flaws by a human.

Recent algorithms for semantic segmentation are based on convolutional neural networks (CNNs) that leverage networks as ResNet101 [He et al. 2015] for basic feature extraction and feature map creation as input step. These feature extractor backbones are available pretrained on image classification tasks with datasets like ImageNet[5] and thus it is not needed to further train these networks. From the extracted basic features the algorithms create more task-specific features and assign a label to each pixel in the output step through softmax classification. For the models in our work we considered ResNet101 as backbone for feature extraction.

We consider two different algorithms for semantic segmentation in our work. First, DeeplabV3+ originated from [Chen et al. 2017] and second Detectron2 by [Wu et al. 2019]. Both of these algorithms implement the approach of a Feature Pyramid Network (FPN) [Lin et al. 2016] to create high level feature map at different scales. The main difference between these algorithms is the application of atrous-spatial convolution in the DeeplabV3+ model and the higher number of concatenations of sub-sampled feature maps to the output feature map that is used for label prediction in Detectron2. Both, as is the inherent task of semantic segmentation, inference on an input image and segment all object classes [Caesar et al. 2016] into semantic labels.

Our algorithms are trained on the Cityscapes dataset [Cordts et al. 2016], a collection of European urban street scenes in the daytime with good to medium weather conditions, collected via a car mounted real photo camera and hand labelled semantic ground truth. The dataset consists of semantic labels for 19 different classes but we are considering only the pedestrian class for evaluation.. The dataset consists of semantic labels for 19 different classes but we are considering only the pedestrian class for evaluation.

To measure the performance of the task at hand we investigated the widely used metrics, mean Intersection over Union (mIoU), frequency weighted intersection over union (fwIoU), mean accuracy (mAcc) and pixel accuracy (pACC) from the COCO semantic segmentation benchmark task [Shelhamer et al. 2016]. All of these performance metrics allow to judge if an algorithm can detect, segment and add a semantic label to the input images for each class it has been trained on. Because we only consider the pedestrian class and ignore the remaining 18 classes this means that mAcc and pACC result into the same value. Similarly, mIoU and fwIoU are the same in this special case. With only one class and one pedestrian per image to evaluate we see no real advantage of mIoU over pACC and their positive correlation to one another if a pedestrian is labelled correctly. This lead to the decision to use only pACC as our evaluation metric.

The pACC is here defined as the the number of correctly classified pedestrian class pixels over the sum of all pixels labelled as pedestrian in the ground truth.

With our trained models for DeeplabV3+, we are getting the following results for the pACC on the test datasets. For the cityscapes dataset the pACC is *95.11* and for the Detectron2 model trained on cityscapes the reached pACC is *95.80*.

[5] www.image-net.org

5.3 Validation flow

Validation is typically used to testify that a system is running with the required performance over a specified boundary of operation in a top-down approach. Machine learning is coming from the opposite: It adjusts a system according to the provided data and extracts its behavior from this data in a bottom-up approach. In terms of engineering this is not quite compatible with established software engineering approaches regarding the quality assurance.

With our approach we aim to provide a tool for an iterative guided process. We suggest identifying and specifying parameters to be varied and analyzed to identify specific insufficiencies. The list of parameters can be increased or a completely new scene or set of parameters can be chosen. This process is iteratively continued until the DNN performance is within the specification or - if not - one next step could be to devise acquisition of new data to overcome the detected weaknesses by re-training.

Another validation approach could be the automated determination of sensitive parameters or (ultimately) an intelligent search through high-dimensional validation parameter spaces. Our validation engine is designed to be automated in this sense in future work.

6 RESULTS

To demonstrate the effectiveness of our evaluation approach, we conducted a number of parameter variation experiments and evaluated the results.

6.1 Generated data base

An urban 3D scene spread across $34.5km^2$ is used as the base scenario for the experiments described in the following, as depicted in Fig. 3. Image frames are rendered in HD-resolution (1920 x 1080). Every frame contains exactly one pedestrian with a feature set unique to its variation. We use two types of pedestrians: one person with a black outfit and other with a yellow shirt. We generated frames from 3 fixed camera viewing angles. For each viewing angle, the samples are generated from the same parameter list consisting of the detailed parameterization with different step sizes and boundary conditions for the individual features. Specifically, we vary:

- **Position**: In a street spanning over a length of 100m, the pedestrian is positioned in the y-direction (along the street) between 5 to 35m and between -2 to 7m in the x-direction (across the street) with step sizes 2 m and 1 m respectively. The pedestrian appears on each side of the pavements and also on the street.
- **Orientation of the pedestrian**: The pedestrian asset is rotated 360^0 with a step size of 45^0
- **sun position (time of the day)**: We use Blender's parametric sky model to alter the sun position.

Images are rendered with Blender's 'cycles' renderer with a raytracing and a ray sample rate of 1024 to trace the light sources. The samples have a gamma correction of 1.5 (images are slightly dark with no gamma correction due to the Blender's filmic view

transform used in color space conversion) and exposure of 0.5. The image is stored in png format for both rendered and ground truth images. The rendering time for each image frame is about 6 mins and about 1-2 seconds for ground truth on a Dual Xeon server. We execute several rendering tasks in parallel on multiple machines.

As evaluation perception function, we are using the two networks trained on the Cityscapes dataset, one based on DeeplabV3+ (called 'Deeplab' in the following) and the other one based on Detectron2 (called 'Detectron' below), as described in section 5.2. Each rendered frame is inferred by this model and performance is evaluated using pACC. Only the pedestrian class is considered for the metric calculation to focus on pedestrian detection task.

For each pedestrian and each camera view angle about 4950 frames are rendered. The number of frames for all the three different camera view angles is about 14850 and about 29700 frames total for the two pedestrians.

As the pedestrians are moving through the scene they are visible in all accessible places, but can be occluded by the occluding parts (trees, lamp posts, etc.). Because of the changing sun angle also a good variation of the illumnation (in shadow or not) is achieved in the data base.

6.2 Evaluation of experiments

The pACC values on our generated dataset in the evaluation are in the order of about 80-90% on average. That is lower than the average evaluation with Cityscape data (close to 100%).

The total mean detection rates for the two networks per pedestrian model are listed in table 2. This table indicates already the huge variations in the perception performance, specifically the span between 0% and 100% minimum to maximum. The lower boundary is expected, as there are always cases where an object is increasingly occluded until the detection fails. Totally, 14.4% of the entire data is > 5% occluded. The dark dressed pedestrian has 16.18% of samples, which are > 5% occluded in the database. Similarly, a brightly dressed pedestrian has 12.6% of samples, which are > 5% occluded.

The database contains huge variations, hence we investigate few important factors and their dependencies in detail like pixel area, occlusion rate and distance of pedestrian from the camera.

The following plots consists of scattered and regression plots. The regression curve is plotted by

fitting a polynomial function of a higher orders. The red regression curve is computed using the data points comprising of dark and bright clothed pedestrians combined. The black and blue regression curves are calculated using isolated data points of dark and bright pedestrian, respectively. These regression plots explain the overall trend of a parameter and its dependency over the accuracy factor. All the parameter variations are validated twice with deeplab and detectron model individually.

6.2.1 Impact of the pixel area on pedestrian's detection rate. The fig. 4 is a plot depicting the impact of pixel area on pACC. The sample data points are densely populated in the region above 60% pACC. With the same variation of scene parameters both the perception models perform differently. The deeplab model detects pedestrians fitted with dark clothes with high probability compared to bright pedestrians. On contrary, the detectron model behaves in a different way. With this model, the pedestrian dawned in bright color clothes have higher detection rates compared to darker clothes. The regression curves give us an initial peak in the dependency factor of cloth shades the pedestrians wear. With the generated dataset, we can device few hypothesis. The deeplab model is stable for all both variations of cloths for pedestrians with pixel area less than 0.48% and detectron is stable until pixel area 0.35%. It behaves differently after these thresholds. The deeplab model performance is better for pedestrians fitted with dark clothes and detectron is better with bright clothed pedestrians. These hypothesis needs extensive testing with humongous datasets.

The plot in fig. 5 depicts the performance of pixel area when categorized by the appending parameter occlusion rates of the corresponding pedestrians. Approximately 28% of the samples have 0% occlusion rate, and 85% of the samples have < 85% occlusion rate. The occlusion rate (in %) ranges from 0 - 73.67 % in the entire dataset of 29697 samples. Visualization becomes harder when data is categorized. Hence, data bins of size five are used on pixel area and occlusion rates respectively to approximate the plots. The plots show that pedestrians are occluded at different pixel area levels and the models behave incoherently. For example, the pedestrian when occluded around 75% has less chance of getting detected under detectron and better in deeplab model. The pixel area dependency on accuracy in deeplab model is linear for pedestrians who are occluded around 55% and it is not entirly linear in detectron model.

6.2.2 Impact of occlusion rate on pedestrian's detection rates. The plots in fig. 6 shows a downwards trend with increasing occlusion rates. Detectron model is comparatively more robust than deeplab. The plot explains that detectron offers stable detection with occlusion rates < 35% and then the performance drops. Deeplabs performance drops after 15% occlusion rate.

6.2.3 Impact of distance on pedestrian's detection rates. The distance considered in the fig. 7 is the Euclidean distance between the camera and pedestrian. The maximum and minimum distance of the pedestrian is 17 - 38.2 m, respectively. The deeplab regression curves depict the performance drop as the distance increases. Whereas, the detectron model's performance is stable until 35m, and then there is a sudden rise near the maximum distance.

6.3 Additional samples showcasing variations

Filtering the data sample points based on sun time did not extensively help to study sun parameters since the varying sun time would cast shadows of city buildings and pedestrians. We would need to measure the light energy at a particular point to study the impact on the detection rates. Due to implementation issues, we tried analyzing the cloth color's reflection dependency over dark and bright clothed pedestrians. We considered a black outfit for dark cloth type and yellow outfit for bright clothed pedestrian type. Every pedestrian is rendered with 2 different materials - shiny and non shiny surfaces as it can be seen in Table 3 and Table 5. It is achieved by factoring the metallic and specular tint properties of a surface material (Principled BSDF function) in blender. Though, the difference in images rendered is not obvious at first glance, these parametric changes have an interesting impact on the detection rates as explained in further reading. This can be categorized as

Table 2: Deeplab and Detectrons results on both the pedestrians

Models	Dark cloth pedestrian	Bright cloth pedestrian
Deeplab (pACC%)	89.27 (min: 0.071; max: 100)	78.82 (min: 0.065; max: 100)
Detectron (pACC%)	87.31 (min: 0.01; max: 100)	92.04 (min: 0.024; max: 100)

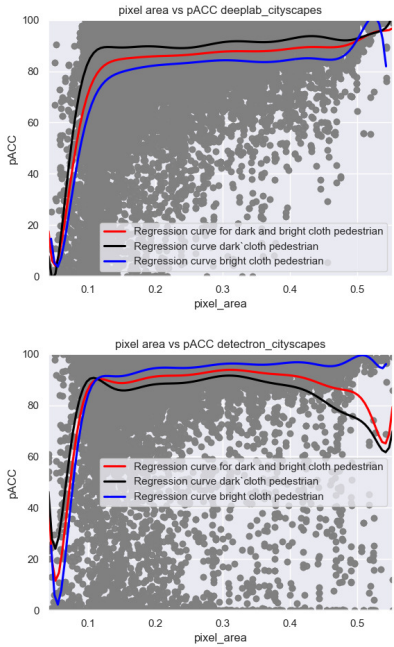


Figure 4: Performance of Deeplab (top) and dectectron (bottom) over pixel area.

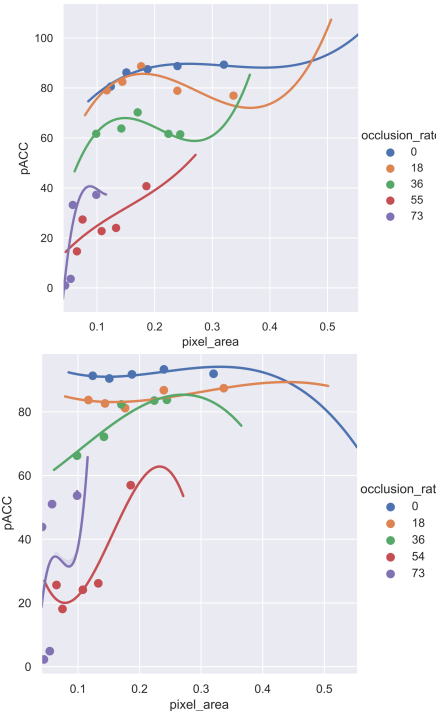


Figure 5: Pixel area based on metadata - occlusion rates, Deeplab (top) and dectectron (bottom).

adversarial attack on the image but in a controlled, precise and parametric environment.

- Dark clothed pedestrian :
  From various plots shown above, a few of the outlier samples are shown below in the Table 3 and Table 5. This table helps understand the massive variation of DNN performance with little or no noticeable tweaks; it causes the network to fluctuate its output. The pedestrians in these images are 10% occluded. All images have the same parameter variation except the sun time difference and material of the pedestrian. We could see the contrast variation of pACC, as shown in Table 4. Sample 2 shows a tremendous drop from 97.85% to 56.97%.
- Bright clothed pedestrian :
  The images shown in Table 5, and the corresponding Table 6 with inference values explain the stability factor of detectron's output. They show fewer variations, especially if the pictures contain bright clothed pedestrians. Detectron as opposed to deeplab is invariant towards the shiny and non-shiny surface of the pedestrian's cloth material.

The above examples are considered to explain the arbitrary features and parameters influencing the neural networks with no clear distinctionacknow. Its is hard to see patterns with our rendered images and strong inferences can only be confirmed by iterating over all

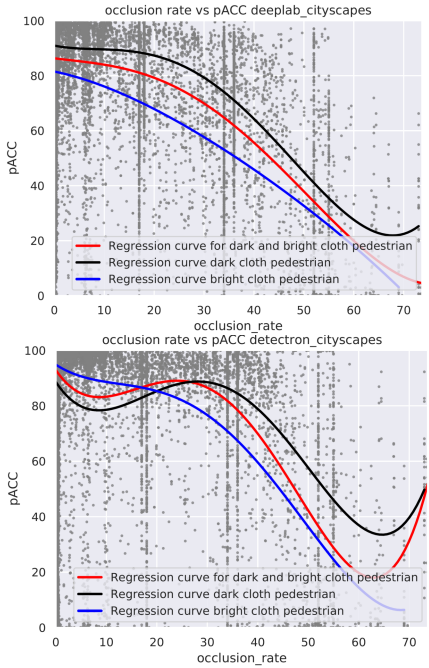


Figure 6: distance evaluation based on pedestrian detection rate, Deeplab (top) and dectectron (bottom).

major color variations. This investigation requires a huge dataset to study these parameters and its dependency over the stability of the network to categorize it as a safe algorithm at critical situations. It is merely to motivate the reader that these parameters do play a important role on the detection rates. These parameters are need to be studied in detailed fashion to gain confidence in neural network based systems.

6.4 Discussion

All these parameters (pixel area, occlusion rate and distance) considered for the evaluation show case a non linear impact on the detection rate. The models behave quite contrast to each other.

The detection rate in detectron model (fig.4) show a downward trend as the pixel area increases, which is not the case when the humans tries to detect the pedestrian. The pedestrian fitted with dark cloths are detected with higher rate by deeplab model. Similarly, pedestrian fitted with bright cloths are detected with higher rate by detectron model. The occlusion plots in fig. 6 follows a clear downward trend as the occlusion rate increases, which is intuitive and as expected. We also see a major dependency on the texture of

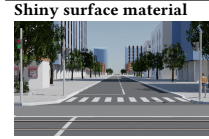
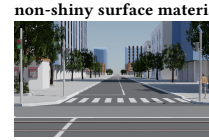
Table 3: Variation of dark clothed pedestrian: shiny and non-shiny materials

sample 1	sample 2
Shiny surface material	
	
Non shiny surface material	
	

Table 4: Variation of detection rate over changes in parameters: Deeplab and Detectrons results on dark clothed pedestrians for images shown in Table 3

Models	sample 1	sample 2
Shiny surface material		
Deeplab (pACC%)	78.82	67.27
Detectron (pACC%)	98.36	97.85
Non shiny surface material		
Deeplab (pACC%)	77.15	65.2
Detectron (pACC%)	95.68	56.97

Table 5: Variation of bright clothed pedestrian: shiny and non-shiny materials

sample1	sample2
Shiny surface material	
	
non-shiny surface material	
	

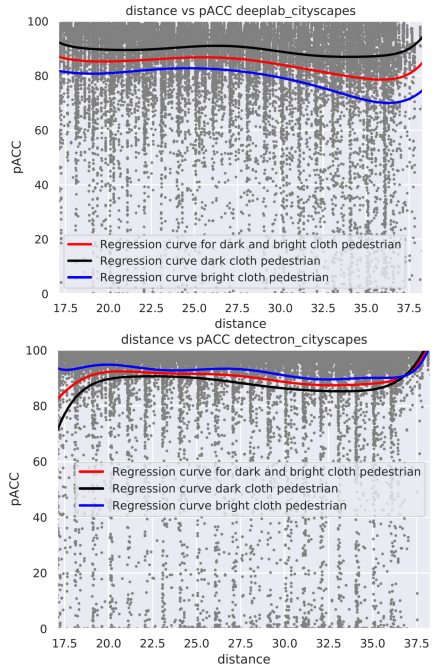


Figure 7: Pixel area based on metadata - occlusion rates, Deeplab (top) and dectectron (bottom).

Table 6: Variation of detection rate over changes in parameters: Deeplab and Detectrons results on dark clothed pedestrians for images shown in Table 5

Models	sample 1	sample 2
Shiny surface material		
Deeplab (pACC%)	76.2	70.3
Detectron (pACC%)	98.52	98.54
Non shiny surface material		
Deeplab (pACC%)	75.72	70.69
Detectron (pACC%)	97.64	98.51

the material dawned by pedestrians. These factors might be dependent to other internal parameters like surface material, irradiance, and shadows. Hence we see these non linear and contrasting behavior between the models. DNNs are powerful in generalizing the objects in a scene. It can also be invariable to the position of the object. From our experiments we see that they are quite sensitive to the parameters (occlusion, pixel area, surface material, irradiance, and shadows) that are not generally taken into consideration. With these plots, we can conclude that the trained models are not stable and not invariant to scene parameters. The analysis of the cause of these effect would require a deeper investigation and more parameter variation runs.

7 CONCLUSIONS AND OUTLOOK

We have presented a new validation approach using computational validation based on synthetic generative data. The approach allows a flexible description of parameters to be varied and to systematically test parameters in our unified validation parameter space. We consider our system as a valuable tool for the analysis of insufficiencies in perception functions.

Although the synthetic content presented in this paper is currently not photorealistic, the evaluation based on pACC is covering a good range and seemed to have good distinguishing power for analysis of the chosen network algorithms (Deeplab+Detectron). The fact that the pACC values are on average slightly lower than on real test data (Cityscape) indicate a domain-gap between the real and synthetic data. We plan to investigate if this is due to rendering fidelity or to other effects (like scene complexity and variation). We get a more detailed analysis on the DNNs by closing in the domain gap of real and virtual images.

The evaluated experiments show the potential to identify insufficiencies and to relate it to scene or other properties.

Moving forward we are implementing more sophisticated analysis methods based on more complex meta-data relations. Further, we will improve the computational efficiency of our validation engine. The latter aspect also includes concepts to enable incremental rendering and separation of computational stages, like computation of sensor errors independent from the rendering, and similar.

Both planned extensions will enhance the tool for validation, which allows to explore a high number of parameters and, therefore, provide wider coverage and higher degree of automation for the validation of DNNs for perception functions.

ACKNOWLEDGMENTS

The research leading to these results was partially funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving".

REFERENCES

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

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2016. COCO-Stuff: Thing and Stuff Classes in Context. arXiv:1612.03716 [cs.CV]

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.

Pegasus Consortium. 2020. Pegasus project home page. https://www.pegasusprojekt.de/en/home.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. arXiv:1604.01685 [cs.CV]

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

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.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2016. Feature Pyramid Networks for Object Detection. arXiv:1612.03144 [cs.CV]

M. Magnor, O. Grau, O. Sorkine-Hornung, and C. Theobalt (Eds.). 2015. *Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality.* A K Peters CRC Press.

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

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2017. On a Formal Model of Safe and Scalable Self-driving Cars. *Arxiv* (2017). https://arxiv.org/abs/1708.06374

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

W. Wachenfeld and H. Winner. 2015. Die Freigabe des autonomen Fahrens. In *Autonomes Fahren. Technische, rechtliche und gesellschaftliche Aspekte.* Springer Vieweg.

Josie Wernecke. 1994. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor.* Addison-Wesley.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

7.2 Publication 2

Improved Sensor Model for Realistic Synthetic Data Generation

Korbinian Hagn and Oliver Grau

Published in

2021 Proceedings ACM Computer Science in Cars Symposium. [HG21]

DOI: 10.1145/3488904.3493383

Improved Sensor Model for Realistic Synthetic Data Generation

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

ABSTRACT

Synthetic, i.e., computer generated-imagery (CGI) data is a key component for training and validating deep-learning-based perceptive functions due to its ability to simulate rare cases, avoidance of privacy issues and easy generation of huge datasets with pixel accurate ground-truth data. Recent simulation and rendering engines simulate already a wealth of realistic optical effects, but are mainly focused on the human perception system. But, perceptive functions require realistic images modeled with sensor artifacts as close as possible towards the sensor the training data has been recorded with.

In this paper we propose a method to improve the data synthesis by introducing a more realistic sensor model that implements a number of sensor and lens artifacts. We further propose a Wasserstein distance (earth mover's distance, EMD) based domain divergence measure and use it as minimization criterion to adapt the parameters of our sensor artifact simulation from synthetic to real images. With the optimized sensor parameters applied to the synthetic images for training, the mIoU of a semantic segmentation network (`DeeplabV3+`) solely trained on synthetic images is increased from 40.36% to 47.63%.

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; *Image segmentation*; *Cross-validation*.

KEYWORDS

datasets, neural networks, sensor simulation, image synthesis, domain adaptation

ACM Reference Format:
Korbinian Hagn and Oliver Grau. 2021. Improved Sensor Model for Realistic Synthetic Data Generation. In *Computer Science in Cars Symposium (CSCS '21), November 30, 2021, Ingolstadt, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488904.3493383

CSCS '21, November 30, 2021, Ingolstadt, Germany

ACM ISBN 978-1-4503-9139-9/21/11…$15.00
https://doi.org/10.1145/3488904.3493383

1 INTRODUCTION

Training of deep neural networks (DNNs) is increasingly resorting towards computer generated imagery (CGI) due to its mitigation of certain issues. First, synthetic data can avoid privacy issues found with recordings of members of the public and on the other hand can automatically produce ground truth data at higher quality and reliability than costly manually labeled data. Moreover, simulations allow synthesis of rare cases and the systematic variation and explanation of critical constellations [Syed Sha et al. 2020] –– a requirement for validation of products targeting safety-critical applications, such as automated driving. Here, the creation of corner cases and scenarios which otherwise could not be recorded in a real-world scenario without endangering other traffic participants is the key argument for validation of perceptive AI with synthetic images.

Despite the advantages in CGI methods, training and validation with synthetic images still has challenges: While training with these images does not guarantee a similar performance on real-world images and validation is only valid if one can make sure that the found weaknesses do not stem from the distribution shift from the real to the synthetic image domain.

To measure and mitigate this domain shift, metrics have been introduced with various applications in the field of domain adaptation or transfer learning. In domain adaptation these metrics are applied to train generative adversarial network (GAN) to adapt on a target feature space [Pan et al. 2009] or to recreate the visual properties of a dataset [Salimans et al. 2016]. On the other hand, the problem of training and validation with synthetic imagery as the source domain is directly related to the predictive performance of a perception algorithm on the target data, which these kinds of metrics struggle to capture [Ravuri and Vinyals 2019]. Additionally, applications of domain adaptation methods often resort to specifically trained DNNs which adapt one domain to the other and therefore add an extra layer of complexity and uncontrollability, whereas the creation of images via a synthesis process allows to understand domain distance influence factors more directly.

Camera-recorded images inherently show visual imperfections or artifacts, such as sensor noise, blur, chromatic aberration or image saturation, as can be seen in an image example from the A2D2 [Geyer et al. 2020] dataset in **Figure 1**. CGI methods, on the other hand, are usually based on idealized models, i.e., pinhole camera model free of sensor artifacts.

In this paper we present an approach to decrease the domain discrepancy of synthetic to real-world imagery for perceptive DNNs by realistically modelling sensor lens artifacts to increase the viability of CGI for training and cross domain validation.

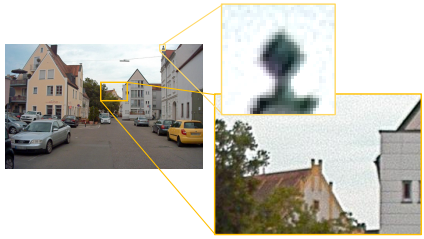


Figure 1: Real-world images (here A2D2) exhibit sensor lens artifacts which have to be closely modelled by an image synthetization process to decrease the domain distance of synthetic to real-world datasets to make them viable for training and validation.

Therefore, a new interpretation of the domain discrepancy by generalization of the distance of two datasets by the per image performance comparison over a dataset utilizing the Wasserstein or earth mover's distance (EMD) is presented.

This domain discrepancy measure is then used to optimize the proposed sensor lens artifact parametrization and effectively minimize the domain discrepancy between a synthetic and a real-world dataset. We demonstrate how this model is able to decrease the EMD domain discrepancy in an optimization of the parameters as depicted in **Figure** 5. Further, we compare the domain discrepancies of random, of extracted parameters from the target real-world dataset and of optimized parameters. Additionally, we show that the model trained with optimized sensor artifacts decreases the domain divergence on a wealth of real-world and synthetic datasets compared to a model trained without sensor artifacts.

2 RELATED WORKS

This work is related to two areas: synthetic data generation for training and validation and domain distance measures, as used in the field of domain adaptation.

Sensor Simulation for Synthetic Image training. The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g., [Burger and Barth 1995]). Recently, specifically for the domain of driving scenarios, games engines have been adapted [Dosovitskiy et al. 2017; Richter et al. 2017].

Although games engines provide a good starting point to simulate environments, they usually only offer a closed rendering setup with many trade-offs balancing between real-time constraints and a subjectively good visual appearance for human observers. Specifically the lighting computation in this rendering pipelines are in-transparent. Therefore it does not produce a physically correct imagery, instead only a fixed rendering quality (as a function of lighting computation and tone mapping), resulting in low dynamic range (LDR) output images (typically 8bit per RGB color channel).

Recently, physical-based rendering techniques have been applied to the generation of data for training and validation, like Synscapes [Wrenninge and Unger 2018]. For our work we use a dataset in high-dynamic range (HDR) resolution created with the physical-based Blender Cycles renderer[1]. We implemented a customized tone mapping to 8bit per color channel and sensor simulation, as described in the next section.

While there is great interest in understanding the domain distance in the area of domain adaptation via generative strategies, i.e., GANs, there has been little research regarding sensor artifact influence on training and validation with synthetic images. Other works [da Costa et al. 2016; Nazaré et al. 2018] add different kinds of sensor noise to their training set and reported a degradation of performance, compared to a model trained with no noise in the training set, due to higher task complexity. Adding noise in training is a common technique for image augmentation and can be seen as regularization technique [Bishop 1995] to prevent overfitting.

Our task of modeling sensor artifacts for synthetic images extracted from camera images is not aiming to improve generalization through random noise, but to tune the parameters of our sensor model to closely replicate the real-world images and improve generalization on the target data.

First results of modeling camera effects to improve synthetic data learning on the task of bounding box detection have been proposed by [Carlson et al. 2018; Liu et al. 2020]. Lin et al. [Liu et al. 2020] additionally state that generalization is an asymmetric measure which should be considered when comparing with symmetric dataset distance measures from literature. Furthermore, Carlson et al. [Carlson et al. 2019] learned sensor artifact parameters from a real-world dataset and applied the learned parameters of their noise sources as image augmentation during training with synthetic data on the task of bounding box detection. However, contrasting our approach, they apply their optimization as style loss on a latent feature vector extracted from a `VGG-16` network trained on ImageNet and evaluate the performance on the task of 2D object detection.

Domain Distance Measures. A key challenge in domain adaptation approaches is the expression of a difference measure between datasets, also called domain shift. A number of methods were developed to mitigate this shift, for example via unsupervised domain adaptation (e.g., see [Ganin and Lempitsky 2015; Long et al. 2015; Tsai et al. 2018; Tzeng et al. 2017]). Similar, [Hoffman et al. 2018] utilize GANs to stylize synthetic source domain images to real-world target domain images. However, these approaches of domain adaptation require another DNN to adapt from source to target domain with little to no tuning capability after the adaptation function has been learned, whereas we want to learn parameters of a sensor simulation, eliminating the need for a complex network.

To measure the domain shift or domain distance, measures based on the classification output of a discriminator network have been proposed [Salimans et al. 2016] based on the `InceptionV3` topology [Szegedy et al. 2016] trained on the ImageNet dataset. The work of [Bińkowski et al. 2018; Heusel et al. 2017] relies on features extracted from the `InceptionV3` network to tune domain adaptation approaches. However, these metrics are not predictive of the classification performance when the data is applied as augmentation

[1] Provided by a project partner (https://www.bit-ts.com/)

or replacement when training a discriminator [Ravuri and Vinyals 2019].

Therefore, to measure performance directly, it is essential to train with the adapted or synthetic data and validate on the target data, i.e., cross-evaluation as done by [Ros et al. 2016; Saleh et al. 2018; Wrenninge and Unger 2018]. In our work we build on this baseline and introduce a measure that mitigates weaknesses of a metric over the dataset as explained in Section 3.2.

Performance Metrics. The mean intersection over union (mIoU) is a widely used performance metric for benchmarking semantic segmentation [Cordts et al. 2016; Varma et al. 2019]. Adaptations and improvements of the mIoU have been proposed which set more weight on the segmentation contour as in [Fernandez-Moral et al. 2018; Rezatofighi et al. 2019]. Performance metrics as the mIoU are computed over the whole validation dataset, i.e., the whole confusion matrix, but there are propositions to apply the mIoU calculation on a per-image basis [Csurka et al. 2013].

A per image comparison mitigates several shortcomings of a single evaluation metric on the whole dataset when used for comparison of classifiers on the same dataset. First, one can distinguish multimodal and unimodal distributions, i.e., strong classification on one half and weak classification on the other half of a set can lead to the same mean as an average classification on all samples. Second, unimodal distributions with the same mean but different shape are also indiscernible under a single dataset averaged metric. This justification led to our choice of a per-image-based mIoU metric as it allows for deeper investigations which are especially helpful when one wants to understand the characteristics that increase or decrease a domain discrepancy.

3 METHODS

Given a synthetic (CGI) dataset of urban street scenes, our goal is to decrease the domain gap to a real-world dataset for semantic segmentation by realistic sensor artifact simulation. Therefore, we systematically analyze the image sensor noise of the target real-world dataset and use these extracted parametrization for our sensor artifact simulation. To compare our source synthetic dataset with the real-world dataset we contrive a novel per image performance-based metric to measure the generalization distance between the two of them. We utilize a `DeeplabV3+` [Chen et al. 2018] semantic segmentation model with a `ResNet101` [He et al. 2016] backbone to train and evaluate on the different datasets throughout this paper. Utilizing this discrepancy measure as optimization criteria, we adapt the parameters with random and extracted parameter starting points of our sensor artifact simulation and therefore further decrease the domain distance between synthetic and real-world images.

3.1 Sensor Simulation

We implemented a simple sensor model with the principle blocks depicted in Figure 2: The module expects images in linear RGB space. Rendering engines like Blender Cycles[2] can provide these images as results in OpenEXR format[3].

[2]https://www.blender.org/
[3]https://www.openexr.com/

We simulate a simple model by applying *sensor noise*, as added Gaussian Noise (zero mean, variance is a free parameter),*chromatic aberration*, and *blur* followed by a simple exposure control (linear tone mapping), finished by non-linear *Gamma correction*.

First, we apply blur by a simple box filter with filter size FxF and a chromatic aberration (CA). The CA is approximated using radial distortions (k1, second order), as defined in OpenCV. The CA is implemented as a per channel (red, green, blue) variation of the k1 radial distortion, i.e., we introduce an incremental parameter ca that affects the radial distortions: $k1(blue) = -ca; k1(green) = 0; k1(red) = +ca$. As next step we apply Gaussian noise to the input image.

Applying a linear function, the pixel values are then mapped and rounded to the target output byte range [0..255] by applying a linear function.

The two parameters of the linear mapping are determined by a histogram evaluation of the input RGB values of image, imitating an auto exposure of a real camera. In our experiments we have set it to saturate 2% (initially) of the brightest pixel values, as these are usually values of very high brightness, like sky or even the sun. Values below the minimum or above the set maximum are mapped to 0 or 255 respectively.

In the last step we apply gamma correction to achieve the final processed synthetic image:

$$x = (\tilde{x})^{\gamma} \tag{1}$$

The parameter γ is an approximation of the sensor non-linear mapping function. For media applications this is usually $\gamma = 2.2$ for the sRGB color space. However, for industrial cameras this is not yet standardized and some vendors do not reveal it[4]. We therefore estimate the parameter as an approximation. Figure 3 depicts the difference of an image with and without simulated sensor artifacts.

3.2 Dataset Discrepancy Measure

Our proposed discrepancy measures quantifies per image performance between same topologies trained on different datasets but evaluated on the same dataset. Considering the task of semantic segmentation we chose the mIoU as our base performance metric. We then modify the original mIoU calculation to be calculated on a per image basis instead of the whole evaluated dataset. Following, we introduce the Wasserstein-1 or EMD metric as our domain discrepancy measure. The measure is calculated on the per-image mIoU distribution of two classifiers, one trained on the source domain, the other trained on the target domain, evaluating the test set of the target domain.

The mIoU is defined as follows:

$$mIoU = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_s}{TP_s + FP_s + FN_s} \times 100\%, \tag{2}$$

With TP_s, FN_s and FP_s being the true-positives, false-negatives, and false-positives of the s^{th} class.

Here, $\mathcal{S} = \{0, 1, ..., S-1\}$ with $S = 11$, as we use the 11 classes as can be seen in Table 1. These classes are common in the real and synthetic datasets we considered for evaluation of the sensor artifact parameter optimization outcome in 4.3.

[4]The providers of the Cityscapes dataset don't document the exact mapping.

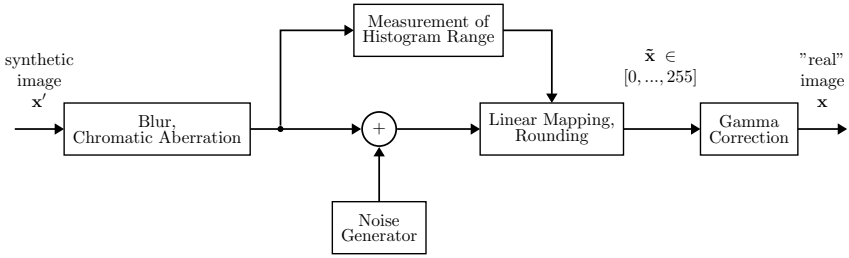


Figure 2: Sensor artifact simulation.

(a) (b)

Figure 3: Left (a): Synthetic images without lens artifacts. Right (b): Applied sensor lens artifacts, including exposure control.

Modifying the mIoU to calculate a distribution over the per-image IoU, it takes the following form:

$$IoU_n = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_{s,n}}{TP_{s,n} + FP_{s,n} + FN_{s,n}} \times 100\%, \tag{3}$$

where n denotes the $n-th$ image in the evaluated dataset. As typically done with the mIoU, IoU_n is measured in %. As we want to compare the distributions of the per image IoU values originating from the evaluation of the target test set by two DNNs, one trained on the source domain and one trained on the target domain, therefore, we apply the Wasserstein distance. The Wasserstein distance as an optimal mass transport metric from [Kolouri et al. 2017] is originally defined for density distributions p and q where inf denotes the infinum, i.e., the lowest transportation cost, $\Gamma(p, q)$ denotes all joint distributions π, i.e., transportation maps, for (X,Y) which have the marginals p and q as follows:

$$W_r(p, q) = (\inf_{\pi \in \Gamma(p,q)} \int_{\mathbb{R} \times \mathbb{R}} |X - Y|^r \, \mathrm{d}\pi)^{1/r}. \tag{4}$$

This distance formulation can be reformulated an was shown by [Ramdas et al. 2017] to be equivalent to the following:

$$W_r(p, q) = (\int_{-\infty}^{\infty} |P(t) - Q(t)|^r)^{1/r} \, \mathrm{d}t. \tag{5}$$

Here P and Q denote the respective cumulative distribution function (CDF) of p and q.

In our application we only calculate the empirical distributions of p and q, further simplifying the formulation in this case to the function of the order statistics:

$$W_r(\hat{p}, \hat{q}) = (\sum_{i=1}^{n} \|\hat{p}_i - \hat{q}_i\|^r)^{1/r}, \tag{6}$$

where $\hat{p}$ and $\hat{q}$ are the empirical distributions of the marginals p and q sorted in ascending order. With $r = 1$ and equal weight distributions we get the EMD which, in other words, measures the area between the respective CDFs with L_1 as ground distance.

We consider a sample size of at least 100 to be sufficient for the EMD calculation to be valid.

In our application the Wasserstein-1 distance is calculated as the distance of the distribution $\hat{p}$, i.e., a model trained on the source domain and evaluated on the target domain, to the distribution $\hat{q}$, i.e., a model trained on the target domain evaluated on the target domain.

When compared to distance measures such as the Fréchet inception distance (FID), which is a special case of the Wasserstein-2 distance when $\hat{p}$ and $\hat{q}$ in 6 are normally distributed and $r = 2$, that is by definition a symmetric distance, our measure is a divergence, i.e., the distance from source dataset A to target dataset B can be different to the distance from target dataset B to source dataset A. A divergence also reflects the characteristic of a classifier having different generalization distance when trained on dataset A and evaluated on dataset B or the other way around.

Because the ground measure of the signatures, i.e., the IoU per image, is bound to $0 \leq IoU_n \leq 100$, the EMD measure is then bounded $0 \leq EMD \leq 100^r$ with r being defined as the Wasserstein norm. For $r = 1$, the measure is bound with $0 \leq EMD \leq 100$.

As we want to ensure that we can be certain in using the performance criterion of a dataset as proxy for its domain distribution, we must be certain that we obtain the same distribution when training

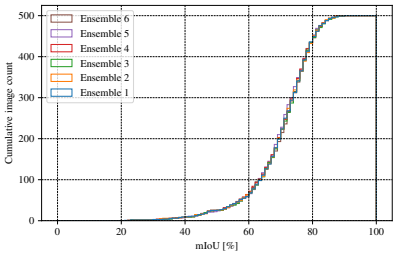


Figure 4: CDF of ensembles of `DeeplabV3+` models trained on Cityscapes and evaluated on the same. Applying the two-sample Kolmogorov-eSmirnov test to each possible pair of the ensemble we get a minimum $p - value > 0.95$.

with the same DNN from different, i.e., random, starting conditions. Therefore, we trained six models of the `DeeplabV3+` network with the same hyperparameters but by different random weight initialization on the Cityscapes dataset and evaluated them on the validation set calculating the mIoU per image distributions. The resulting distributions of each model in the ensemble are then converted into a CDF as is shown in Figure 4. When comparing CDFs, and to reinforce the claim that the mIoU per image performance distribution is constant when training a model on a dataset, we apply the two-sample Kolmogorov-Smirnov test [Massey Jr 1951] on each pair of distributions in the ensemble. The resulting p-values of the Kolmogorov-Smirnov tests are at least > 0.95, hence supporting our hypothesis.

4 RESULTS AND DISCUSSION

4.1 Sensor Parameter Extraction

As a baseline for our sensor artifact simulation we analyzed images from the Cityscapes training data and measured the parameters. Sensor noise was extracted from images with uniformly coloured areas ranging from dark to light colors. Chromatic aberration was extracted from images with traffic signs on the outmost edges of the image in horizontal and vertical direction, due to their favorable high contrast of black and white of the traffic signs. Saturation was measured as relative number of pixels in percent that are clipped at the value 255 in all color channels in the RGB images.

The resulting measured parameters are then as follows:

$saturation = 2.0\%$, $noise \sim \mathcal{N}(0, 3)$, $\gamma = 0.8$, $F = 4$, and $ca = 0.08$. These parameters are later also used as a starting point of our parameter optimization approach.

4.2 Sensor Artifact Optimization Experiment

As we want to show that by applying sensor lens artifacts we can effectively decrease the domain discrepancy between synthetic and real-world domain, we further utilize the EMD as dataset discrepancy measure and the extracted sensor parameters from camera images of Cityscapes, we then apply an optimization strategy to iteratively decrease the gap between the Cityscapes and the synthetic dataset [Consortium 2021] receiving the optimal parameters for the sensor simulation. For optimization, we chose to use the trust region method. Specifically, we use the trust region reflective (trf) method as implemented in `SciPy` [Virtanen et al. 2020]. The trf is a least squares minimization method to find the local minimum of a cost function given certain input variables. As cost function we use the EMD from synthetic model and real-world model predictions on the target test set. The variables as input to the cost function are the parameters of the sensor artifact simulation. The `trf` method has the capability of bounding the variables to meaningful ranges as this will prevent the parameters to increase into unreasonable values, e.g., increase the additive gaussian noise to values > 10. The stop criterion is met when the increase of parameter step size or decrease of the cost function is below 10^{-6}.

The overall description of our optimization method is depicted in Figure 5. **Step 1:** Initial parameters from the optimization method are applied in the sensor artifact simulation to the synthetic images. **Step 2:** The `DeeplabV3+` model with `ResNet101` backbone is pretrained on 15 epochs on the original synthetic dataset is loaded and trained for one epoch on the synthetic dataset with a learning rate of 0.1. **Step 3:** The model parameters are frozen and set to evaluate. **Step 4:** The model predicts on the validation set of the Cityscapes dataset. **Step 5:** The remaining domain discrepancy is measured by evaluation of the mIoU per image and calculation of the EMD to the evaluations of a model trained on Cityscapes. **Step 6:** The resulting EMD is fed as cost to the optimization method. **Step 7:** New parameters are set for the sensor artifact simulation, or the optimization ends if the stop criteria are met.

After iterating the parameter optimization with the trf method we compare our optimized lens artifacts trained model with the unmodified synthetic trained model by their EMDs with a model trained on the Cityscapes dataset. Figure 6 depicts the distributions resulting from this evaluation. The `DeeplabV3+` model trained with the optimized sensor artifact simulation applied on the synthetic dataset outperforms the baseline and achieves an EMD score of 26.48, while decreasing the domain gap by 6.19. The resulting parameters are $saturation = 2.11\%$, $noise \sim \mathcal{N}(0, 3.0000005)$, $\gamma = 0.800001$, $F = 4$ and $ca = 0.008000005$. The parameters changed only slightly from the starting point, indicating the extracted parameters as good first choice as we can confirm when we optimize from a random initialized starting point.

An exemplary visual inspection of the results in Figure 7 helps to understand the distribution shift and therefore the decreased EMD. While the best prediction performance image (TOP) increased only slightly from the synthetic trained model (c) to the sensor artifact optimized model (d), the worst prediction case (Bottom) shows improved segmentation performance for the sensor artifact optimized model (d), in this case even better than the Cityscapes trained model (b).

We compare the overall mIoU performance on the Cityscapes datasets between models trained with the initial unmodified synthetic dataset, the synthetic dataset with random initialized lens artifact parameters and the synthetic dataset with extracted parameters from Cityscapes with the baseline of a model trained on

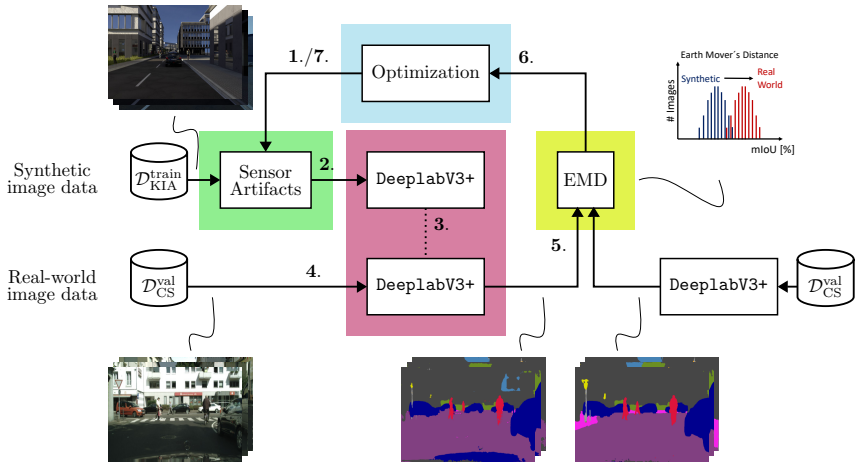


Figure 5: Optimization of sensor artifacts to decrease discrepancy between real and synthetic datasets.

Table 1: Performance results as per class mIoU, overall mIoU and EMD domain divergence evaluated on Cityscapes with models trained on Cityscapes, our synthetic, synthetic with random parameterized lens artifacts and synthetic extracted parameterized lens artifacts. For the latter two, there are models evaluated on Cityscapes with and without optimization of the parameters for the sensor lens artifact simulation. The model trained with optimized extracted parameters achieves highest performance on the Cityscapes dataset.

Model trained on	optimized	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	human	car	truck	mIoU ↑	EMD ↓
Cityscapes	no	97.50	81.21	92.22	54.54	57.28	70.26	92.33	93.98	82.55	93.67	81.62	81.56	-
w/o artifacts	no	60.46	23.51	71.99	10.70	26.00	13.47	75.72	74.70	51.27	34.46	1.74	40.37	32.67
w/ random artifacts	no	77.86	20.56	60.66	8.21	17.83	7.36	72.18	68.21	50.53	74.01	4.68	41.58	30.03
w/ extracted artifacts	no	80.65	27.17	66.54	10.43	21.64	11.87	68.82	67.99	59.22	70.06	**7.39**	44.71	29.84
w/ random artifacts	yes	**82.89**	**34.69**	71.11	11.29	16.66	9.12	69.81	**76.77**	**61.48**	**79.50**	5.30	45.76	26.58
w/ extracted artifacts	yes	79.62	30.30	**75.74**	**16.30**	**28.31**	**13.86**	**78.75**	70.88	59.74	64.06	6.42	**47.63**	**26.48**

the Cityscapes dataset and results are listed in Table 1. Additionally, for the random and the extracted parameters we evaluate the performance with initial and optimized parameters, where the parameters have been optimized by our EMD minimization. While the model without any sensor simulation achieves the lowest overall and per class performance, the model with random parameter initialization achieves slightly higher performance and is on its part surpassed by the model with the Cityscapes extracted parameters. For the optimized parameter trained models, they outperform all non-optimized models on the evaluation with the model that has been trained on the optimized extracted starting parameters outperforming all other models. The model trained with optimized random starting parameters achieves even higher performance on classes road, sidewalk, human and even significantly on the car class but still falls behind on five of the remaining classes and the overall performance on the Cityscapes dataset. Further, the random parameter optimized model took over 22 iterations to converge to its local minimum, whereas the optimization of extracted starting parameters only took 6 iterations until reaching a local minimum, making it more than 3 times faster to converge. Furthermore, it is

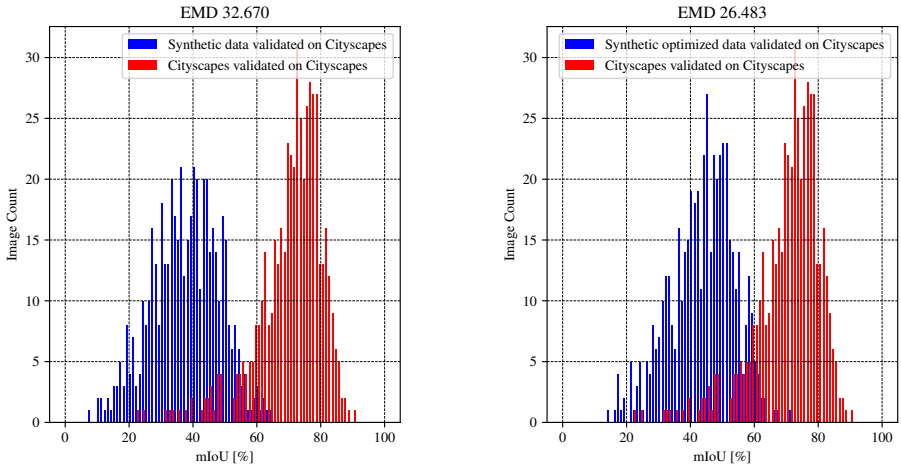
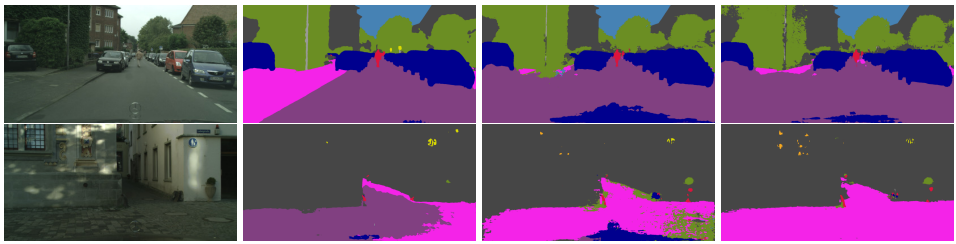


Figure 6: EMD domain divergence calculation of synthetic and optimized synthetic data to real-world images. (a) Comparison of synthetic data with Cityscapes data, (b) Synthetic sensor artifact optimized dataset compared to the target dataset Cityscapes.



(a) Original (b) CS Pred (c) Synthetic Pred (d) Synthetic Optimized Pred

Figure 7: Top row: Best performance predictions. Bottom row: Worst Performance Predictions. Original (a), Cityscapes trained model prediction (b), synthetic trained model prediction (c) and sensor artifact optimized trained model prediction (d). While top performance increased only slightly, the optimization lead to more robust predictions in worst case, i.e., harder examples.

shown all models with applied sensor lens artifacts outperform the model trained without additional lens artifacts.

4.3 EMD Cross-Evaluation

To examine if the domain divergence decreases with other real-world and synthetic datasets with our sensor lens artifact simulation as well, we evaluate our EMD results on a range of real-world and synthetic datasets for semantic segmentation. Including real-world datasets A2D2 [Geyer et al. 2020], Berkeley Deep Drive (BDD100K) [Yu et al. 2020], Cityscapes [Cordts et al. 2016], as well as synthetic GTAV [Richter et al. 2016], India Driving Dataset (IDD) [Varma et al. 2019], Mapillary Vistas (MV) [Neuhold et al. 2017], the synthetic Synscapes (SYNS) [Wrenninge and Unger 2018], our synthetic (Synth) [Consortium 2021] and our synthetic with optimized sensor

Table 2: Cross domain discrepancy results of models trained with and without optimized lens artifacts, evaluated on different real-world and synthetic datasets. The domain divergence is measured with our proposed EMD measure. The model trained with optimized lens artifacts applied to the synthetic images exhibits a smaller domain divergence than the model trained without lens artifacts.

	EMD ↓	Model trained on	
		Synthetic	**Synthetic Optimized**
Model evaluated on	**A2D2**	34.95	**29.32**
	BDD100K	26.43	**21.54**
	CS	32.66	**26.48**
	GTAV	36.08	**32.94**
	IDD	41.64	**36.95**
	MV	30.35	**27.03**
	SYNS	43.76	**43.56**

lens artifacts (SynthOpt) datasets. Therefore, for every considered dataset a `DeeplabV3+` model has been trained on its training set and was then evaluated with the per-image mIoU on its test set to get the target distributions. Then, all models are evaluated on each of the other datasets test set they have not been trained with and for each resulting distribution the EMD to the beforehand calculated datasets target distribution is calculated. In Table 2 the results of this cross-domain analysis measured with the EMD score are depicted. Each of the columns denote a `DeeplabV3+` model that has been trained on the corresponding dataset, whereas the rows denote the target datasets, e.g., column entry Synthetic and row entry A2D2 denote a model that was trained with our synthetic data, then evaluated on the A2D2 test set and the EMD was calculated by comparison with the target distribution of a model trained and evaluated on the A2D2 dataset. Bold results denote the lower EMD score between our synthetic trained baseline model and our optimized parameter applied sensor lens artifacts trained model. Our model trained with optimized parameters applied to the synthetic dataset achieves lower EMD scores on all considered datasets compared to the synthetic trained baseline, and while the domain discrepancy decrease is high on real datasets, the discrepancy only decreased marginally for the other synthetic datasets, i.e., for the GTAV and the Synscapes datasets. The lower EMD scores on the real-world datasets can be attributed to the applied sensor lens artifacts when training the model as these dataset all exhibit some form of sensor lens artifacts due to the nature these images where captured compared to the synthetic generated datasets.

CONCLUSIONS

In this paper we could demonstrate that we are able to decrease the generalization distance and domain divergence between perceptive AI models trained with synthetic datasets and models trained with real-world datasets by realistically modelling sensor lens artifacts on the synthetic dataset for training. To achieve this we utilize the performance metric mIoU per image as a proxy distribution for a dataset and the EMD as a domain discrepancy measure between distributions and can then decrease visual differences of a synthetic dataset through optimization, i.e., minimization of this EMD measure and altogether increase the viability of CGI for training and cross domain validation purposes of perceptive AI. We reinforce our argument for a per-image performance measure as a proxy distribution and showed that training an ensemble of a fixed DNN model with different random initialized weights but with the same hyperparameters, will lead to the same per-image performance distributions when these ensemble models are evaluated on the test set of the corresponding training dataset.

Furthermore, we show that by application of sensor lens artifacts on the source synthetic dataset the domain divergence can effectively be decreased by either random initialized parameters or parameters extracted from the target dataset. Further decreasing the gap by optimization of the parameters with our proposed method. Last, we showed that a model trained on synthetic images with optimized sensor artifacts exhibits a smaller domain divergence, measured by the EMD, on a range of real-world and other synthetic datasets than a model trained solely with synthetic images. However, application of realistically modeled sensor artifacts to synthetic images can reduce the domain divergence only to a limited value, the remaining gap can only be closed by adaptation of other factors such as color grading, object mismatches and scene complexity.

When utilizing synthetic imagery for training and validation, the domain gap, due to visual differences of real and computer generated images limits the applicability of the same. The introduced sensor simulation and the domain discrepancy measure as well as its application in minimization of the visual difference between synthetic and real-world datasets is one step further towards fully utilizing CGI for validation of perceptive AI functions.

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

C. M. Bishop. 1995. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation* 7, 1 (1995), 108–116. https://doi.org/10.1162/neco.1995.7.1.108

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lUOzWCW

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

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*.

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.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 833–851.

KI-A Consortium. 2021. KI-A homepage. https://www.ki-absicherung-projekt.de/en/ Accessed: 2021-10-27.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. 2013. What is a good evaluation measure for semantic segmentation?.. In *Bmvc*, Vol. 27. 10–5244.

Gabriel B. Paranhos da Costa, Welinton A. Contato, Tiago S. Nazaré, João do E. S. Batista Neto, and Moacir Ponti. 2016. An empirical study on the effects of different types of noise in image classification tasks. *CoRR* abs/1609.02781 (2016). arXiv:1609.02781 http://arxiv.org/abs/1609.02781

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

E. Fernandez-Moral, R. Martins, D. Wolf, and P. Rives. 2018. A New Metric for Evaluating Semantic Segmentation: Leveraging Global and Contour Accuracy. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 1051–1056. https://doi.org/10.1109/IVS.2018.8500497

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

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. 2020. A2D2: Audi Autonomous Driving Dataset. arXiv:2004.06320 [cs.CV]

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

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.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*. PMLR, 1989–1998.

S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde. 2017. Optimal Mass Transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine* 34, 4 (2017), 43–59. https://doi.org/10.1109/MSP.2017.2695801

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

Z. Liu, T. Lian, J. Farrell, and B. A. Wandell. 2020. Neural Network Generalization: The Impact of Camera Parameters. *IEEE Access* 8 (2020), 10443–10454. https://doi.org/10.1109/ACCESS.2020.2965089

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 97–105.

Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951), 68–78.

Tiago S. Nazaré, Gabriel B. Paranhos da Costa, Welinton A. Contato, and Moacir Ponti. 2018. Deep Convolutional Neural Networks and Noisy Images. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Marcelo Mendoza and Sergio Velastín (Eds.). Springer International Publishing, Cham, 416–424.

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

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2009. Domain Adaptation via Transfer Component Analysis. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence* (Pasadena, California, USA) *(IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1187–1192.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. 2017. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy* 19, 2 (2017), 47.

Suman V. Ravuri and Oriol Vinyals. 2019. Seeing is Not Necessarily Believing: Limitations of BigGANs for Data Augmentation.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. 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*. 658–666.

S. R. Richter, Z. Hayder, and V. Koltun. 2017. Playing for Benchmarks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2232–2241. https://doi.org/10.1109/ICCV.2017.243

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.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3234–3243. https://doi.org/10.1109/CVPR.2016.352

Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. 2018. Effective use of synthetic data for urban scene semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 84–100.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29 (2016), 2234–2242.

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

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker. 2018. Learning to Adapt Structured Output Space for Semantic Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7472–7481. https://doi.org/10.1109/CVPR.2018.00780

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2962–2971. https://doi.org/10.1109/CVPR.2017.316

Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. 2019. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1743–1751.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stefan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antonio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* (2020).

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

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2636–2645.

7.3 Publication 3

Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation

Korbinian Hagn and Oliver Grau

Published in:

Fingscheidt, T., Gottschalk, H., Houben, S. (eds) Deep Neural Networks and Data for Automated Driving. Springer, Cham. [HG22b]

DOI: 10.1007/978-3-031-01233-4_4

Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation

Korbinian Hagn and Oliver Grau

Abstract Synthetic, i.e., computer-generated imagery (CGI) data is a key component for training and validating deep-learning-based perceptive functions due to its ability to simulate rare cases, avoidance of privacy issues, and generation of pixel-accurate ground truth data. Today, physical-based rendering (PBR) engines simulate already a wealth of realistic optical effects but are mainly focused on the human perception system. Whereas the perceptive functions require realistic images modeled with sensor artifacts as close as possible toward the sensor, the training data has been recorded. This chapter proposes a way to improve the data synthesis process by application of realistic sensor artifacts. To do this, one has to overcome the domain distance between real-world imagery and the synthetic imagery. Therefore, we propose a measure which captures the generalization distance of two distinct datasets which have been trained on the same model. With this measure the data synthesis pipeline can be improved to produce realistic sensor-simulated images which are closer to the real-world domain. The proposed measure is based on the Wasserstein distance (earth mover's distance, EMD) over the performance metric mean intersection-over-union (mIoU) on a per-image basis, comparing synthetic and real datasets using deep neural networks (DNNs) for semantic segmentation. This measure is subsequently used to match the characteristic of a real-world camera for the image synthesis pipeline which considers realistic sensor noise and lens artifacts. Comparing the measure with the well-established Fréchet inception distance (FID) on real and artificial datasets demonstrates the ability to interpret the generalization distance which is inherent asymmetric and more informative than just a simple distance measure. Furthermore, we use the metric as an optimization criterion to adapt a synthetic dataset to a real dataset, decreasing the EMD distance between a synthetic and the Cityscapes dataset from 32.67 to 27.48 and increasing the mIoU of our test algorithm (`DeeplabV3+`) from 40.36 to 47.63%.

K. Hagn ( ) · O. Grau
Intel Deutschland GmbH, Lilienthalstraße 15, 85579 Neubiberg, Germany
e-mail: korbinian.hagn@intel.com

O. Grau
e-mail: oliver.grau@intel.com

T. Fingscheidt et al. (eds.), *Deep Neural Networks and Data for Automated Driving*,
https://doi.org/10.1007/978-3-031-01233-4_4

1 Introduction

Validation of deep neural networks (DNNs) is increasingly resorting toward computer-generated imagery (CGI) due to its mitigation of certain issues. First, synthetic data can avoid privacy issues found with recordings of members of the public and, on the other hand, can automatically produce vast amounts of data at high quality with pixel-accurate ground truth data and reliability than costly manually labeled data. Moreover, simulations allow synthesis of rare cases and the systematic variation and explanation of critical constellations [SGH20]—a requirement for validation of products targeting safety-critical applications, such as automated driving. Here, the creation of corner cases and scenarios which otherwise could not be recorded in a real-world scenario without endangering other traffic participants is the key argument for the validation of perceptive AI with synthetic images.

Despite the advantages of CGI methods, training and validation with synthetic images still have challenges: Training with these images does not guarantee a similar performance on real-world images and validation is only valid if one can verify that the found weaknesses in the validation do not stem from the synthetic-to-real distribution shift seen in the input.

To measure and mitigate this domain shift, metrics have been introduced with various applications in the field of domain adaptation or transfer learning. In domain adaptation, the metrics such as FID, kernel inception distance (KID), and maximum mean discrepancy (MMD) are applied to train generative adversarial networks (GANs) to adapt on a target feature space [PTKY09] or to re-create the visual properties of a dataset [SGZ+16]. However, the problem of training and validation with synthetic imagery is directly related to the predictive performance of a perception algorithm on the target data, and these kinds of metrics struggle to correlate with the predictive performance [RV19]. Additionally, applications of domain adaptation methods often resort to specifically trained DNNs, e.g., GANs, which adapt one domain to the other and therefore add an extra layer of complexity and uncontrollability. This is especially unwanted if a validation goal is tested, e.g., to detect all pedestrians, and the domain adaption by a GAN would add additional objects into the scene (e.g., see [HTP+18]) making it even harder to attribute detected faults of the model to certain specifics of the tested scene. Here, the creation of images via a synthesis process allows to understand domain distance influence factors more directly as all parameters are under direct control.

Camera-recorded images inherently show visual imperfections or artifacts, such as sensor noise, blur, chromatic aberration, or image saturation, as can be seen in an image example from the A2D2 [GKM+20] dataset in Fig. 1. CGI methods, on the other hand, are usually based on idealized models; for example, the pinhole camera model [Stu14] which is free of sensor artifacts.

In this chapter, we present an approach to decrease the domain divergence of synthetic to real-world imagery for perceptive DNNs by realistically modeling sensor lens artifacts to increase the viability of CGI for training and validation. To achieve this, we first introduce a model of sensor artifacts whose parameters are extracted

Fig. 1 Real-world images (here A2D2) exhibit sensor lens artifacts which have to be closely modeled by an image synthesization process to decrease the domain distance of synthetic to real-world datasets to make them viable for training and validation

from a real-world dataset and then apply it on a synthetic dataset for training and measuring the remaining domain divergence via validation. Therefore, a new interpretation of the domain divergence by generalization of the distance of two datasets by the per-image performance comparison over a dataset utilizing the Wasserstein or earth mover's distance (EMD) is presented. Next, we demonstrate how this model is able to decrease the domain divergence further by optimization of the initial extracted sensor camera simulation parameters as depicted in Fig. 6. Additionally, we compare our results with randomly chosen parameters as well as with randomly chosen and optimized parameters. Last, we strengthen the case for the usability of our EMD domain divergence measure by comparison with the well-known Fréchet inception distance (FID) on a set of real-world and synthetic datasets and highlight the advantage of our asymmetric domain divergence against the symmetric distance.

2 Related Works

This chapter is related to two areas: domain distance measures, as used in the field of domain adaptation and synthetic data generation for training and validation.

Domain distance measures: A key challenge in domain adaptation approaches is the expression of a distance measure between datasets, also called domain shift. A number of methods were developed to mitigate this shift (e.g., see [LCWJ15, GL15, THSD17, THS+18]).

To measure the domain shift or domain distance, the inception score (IS) has been proposed [SGZ+16], where the classification output of an `InceptionV3`-

based [SVI+16] discriminator network trained on the ImageNet dataset [DDS+09] is used. The works of [HRU+17, BSAG21] rely on features extracted from the `InceptionV3` network to tune domain adaptation approaches, i.e., the FID and KID. However, these metrics cannot predict if the classification performance increases when adapted data is applied as training data for a discriminator [RV19].

Therefore, to measure performance directly, it is essential to train with the adapted or synthetic data and validate on the target data, i.e., cross-evaluation as done by [RSM+16, WU18, SAS+18].

Performance metrics: The mean intersection-over-union (mIoU) is a widely used performance metric for benchmarking semantic segmentation [COR+16, VSN+18]. Adaptations and improvements of the mIoU have been proposed which put more weight on the segmentation contour as in [FMWR18, RTG+19]. Performance metrics as the mIoU are computed over the whole validation dataset, i.e., the whole confusion matrix, but there are propositions to apply the mIoU calculation on a per-image basis and compare the resulting empirical distributions [CLP13].

A per-image comparison mitigates several shortcomings of a single evaluation metric on the whole dataset when used for comparison of classifiers on the same dataset. First, one can distinguish multimodal and unimodal distributions, i.e., strong classification on one half and weak classification on the other half of a set can lead to the same mean as an average classification on all samples. Second, unimodal distributions with the same mean but different shape are also indiscernible under a single dataset averaged metric. This justification led to our choice of a per-image-based mIoU metric as it allows for deeper investigations which are especially helpful when one wants to understand the characteristics that increase or decrease a domain divergence.

Sensor simulation for synthetic image training: The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g., [BB95]). Recently, specifically for the domain of driving scenarios, games engines have been adapted [RHK17, DRC+17].

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 for human observers. Specifically the lighting computation in the rendering pipelines is intransparent. Therefore, it does not produce a physically correct imagery; instead only a fixed rendering quality (as a function of lighting computation and tone mapping), resulting in output of images having a low dynamic range (LDR) (typically 8-bit per RGB color channel).

Recently, physical-based rendering techniques have been applied to the generation of data for training and validation, like Synscapes [WU18]. For our chapter we use a dataset in high dynamic range (HDR) created with the physical-based `Blender Cycles` renderer.[1] We implemented a customized tone mapping to 8-bit per color channel and sensor simulation, as described in the next section.

[1] Provided by the KI Absicherung project [KI 20].

While there is great interest in understanding the domain distance in the area of domain adaptation via generative strategies, i.e., GANs, there has been little research regarding sensor artifact influence on training and validation with synthetic images. Other works [dCCN+16, NdCCP18] add different kinds of sensor noise to their training set and report a degradation of performance, compared to a model trained with no noise in the training set, due to training of a harder, i.e., noisier, visual task. Adding noise in training is a common technique for image augmentation and can be seen as a regularization technique [Bis95] to prevent overfitting.

Our task of modeling sensor artifacts for synthetic images extracted from camera images is not aimed at improving the generalization through random noise, but to tune the parameters of our sensor model to closely replicate the real-world images and improve generalization on the target data.

First results of modeling camera effects to improve synthetic data learning on the perceptive task of bounding box detection have been proposed by [CSVJR18, LLFW20]. Lin et al. [LLFW20] additionally state that generalization is an asymmetric measure which should be considered when comparing with symmetric dataset distance measures from literature. Furthermore, Carlson et al. [CSVJR19] learned sensor artifact parameters from a real-world dataset and applied the learned parameters of their noise sources as image augmentation during training with synthetic data on the task of bounding box detection. However, contrasting our approach, they apply their optimization as style loss on a latent feature vector extracted from a `VGG-16` network trained on ImageNet and evaluate the performance on the task of 2D object detection.

3 Methods

Given a synthetic (CGI) dataset of urban street scenes, our goal is to decrease the domain gap to a real-world dataset for semantic segmentation by realistic sensor artifact simulation. Therefore, we systematically analyze the image sensor artifacts of the real-world dataset and use this extracted parametrization for our sensor artifact simulation. To compare our synthetic dataset with the real-world dataset we contrive a novel per-image performance-based metric to measure the generalization distance between the datasets. We utilize a `DeeplabV3+` [CZP+18] semantic segmentation model with a `ResNet101` [HZRS16] backbone to train and evaluate on the different datasets throughout this paper. To show the valuable properties of our measure we compare it with the established domain distance, i.e., Fréchet inception distance (FID). Lastly, we use our measure as optimization criteria for adapting the parameters of our sensor artifact simulation with the extracted parameters as starting point and show that we can further decrease the domain distance from synthetic images to real-world images.

3.1 Sensor Simulation

We implemented a simple sensor model with the principle blocks depicted in Fig. 2: The module expects images in linear RGB space. Rendering engines like `Blender Cycles`[2] can provide these images as results in `OpenEXR` format.[3]

We simulate a simple model by applying *chromatic aberration*, *blur*, and *sensor noise*, as additive Gaussian noise (zero mean, variance is a free parameter), followed by a simple exposure control (linear tone mapping), finished by non-linear *gamma correction*.

First, we apply blur by a simple box filter with filter size $F \times F$ and a chromatic aberration (CA). The CA is approximated using radial distortions (k1, second order), e.g., [CV14], as defined in OpenCV. The CA is implemented as a per channel (red, green, blue) variation of the k1 radial distortion, i.e., we introduce an incremental parameter ca that affects the radial distortions: $\text{k1(blue)} = -ca$; $\text{k1(green)} = 0$; $\text{k1(red)} = +ca$. As the next step, we apply Gaussian noise to the input image.

Applying a linear function, the pixel values are then mapped and rounded to the target output byte range [0, ..., 255].

The two parameters of the linear mapping are determined by a histogram evaluation of the input RGB values of the respective image, imitating an auto exposure of a real camera. In our experiments we have set it to saturate 2% (initially) of the brightest pixel values, as these are usually values of very high brightness, induced by sky or even the sun. Values below the minimum or above the set maximum are mapped to 0 or 255, respectively.

In the last step we apply gamma correction to achieve the final processed synthetic image:

$$\mathbf{x} = (\tilde{\mathbf{x}})^{\gamma} \tag{1}$$

The parameter γ is an approximation of the sensor non-linear mapping function. For media applications this is usually $\gamma = 2.2$ for the sRGB color space [RD14]. However, for industrial cameras, this is not yet standardized and some vendors do

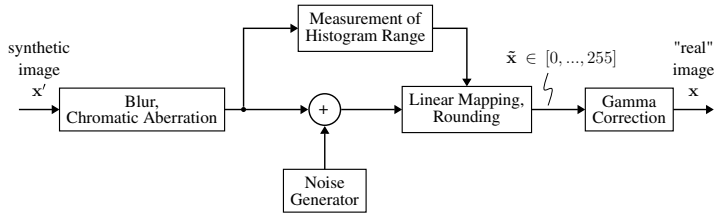
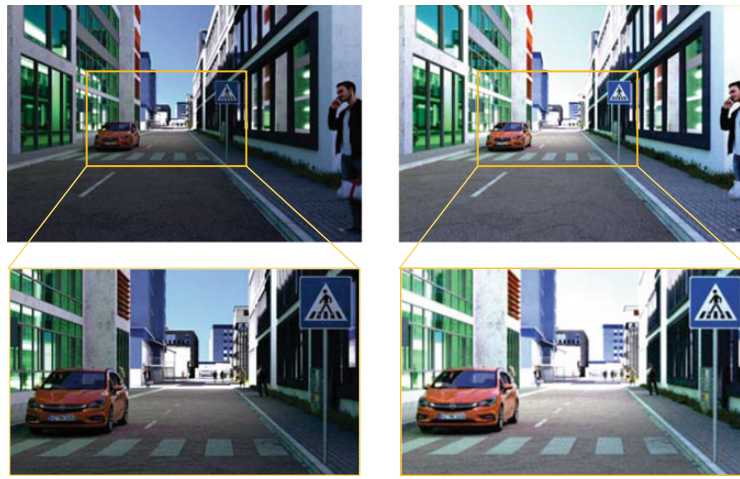


Fig. 2 Sensor artifact simulation

[2] https://www.blender.org/.
[3] https://www.openexr.com/.



(a) (b)

Fig. 3 Left **a**: Synthetic images without lens artifacts. Right **b**: Applied sensor lens artifacts, including exposure control

not reveal it.[4] We therefore estimate the parameter as an approximation. Figure 3 depicts the difference of an image with and without simulated sensor artifacts.

3.2 Dataset Divergence Measure

Our proposed distance quantifies per image performance between models trained on different datasets but evaluated on the same dataset. Considering the task of semantic segmentation we chose the mIoU as our base metric. We then modify the mIoU to be calculated per image instead of the confusion matrix on the whole evaluated dataset. Next, we introduce the Wasserstein-1 or earth mover's distance (EMD) metric as our divergence measure between the per-image mIoU distribution of two classifiers trained on distinct datasets, i.e., synthetic and real-world datasets, but evaluated on the same real-world dataset the second classifier has been trained with.

The mIoU is defined as follows:

$$mIoU = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_s}{TP_s + FP_s + FN_s} \times 100\%, \tag{2}$$

[4] The providers of the Cityscapes dataset don't document the exact mapping.

with TP_s, FN_s, and FP_s being the amount of true-positives, false-negatives, and false-positives of the sth class over all images of the evaluated dataset.

Here, $\mathcal{S} = \{0, 1, ..., S-1\}$, with $S = 11$, as we use the 11 classes defined in Table 1. These classes are the maximal overlap of common classes in the real and synthetic datasets considered for cross-evaluation and comparison of our measure with the Fréchet inception distance (FID), as can be seen later in Sect. 4.3, Tables 3 and 4.

A distribution over the per-image IoU takes the following form:

$$IoU_n = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_{s,n}}{TP_{s,n} + FP_{s,n} + FN_{s,n}} \times 100\%, \tag{3}$$

where n denotes the nth image in the validation dataset. Here, IoU_n is measured in %. We want to compare the distributions of per-image IoU values from two different models; therefore, we apply the Wasserstein distance. The Wasserstein distance as an optimal mass transport metric from [KPT+17] is defined for density distributions p and q where inf denotes the infinium, i.e., lowest transportation cost, $\Gamma(p, q)$ denotes the set containing all joint distributions π, i.e., transportation maps, for (X,Y) which have the marginals p and q as follows:

$$W_r(p, q) = \left(\inf_{\pi \in \Gamma(p,q)} \int_{\mathbb{R} \times \mathbb{R}} |X - Y|^r \mathrm{d}\pi \right)^{1/r}. \tag{4}$$

This distance formulation is equivalent to the following [RTC17]:

$$W_r(p, q) = \left(\int_{-\infty}^{\infty} |P(t) - Q(t)|^r \right)^{1/r} \mathrm{d}t. \tag{5}$$

Here P and Q denote the respective cumulative distribution functions (CDFs) of p and q.

In our application we calculate the empirical distributions of p and q, which simplifies in this case to the function of the order statistics:

$$W_r(\hat{p}, \hat{q}) = \left(\sum_{i=1}^{n} |\hat{p}_i - \hat{q}_i|^r \right)^{1/r}, \tag{6}$$

where $\hat{p}$ and $\hat{q}$ are the empirical distributions of the marginals p and q sorted in ascending order. With $r = 1$ and equal weight distributions we get the earth mover's distance (EMD) which, in other words, measures the area between the respective CDFs with L_1 as ground distance.

We assume a sample size of at least 100 to be enough for the EMD calculation to be valid, as fewer samples might not guarantee a sufficient sampling of the domains. In our experiments we use sample sizes ≥ 500.

Table 1 Due to differences in label definition of real-world datasets, the class mapping for training and evaluation is decreased to 11 classes that are common in all considered datasets: A2D2 [GKM+20], Cityscapes [COR+16], Berkeley Deep Drive (BDD100K) [YCW+20], Mapillary Vistas (MV) [NOBK17], India Driving Dataset (IDD) [VSN+18], GTAV [RVRK16], our synthetic dataset [KI 20], and Synscapes [WU18]

s	0	1	2	3	4	5	6	7	8	9	10
Label	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	human	car	truck

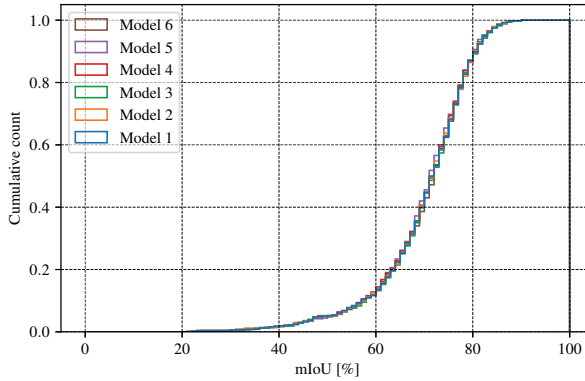


Fig. 4 CDF of ensembles of `DeeplabV3+` models trained on Cityscapes and evaluated on its validation set. Applying the 2-sample Kolmogorov-Smirnov test to each possible pair of the ensemble, we get a minimum $p-$value > 0.95

The FID is a special case of the Wasserstein-2 distance derived from (6) with $p = 2$ and $\hat{p}$ and $\hat{q}$ being normally distributed, leading to the following definition:

$$\text{FID} = ||\boldsymbol{\mu} - \boldsymbol{\mu}_w||^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}), \tag{7}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_w$ are the means, and Σ and Σ_w are the covariance matrices of the multivariate Gaussian-distributed feature vectors of synthetic and real-world datasets, respectively.

Compared to distance metrics such as the FID which by definition is symmetric, our measure is a divergence, i.e., the distance from dataset A to dataset B can be different to the distance from dataset B to dataset A. Being a divergence also reflects the characteristic of a classifier having different generalization distance when trained on dataset A and evaluated on dataset B or the other way around.

Because the ground measure of the signatures, i.e., the IoU per image, is bounded to $0 \leq IoU_n \leq 100$, the EMD measure is then bounded to $0 \leq EMD \leq 100^r$ with r being the Wasserstein norm. For $r = 1$, the measure is bound with $0 \leq EMD \leq 100$.

To verify whether the per-image IoU of a dataset is a good proxy of a dataset's domain distribution, we need to verify that the distribution stays (nearly) constant when training from different starting conditions. Therefore, we trained six models of the `DeeplabV3+` network with the same hyperparameters but different random initialization on the Cityscapes dataset and evaluated them on the validation set calculating the mIoU per image. The resulting distributions of each model in the ensemble are converted into a CDF as is shown in Fig. 4. To have a stronger empirical evidence of the per-image mIoU performance distribution being constant for a dataset, we apply the two-sample Kolmogorov-Smirnov test on each pair of distri-

bution in the ensemble. The resulting p-values are at least > 0.95, hence supporting our hypothesis.

3.3 Datasets

For our sensor parameter optimization experiments we consider two datasets. First, the real-world Cityscapes dataset, which consists of 2,975 annotated images for training and 500 annotated images for validation. All images were captured in urban street scenes in German cities. Second, the synthetic dataset provided by the KI-A project [KI 20]. This dataset consists of 21,802 annotated training images and 5,164 validation images. The KI-A synthetic dataset comprises urban street scenes, similar to Cityscapes, and suburban to rural street scenes which are characterized by less traffic and less dense house placements, therefore more vegetation and terrain objects.

4 Results and Discussion

4.1 Sensor Parameter Extraction

As a baseline for our sensor simulation, we analyzed images from the Cityscapes training data and measured the parameters. Sensor noise was extracted from about 10 images with uniformly colored areas ranging from dark to light colors. Chromatic aberration was extracted from 10 images with traffic signs on the outmost edges of the image, as can be seen in Fig. 5. The extracted values have been averaged over the count of images. The starting parameters of our optimization approach are then as follows: saturation $= 2.0\%$, noise $\sim \mathcal{N}(0,\ 3)$, $\gamma = 0.8$, $F = 4$, and $ca = 0.08$.

4.2 Sensor Artifact Optimization Experiment

Utilizing the EMD as dataset divergence measure and the extracted sensor parameters from camera images of Cityscapes, we apply an optimization strategy to iteratively decrease the gap between the Cityscapes and the synthetic dataset [KI 20]. For optimization, we chose to use the trust region reflective (trf) method [SLA+15] as implemented in `SciPy` [VGO+20]. The trf is a least-squares minimization method to find the local minimum of a cost function given certain input variables. The cost function is the EMD from synthetic model and real-world model predictions on the same real-world validation dataset. The variables as input to the cost function are the parameters of the sensor artifact simulation. The trf method has the capability

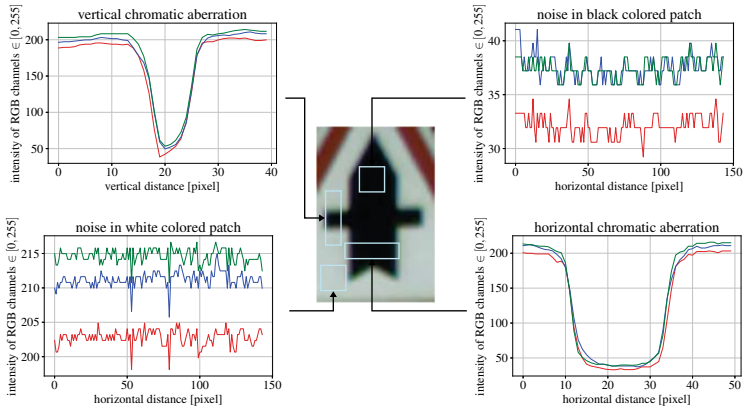


Fig. 5 Exemplary manual extraction of sensor parameters from an extracted patch of a Cityscapes image on a traffic sign in the top right corner. Diagrams clockwise beginning top left: vertical chromatic aberration, noise level on black area, horizontal chromatic aberration, noise level on a plain white area

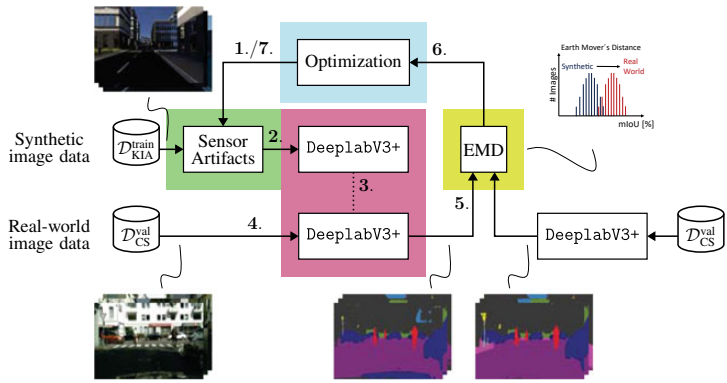


Fig. 6 Optimization of sensor artifacts to decrease divergence between real and synthetic datasets

of bounding the variables to meaningful ranges. The stop criterion is met when the increase of parameter step size or decrease of the cost function is below 10^{-6}.

The overall description of our optimization method is depicted in Fig. 6. *Step 1*: Initial parameters from the optimization method are applied in the sensor artifact simulation to the synthetic images. *Step 2*: The `DeeplabV3+` model with `ResNet101` backbone is pre-trained on 15 epochs on the original unmodified synthetic dataset and finetuned for one epoch on the synthetic dataset with applied sensor artifacts and

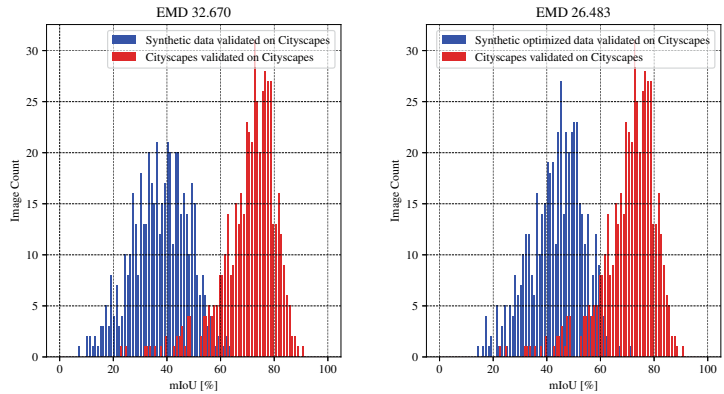


Fig. 7 EMD domain divergence calculation of synthetic and optimized synthetic data to real-world images. **a** Comparison of synthetic data with Cityscapes data, **b** synthetic sensor artifact optimized dataset compared to the target dataset Cityscapes

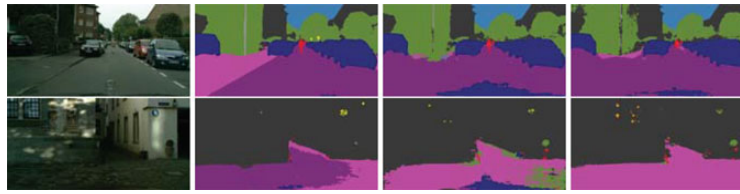
a learning rate of 0.1. *Step 3*: The model parameters are frozen and set to evaluate. *Step 4*: The model predicts on the validation set of the Cityscapes dataset. *Step 5*: The remaining domain divergence is measured by evaluation of the mIoU per image and calculation of the EMD to the evaluations of a model trained on Cityscapes. *Step 6*: The resulting EMD is fed as cost to the optimization method. *Step 7*: New parameters are set for the sensor artifact simulation, or the optimization ends if the stop criteria are met.

After iterating the parameter optimization with the trf method, we compare our optimized trained model with the unmodified synthetic dataset by their per-image mIoU distributions on the Cityscapes dataset. Figure 7 depicts the distributions resulting from this evaluation. The `DeeplabV3+` model trained with the optimized sensor artifact simulation applied on the synthetic dataset outperforms the baseline and achieves an EMD score of 26.48, while decreasing the domain gap by 6.19. The resulting parameters are saturation $= 2.11\%$, noise $\sim \mathcal{N}(0,\ 3.0000005)$, $\gamma = 0.800001$, $F = 4$ and $ca = 0.008000005$. The parameters changed only slightly from the starting point, indicating the extracted parameters as good first choice.

An exemplary visual inspection of the results in Fig. 8 helps to understand the distribution shift and therefore the decreased EMD. While the best prediction performance image (top row) increased only slightly from the synthetic trained model (c) to the sensor artifact optimized model (d), the worst prediction case (bottom row) shows improved segmentation performance for the sensor-artifact-optimized model (d), in this case even better than the Cityscapes trained model (b).

Table 2 Performance results as per-class mIoU, overall mIoU, and EMD domain divergence evaluated on Cityscapes with models trained on Cityscapes, our synthetic only, synthetic with random parameterized lens artifacts, and synthetic extracted parameterized lens artifacts datasets. For the latter two, there are models evaluated on Cityscapes with and without optimization of the parameters for the sensor lens artifact simulation. The model trained with optimized extracted parameters achieves the highest performance on the Cityscapes dataset

Model trained on	Optimized	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	human	car	truck	mIoU ↑	EMD ↓
Cityscapes	no	97.50	81.21	92.22	54.54	57.28	70.26	92.33	93.98	82.55	93.67	81.62	81.56	–
w/o artifacts	no	60.46	23.51	71.99	10.70	26.00	13.47	75.72	74.70	51.27	34.46	1.74	40.37	32.67
w/ random artifacts	no	77.86	20.56	60.66	8.21	17.83	7.36	72.18	68.21	50.53	74.01	4.68	41.58	30.03
w/ extracted artifacts	no	80.65	27.17	66.54	10.43	21.64	11.87	68.82	67.99	59.22	70.06	**7.39**	44.71	29.84
w/ random artifacts	yes	**82.89**	**34.69**	71.11	11.29	16.66	9.12	69.81	**76.77**	**61.48**	**79.50**	5.30	45.76	26.58
w/ extracted artifacts	yes	79.62	30.30	**75.74**	**16.30**	**28.31**	**13.86**	**78.75**	70.88	59.74	64.06	6.42	**47.63**	**26.48**



(a) Original **(b)** CS Pred **(c)** Synth Pred **(d)** Synth Opt Pred

Fig. 8 Top row: Best performance predictions. Bottom row: Worst performance predictions. **a** Original, **b** Cityscapes-trained model prediction, **c** synthetically trained model prediction, and **d** sensor artifact optimized trained model prediction. While top performance increased only slightly, the optimization lead to more robust predictions in worst case, i.e., harder examples

We compare the overall mIoU performance on the Cityscapes datasets between models trained with the initial unmodified synthetic dataset, the synthetic dataset with random initialized lens artifact parameters, and the synthetic dataset with extracted parameters from Cityscapes with the baseline of a model trained on the Cityscapes dataset. Results are listed in Table 2 (rows 1–4). Additionally, for the random and the extracted parameters, we evaluate the performance with initial and optimized parameters, where the parameters have been optimized by our EMD minimization (rows 5 and 6). While the model without any sensor simulation achieves the lowest overall performance (row 2), the model with random parameter initialization achieves a slightly higher performance (row 3) and is surpassed by the model with the Cityscapes extracted parameters (row 4). Next, we take the models trained with optimized parameters into account (rows 5 and 6). Both models outperform all non-optimized experiment settings in terms of overall mIoU, with the model using optimized extracted parameters from Cityscapes showing the best overall mIoU (row 6). Concretely, the model trained with optimized random starting parameters achieves higher performance on classes road, sidewalk, human, and even significantly on the car class but still falls behind on five of the remaining classes and the overall performance on the Cityscapes dataset (row 5). Further, the random parameter optimized model took over 22 iterations to converge to its local minimum, whereas the optimization of extracted starting parameters only took six iterations until reaching a local minimum, making it more than three times faster to converge. Furthermore, it is shown that all models with applied sensor lens artifacts outperform the model trained without additional lens artifacts.

4.3 EMD Cross-evaluation

To get a deeper understanding of the implications of our EMD score, we evaluate our EMD results on a range of real-world and synthetic datasets for semantic segmentation. Including real-world datasets A2D2 [GKM+20], Cityscapes (CS) [COR+16],

Table 3 Cross-domain divergence results of models trained on different real-world and synthetic datasets and evaluated on various validation or test sets of an average size of 1000 images. The domain divergence is measured with our proposed EMD measure; boldface values indicate the lowest divergence comparing our synthetic (Synth) and synthetic-optimized (SynthOpt) datasets, whereas underlined values indicate the lowest divergence values over all the datasets. The model trained with optimized lens artifacts applied to the synthetic images exhibits a smaller domain divergence than the model trained without lens artifacts

EMD ↓	A2D2	BDD100K	CS	GTAV	IDD	MV	SYNS	Synth	SynthOpt
A2D2	–	18.70	23.46	37.84	20.03	10.72	46.78	34.95	**29.32**
BDD100K	6.36	–	9.45	22.14	7.26	1.42	36.33	26.43	**21.54**
CS	10.90	12.09	–	36.42	13.01	4.08	20.62	32.66	**26.48**
GTAV	33.28	28.37	29.72	–	30.55	23.30	37.53	36.08	**32.94**
IDD	24.37	19.83	24.71	34.71	–	12.81	46.23	41.64	**36.95**
MV	10.63	10.36	14.34	28.35	9.20	–	35.97	30.35	**27.03**
SYNS	25.45	31.46	23.64	45.16	25.12	23.45	–	43.76	**43.56**

Berkeley Deep Drive (BDD100K) [YCW+20], Mapillary Vistas (MV) [NOBK17], India Driving Dataset (IDD) [VSN+18], as well as synthetic GTAV [RVRK16], our synthetic (Synth and SynthOpt) [KI 20], and Synscapes (SYNS) [WU18] datasets. In Table 3 the results of cross-domain analysis measured with the EMD score are depicted. The columns denote that a `DeeplabV3+` model has been trained on the corresponding dataset, i.e., the source dataset, whereas the rows denote the datasets it was evaluated on, i.e., the target datasets. Our optimized synthetic dataset achieves lower EMD scores, shown in boldface, than the synthetic baseline. While the domain divergence decrease is high on real datasets, the divergence decreased only marginally for the other synthetic datasets. Inspecting the EMD result on all datasets, the lowest divergence values are indicated by underline; the MV dataset shows to be closest to all the other evaluated datasets.

To set our measure in relation to established domain distance measures, we calculated the FID from each of our considered datasets to one another. The results are shown in Table 4. The FID, defined in (7), is the Wasserstein-2 distance of feature vectors from the `InceptionV3` [SVI+16] network sampled on the two datasets to be compared with each other.

Again, boldface values indicate the lowest FID values between the synthetic (Synth) and synthetic-optimized (SynthOpt) datasets, whereas underlined values indicate the lowest values of all datasets. Here, only 4 out of the 7 datasets are closer, measured by the FID, to the synthetic-optimized dataset than to the original dataset. Furthermore, the FID sees the CS and the SYNS dataset closer to one another than the EMD divergence measure, while the MV dataset shows the lowest FID among the other evaluated datasets.

FID and EMD somewhat agree, if we evaluate the distance as minimum per-row in both tables, that the Mapillary Vistas dataset is in most cases the dataset that is closest to all other datasets.

Table 4 Cross-domain distance results measured with the Fréchet inception distance (FID). Lowest FID between synthetic (Synth) and synthetic optimized (SynthOpt) datasets are in boldface, whereas the lowest FID values over all datasets are underlined

FID ↓	A2D2	BDD100K	CS	GTAV	IDD	MV	SYNS	Synth	SynthOpt
A2D2	–	60.16	98.46	78.16	58.75	41.84	109.35	**116.54**	121.55
BDD100K	60.16	–	59.90	62.42	52.15	29.66	74.871	115.51	**109.08**
CS	98.46	59.90	–	85.81	68.92	59.69	43.87	119.97	**112.42**
GTAV	78.16	62.42	85.81	–	74.08	51.00	89.62	92.513	**92.24**
IDD	58.75	52.15	68.92	74.08	–	37.36	64.09	**118.06**	125.30
MV	41.84	29.66	59.69	51.00	37.36	–	70.24	**74.46**	78.66
SYNS	109.35	74.87	43.87	89.62	64.09	70.24	–	113.77	**108.3**

Now, calculating the minimum per-column in both tables, the benefit of our asymmetric EMD comes to the light. The minimum per-column values of the FID are unchanged due to the diagonal symmetry of the cross-evaluation matrix stemming from the inherent symmetry of the measure. However, the EMD regards the BDD100K as the closest dataset. An intuitive explanation for the different minimum observations of the EMD is as follows: Training with many images exhibiting different geospatial and sensor properties of the Mapillary Vistas dataset covers a very broad domain and results in good generalization capability and therefore evaluation performance. Training with any of the other datasets cannot generalize well to the vast domain of Mapillary Vistas but to the rather constrained domain of BDD100K, which consists of lower resolution images with heavy compression artifacts, where even a model that has been trained on BDD100K does not generalize well on.

The asymmetric nature of our EMD allows for a more thorough and complex analysis of dataset discrepancies, when applied to the tasks of visual understanding, e.g., semantic segmentation, which otherwise cannot be captured by inherently symmetric distance metrics such as FID. Contrasting to [LLFW20], we could with our evaluation method not identify a consistency between FID and the generalization divergence, i.e., our EMD measure.

5 Conclusions

In this chapter, we could demonstrate that by utilizing the performance metric per image as a proxy distribution for a dataset and the earth mover's distance (EMD) as a divergence measure between distributions, one can decrease visual differences of a synthetic dataset through optimization and increase the viability of CGI for training and validation purposes of perceptive AI. To reinforce our argument for per-image performance measures as proxy distributions, we showed that training an ensemble of a fixed model with different random starting conditions but with the same

hyperparameters leads to the same per-image performance distributions when these ensemble models are evaluated on the validation set of the training dataset. When utilizing synthetic imagery for validation, the domain gap, due to visual differences between real and computer-generated images, is hindering the applicability of these datasets. As a step toward decreasing the visual differences, we apply the proposed divergence measure as a cost function to an optimization which varies the parameters of the sensor artifact simulation, while trying to re-create the sensor artifacts that the real-world dataset exhibits. As starting point of the sensor artifact parameters, we extracted empirically the values from chosen images of the real-world dataset. The optimization improved the visual difference between the real-world and the optimized synthetic dataset measurably by the EMD and we could show that even when starting with random initialized parameters we can decrease the EMD and increase the mIoU on the target datasets. When measuring the divergence after parameter optimization to other real-world and synthetic datasets, we could show that the EMD decreases for all considered datasets but when measured by the FID only four of the datasets are closer. As the EMD is derived from the mIoU per image, it is an indicator of performance on the target dataset, whereas the FID fails to relate with performance. Effective minimization of the visual difference between synthetic and real-world datasets with the EMD domain divergence measure is one step further toward fully utilizing CGI for validation of perceptive AI functions.

Acknowledgements 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

[BB95] W. Burger, M.J. Barth, Virtual reality for enhanced computer vision, in J. Rix, S. Haas, J. Teixeira (eds.), *Virtual Prototyping: Virtual Environments and the Product Design Process* (Springer, 1995), pp. 247–257

[Bis95] M. Christopher Bishop, Training with noise is equivalent to Tikhonov regularization. Neural Comput. **7**(1), 108–116 (1995)

[BSAG21] M. Binkowski, D.J. Sutherland, M. Arbel, A. Gretton, *Demystifying MMD GANs*, Jan. 2021, pp. 1–36. arxiv:1801.01401

[CLP13] G. Csurka, D. Larlus, F. Perronnin, What is a good evaluation measure for semantic segmentation? in *Proceedings of the British Machine Vision Conference (BMVC)*, Bristol, UK, Sept. 2013, pp. 1–11

[COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, 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, June 2016, pp. 3213–3223

[CSVJR18] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, Modeling camera effects to improve visual learning from synthetic data, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, Munich, Germany, Aug. 2018, pp. 505–520

[CSVJR19] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, *Sensor Transfer: Learning Optimal Sensor Effect Image Augmentation for Sim-to-Real Domain Adaptation*, Jan. 2019, pp. 1–8. arxiv:1809.06256

[CV14] V. Chari, A. Veeraraghavan, L. Distortion, Radial distortion, in *Computer Vision: A Reference Guide*. ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 443–445

[CZP+18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with atrous separable convolution for semantic image segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 833–851

[dCCN+16] G.B.P. da Costa, W.A. Contato, T.S. Nazaré, J.E.S. do Batista Neto, M. Ponti, *An Empirical Study on the Effects of Different Types of Noise in Image Classification Tasks*, Sept. 2016, pp. 1–6. arxiv:1609.02781

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: a large-scale hierarchical image database, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, June 2009, pp. 248–255

[DRC+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in *Proceedings of the Conference on Robot Learning CORL*, Mountain View, CA, USA, Nov. 2017, pp. 1–16

[FMWR18] E. Fernandez-Moral, R. Martins, D. Wolf, P. Rives, A new metric for evaluating semantic segmentation: leveraging global and contour accuracy, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, June 2018, pp. 1051–1056

[GKM+20] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V.H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, P. Schuberth, *A2D2: Audi Autonomous Driving Dataset*, April 2020, pp. 1–10. arxiv:2004.06320

[GL15] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1180–1189

[HRU+17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6626–6637

[HTP+18] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, CyCADA: cycle-consistent adversarial domain adaptation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 1989–1998

[HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778

[KI 20] KI Absicherung Consortium. KI Absicherung: Safe AI for Automated Driving (2020). Assessed 18 Nov. 2021

[KPT+17] S. Kolouri, S.R. Park, M. Thorpe, D. Slepcev, G.K. Rohde, Optimal mass transport: signal processing and machine-learning applications. IEEE Signal Process. Mag. **34**(4), 43–59 (2017)

[LCWJ15] M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, July 2015, pp. 97–105

[LLFW20] Z. Liu, T. Lian, J. Farrell, B. Wandell, Neural network generalization: the impact of camera parameters. IEEE Access **8**, 10443–10454 (2020)

[NdCCP18] T.S. Nazaré, G.B.P. da Costa, W.A. Contato, M. Ponti, Deep convolutional neural networks and noisy images, in *Proceedings of the Iberoamerican Congress on Pattern Recognition (CIARP)*, Madrid, Spain, Nov. 2018, pp. 416–424

[NOBK17] G. Neuhold, T. Ollmann, S. Rota Bulò, P. Kontschieder, The mapillary vistas dataset for semantic understanding of street scenes, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 4990–4999

[PTKY09] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Pasadena, CA, USA, July 2009, pp. 1187–1192

[RD14] R. Ramanath, M.S. Drew, Color Spaces, in *Computer Vision: A Reference Guide*. ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 123–132

[RHK17] S.R. Richter, Z. Hayder, V. Koltun, Playing for benchmarks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2232–2241

[RSM+16] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 3234–3243

[RTC17] A. Ramdas, N. García Trillos, M. Cuturi, On Wasserstein two sample testing and related families of nonparametric tests. Entropy **19**(2), 47 (2017)

[RTG+19] H. Rezatofighi, N. Tsoi, J.Y. Gwak, A. Sadeghian, I. Reid, S. 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)*, Long Beach, CA, USA, June 2019, pp. 658–666

[RV19] S.V. Ravuri, O. Vinyals, Seeing is not necessarily believing: limitations of BigGANs for data augmentation, in: *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, New Orleans, LA, USA, June 2019, pp. 1–5

[RVRK16] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: ground truth from computer games, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 102–118

[SAS+18] F.S. Saleh, M.S. Aliakbarian, M. Salzmann, L. Petersson, J.M. Alvarez, Effective use of synthetic data for urban scene semantic segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 84–100

[SGH20] Q.S. Sha, O. Grau, K. Hagn, DNN analysis through synthetic data variation, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)*, virtual conference, Dec. 2020, pp. 1–10

[SGZ+16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, Xi Chen, *Improved Techniques for Training GANs*, June 2016, pp. 1–10. arxiv:1606.03498

[SLA+15] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1889–1897

[Stu14] P. Sturm, Pinhole Camera Model, in *Computer Vision: A Reference Guide*. ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 610–613

[SVI+16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 2818–2826

[THS+18] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 7472–7481

[THSD17] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 2962–2971

[VGO+20] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson,

K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, l. Polat, Y. Feng, E.W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributers. Fundamental algorithms for scientific computing in python. SciPy 1.0. Nat. Methods **17**, 261–272 (2020)

[VSN+18] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, C.V. Jawahar, *IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments*, Nov. 2018, pp. 1–9. arxiv:1811.10200

[WU18] M. Wrenninge, J. Unger, *Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing*, Oct. 2018, pp. 1–13. arxiv:1810.08705

[YCW+20] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, BDD100K: a diverse driving dataset for heterogeneous multitask learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 2636–2645

7.4 Publication 4

A Variational Deep Synthesis Approach for Perception Validation

Oliver Grau, Korbinian Hagn and Qutub Syed Sha

Published in:

Fingscheidt, T., Gottschalk, H., Houben, S. (eds) Deep Neural Networks and Data for Automated Driving. Springer, Cham. [GHS22]

DOI: 10.1007/978-3-031-01233-4_13

A Variational Deep Synthesis Approach for Perception Validation

Oliver Grau, Korbinian Hagn, and Qutub Syed Sha

Abstract This chapter introduces a novel data synthesis framework for validation of perception functions based on machine learning to ensure the safety and functionality of these systems, specifically in the context of automated driving. The main contributions are the introduction of a generative, parametric description of three-dimensional scenarios in a validation parameter space, and layered scene generation process to reduce the computational effort. Specifically, we combine a module for probabilistic scene generation, a variation engine for scene parameters, and a more realistic sensor artifacts simulation. The work demonstrates the effectiveness of the framework for the perception of pedestrians in urban environments based on various deep neural networks (DNNs) for semantic segmentation and object detection. Our approach allows a systematic evaluation of a high number of different objects and combined with our variational approach we can effectively simulate and test a wide range of additional conditions as, e.g., various illuminations. We can demonstrate that our generative approach produces a better approximation of the spatial object distribution to real datasets, compared to hand-crafted 3D scenes.

1 Introduction

This chapter introduces an automated data synthesis approach for the validation of perception functions based on a generative and parameterized synthetic data generation. We introduce a multi-stage strategy to sample the input domain of the possible generative scenario and sensor space and discuss techniques to reduce the required vast amount of computational effort. This concept is an extension and generaliza-

O. Grau ( ) · K. Hagn · Q. Syed Sha
Intel Deutschland GmbH, Lilienthalstraße 15, 85579 Neubiberg, Germany
e-mail: oliver.grau@intel.com

K. Hagn
e-mail: korbinian.hagn@intel.com

Q. Syed Sha
e-mail: syed.qutub@intel.com

T. Fingscheidt et al. (eds.), *Deep Neural Networks and Data for Automated Driving*,
https://doi.org/10.1007/978-3-031-01233-4_13

tion of our previous work on parameterization of the scene parameters of concrete scenarios, called validation parameter space (VPS) [SGH20]. We extend this parameterization by a probabilistic scene generator to widen the coverage of the generated scenarios and a more realistic sensor simulation, which also allows to variate and simulate different sensor characteristics. This 'deep' synthesis concept overcomes currently available systems (as discussed in the next section) or manually, i.e., by human-operator-generated synthetic data. We describe, how our synthetic data validation engine makes use of the parameterized, generative content to implement a tool supporting complex and effective validation strategies.

Perception is one of the hardest problems to solve in any automated system. Recently, great progress has been made in applying machine learning techniques to deep neural networks to solve perceptional problems. Automated vehicles (AVs) are a recent focus as an important application of perception from cameras and other sensors, such as LiDAR and RaDAR [YLCT20]. 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. Universally accepted methodologies for validating safety of machine learning-based systems are still an open research topic.

Techniques to capture and render models of the real world have matured significantly over the last decades and are now able to synthesize virtual scenes in a visual quality that is hard to distinguish from real photographs for human observers. Computer-generated imagery (CGI) is increasingly popular for training and validation of deep neural networks (DNNs) (see, e.g., [RHK17, Nik19]). 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.

Due to the progress in visual and multi-sensor synthesis, building systems for validation of these complex systems in the data center becomes feasible now and offers more possibilities for the integration of intelligent techniques in the engineering process of complex applications. We compare our approach with methods and strategies targeting testing of automated driving [JWKW18].

The remainder of this chapter is structured as follows: The next section will give an outline of related work in the field. In Sect. 3 we give an overview of our approach. Section 4 describes an outline of our synthetic data validation engine, our parameterization, including a realistic sensor simulation, and the effective computation of the required variations. In Sect. 5 we present evaluation results, followed by Sect. 6 with some concluding remarks.

2 Related Work

The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g., [BB95]). Several methodologies for verification and validation of AVs have been developed [KP16, JWKW18, DG18] and commercial options exist.[1] These tools were originally designed for virtual testing of automotive functions, such as 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 system.

What is getting presented in this chapter is focusing on the validation of perception functions, which is an essential module of automated systems. However, by separating the perception as a component, the validation problem can also be decoupled from the validation of the full driving stack. Moreover, this separation allows, on the one hand, the implementation of various more specialized validation strategies and, on the other hand, there is no need to simulate dynamic actors and the connected problem of interrelations between them and the ego-vehicle. The full interaction of objects is targeted by upcoming standards like OpenScenario.[2]

Recently, specifically in the domain of driving scenarios, game engines have been adopted for synthetic data generation by extraction of in-game images and labels from the rendering pipeline [WEG+00, RVRK16]. Another virtual simulator system, which gained popularity in the research community, is CARLA [DRC+17], also based on a commercial game engine (`Unreal4` [Epi04]). Although game engines provide a good starting point to simulate environments, they usually only offer a closed rendering setup 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 8 bit 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, as in the Synscapes dataset [WU18]. For our experimental deep synthesis work, we use the physical-based open-source `Blender Cycles` renderer[3] in high dynamic range (HDR) resolution, which allows realistic simulation of illumination and sensor characteristics increasing the coverage of our synthetic data in terms of scene situations and optical phenomena occurring in real-world scenarios.

The effect of sensor and lens effects on perception performance has not been studied a lot. In [CSVJR18, LLFW20], 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 [LLFW20] and

[1] For example, `Carmaker` from IPG or `PreScan` from TASS International.

[2] https://www.asam.net/standards/detail/openscenario/.

[3] https://www.blender.org/.

[CSVJR19]. A sensor model including sensor noise, lens blur, and chromatic aberration was developed based on real datasets [HG21] and integrated into our validation framework.

Looking at virtual scene content, the most recent simulation systems for validation of a 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 aimed to simulate environments spanning a huge virtual space and are then virtually driving a high number of test miles (or km) in the virtual world provided [MBM18, WPC20, DG18]. Although this might be a good strategy to validate full AD stacks, one remaining problem for validation of perception systems is the limited coverage of data testing critical scene constellations (sometimes called 'corner cases') and parameters that lead to drop in performance of the DNN perception.

A more suitable approach is to use probabilistic grammar systems [DKF20, WU18] 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. In this chapter we demonstrate the effectiveness of a simple probabilistic grammar system together with our previous scene parameter variation [SGH20] with a novel multi-stage strategy. This approach allows to systematically test conditions and relevant parameters for validation of perceptional function in a structured way.

3 Concept and Overview

The novelty of the framework introduced in this chapter is the combination of modules for parameterized generation and testing of a wide range of scenarios and scene parameters as well as sensor parameters. It is tailored towards exploration of factors that (hypothetically) define and limit the performance of perception modules.

A core design feature of the framework is the consequent parameterization of the scene composition, scene, and sensor parameters into a *validation parameter space* (VPS) as outlined in Sect. 4.2. This parameterization only considers the near proximity of the ego-car or sensor; in other words, only the objects visible to the sensor are generated. This allows a much more well-defined test of constellations involving a specific number of object types, environment topology (e.g., types and dimensions of streets), and relation of objects, usually as an implicit function of where objects are positioned relative in the scene.

This leads to a different data production and simulator pipeline than for conventional AV validation which typically provides a virtual world with a large extent to simulate and test the driving functions down to a physical level, inspired by real-world validation and test procedures [KP16, MBM18, DG18, JWKW18, WPC20].

Figure 1 shows the building blocks of our `VALERIE` system. The system runs an expansion of the VPS specified in the 'validation task' description. Our current implementation is based on a probabilistic description of how to generate the scene

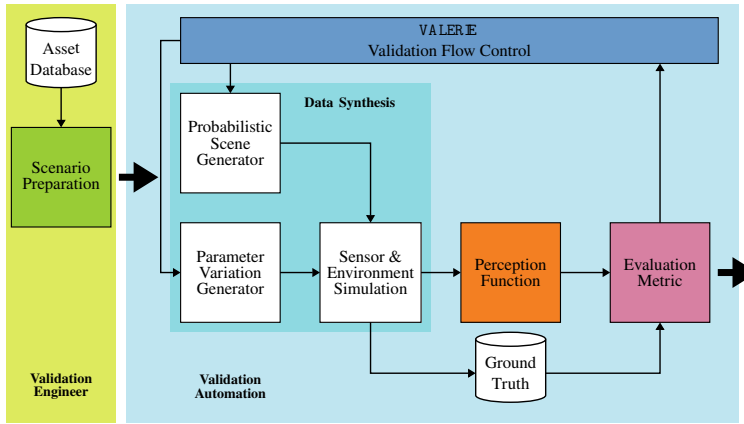


Fig. 1 Block diagram of the proposed validation approach

and defines the parameter variations in the parameter space. In the future, the validation task should also include a more abstract target description of the evaluation metrics.

The *data synthesis* block consists of three sub-components: The *probabilistic scene generator* generates a scene constellation, including a street layout, and places three-dimensional objects from the *asset database* according to probabilistic placement rules laid out in the scenario preparation. The *parameter variation generator* produces variations of that scene, including sun and light settings and variations of the placement of objects (see Fig. 2 for some examples). The *sensor & environment simulation* is using a rendering engine to compute a realistic simulation of the sensor impressions.

Further, ground truth data is provided through the rendering process, which can be used for a pixel-accurate depth map (distance from camera to scene object) or meta data, like pixel-wise label identifiers of classes or object instances. Depending on the perception task, this information is specifically used for training and evaluation of semantic segmentation (see Sect. 4.5).

The output of the sensor simulation is passed to the *perception function* under test and the response to that data is computed. An evaluation metric specific to the validation task is based on the perception response. The *ground truth* data, as generated by the rendering process is usually required here, e.g., to compute the similarity to the known appearance of objects. In the experiments presented in this chapter we used known performance metrics for DNNs, such as the mean intersection-over-union (mIoU) metric, as introduced by [EVGW+15].

The parameterization along with the computation flow are described in detail in the next section.

4 VALERIE: Computational Deep Validation

The goal of validation is usually to demonstrate that the perception function (DNN) is performing to a level defined by the validation goal for all cases included and specified in the ODD (operational design domain) [SAE18].

The framework presented in this contribution supports the validation of perception with the data synthesis modules outlined above. Further, we suggest to consider three levels or layers, which are supported in our framework:

1. The *scene variation* generates 3D scenes using a probabilistic grammar.
2. The *scene parameter variation* generates variations of scene parameters, such as moving objects or changing the illumination of the scene by changing the time of the day.
3. The *sensor variation* generates sensor faults and artifacts.

For an actual DNN validation, an engineer or team would build various scenarios and specify variations within these scenarios. The variations can contain lists of alternative objects (assets), different topology and poses (including position and orientation of objects) and expansions of streets and object and global parameters such as direction of the sun. Our modular multi-level approach enables strategies to sample the input space, typically by applying scene variation and this can be then combined with more in-depth parameter variation runs and variation of sensor parameters, as required.

Specifically, the ability to either combine two or all three levels of our framework allows to cover a wider range of object and scene constellations. In particular, with our integrated asset management approach, a validation engineer can ensure that certain, e.g., known critical object classes are included in the validation runs, i.e., he can explicitly control the coverage of these object classes. By combination with our parameter variation sub-system, local changes are varied, including relative positioning of critical objects, the positioning of the ego-vehicle or camera, global scene parameters such as the sun angle, etc. can be achieved.

4.1 Scene Generator

In computer graphics, a scene is considered as a collection $\mathcal{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots\}$ of objects $\mathbf{o}_i$ and these are usually organized in scene graphs (see, e.g., [Wer94]) and this model is also basis for file format specifications to exchange 3D models and scenes, e.g., VRML[4] or glTF.[5]

Each object in this graph can have a position and orientation and scale in a (world) coordinate system. These are usually combined into a transformation matrix $\mathbf{T}$. Several parameterizations for position and orientations are possible, for the position

[4] ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2004 https://www.web3d.org.
[5] www.khronos.org/gltf.

Table 1 Example placement description (`json` format)

```
{ "class_id" : "tree",
  "copies" : 5,
  "target": { "v1" : [59.0, 20.0, 0.15],   "v2": [59.8, 72.50, 0.15] },
  "rot_range" : [0.0, 360.0],
  "asset_list" : [
   "e649326c-061e-4881-b0a8-f2ad6297124c",
   "3a742848-4166-4256-a045-c4a8f8ef94bc",
   "4e153905-0ed6-4077-be33-74ef4c7509f5" ] }
```

usually a Cartesian 3D vector for orientation notations such as Euler angles or quaternions are common.

Objects $\mathbf{o}_i$ are described as geometry, e.g., as a triangular mesh and appearance (material).

Sensors, such as a camera, can also be represented in a scene graph and so can be light sources. Both also have a position and orientation, and accordingly, the same transformation matrices as for objects can be applied (except scaling).

The probabilistic scene generator (depicted in Fig. 1) places objects $\mathbf{o}_i$ according to a grammar file that specifies rules for placements and orientations in specific areas of the 3D scene. The example `json` file in Table 1 specifies the placement of tree objects in a rectangular area of the scene: The `tree` objects are randomly drawn from a database, the field `assets_list` specifies a list of possible assets in universally unique identifier (UUID) notation. The 3D models in the database are tagged with a `class_id`, which specifies the type of objects, e.g., humans or buildings. The class information will be used in the generation of meta-data, semantic class labels, and an instance object identifier which allows to determine on a pixel level the originating 3D object.

The placement and orientation are determined by a pseudo random number generator, with a controlled seed for the ability to exactly re-run experiments if required. The scene generator handles other constraints, such as specific target densities in specific areas, distant ranges between objects, and it finally checks that the placed object neither collides nor intersects with other objects in the scene.

The street base is also part of the input description for the scene generator. A street can be varied in width, type of crossings, and textures for street and sidewalk. In the current simplistic implementation, the street base is limited to a flat 'lego world', i.e., only rectangular structures are implemented. Each call of the scene generator generated a different randomized scene according to the rules in the generation description. Figure 2 shows scenes generated by a number of runs of the scene generator.

Fig. 2 Examples scenes with randomly selected and placed objects

4.2 Validation Parameter Space (VPS)

Another core aspect of our validation approach is to parameterize all possible variations of scene, sensor parameters, and states in a unified validation parameter space (VPS): Objects in a scene graph can be manipulated by considering their properties or attributes as a list of variable parameters. A qualitative overview of those parameters is given in Table 2. Most attributes are of geometrical nature, but also materials or properties of light sources can be varied, as depicted in Fig. 3.

In addition to static properties, a scene graph can include object properties that vary over time. Some of them are already included in Table 2, such as the trajectories of objects and sensors, indicated as $\mathcal{T} = (\mathbf{T}(t))$, with discrete time instants t. Computer

Table 2 Overview of parameters to vary in a scene

Object class	Variable parameters
Static object, e.g., buildings	Limited to position, orientation, and size
Streets, roads	Geometry (e.g., position, size of lanes, etc.), friction (as function of weather conditions)
Vehicles	$\mathbf{T}_\mathrm{v}$ = (position, orientation), trajectory $\mathcal{T}_\mathrm{v} = (\mathbf{T}_\mathrm{v}(t))$
Humans (pedestrian)	$\mathbf{T}_\mathrm{p}$ = (position, orientation), trajectory $\mathcal{T}_\mathrm{p} = (\mathbf{T}_\mathrm{p}(t))$
Environment	Light, weather conditions
Sensors	$\mathbf{T}_\mathrm{s}$ = (position, orientation), trajectory $\mathcal{T}_\mathrm{s} = (\mathbf{T}_\mathrm{s}(t))$, sensor attributes

Fig. 3 Example of scene parameter variation; in this case the time of the day is varied, causing dramatic changes in the scene illumination according to contrast variations

graphic systems handle these temporal variations, also known as animations, and in principle, any attribute can be varied over time by these systems.

We introduce an important restriction in the current implementation of our validation and simulation engine: Our animations are fixed, i.e., they do not change during the runtime of the simulation in order to allow deterministic and repeatable object appearance, like poses of characters. This could be different for example when a complete autonomous system is simulated, as the actions of the system might change the way other scene agents react. We will include these aspects in the discussion and outlook and will discuss how these aspects could be mitigated.

For the use in our validation engine, as described in the next section, we augment a description of the scene in a scene graph (the asset) as outlined above, with an explicit description of those parameters which are variable in a validation run. Currently, our engine considers a list of numerical parameters with the following attributes:

```
parameter_name, scene_graph_ref, type, minimum, maximum
```

A specific example to describe variations of the position of a person in a 2-D plane in pseudo markup notation is

```
{p1, scene.person-1.pos.x, FLOAT, 0.0, 20.0}
```

Parameters, such as `p1`, are unique parameter identifiers used in the validation engine to produce and test variations of the scene.

4.3 Computation of Synthetic Data

Synthetic data is generated using computer graphics methods. Specifically for color (RGB) images, there are many software systems available, both commercially and as open source. For our experiments in this chapter, we are using `Blender`,[6] as this tool allows importing, editing, and rendering of 3D content, including scripting.

The generation of synthetic data involves the following steps: First, a 3D scene model with a city model and pedestrians is generated using the probabilistic scene generator and is stored in one or more files.

The scene files are loaded into one scene graph and objects have a unique identifier and can be addressed by the following naming convention:

```
root_object.{subcomponent}.attribute
```

For the example used in Sect. 4.2, `scene.person-1.pos.x` refers to a path from the root object `scene` to the object `person-1` and addresses the attribute `pos.x` in `person-1`. The object names are composed of `ObjectClass-ObjectInstanceID`. These conventions are used to assign a class or instance labels during ground truth generation.

The labels for object classes will be mapped to a convention used in annotation formats (i.e., as used with in the Cityscapes dataset [COR+16]) 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.

Using a second parameter `pos.y`, as included in the example in Sect. 4.2, would allow the positioning of the person in a plane, spanned by x- and y-axis of the coordinate system defined by the scene graph.

4.4 Sensor Simulation

We implemented a sensor model with the function blocks described in the chapter 'Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation' [HG22], Sect. 3.1 'Sensor Simulation', and depicted in Fig. 2. The module expects images in linear RGB space and floating point resolution as provided by the state-of-the-art rendering software.

We simulate a camera error model by applying *sensor noise*, as additive Gaussian noise (with zero mean and freely selectable variance) and an automatic, histogram-

[6] www.blender.org.

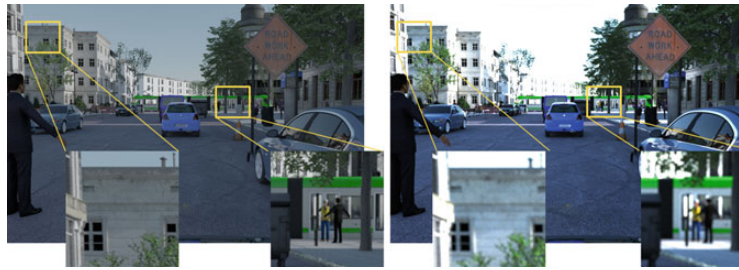


Fig. 4 Realistic sensor effect simulation: Standard `Blender` tone-mapped output (left), and the sensor simulation output (right)

based exposure control (linear tone-mapping), followed by non-linear *Gamma correction*. Further, we simulate the following lens artifacts *chromatic aberration* and *blur*. Figure 4 shows a comparison of the standard tone-mapped 8-bit RGB output of `Blender` (left) with our sensor simulation. The parameters were adapted to match the camera characteristic of Cityscape images. The images do not only look more realistic to the human eye, they also are closing the domain gap between the synthetic and real data (for details see the chapter 'Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation' [HG22]).

4.5 Computation and Evaluation of Perceptional Functions

Perception functions consist of a multitude of different approaches considering the wide range of different tasks. For experiments presented in this chapter, we are considering the tasks of semantic segmentation and 2D bounding box detection. In the first 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.

For semantic segmentation we consider two different topologies: `DeeplabV3+` as proposed in [CPK+18] and `Detectron2` [WKM+19], both are utilizing `ResNet101` [HZRS16] backbones.

These algorithms are trained on three different datasets to create three different models for evaluation. The first dataset is the Cityscapes dataset [COR+16], a collection of European urban street scenes during daytime with good to medium weather conditions. The second dataset is A2D2 [GKM+20]. Similar to the Cityscapes dataset it is a collection of European urban street scenes and additionally it has sequences from driving on a motorway. The last dataset, KI-A tranche 3, is a synthetic dataset

provided by BIT-TS, a project partner of the KI-Absicherung project,[7] consisting of urban street scenes inspired by the preceding two real-world datasets. 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.

For the second task, the 2D-bounding box detection, we utilize the single-shot multibox detector (SSD) by [LAE+16], a 2D-bounding box detector trained on the synthetic data for pedestrian detection. This bounding box detector is applied on our variational data in Sect. 5.

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 [LSD15]. 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. Another performance measure utilized is the pixel accuracy ($pAcc$) which is defined as follows:

$$pAcc = \frac{TP + TN}{TP + FP + FN + TN}. \tag{1}$$

The number of true positives (TP), true negatives (TN), false positives (FP), and true negatives (TN) are used to calculate $pAcc$, which can also be seen as a measure for correctly predicted pixels over all pixels considered for evaluation.

For the 2D-bounding box detection we are interested in cases where, according to our definition, the performance-limiting factors are within bounds where the network should still be able to correctly predict a reasonable bounding box for each object to detect. For each synthesized and inferred image, the true positive rate (TPR) is calculated. The TPR is defined as the number of correctly detected objects (TP) over the sum of correctly detected and undetected objects (TP+FN). As we are interested in prediction failure cases we can then filter out all images with a true positive rate (TPR) of 1 and are left with images where the detection has omitted objects to detect.

4.6 Controller

The `VALERIE` controller (as depicted in Fig. 1, validation flow control) executes the validation run. This run can be configured in multiple ways depending on how much synthetic data is generated and evaluated. Two aspects have a major influence on this: First, the specification of parameters to be varied, and second, the used sampling strategy, which also depends on the validation goal. Both aspects are briefly described in the following.

Specification of variable validation parameters: As outlined in Sect. 4.2, the approach depends on the provision of a generative scene model. This consists of a parameterized 3D scene model and includes 3D assets in the form of static and

[7] https://www.ki-absicherung-projekt.de/.

dynamic objects. On top of this, we define variable parameters in this scene as an explicit list, as explained in Sect. 4.2.

For the specification of a validation run, all or a subset of these parameters are selected and a range and sampling distribution for that specific parameter is added. For example, to vary the x-position of a person in the scene along a line with the uniform or homogeneous distribution and a step size of 1 m, we define

```
{p1, UNIFORM, 1.5, 5.5, 1.0}
```

The parameters refer to the following: Parameter `p1` refers to parameter declarations of x position of `person-1` in the example of Sect. 4.2. The field `UNIFORM` refers to a uniform sampling distribution. Other modes include `GAUSSIAN` (Gaussian distribution). The parameters 1.5, 5.5, 1.0 refer to the parameter range [1.5...5.5] and the initial step size of 1m.

Sampling of variable validation parameters: The actual expansion or sampling of the validation parameter space can be further configured and influenced in the `VALERIE` controller by selecting a sampler and validation strategy or goal.

The *sampler* object provides an interface to the *controller* to the validation parameter space, considering the parameter ranges and optionally the expected parameter distribution. We support uniform and Gaussian distributions.

In our current implementation, the *controller* can be configured to either sample the validation parameter space by a full grid search, or by a Monte-Carlo random sampling.

However, the step size can be iteratively adapted depending on the validation goal. One option here is to automatically refine the search for edge cases (or corner cases) in the parameter space: As an edge case, we consider here a parameter instance, where the evaluation function is changing between an 'acceptable' state to a 'failed' state (using a continuous performance metric). For our use case of person detection, that means a drop in the performance metric below a threshold.

Other validation goals we are planning to implement could be the automated determination of sensitive parameters or (ultimately) more intelligent search through high-dimensional validation parameter spaces.

4.7 Computational Aspects and System Scalability

Our approach is designed for execution in data centers. The implementation of the components described above is modular and makes use of containerized modules using `docker`.[8] For the actual execution of the modules we use the `Slurm`[9] scheduling tool, which allows running our validation engine with a high number of variants in parallel, allowing the exploration of many states in the validation parameter space.

[8] www.docker.com.

[9] https://slurm.schedmd.com.

The results presented here are produced on an experimental setup using six dual `Xeon` server nodes, each equipped with 380 GB RAM. The runtime of the rendering process as outlined above is mainly determined by the rendering and in the order of 10...15 min per frame, using the high-quality physically based rendering (PBR) `Cycles` render engine.

5 Evaluation Results and Discussion

To evaluate the effectiveness of our data synthesis approach, we conducted experiments in generating scenes, variation of a few important parameters, and then we evaluated the perception performance including an analysis of performance-limiting factors, such as occlusions and distance to objects.

We used our scene generator to generate variations of street crossings, as depicted in Fig. 2. For these examples a base ground is generated first, with flexible topology (crossings, t-junction) and dimensions of streets, sidewalks, etc. In the next step, buildings, persons, and objects, including cars, traffic signs, etc. , are selected from a database and randomly placed by the scene generator, taking into account the probabilistic description and rules. The approach can handle any number of object assets. The current experimental setup includes a total of about 500 assets, with about 60 different buildings, 180 different person models, and other objects, including vegetation, vehicles, and so on.

Scene parameter variation: Within the generated scenes, we vary the position and orientations of persons and some occluding objects.

Further, we change the illumination by changing the time of the day. This has two main effects: First, it is changing the illumination intensity and color (dominant at sunset and sunrise), and second, it is generating a variation of shadows casted into the scene. In particular, from our experience, the latter creates challenging situations for the perception.

Comparison of object distribution: Fig. 5 shows the spatial distribution of persons in a) the Cityscapes dataset, b) KI-A tranche 3 dataset, and c) a dataset using our generative scene generator, as depicted in Figs. 2 and 3. The diagrams present a top-view of the respective sensor (viewing cone) and the color encodes the frequency of persons within the sensor viewing cone, i.e., they give a representation of distance and direction of persons in all considered frames of the dataset. The real-world Cityscapes dataset has a distribution that corresponds with most persons located left and right of the center, i.e., the street. There are slightly more persons on the right side, which can be explained by the fact that often sidewalks on the left hand are occluded by vehicles from the other road side. The distribution of our dataset resembles as expected this distribution, with slightly less occupation in the distance. In contrast, the distribution of the KI-A tranche 3 dataset shows a very sharp cumulation of the distribution on what corresponds to a narrow band on the sidewalks of their 3D simulation.

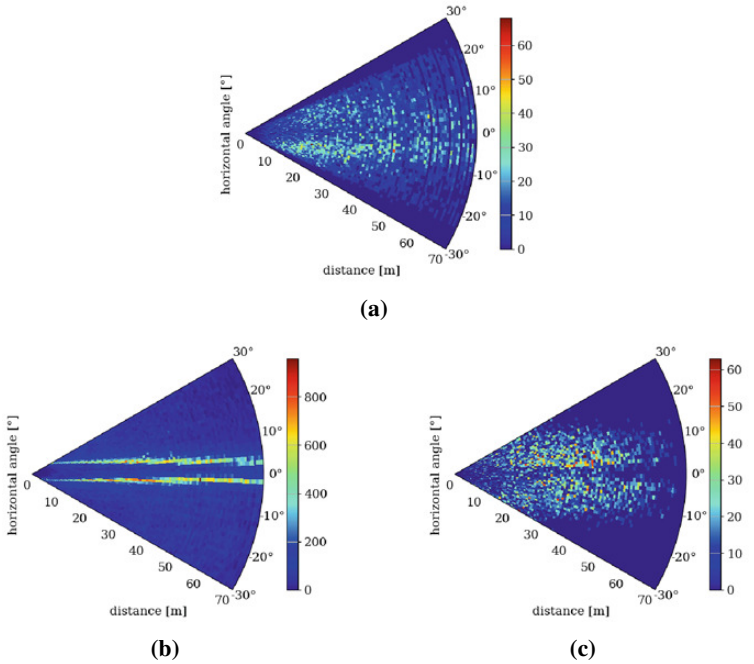


Fig. 5 Pedestrian distribution over horizontal angle and distance. **a**: Cityscapes. **b**: KI-A tranche 3. **c**: Our synthetic data

Influence of different occluding objects on detection performance: A number of object and attribute variations are depicted in Fig. 6. On the left side, the SSD bounding box detector [LAE+16] is applied to the three images with different occluding objects in front of a pedestrian. In all three images, two bounding boxes are predicted for the same pedestrian. While one bounding box includes the whole body, the second bounding box only covers the non-occluded upper part of the pedestrian. On the right side, the `DeeplabV3+` model trained on the KI-A tranche 3 is used to create a semantic map of the same three images. Besides the arms, the pedestrian is detected, even partially through the occluding fence. However, another interesting observation can be made: The ground the pedestrian stands on is always labeled as sidewalk. We interpret this as an indication to a bias in the training data, as the training data does not include enough images of pedestrians on the road, just on the sidewalk. This hypothesis can be further strengthened when we inspect the pedestrian distributions in Fig. 5b, where the pedestrians are distributed narrowly left and right off the street in the middle. Additionally, both bounding box prediction and the

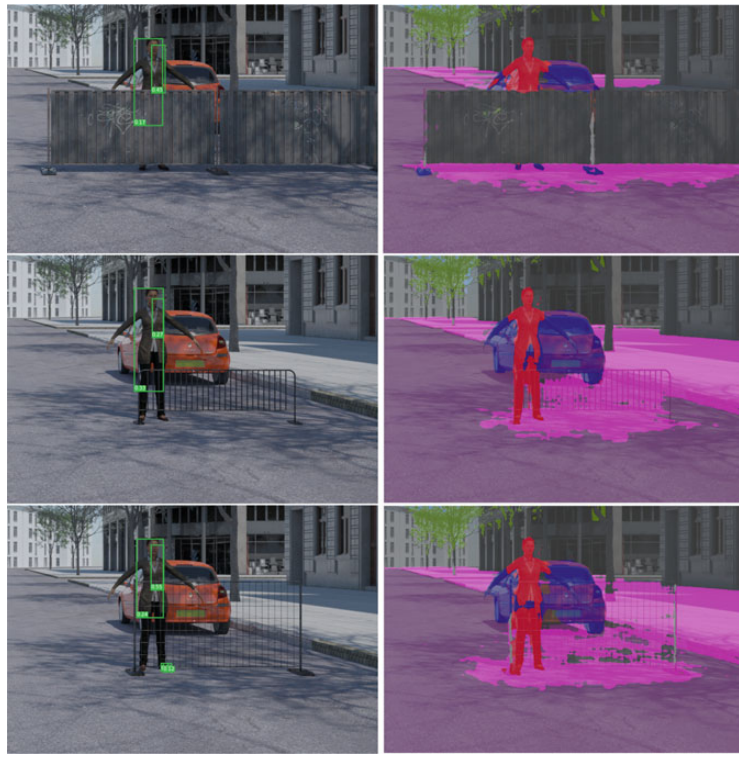


Fig. 6 Scene with variation of occluding objects. Left: 2D bounding box detection. Right: semantic segmentation

semantic segmentation do not include the pedestrian's arms in their predictions. This can also be attributed to a bias in the training data.

Influence of noise on detection performance: An experiment demonstrating our sensor simulation determines the influence of sensor noise on the predictive performance. In Fig. 7, Gaussian noise with increasing variance is applied to an image, and three `DeeplabV3+` models trained on A2D2, Cityscapes, and a synthetic dataset, respectively, are used to predict on the data. While image color pixels are represented in the range $x_i \in [0, 255]$, the noise variance is in the range of $\sigma^2 \in [0, 20]$ with a step size of 1. For each noise variance step, the mIoU performance metric on the image prediction per model is calculated. While initially the models trained on Cityscapes and the synthetic dataset increase in performance, all models' predictive

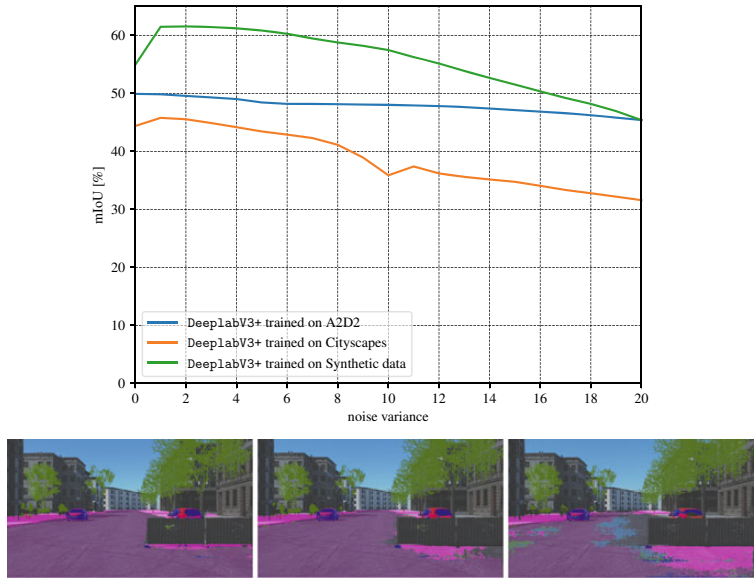


Fig. 7 Top: mIoU performance decreases with increasing noise variance. Bottom (left to right): segmentation maps with increasing noise variance $\sigma^2 \in \{0, 10, 20\}$, image pixels $x_i \in [0, 255]$

performance ultimately decreases with an increasing level of sensor noise. The initial increase can be explained to stem from the domain shift of training to validation data, where in the training data a small noise variance can be observed.

Analysis of performance-limiting factors: Some scene parameters have a major influence on the perception performance. This includes the occlusion rate of objects, with totally occluded objects that are obviously not detectable or the object size (in pixels) in the images, also with a natural boundary where detection breaks up if the object size is too small. Other performance-limiting factors include contrast and other physically observable parameters.

To measure the influence or sensitivity of perception functions against performance-limiting factors we designed an experiment using about 30,000 frames containing one person each. The person is moved and rotated on the sidewalk and on the street. The occlusions are determined by rendering a mask of the person and comparison with the actual instance mask considering occluding objects. A degree of 100% represents a fully occluded object. Figure 8 shows results of this experiment, each gray dot representing one frame and the colored curves showing regression plots with differently clothed persons.

The figure shows a $pAcc$ downwards trend with increasing occlusion rates. The `Detectron2` model (trained on Cityscapes) is comparatively more robust than

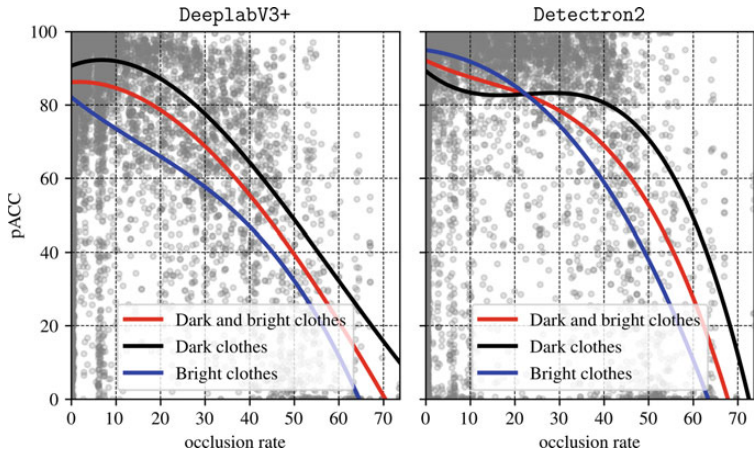


Fig. 8 Polynomial regression curves (of order 3) on pedestrian detection rate $pAcc$ of `DeeplabV3+` (left) and `Detectron2` (right) for various occlusion rates of a pedestrian wearing dark or bright clothes

`DeeplabV3`. The plot shows that `Detectron2` offers stable detection with occlusion rates $<35\%$ and then the performance drops. `DeeplabV3's` (also trained on Cityscapes) performance drops after 15% occlusion rate. The curves are not linearly following a trend due to the fact that there are other scene parameters (sunlight, shadow, direction of pedestrian) which are not constant across the rendered images.

What can also be seen in the figures is that, despite the trend of the regression curves, there is a great variation in the data—visible by the widely scattered grey points. That means that the performance depends also on other factors besides the occlusion rate. Figure 8 is showing one example of analysis possible with the metadata provided by our framework. More parameters are considered in our previous work [SGH20].

Data bias analysis: Another experiment we conducted considers failure cases, i.e., false negatives (FN) of the SSD 2D-bounding box detector regarding pedestrian detection. To accomplish this, we rendered 2640 images with our variational data synthesis engine. These images are then inferred by the SSD model and evaluated. Only pedestrians with a bounding box width greater than $0.1 \times$ image width and a height of $0.1 \times$ image height are considered valid for evaluation. Additionally, only objects with an occlusion rate below 25% are considered valid. These restrictions guarantee that pedestrians in the validation are of sufficient size, i.e., close to the camera, and clearly visible due to little occlusion and would therefore be easy to detect.

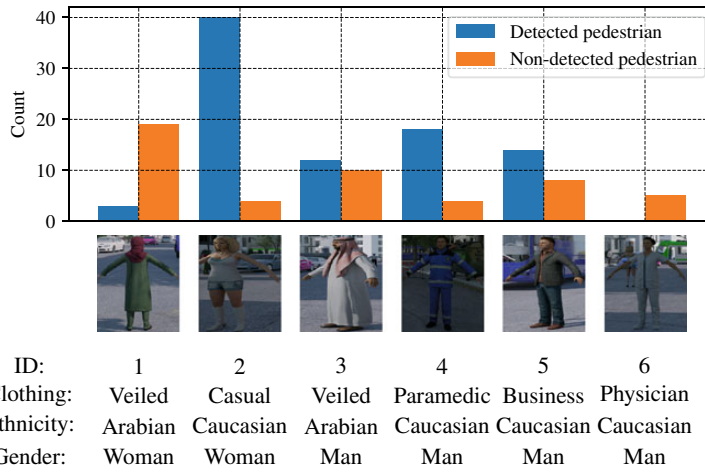


Fig. 9 Count of detected and non-detected pedestrians for different pedestrian assets, i.e., different clothing, ethnicity, and gender

With these restrictions in place we found that from all the pedestrian assets the synthesis engine placed in the scene, there were six assets that were omitted by the SSD model as can be seen in Fig. 9.

The asset ID 1 is an Arabian woman wearing traditional clothes effectively veiling the person. Asset ID 2 is a Caucasian woman clothed in summer casual, i.e., short pants and short sleeves, revealing parts of her skin. The second Arabian ethnicity asset with the ID 3 is similar to asset 1 clothed in traditional veiling clothes but of male gender. The remaining assets 4, 5, and 6 are of male gender and Caucasian ethnicity wearing different work clothes, i.e., a blue paramedical outfit for ID 4, business casual jeans and jacket for ID 5, and white physician clothes for ID 6.

The asset ID 2 with the summer casual clothed woman is only miss-detected a few times, in most cases the detection worked well, indicating no data bias for this asset. In contrast, the pedestrian asset ID 6 of a physician dressed in white hospital clothing has not been detected at all. Additionally, two of the assets that were relatively most often overlooked by the network are the Arabian clothed woman with asset ID 1, as well as an Arabian clothed man with the ID 3. This result would suggest that these kind of pedestrian assets, i.e., IDs 1, 3, and 6, were not present in the data for training the model and adding them to it will lead to a mitigation of this exact failure case.

6 Outlook and Conclusions

This chapter has introduced a new generative data synthesis framework for the validation of machine learning-based perception functions. The approach allows a very flexible description of scenes and parameters to be varied and systematical tests of parameter variations in our unified validation parameter space.

The conducted experiments demonstrate the benefits of splitting the validation process into scene variation that looks into randomized placement of objects and a variation of scene parameters and sensor simulation. Our simple probabilistic scene generator is scalable and able to produce scenes with a high number of different objects—as provided by an asset database. The spatial distribution of the positioned objects, as demonstrated for persons in Fig. 5, is more realistic compared to manually crafted 3D scenes. Along with our sensor simulation (results discussed in the chapter 'Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation' [HG22]), we present a step to close the domain-gap between synthetic and real data. Future work will continue to analyze the influence of other factors, such as rendering fidelity, scene complexity, and composition, to further improve the capabilities of the framework and make it even more applicable for the validation of real-world AI functions.

Our experiments with performance-limiting factors, as shown for occlusion rates and object size (as a function of distance to the camera) in the previous section gives clear evidence that the performance of perception functions cannot be characterized by only a few factors. It is, however, a complex function of many parameters and aspects, including scene complexity, scene lighting and weather conditions, and the sensor characteristics. The deep validation approach described in this chapter is addressing this multi-dimensional complexity problem and we designed a system and methodology for flexible validation strategies to span all these parameters at once.

Our validation parameterization, as demonstrated in the results section, is an effective way to detect performance problems in perception functions. Moreover, it allows in its flexible design the sampling and a practical computation at scale allowing for deep exploration of the multi-variate validation parameter space. Therefore, we see our system as a valuable tool for the validation of perception functions.

Moving forward we are looking into using the deep synthesis approach to implement sophisticated algorithms to support more complex validation strategies. As another key direction we target improvements in the computational efficiency of our validation approach, allowing coverage of more complexity and parameter dimensions.

Acknowledgements 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

[BB95] W. Burger, M.J. Barth, Virtual reality for enhanced computer vision, in *Virtual Prototyping: Virtual Environments and the Product Design Process*, ed. by J. Rix, S. Haas, J. Teixeira (Springer, 1995), pp. 247–257

[COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, 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, 2016), pp. 3213–3223

[CPK+18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **40**(4), 834–848 (2018)

[CSVJR18] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, Modeling camera effects to improve visual learning from synthetic data, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (Munich, Germany, 2018), pp. 505–520

[CSVJR19] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, Sensor transfer: learning optimal sensor effect image augmentation for sim-to-real domain adaptation, pp. 1–8 (2019). arXiv:1809.06256

[DG18] W. Damm, R. Galbas, Exploiting learning and scenario-based specification languages for the verification and validation of highly automated driving, in *Proceedings of the IEEE/ACM International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)* (Gothenburg, Sweden, 2018), pp. 39–46

[DKF20] J. Devaranjan, A. Kar, S. Fidler, Meta-Sim2: learning to generate synthetic datasets, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Virtual conference, 2020), pp. 715–733

[DRC+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in *Proceedings of the Conference on Robot Learning CORL* (Mountain View, CA, USA, 2017), pp. 1–16

[Epi04] Epic Games, Inc. Unreal Engine Homepage (2004). [Online; accessed 2021-11-18]

[EVGW+15] M. Everingham, L.V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. (IJCV) **111**(1), 98–136 (2015)

[GKM+20] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V.H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, P. Schuberth, A2D2: Audi autonomous driving dataset (2020), (pp. 1–10). arXiv:2004.06320

[HG21] K. Hagn, O. Grau, Improved sensor model for realistic synthetic data generation, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)* (Virtual Conference, 2021), pp. 1–9

[HG22] K. Hagn, O. Grau, Optimized data synthesis for DNN training and validation by sensor artifact simulation, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety* ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, 2022), pp. 149–170

[HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, 2016), pp. 770–778

[JWKW18] P. Junietz, W. Wachenfeld, K. Klonecki, H. Winner, Evaluation of different approaches to address safety validation of automated driving, in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)* (Maui, HI, USA, 2018), pp. 491–496

[KP16] Nidhi Kalra, Susan M. Paddock, Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? Transp. Res. Part A: Policy Pract. **94**, 182–193 (2016)

[LAE+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Amsterdam, The Netherlands, 2016), pp. 21–37

[LLFW20] Zhenyi Liu, Trisha Lian, Joyce Farrell, Brian Wandell, Neural network generalization: the impact of camera parameters. IEEE Access **8**, 10443–10454 (2020)

[LSD15] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA, USA, 2015), pp. 3431–3440

[MBM18] T. Menzel, G. Bagschik, M. Maurer, Scenarios for development, test and validation of automated vehicles, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)* (Changshu, China, 2018), pp. 1821–1827

[Nik19] S.I. Nikolenko, *Synthetic Data for Deep Learning* (2019), pp. 1–156. arXiv:1909.11512

[RHK17] S.R. Richter, Z. Hayder, V. Koltun, Playing for benchmarks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, 2017), pp. 2232–2241

[RVRK16] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: ground truth from computer games, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Amsterdam, The Netherlands, 2016), pp. 102–118

[SAE18] SAE International. SAE J3016: surface vehicle recommended practice—taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE Int. (2018)

[SGH20] Q.S. Sha, O. Grau, K. Hagn, DNN analysis through synthetic data variation, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)* (Virtual Conference, 2020), pp. 1–10

[WEG+00] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, A. Sumner, et al., TORCS—The Open Racing Car Simulator (2000). [Online; accessed 2021-11-18]

[Wer94] J. Wernecke, *The Inventor Mentor: Programming Object-Oriented 3D Graphics With Open Inventor* (Addison-Wesley, 1994)

[WKM+19] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2 (2019). [Online; assessed 2021-11-18]

[WPC20] Mingyun Wen, Jisun Park, Kyungeun Cho, A scenario generation pipeline for autonomous vehicle simulators. HCIS **10**, 1–15 (2020)

[WU18] M. Wrenninge, J. Unger, Synscapes: a photorealistic synthetic dataset for street scene parsing (2018), pp. 1–13. arXiv:1810.08705

[YLCT20] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, Kazuya Takeda, A survey of autonomous driving: common practices and emerging technologies. IEEE Access **8**, 58443–58469 (2020)

7.5 Publication 5

Validation of pedestrian detectors by classification of visual detection impairing factors

Korbinian Hagn and Oliver Grau

Published in

Proceedings of the 17th European Conference on Computer Vision Workshops (ECCVW 2022), 2022 [HG23]

DOI: 10.1007/978-3-031-25072-9_33

Validation of Pedestrian Detectors by Classification of Visual Detection Impairing Factors

Korbinian Hagn( ) and Oliver Grau

Intel Deutschland GmbH, Lilienthalstraße 15, 85579 Neubiberg, Bayern, Germany
{korbinian.hagn,oliver.grau}@intel.com

Abstract. Validation of AI based perception functions is a key cornerstone of safe automated driving. Building on the use of richly annotated synthetic data, a novel pedestrian detector validation approach is presented, enabling the detection of training data biases, like missing poses, ethnicities, geolocations, gender or age. We define a range of visual impairment factors, e.g. occlusion or contrast, which are deemed to be influential on the detection of a pedestrian object. A classifier is trained to distinguish a pedestrian object only by these visual detection impairment factors which enables to find pedestrians that should be detectable but are missed by the detector under test due to underlying training data biases. Experiments demonstrate that our method detects pose, ethnicity and geolocation data biases on the CityPersons and the EuroCity Persons datasets. Further, we evaluate the overall influence of these impairment factors on the detection performance of a pedestrian detector.

1 Introduction

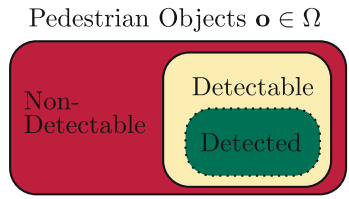


Fig. 1. A classifier can be trained to distinguish detectable from non-detectable pedestrians according to a set of visual impairment factors. Objects in the detectable set which have not been detected indicate a data bias of the pedestrian detector.

Modern deep learning-based object detectors are capable of detecting objects with unprecedented accuracy. The detection boundary of a vision task is often

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-25072-9_33.

L. Karlinsky et al. (Eds.): ECCV 2022 Workshops, LNCS 13805, pp. 476–491, 2023.
https://doi.org/10.1007/978-3-031-25072-9_33

compared to human detection capabilities. While closing in on reaching this detection limit it is necessary to understand what properties of an object in an image does make it difficult for a human to detect this very object and how we are able to exploit this detection boundary for validation. In Fig. 1 we argue that for a set of pedestrian objects in a dataset, there is (i) a non-detectable subset, defined as persons that are not detectable due to their visual impairment factors that we will introduce in the course of this paper, (ii) the detectable subset, which is the set of pedestrian objects that are detectable by a human observer or state-of-the-art object detector trained on a global super set without any data biases, and (iii) a subset consisting of objects that are detected by a state-of-the-art detector trained on a Geo specific domain with likely data biases.

The difference between detectable and detected subsets is then mainly defined due to differences in the training data of the detector. The boundary of the detectable subset itself is defined by visual detection impairment factors, like occlusion and contrast. If for example the contrast is close to 0 or the occlusion is close to 100% then the object, while still being present in the image, simply cannot be detected even by a human observer.

Our validation approach makes use of all these three subsets by identifying pedestrians that should be detectable according to their visual impairment factors, i.e., classifying if a pedestrian object is in the non-detectable or the detectable set. We then evaluate the difference between the detectable subset and the actually detected subset which we receive from predictions of a pedestrian detector. We demonstrate by investigation of this difference that we can reveal data biases in the pedestrian detector under test, like missing poses, ethnicity or geolocations originating from the used training dataset.

This validation strategy is majorly enabled by the recent progress in synthetic data generation giving new possibilities to produce high-realism data for computer vision tasks. Especially for understanding the prediction decision-making of a detector, the pixel-accurate ground-truth and the rich meta-data that can be extracted from the data generation process are highly beneficial to find relations of a visual factor, e.g., contrast, of an object and its detection.

The contributions of this paper are as follows:

- We identify factors that are impairing the visual detection capability of a pedestrian object and investigate each factor's importance to the detectability.
- We demonstrate the usage of these factors by training a classifier to distinguish detectable from non-detectable pedestrian objects and utilizing this classifier to validate a pedestrian detector for biases of its underlying training dataset.

2 Related Works

Validation Strategies. A validation strategy of a perception function which is part of the automotive driving stack is of high importance to guarantee a safe driving function. Our method hereby detects pedestrian detection faults, i.e. a

functional insufficiency as defined by [12], addressing the lack of generalization as defined by [22]. To validate for a functional insufficiency, the detection, creation and testing with corner cases has been addressed by several previous works [1,2]. Bolte et al. [3] define a corner case as a *"non-predictable relevant object in relevant location"*. By this definition, we propose a new method validating a pedestrian detector through the detection of corner cases by learning to classify the *"non-predictable"* property from visual detection impairment factors.

Validation with Synthetic Data. Several synthetic datasets for automotive perception functions have been proposed [19,21,25]. While these datasets are very valuable for benchmarking, methods for validation should allow to directly control the generation of synthetic data, especially for automotive tasks like pedestrian detection, and here the most notably to mention are Carla [10] and the LGSVL Simulator [20]. But, instead of a whole simulation of an automotive driving scenario also variational automotive scene creation approaches have been introduced for detector evaluation [11,18,23]. These works could already show validation results with detecting training and validation dataset domain shifts. Our method not only detects data biases with semantic relevance, i.e. geolocation, ethnicity, etc., but also missing pose information in the training data by utilizing a classifier to distinguish detectable from non-detectable pedestrians through analyzing the proposed visual detection impairment factors.

Visual Detection Impairing Factors. Besides label noise and sensor noise, detection impairing factors such as occlusion rate and contrast were analyzed for their effect on object detection performance. While there are factors that are believed to have no influence on the detection, for example contrast [26], we found evidence that contrast is still relevant to the object detection performance. Distance and occlusion on the other side is acknowledged to be one of the most influential factors and has already been addressed by several works [7,8] and incorporated into a safety relevance distance metric [17]. We not only address these factors and their influence on the detection performance, but add several new factors relevant to the detection performance and investigate their influence.

Pedestrian Detection Models. While there is a recent popularity increase of models based on the Transformer architecture [9], most automotive detection benchmarks[1,2] are still dominated by convolutional neural network (CNN) based methods. In this work we are using the Cascade R-CNN [5] pedestrian detector, a development from the R-CNN [14] and Faster R-CNN [13] model. The HRNet [24] feature extraction backbone is pre-trained on ImageNet [6].

[1] CityPersons: https://github.com/cvgroup-njust/CityPersons.
[2] EuroCity Persons: https://eurocity-dataset.tudelft.nl/eval/benchmarks/detection.

3 Methodology

Our method is described in several sub steps: First, the generation of synthetic training and validation data. Next, the definition of the visual detection impairment factors and their extraction from synthetic data. Last, the definition and training of a detectability classifier and validation of a pedestrian detector for data biases.

3.1 Synthetic Data Generation

The synthetic data used in this contribution is generated by a data synthesis pipeline and includes special modules to compute meta- or ground-truth data which is hard or impossible to observe and measure in real data. One example is the pixel-accurate occlusion rate of an object, by differencing the mask of the unoccluded object, computed in a separate rendering pass from the occluded object in the complete scene. To achieve a representative calibration of the detector, the synthetic data should have similar characteristics than real scenes. This is also described as domain gap between the real and synthetic data. We achieve highly realistic synthetic data by three levels: i) An automated scene generator produces scenes with similar complexity as those in real data, ii) we use similar 3D objects from an asset database and iii) a realistic sensor simulation building on the work described in [15]. The rendering process delivers realistic scene illumination in linear color space with floating accuracy based on the Blender Cycles path-tracing rendering engine[3], followed by a sensor simulation that includes simulation of effects like sensor noise, lens distortions, and chromatic aberrations and a tone mapping to integer sRGB color space. The parameters of the sensor simulation are tuned to match the characteristics of the Cityscapes data similar to [16].

For the purpose of this paper, we synthesize a dataset that contains complex urban street scenes with a variation of objects (about 300), such as different houses, vehicles, street elements, and about 150 different human characters automatically placed from an asset database. Figure 2 depicts some example frames from that data set. The synthetic data generation pipeline also computes various metadata and ground-truth, including semantic and instance segmentation, the distance of objects to the camera, occlusion rates, 2D + 3D bounding boxes, radiometric object features including contrast measures as introduced above. The dataset is split into a training set to calibrate the detectability classifier and a validation set to detect data biases of the pedestrian detector under test.

One of the advantages of synthetic data in our method is the precision and deterministic nature of the label and bounding box meta-data, which is free from noise, as all the generated labeling data is pixel accurate.

For evaluations with this dataset, the pedestrians in the images are prefiltered to guarantee that only the considered impairment factors are influential on a pedestrian detection. This means that pedestrians that are too close to

[3] blender.org.

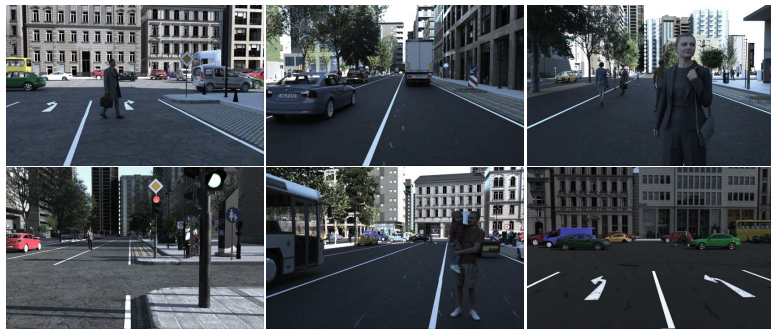


Fig. 2. Our fully parameterizable generation pipeline allows rendering pedestrians at any size, occlusion, time of day, and distance to the camera.

one another cannot be detected due to non-maximum suppression (NMS) and are therefore ignored. The resulting synthetic dataset $\mathcal{D}_{synth}$ consists of 17012 images of pedestrians for training and 26745 images of pedestrians for evaluation.

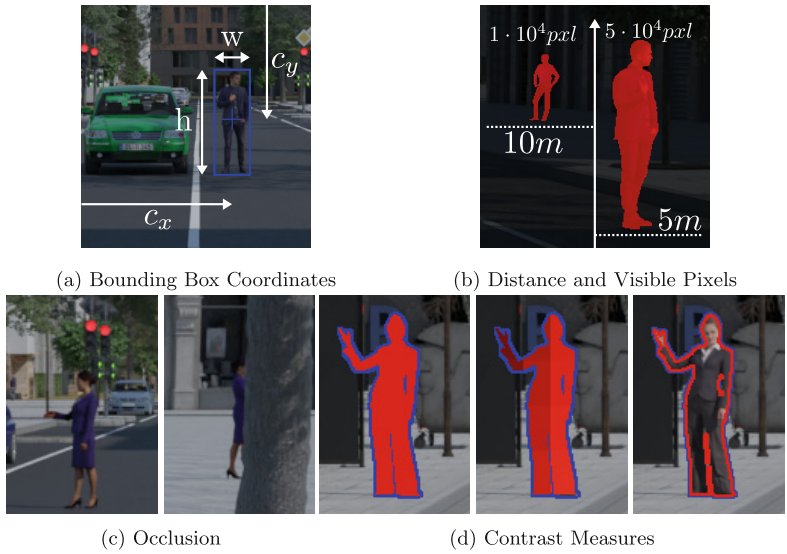
3.2 Visual Detection Impairing Factors

The major factors of a pedestrian object we consider to be influential of the detection capability of a detector are visualized in Fig. 3. Beginning with the placement of a pedestrian in an image. This information is extracted from the bounding box coordinates of the ground-truth. The coordinates are defined by the center o_{cx}, o_{cy} coordinate and the width o_w and height o_h of the box in pixels.

Next, distance to the observer o_d, i.e., camera, in $[m]$ and the number of visible pixels o_{vp} of the object. The distance to the observer is extracted from the 3D placement in the rendered scene. The information about the number of visible pixels is extracted from the instance segmentation label, by counting the pixels that belong to the person.

Determining the occlusion rate o_{ocl}, i.e., the ratio of visible pixels to the whole pixels of an object, is done by extracting the number of occluded and non-occluded pixels from the instance segmentation ground-truth pixels with, and without occluding objects.

Last, we define different contrast measures as visual impairment factors. The first contrast measure o_{cfull} is defined as the euclidean distance of the mean object RGB color to its surrounding background. This is done by dilating the instance segmentation mask of the person and subtract the undiluted segmentation mask so that we get the surrounding 5 pixel border of the object. The contrast is now calculated by the euclidean distance of the mean object color to the mean surrounding background color. Another contrast measure o_{cmean} is defined by segmenting the object into 12 smaller segments and calculate the



(a) Bounding Box Coordinates (b) Distance and Visible Pixels

(c) Occlusion (d) Contrast Measures

Fig. 3. The potential detection performance impairing factors we consider in this work: (a) bounding box coordinates (o_{cx}, o_{cy}, o_h, o_w), (b) distance and number of visible pixels of a pedestrian (o_d, o_{vp}), (c) rate of occlusion (o_{ocl}), (d) contrast of a pedestrian (red) to its background (blue) calculated by the full pedestrian silhouette (o_{cfull}), segment wise (o_{cmean}) and edge wise (o_{cedge}).

mean RGB color of a segment and the euclidean distance to its neighboring mean background color. The resulting contrast measure is then derived by averaging over all segments. The last considered contrast measure o_{cedge} is calculated with only the 5 pixels on the edge of the instance label, then calculating the mean color and the euclidean distance to the mean of the surrounding background.

3.3 Classification of Detectable Pedestrians

We define the classification loss to train a classifier to distinguish persons being detectable or not. This classifier is then used to analyze a pedestrian detector for data biases due to missing data in the training data.

Classification Loss. We begin with a synthetic image dataset of pedestrian objects $\mathbf{o} \in \Omega = \{\mathbf{o}_1, \ldots, \mathbf{o}_O\}$ where O is the number of pedestrians in a set of all pedestrian objects defined as Ω. Each pedestrian was either detected (1) or missed (0) by the pedestrian detector under test. In the accumulation step each pedestrian object is enriched with the previously defined detection impairment

factors, and thus we obtain the sample vector of the objects metadata $\mathbf{o}$ defined as follows:
$\mathbf{o} = (o_{cx}, o_{cy}, o_h, o_w, o_d, o_{vp}, o_{ocl}, o_{cfull}, o_{cmean}, o_{cedge})$, where every entry in this vector is normalized to $\mu = 0$ and $\sigma^2 = 1$ and equals to the detection impairment factors previously described.

With the enriched pedestrian objects and their corresponding target class (detected, missed) we then train a supervised deep neural network as classifier. The classification loss is the default cross entropy loss defined as,

$$J^{cls}(p, u) = -\sum_{i \in \mathcal{S}} u_i log(p_i). \tag{1}$$

The classification output is a discrete probability distribution $p = (p_0, \ldots, p_s)$, i.e. confidence, computed by a softmax with s being the predicted class $s \in \mathcal{S} = \{0, \ldots, S-1\}$, Here, $S = 2$, i.e., the pedestrian is detectable ($s = 1$) or the pedestrian is non-detectable ($s = 0$). The respective target class is defined as $u = (u_0, \ldots, u_s)$, with the detectable class defined as 1 and the non-detectable target class as 0.

The overall function g of our classifier is then described as follows,

$$g : \Omega \rightarrow \mathcal{S}. \tag{2}$$

The classifier learns a mapping of pedestrian objects Ω to a corresponding detectability class $\mathcal{S}$.

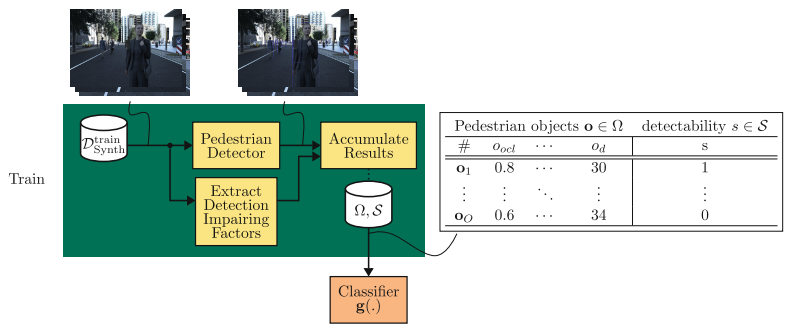


Fig. 4. Training a classifier to distinguish between detectable and non-detectable pedestrian objects.

Training the Classifier. Our approach to train a classifier to distinguish between detectable and non-detectable pedestrian objects is sketched in Fig. 4. We use the synthesized training dataset $\mathcal{D}^{train}_{Synth}$ as described in Sect. 3.2. These

images are inferred by a state-of-the-art Cascade R-CNN [5] pedestrian detector with a HRNet [24] backbone pretrained on ImageNet [6]. This detector was further trained on a sub-set of our synthetic dataset. In parallel to the inference process, we extract our defined visual detection impairment factors on a per pedestrian object basis from each synthetic data frame. Next, the inference results of the detector and the extracted impairment factors are accumulated to the pedestrian object set Ω and target class set $\mathcal{S}$. Ω is the classification input with each row being a pedestrian object with its visual impairment factors $\mathbf{o}$ vectorized and a corresponding detectability class s in $\mathcal{S}$ as target class. These inputs are then used to train a fully connected deep neural network with 5 hidden layers to learn the mapping $\mathbf{g}(.)$, i.e. the detectability classification. Training this classifier results in a high classification F1 score of 0.93. The F1 score is the harmonic mean of precision and recall and regularly used to evaluate binary classification tasks.

Detection of Validation Samples. The trained detectability classifier is used to validate another pedestrian detector as described in Fig. 5. The detectors under test in our validation experiments are Cascade R-CNN detectors trained on the CityPersons [27] (CP) dataset and another trained on the EuroCity Persons [4] (ECP) dataset. For validation, we use the previously rendered synthetic validation dataset $\mathcal{D}^{val}_{Synth}$. In this dataset we integrated our validation samples, i.e., if we want to validate a pedestrian detector for biases due to geolocation of the training data we have to add different pedestrian assets from various other geolocations. Similar we can add different posed pedestrians, gender, age-group or ethnicity assets to the validation images. The Pedestrian detector under test generates inference results on the validation set images. In parallel the visual impairment factors per pedestrian object are extracted from each image. The detections and the impairment factors are accumulated into the sets Ω and $\mathcal{S}$. The previously trained classifier $\mathbf{g}(.)$ classifies the input objects into detectable or non-detectable. The results, i.e. confidence score, per object prediction and target detectability class, are stored in the validation result dataset $\mathcal{D}^{val\_cl}_{Synth}$. The validation results are then enhanced by the per pedestrian meta information from $\mathcal{D}^{val}_{Meta}$ which we obtain from the image synthesis process. This meta information consists of the asset ID, the geolocation, pose, gender, age-group and ethnicity and is useful to draw conclusions of underlying data biases in the training data.

The validation result $\mathcal{D}^{val\_cl}_{Synth}$ is now used to find detection biases of the pedestrian detector under test. Therefore, we simply filter the dataset for pedestrian objects that are classified as detectable (prediction=1) but were missed by the pedestrian detector under test (detectability=0). From this filtered validation result the data biases are extracted by statistical analysis of occurrences as described in section Results & Discussion.

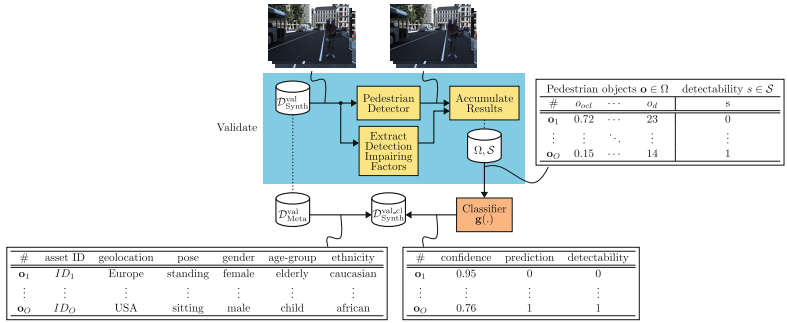


Fig. 5. Generation of validation data to detect data biases in the pedestrian detector.

4 Results and Discussion

We present two different results: First, we analyze the validation results from pedestrian detectors trained on the CityPersons (CP) and on the EuroCity Persons (ECP) dataset. This step is separated into the detection of geolocation, gender, age-group and ethnicity data biases and into the detection of data biases due to missing poses.

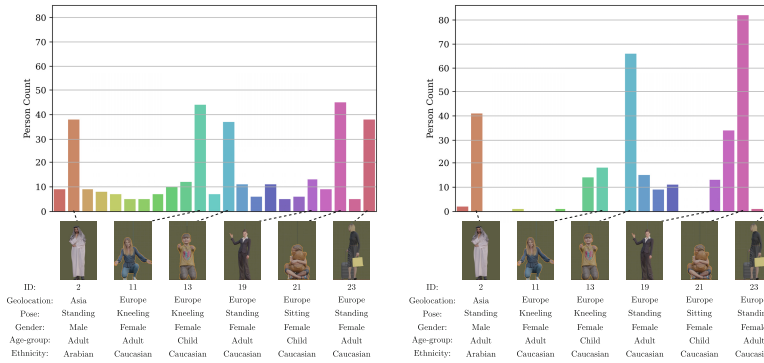
Last, we analyze the influence of each individual visual detection impairment factor on the detection performance, i.e. miss rate.

4.1 Evaluation of Pedestrian Detection Data Biases

Found data biases in our approach can mainly be categorized in two categories. The first category are data biases attributed to missing semantic characteristics in the training dataset and therefore resulting data bias. These errors can occur due to differences in pedestrians from different geolocations, missing gender, ethnicity or age-group in train and validation set. The second category of data biases are missing poses, which may not only originate from missing training data, but also due to incapability of the detector, i.e. missing aspect ratios of anchor boxes.

Validation of Data Bias Characteristics. To generate meaningful results from the validation result set $\mathcal{D}_{Synth}^{val\_cl}$ we filter all pedestrians according to their frequency. We deem a pedestrian to be meaningful if the ratio of frequency in the result set to overall frequency in the synthetic validation data is above 10%. Following the pre-filtering of the validation results we can plot the frequency of each pedestrian for the (a) CP and the (b) ECP dataset, as depicted in Fig. 6.

For the CP dataset 23 persons cause a data bias error comparing to only 15 persons for the ECP dataset. In the CP dataset we found that pedestrians from non-European geolocation and with a different ethnicity than Caucasian



(a) CityPersons (b) EuroCity Persons

Fig. 6. Distribution of found data biases.

will have a significant influence to miss-detections due to the data bias in the training data. This bias is very prominent if the person wears clothing typical for its geolocation but uncommon in Europe as can be seen with ID 2, a male veiled in traditional Arabian clothing. A similar observation can also be made with the ECP dataset. While many occurrences of persons with female gender can be observed, the originating cause is often in combination with an uncommon pose as in ID 19 with the waving hand or in ID 23 carrying an additional trolley. Sitting or kneeling poses, especially in combination with the child age-group, are the most common sources of data bias found in both datasets as can bee seen with the IDs 11, 13 and 21. For the child age-group as well as for the kneeling and sitting poses the overall height of the person is smaller than usual. This strongly suggests there are too few persons in sitting or kneeling pose included in both datasets. Comparing the observations from the CP and the ECP datasets we see the CP dataset to have a marginally smaller data bias on sitting and kneeling children but a higher data bias on uncommon poses of standing persons.

Validation of Pose Data Bias. To further filter the observations of found data biases we can evaluate the persons in the previous result set if they have at least a bounding box prediction with a ground-truth intersection over union (IoU) of > 0.25. Because a detection is only counted as a valid detection with an IoU threshold of > 0.5 we can lower this threshold in the validation to inspect if there was at leas at partial detection of a person. Exemplary predictions where the person was not counted as detected, but a partial detection was predicted nonetheless can be seen in Fig. 7.

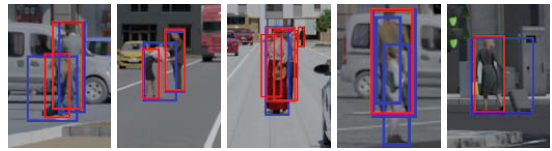
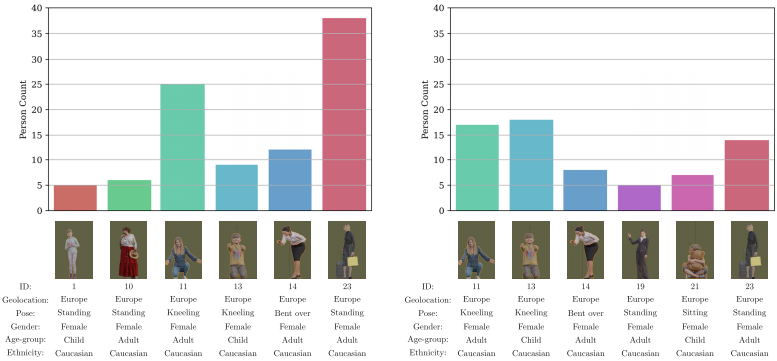


Fig. 7. Non-detected persons (blue) with partial predictions (red), i.e. 50% > IoU > 25%. (Color figure online)

If we now further filter the previous validation result sets to only include persons with a partial detection of IoU > 0.25 we get the frequency distributions as depicted in Fig. 8.



(a) CityPersons (b) EuroCity Persons

Fig. 8. Distribution of found data biases due to missing pose data.

Again, the IDs 11, 13, 14 and 23 can be found in both datasets, suggesting that the characteristics, i.e. geolocation, age-group, gender and ethnicity, are known by the detector but the pose information is missing. The ID 19 and 21 were partially detected with the ECP dataset, but they are missing entirely now in the CP dataset, i.e. while the ECP dataset suffers from missing pose information on these persons, the CP dataset suffers from a data bias because not even a partial prediction was made. Similarly, the ID 1 and 10 are partially detected with the CP dataset but not with the ECP dataset.

Additionally, with the ID 23 person we found that the CP dataset did miss the detections because the luggage trolley is defined to be part of the bounding box of this person. While the ECP prediction would, in most cases, predict a loose bounding box for the person that is sufficient to count as detected, the

CP bounding box prediction was tighter on the person. Careful evaluation and definition of the ground-truth is therefore a key to produce meaningful validation results without any ambiguity.

Summarizing, the CP dataset is missing more pedestrian characteristics than the ECP dataset. Both datasets have a strong bias to European persons and both datasets are missing pose information on kneeling, sitting or bent over persons especially if they are children.

4.2 Detection Impact of Visual Impairment Factors

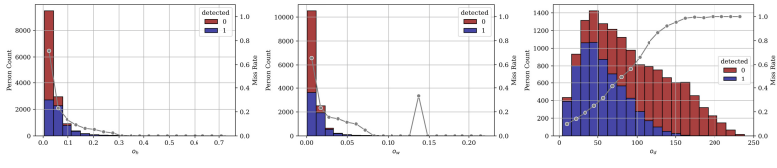
To understand if the visual impairment factors are meaningful for a detectable or non-detectable classification we have to investigate the relation of detection performance and each individual factor. Therefore, the predictions on the synthetic training set $\mathcal{D}_{Synth}^{train}$ are evaluated along with each individual impairment factor visualized as a histogram. Figure 9 shows the count of detected (blue) and non-detected (red) persons per bin. Additionally, the miss rate of each factor is calculated per bin and plotted into each histogram. The miss rate is defined as the number of missed detections (detected = 0) over the number of all persons, i.e., detected and non-detected.

A major influence factor is obviously the size of an object and therefore the number of visible pixels of a pedestrian in the image which is evident in several interrelated impairment factors. Small height, small width, high distance and a low number of visible pixels lead to a high miss rate. Increasing the occlusion rate leads to a reduced number of visible pixels and increased miss rate as well. The contrast measures show a decreasing tendency for increased values of contrast but not as pronounced as the other size related factors and with occasional outliers at higher values. Calculating the Spearman correlation of miss rate and the individual impairment factors emphasizes these observations as shown in Table 1. The size related factors (o_h, o_w, o_d, o_{vp}, o_{ocl}) show high positive or negative correlations, where the width o_w has the lowest absolute correlation to the miss rate. For the contrast measures we see a high negative correlation on the o_{cfull} contrast measure and lower negative correlations for the other two contrast measures.

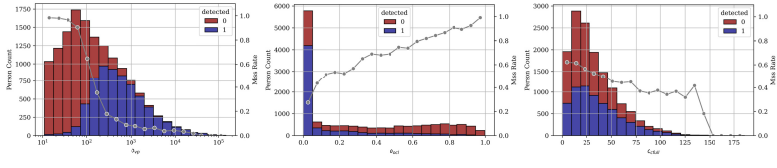
Table 1. Spearman correlation of the visual impairment factors and miss rate.

	o_h	o_w	o_d	o_{vp}	o_{ocl}	o_{cfull}	o_{cmean}	o_{cedge}	o_{cy}	o_{cx}
ρ_s	−0.886	−0.744	0.995	−0.995	0.993	−0.953	−0.451	−0.343	0.093	0.017

Last, we visualize the remaining factors o_{cy} and o_{cx}, i.e. the horizontal and vertical placement of a person in the image, on a heatmap with their individual marginal distributions and the corresponding miss rate plots in Fig. 10.



(a) Height (o_h) (b) Width (o_w) (c) Distance (o_d)



(d) Visible Pixels (o_{vp}) (e) Occlusion (o_{ocl}) (f) Full contrast (o_{cfull})

(g) Mean contrast (o_{cmean}) (h) Edge contrast (o_{cedge})

Fig. 9. Influence of visual impairing factors of a pedestrian on the detection performance, i.e. miss rate (gray).

The horizontal placement has no influence on the miss rate, which is confirmed by the Spearman correlation of 0.017. The vertical placement with a similar low correlation however shows several high miss rate values closer to the bottom of the image which we attribute to outliers. Additionally, a strong increase of the miss rate at the center of the image can be observed. This is due to persons at great distances which are hard to detect aligning at a vertical position around the horizon. Similarly, the sharp distribution of o_{cy} at around 0.42 stems from the fixed vertical camera angle in our synthetic data. Finally, due to missing data at lower o_{cy} values we cannot conclude that this factor has no influence.

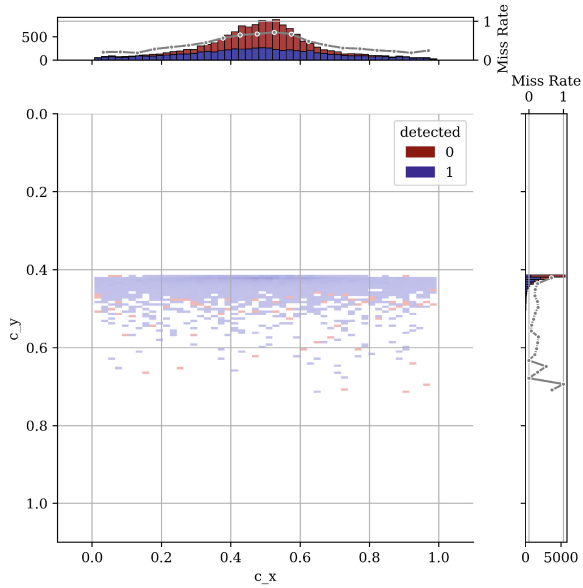


Fig. 10. Influence of person center point position in the image on detection performance, i.e. miss rate (gray).

5 Conclusions

Validation of AI based perception functions is an essential building block for safe automated driving. A failed pedestrian detection due to an underlying data bias in the training data can lead to grave consequences. In this work we presented a method to find data biases of a detector and its underlying real-world dataset by classifying a person according to several visual detection impairment factors into a detectable and non-detectable class through the use of rich annotated synthetic data. We demonstrated the effectiveness of this approach for the real-world datasets CityPersons (CP) and EuroCity Person (ECP) by detecting 23 ethnicity and geolocation based data biases in the CP and 15 in the ECP dataset. Further, 6 missing pedestrian poses were identified in both datasets. Here we remarked that the ground-truth of the validation data has to be carefully defined to generate meaningful data bias findings. Last, we investigated the influence of each visual impairment factor on the overall pedestrian detection performance and came to the conclusion that the visible pixels and distance to the observer have the highest influence with very high Spearman correlations ($|\rho_s| = 0.995$) to the miss rate. The contrast measure o_{cfull} showed a very good correlation ($|\rho_s| = 0.953$) while the horizontal position in the image has no effect ($|\rho_s| =$

0.017). The influence of the vertical position indicates no influence as well, but due to a lack of data the result is inconclusive.

Acknowledgement. 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. Abrecht, S., Gauerhof, L., Gladisch, C., Groh, K., Heinzemann, C., Woehrle, M.: Testing deep learning-based visual perception for automated driving. ACM Trans. Cyber-Phys. Syst. (TCPS) **5**(4), 1–28 (2021)
2. Bernhard, J., Schulik, T., Schutera, M., Sax, E.: Adaptive test case selection for DNN-based perception functions. In: 2021 IEEE International Symposium on Systems Engineering (ISSE), pp. 1–7. IEEE (2021)
3. Bolte, J.A., Bar, A., Lipinski, D., Fingscheidt, T.: Towards corner case detection for autonomous driving. In: 2019 IEEE Intelligent vehicles symposium (IV), pp. 438–445. IEEE (2019)
4. Braun, M., Krebs, S., Flohr, F.B., Gavrila, D.M.: Eurocity persons: a novel benchmark for person detection in traffic scenes. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1844–1861 (2019). https://doi.org/10.1109/TPAMI.2019.2897684
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: high quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1483–1498 (2019)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
7. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: a benchmark. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 304–311 (2009). https://doi.org/10.1109/CVPR.2009.5206631
8. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2011)
9. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net (2021). https://openreview.net/forum?id=YicbFdNTTy
10. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: an open urban driving simulator. In: Conference on Robot Learning, pp. 1–16. PMLR (2017)
11. Gannamaneni, S., Houben, S., Akila, M.: Semantic concept testing in autonomous driving by extraction of object-level annotations from Carla. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1006–1014 (2021)
12. Gauerhof, L., Munk, P., Burton, S.: Structuring validation targets of a machine learning function applied to automated driving. In: Gallina, B., Skavhaug, A., Bitsch, F. (eds.) SAFECOMP 2018. LNCS, vol. 11093, pp. 45–58. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99130-6_4
13. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, pp. 580–587 (2014)
15. Grau, O., Hagn, K., Sha, Q.S.: A Variational synthesis approach for deep validation. In: Fingscheidt, T., Gottschalk, H., Houben, S. (eds.) Deep Neural Networks and Data for Automated Driving, pp. 359–381. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-01233-4_13
16. Hagn, K., Grau, O.: Improved sensor model for realistic synthetic data generation. In: Computer Science in Cars Symposium. CSCS 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3488904.3493383
17. Lyssenko, M., Gladisch, C., Heinzemann, C., Woehrle, M., Triebel, R.: From evaluation to verification: towards task-oriented relevance metrics for pedestrian detection in safety-critical domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 38–45 (2021)
18. Lyssenko, M., Gladisch, C., Heinzemann, C., Woehrle, M., Triebel, R.: Instance segmentation in Carla: methodology and analysis for pedestrian-oriented synthetic data generation in crowded scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 988–996 (2021)
19. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 102–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_7
20. Rong, G., et al.: LGSVL simulator: a high fidelity simulator for autonomous driving. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE (2020)
21. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The Synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243 (2016)
22. Sämann, T., Schlicht, P., Hüger, F.: Strategy to increase the safety of a DNN-based perception for had systems. arXiv preprint arXiv:2002.08935 (2020)
23. Syed Sha, Q., Grau, O., Hagn, K.: DNN analysis through synthetic data variation. In: Computer Science in Cars Symposium. CSCS 2020, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3385958.3430479
24. Wang, J., et al.: Deep high-resolution representation learning for visual recognition. TPAMI **43**, 3349–3364 (2019)
25. Wrenninge, M., Unger, J.: Synscapes: A photorealistic synthetic dataset for street scene parsing (2018)
26. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: How far are we from solving pedestrian detection? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
27. Zhang, S., Benenson, R., Schiele, B.: Citypersons: a diverse dataset for pedestrian detection. In: CVPR (2017)

7.6 Publication 6

Increasing pedestrian detection performance through weighting of detection impairing factors

Korbinian Hagn and Oliver Grau

Published in

2022 Proceedings ACM Computer Science in Cars Symposium. [HG22a]

DOI: 10.1145/3568160.3570225

Increasing pedestrian detection performance through weighting of detection impairing factors

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

Figure 1: Improving pedestrian detectors for objects at low contrast, medium occlusion and greater distance. Previous methods (green boxes) missed pedestrians are now detected with our method (blue & green boxes).

ABSTRACT

Object detection is a matured technique, converging to the detection performance of human vision. This paper presents a method to further close the remaining gap of detection capability by investigating visual factors impairing the detectability of objects. As some of these factors are hard or impossible to measure in real sensor data, a detector is trained on synthetic data making perfect measurements and ground truth data available at a large scale. The resulting detector is then used to calibrate an empirical weighting loss, which weights samples of real training data and their corresponding detection impairing factors. The method is applied to the task of pedestrian detection in traffic scenes. The effectiveness of the empirical detection impairment weighting loss (DIW loss) is demonstrated on a detector trained on the CityPersons dataset and reaches a new state-of-the-art detection performance on this benchmark, improving the previous by 1.88%.

CSCS '22, December 8, 2022, Ingolstadt, Germany

ACM ISBN 978-1-4503-9786-5/22/12…$15.00
https://doi.org/10.1145/3568160.3570225

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; **Learning settings**; *Rendering*.

KEYWORDS

Pedestrian Detection, Object Detection, 2D-Bounding Box Detection, Synthetic Data

ACM Reference Format:
Korbinian Hagn and Oliver Grau. 2022. Increasing pedestrian detection performance through weighting of detection impairing factors. In *Computer Science in Cars Symposium (CSCS '22), December 8, 2022, Ingolstadt, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3568160.3570225

1 INTRODUCTION

Modern deep learning-based object detectors are capable of detecting objects with unprecedented accuracy. The detection boundary of a vision task is often compared to one of the human detection capabilities. While closing in on reaching this detection limit it is necessary to understand what properties of an object in an image make it difficult for a human to detect this very object. In Figure 2 we argue that for a set of data objects in a dataset, there is (i) a non-detectable subset, defined as objects that are not detectable due to visual impairment factors introduced in the course of this paper, (ii) the detectable subset, which is the set of objects that are detectable by a human observer, and (iii) the set of detectable objects consisting of objects that can be detected by state-of-the art approaches.

This detectable subset marks the region of improvement that we try to achieve by investigating the factors that objects in this subset have in common, e.g., occlusions, contrast, and other measures, and we argue that by retraining a detector to be specifically aware of these factors will improve the detection performance.

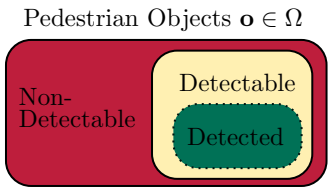


Figure 2: The detected subset can be improved to equal the detectable subset.

Driven by recent progress in synthetic data generation, new possibilities to produce high-realism data for computer vision tasks are given. Automotive applications especially benefit from the generation of rare cases and can potentially augment their training data to improve their performance at correctly detecting them. Especially for understanding the prediction decision-making of a detector, the pixel-accurate ground truth and the rich meta-data that can be extracted from the synthetic data generation process are highly beneficial to find relations of a visual factor, e.g., contrast, of an object and its detection.

The contribution of this paper are as follows:

- We define factors that are impairing the visual detection capability of an object and identify the influence of each factor in an ablation study of our method
- We build an empirical loss function taking these factors into account by weighting objects and their associated impairment factor according to their vicinity on a set of objects that are in the detectable region of a dataset. We demonstrate the efficacy of our method by evaluating the trained pedestrian detector performance on the CityPersons benchmark. Further, we compare our approach to two popular weight sampling methods and show that we can outperform both these methods.

2 RELATED WORKS

2.0.1 Sample weighting. The weighting of training samples is a well-researched topic in the literature. Importance sampling [25] for example, is a well-known statistical method assigning weights to samples to match distributions. Hard examples mining [33, 39], is another representative of sample weighting done by oversampling harder, i.e., higher gradient samples or boosting algorithms as AdaBoost [12] sample harder examples for subsequent classifier training. Another method giving attention to special or harder samples is done through the focal loss [30], by disentangling close objects in the repulsion loss [45] and the aggregation loss [50] which enforces object proposals to locate compactly to the designated ground truth object.

Contrasting these loss terms is self-paced learning [27], where sample weights are obtained through optimizing the weighted training loss to encourage learning of easier examples first.

To prevent these methods overfitting various regularization techniques have been explored [23, 27, 31]. These strategies were found to have the advantage of being more robust against the training set bias [38].

The class imbalance problem, which is tackled by using dataset resampling [4, 9] or cost-sensitive weighting [26, 42] is not a concern in our work as for the case of pedestrian detection, we only have two different classes, pedestrian and background. Multi-stage detectors, as we are utilizing in this work, implement a region proposal network (RPN) with a subsequent sampling of foreground and background proposals, efficiently balance the classes that are used for training the detector.

Annotated real-world data is subject to inaccuracies introduced by human annotators, which can be described as noise and has been thoroughly studied [34] and mitigation strategies have been proposed [24, 29, 37], [21, 43]. For synthetically generated datasets, as we are utilizing in this work, the noisy label problem is not relevant.

Contrasting our weighting approach to meta-learning [38, 41], [1, 28] we are not adapting the weights in an online fashion but rather steer the attention of the network to train, i.e., the weighting of samples, towards samples that are hard to detect by definition of their visual impairment factors.

Differing from the previous approaches, our weighting method not only considers hard examples, found by filtering a detector false negatives but combines these with a new range of visual impairment factors. The weights are computed offline and adding this weighting loss term to the overall loss only requires simple weight multiplication with the per-object loss term without tuning any additional hyperparameters.

In other words, our approach puts the training focus on objects that are hard to detect by measures that reflect the human vision.

2.0.2 Synthetic data generation. Recent methods have evolved that allow for the generation of synthetic data, especially for automotive tasks, i.e. pedestrian detection, most notably to mention is Carla [11]. Instead of a whole simulation of an automotive driving scenario also variational automotive scene creation approaches have been introduced for detector evaluation [13].

2.0.3 Detection models. While there is a recent popularity increase of models based on the Transformer architecture [10], stemming from Natural Language Processing (NLP), that are applied to visual computing tasks, most automotive detection benchmarks are still dominated by convolutional neural network (CNN) based methods. More recent and advancing the R-CNN [15] and Faster R-CNN [14] model is the Cascade R-CNN [3] we are using throughout this work. These multi-stage detector architectures commonly utilize backbones trained on ImageNet [6], such as HRNet [44] or MobileNet [22].

2.0.4 Detection impairing factors. Besides label noise and sensor noise, detection impairing factors such as occlusion rate and contrast were analyzed for their effect on object detection performance.

While there are factors that are believed to not influence the detection performance, for example, contrast [48] of an object, we found evidence that contrast is still relevant to the object detection performance. Occlusion on the other side is acknowledged to be one of the most influential factors and has already been addressed by several works [7, 8]. While we not only address these factors as influential for detection performance, we also add several other factors with relevance to the detection performance, albeit being correlated, and investigate their influence.

3 METHODOLOGY

3.1 Detection impairing factors

The major factors of an object we consider to be influential of the detection capability of a detector are visualized in Figure 3. Beginning with the placement of a pedestrian in an image, we get this information from the bounding box coordinates from the ground truth labels of the dataset. The coordinates are defined by the center o_{cx}, o_{cy} coordinated and the width o_w and height o_h of the bounding box in $[pixels]$.

Next, distance to the observer o_d, i.e., camera, in $[m]$ and the number of visible pixels o_{vp} of the object. The distance to the observer is extracted for the synthetic data from the 3D placement in the rendered scene. The information about the number of visible pixels is extracted from the instance segmentation label, by counting the pixel that belong to the object. For real-world datasets without any additional sensor information, e.g., LIDAR or Radar, the distance can be estimated by first normalizing the bounding box diagonal of an object by the diagonal of the image. Then, the focal length in $[m]$, that we get from multiplying the focal length in pixel by the size of a pixel on the sensor, is divided by the product of the normalized bounding box diagonal and the diagonal of the sensor measured in $[m]$. The result is multiplied by an estimated average-sized pedestrian of $1.75m$ height. The intuition behind this method is following a simple intercept theorem, i.e., the ratio of the object on the sensor diagonal to the focal length is equivalent to the ratio of an average-sized pedestrian with $1.75m$ over the distance to the camera. This method is used in this work to extract distance information for the real-world dataset.

For determining the occlusion rate o_{ocl}, i.e., the ratio of non-occluded pixels to the whole pixels of an object, it is again advantageous to use synthetic data. Here, one can extract the number of occluded and non-occluded pixels by counting the instance segmentation ground truth pixels with, and without occluding objects. For real-world images, an estimation of this value is necessary. The ground truth of the dataset should provide two different bounding box annotations, one which annotates only the visible part of the object and a second one which annotates the estimated whole object. Now, the occlusion rate can be estimated by the quotient of the visible bounding box over the whole bounding box.

Last, we extract different contrast measures of an object. The first contrast measure o_{cfull} is defined by calculation of the euclidean distance of the mean object color to its surrounding background. This is done by dilating the instance segmentation mask of the object and subtracting the undiluted segmentation mask so that we get the surrounding 5 pixel border of the object. The contrast is calculated by the euclidean distance of the mean RGB object color to the mean surrounding background color. A more sophisticated contrast measure o_{cmean} is by segmenting the object into 12 smaller segments and again calculating the mean color of a segment and the euclidean distance to its neighboring mean background color. The resulting measure is then derived by averaging over all segments. Another considered contrast measure o_{cedge} is calculated by taking only the 5 pixels on the edge of the instance label, then again calculating the mean color and the euclidean distance to the mean of the surrounding background.

3.2 Synthetic data generation

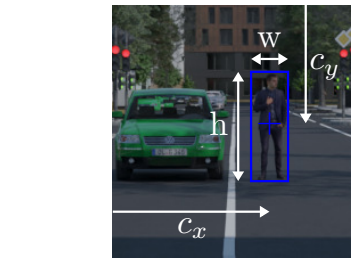
The synthetic data used in this contribution is generated by a data synthesis pipeline and includes special modules to compute meta- or ground truth data which is hard or impossible to observe and measure in real data. One example is the pixel-accurate occlusion rate of an object, by differencing the mask of the un-occluded object, computed in a separate rendering pass from the occluded object in the complete scene.

To achieve a representative calibration of the detector, the synthetic data should have similar characteristics than real scenes. This is also described as domain gap between the real and synthetic data. We achieve highly realistic synthetic data by three levels: i) An automated scene generator produces scenes with similar complexity as those in the used real data, ii) we use similar 3D objects from an asset database and iii) a realistic sensor simulation building on the work described in [16]. The rendering process delivers realistic scene illumination in linear color space with floating accuracy based on the Blender Cycles path-tracing rendering engine[1], followed by a sensor simulation that includes simulation of effects like sensor noise, lens distortions, and chromatic aberrations and a tone mapping to integer sRGB color space. The parameters of the sensor simulation are tuned to match the characteristics of the Cityscape data similar to [17, 18].

For the purpose of this paper, we use a dataset that contains complex urban street scenes with a variation of objects (about 300), such as different houses, vehicles, street elements, and human characters (about 150) automatically placed from an asset database. Figure 4 depicts some example frames from that data set. The synthetic data generation pipeline also computes various metadata and ground truth, including semantic and instance segmentation, the distance of objects to the camera, occlusion rates, 2D + 3D bounding boxes, radiometric object features including contrast measures as introduced above.

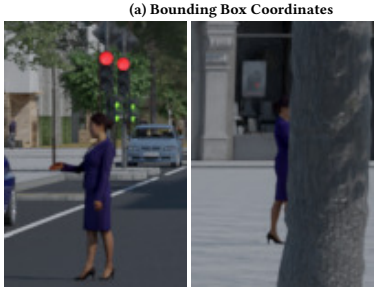
One of the advantages of synthetic data in our method is the precision and deterministic nature of the label and bounding box meta-data, which is free from noise, as all the generated labeling data is pixel accurate. For evaluations with this dataset, the pedestrians in the images are filtered to guarantee that only the considered impairment factors are influential on the detection performance. This means that pedestrians that are too close to one another and would not be detected due to non-maximum suppression (NMS) are ignored. The resulting synthetic dataset $\mathcal{D}_{synth}$ consists of 17012 pedestrians for training and 26745 pedestrians for evaluation.

[1] blender.org



(a) Bounding Box Coordinates

(b) Distance and Visible Pixels



(c) Occlusion

(d) Contrast Measures

Figure 3: The potential detection performance impairing factors we consider in this work are from top left (a) to bottom right (d): (a) bounding box coordinates (o_{cx}, o_{cy}, o_h, o_w), (b) distance and number of visible pixels of a pedestrian (o_d, o_{vp}), (c) rate of occlusion (o_{ocl}), (d) contrast of a pedestrian (red) to its background (blue) calculated by the full pedestrian silhouette, segment wise and edge wise($o_{cfull}, o_{cmean}, o_{cedge}$).

We are utilizing the 2D bounding box enhancement of the automotive segmentation dataset of Cityscapes (CS) [5] named CityPersons (CP) [49]. CS is an inner-city automotive real-world dataset, recorded mainly in Germany and some smaller parts in Austria, Switzerland, and France. CP consists of 19654 pedestrians for training and 3938 pedestrians for evaluation. The reasoning behind choosing this dataset is the availability of instance segmentation labels, which are essential to calculate the contrast factors and the precise number of pixels. To the best of our knowledge, the combination of CS and CP labels are the only pedestrian detection datasets with instance and bounding box labels, including occlusion information. BDD100k [46] would be another viable candidate for our method but is missing occlusion information.

3.3 Sample weighting

Following, we define the distance measure and the weighing loss that we utilize to improve the detection capability of a 2D bounding box detector.

3.3.1 Distance Measure. We begin with a dataset of pedestrian objects $\mathbf{o} \in \Omega = \{\mathbf{o}_1, \ldots, \mathbf{o}_O\}$ where O is the number of pedestrians in a dataset. To get all the objects that were missed by the detector, i.e., false negatives, we filter these pedestrians to $\Omega_m = \mathbf{f}(\Omega)$. A sample vector of the objects metadata $\mathbf{o}$ is defined as follows: $\mathbf{o} = (o_{cx}, o_{cy}, o_h, o_w, o_d, o_{vp}, o_{ocl}, o_{cfull}, o_{cmean}, o_{cedge})$, with every entry in this vector normalized to $\mu = 0$ and $\sigma^2 = 1$ and equaling to the detection impairment factors previously described.

The weighting of a new object $\mathbf{o}$ is then done by calculating the Mahalanobis distance [32] from its impairment factors to the objects in Ω_m. The distance for a new object sample is then defined as follows:

$$\bar{\mathbf{o}}_m = \frac{1}{|\Omega_m|} \sum_{i \in \Omega_m} \mathbf{o}_i, \tag{1}$$

$$d_{mh}(\mathbf{o}, \bar{\mathbf{o}}_m) = \sqrt{(\bar{\mathbf{o}}_m - \mathbf{o})^T V_m^{-1} (\bar{\mathbf{o}}_m - \mathbf{o})}. \tag{2}$$

With V_m^{-1} being the inverse covariance matrix of Ω_m.

3.3.2 Detection impairment weighting loss (DIW loss). The classification output is a discrete probability distribution $p = (p_0, \ldots, p_s)$ computed by a softmax with s being the predicted class $s \in \mathcal{S} = \{0, \ldots, S-1\}$ and here, $S = 2$, i.e., we differentiate between pedestrian ($s = 1$) and background class ($s = 0$). The respective target

Figure 4: Our fully parameterizable generation pipeline allows rendering pedestrians at any size, occlusion, time of day, and distance to the camera.

class is defined as $u = (u_0, \ldots, u_s)$ where the target class is again 1 and background classes are set to 0.

The categorical cross entropy loss, i.e, classification loss is then defined as follows:

$$J^{cls}(p, u) = -\sum_{i \in S} u_i log(p_i). \tag{3}$$

The bounding box regression outputs for a prediction of class s is defined as $t^s = (t^s_x, t^s_y, t^s_w, t^s_h)$ and the respective target bounding box is defined as $v = (v_x, v_y, v_w, v_h)$. Now, using these two to calculate the bounding box regression loss, i.e., localization loss:

$$J^{loc}(t^s, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t^s_i - v_i), \tag{4}$$

in which the $smooth_{L_1}$ loss from [14] is defined as

$$smooth_{L_1}(x, y) = \begin{cases} 0.5 \cdot (x-y)^2/\beta, & \text{if } |x-y| < \beta \\ |x-y| - 0.5 \cdot \beta, & \text{otherwise.} \end{cases} \tag{5}$$

We apply $\beta = \frac{1}{9}$ as was used in training Mask R-CNN [20].

Adding the localization loss and the classification loss together with the distance measure to our detection impairment weighting loss (DIW loss):

$$J^{total}(p, u, t^s, v) = \alpha \cdot (J^{cls}(p, u) + \lambda[s = 1] J^{loc}(t^s, v)),$$
$$where\ \alpha = \begin{cases} \frac{1}{1+d_{mh}(\mathbf{o}_s, \overline{\mathbf{o}}_m)}, & \text{if } s = 1 \\ \gamma, & \text{otherwise.} \end{cases} \tag{6}$$

True positives (TP) and false negatives (FN) with their respective impairing factors $\mathbf{o}_s$ receive higher weights the smaller the distance to the missed observations $\overline{\mathbf{o}}_m$. The weighting of false positives (FP) and true negatives (TN) can be steered by γ and is throughout this paper set to $\gamma = 0.5$. λ is a weighting parameter for weighting the contribution of classification and localization loss, here $\lambda = 1$. $[s = 1]$ indicates the iverson bracket, evaluating to 1 if the predicted class is correctly classified.

3.3.3 Weight computation. Figure 5 shows the experimental setup for the weight computation. First, we use our synthetic validation dataset and extract the detection impairing factors on a per-object level from the individual images. This synthetic dataset is pre-filtered to only include pedestrians of height above 33 pixels, occlusion rates below 0.8, and overlapping by less than 0.5 with

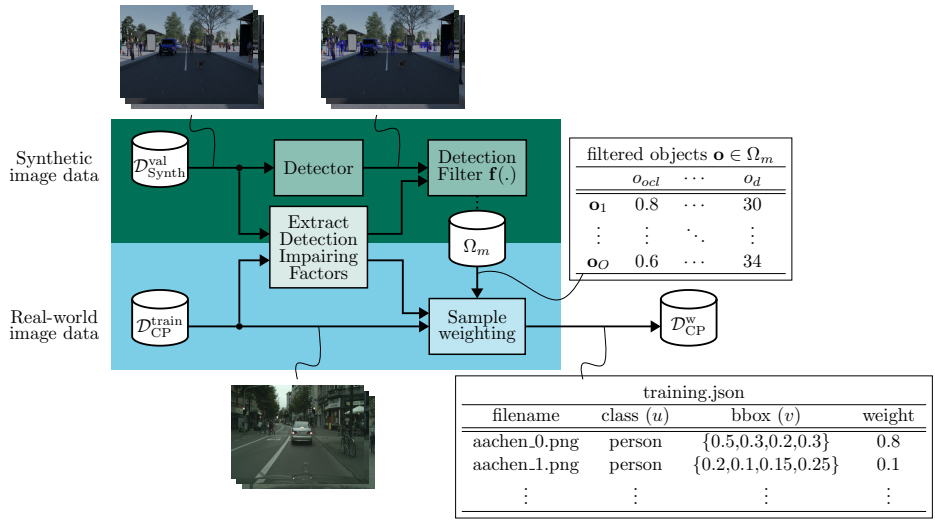


Figure 5: Generation of the training weights for pedestrian objects of a real-world CityPersons dataset.

one another to guarantee that only the detector capabilities are significant for the detection of an object and not other criteria, such as non-maximum suppression (NMS). In parallel, we use a detector that has been trained on the synthetic training dataset to generate bounding box predictions on this validation set. Next, both the predictions and the extracted impairment factors are presented to the detection filter $\mathbf{f}(.)$, which sorts out all TP objects. The remaining object's impairment factors, are stored in the filtered Objects set Ω_m. The distribution of the set Ω_m could at this stage be approximated by a multidimensional distribution function and used to compare and weight new samples to it. But due to the added complexity and inaccuracies of the approximation to the real distribution function of Ω_m we apply the straightforward approach to use the samples in the set to calculate the weighting. Now, we take our real-world training dataset CityPersons ($\mathcal{D}^{train}_{CP}$) and extract the detection impairing factors per-object. This per-object information is now weighted according to its distance to the non-detected samples of the synthetic dataset as defined in 2. The resulting bounding box information with the corresponding weights are fed into a new Dataset $\mathcal{D}^{DIW}_{CP}$ and then used for training a detector.

3.3.4 Visualization of weight surface. In Figure 6 the weighting surface, i.e., $\frac{1}{1+d_{mh}(\mathbf{o}_s,\bar{\mathbf{o}}_m)}$, for two of the ten considered factors is exemplary depicted. The factors occlusion and visible pixels per-object are normalized. The peak, i.e., the highest weight is where occlusion and number of visible pixels in the synthetic validation set were decisive for a missed detection, here we can see high occlusions and a low number of visible pixels. Descending from the

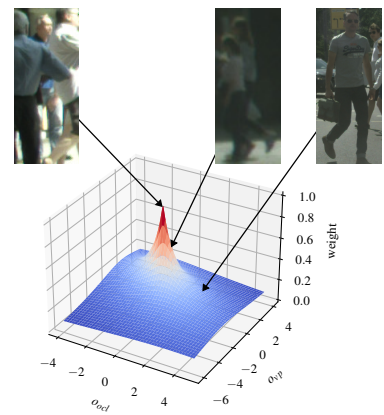


Figure 6: Visualizing the weight distribution surface with two of the visual impairment factors.

peak the occlusions diminish until the whole pedestrian is visible

with a high number of visible pixels. At this point, the weighting of the pedestrian lowered to around 0.1.

3.3.5 Training the detector. The detector we are utilizing in this work is the Cascade R-CNN [3] detector. 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 $lr = 0.02$, with a weight decay of 0.0001 and momentum of 0.9 for the stochastic gradient descent (SGD) optimizer. Image augmentations like random flipping and cropping, as well as brightness and saturation distortions, were used, as is common practice. To stabilize training we apply weight averaging as introduced by [40].

3.3.6 Evaluation. When evaluating the CP dataset, we are distinguishing between four different subsets as were defined in [49]. To evaluate the performance we use the log-average miss rate [2, 8] that is computed by averaging the miss rate (laMR) at nine false positive per image (FPPI) rates which are evenly spaced in log-space in the range 10^{-2} to 10^{0}.

3.3.7 Evaluation metric definition. In the main paper we use the log-average miss rate (laMR) as performance metric to compare our approach to state-of-the-art detectors. This evaluation metric is commonly used for automotive datasets such as CityPersons. Following definitions utilize the false positive (FP), false negative (FN), true positive (TP) and false positive (FP) at different evaluation confidence thresholds c. The definition of the miss rate (MR) for different confidence thresholds c is as follows:

$$\mathrm{MR}(c) = \frac{\mathrm{FN}(c)}{\mathrm{TP}(c) + \mathrm{FN}(c)}, \tag{7}$$

and following the definition of FPPI per confidence threshold c:

$$\mathrm{FPPI}(c) = \frac{\mathrm{FP}(c)}{N}. \tag{8}$$

For $n \in \mathcal{N} = \{1, \ldots, N\}$ and here N is the number of images in the respective evaluation set. Combining these measures to the log-average miss rate (laMR):

$$\mathrm{laMR} = exp\Big(\frac{1}{9}\sum_{f \in \mathcal{F}} log(\mathrm{MR}(\underset{\mathrm{FPPI}(c) \leq f}{\mathrm{argmax}}\ (\mathrm{FPPI}(c))))\Big). \tag{9}$$

With f being equally spaced in the interval $f \in \mathcal{F} = [10^{-2}, 10^{0}]$.

4 RESULTS AND DISCUSSION

A comparison of our method to state-of-the-art results on the CityPersons (CP) test set benchmark[2] is shown in Table 1. Reaching state-of-the-art results we utilized the EuroCity Persons (ECP) [2] dataset for pre-training the detector and then fine-tune the model on the weighted CP training set. The performance improves 1.08% on the *Reasonable* subset with our method compared to the recent best detector APD [47] pre-trained on ECP. On the *Reasonable Small* and *All* subset, our DIW loss improves 1.80% and 1.88% respectively against the previous leader Pedestron [19], which applies the same backbone, detector and pre-training datasets. With a decrease of 1.29% on the *Heavy* subset, the performance compared to Pedestron is worse.

[2]https://github.com/cvgroup-njust/CityPersons

4.1 Pre-training influence

State-of-the-art results were achieved by pre-training the detector on the ECP dataset, however, we can show improvements with our method compared to the non-weighted baseline even without pre-training the detector as is shown in Table 2.

Improvements of up to 2% on the *All* subset are achieved by our method if the detector is not pre-trained on a similar automotive dataset.

4.2 Backbone influence

The DIW loss method improves the detection performance independent of the used backbone as can be seen in Table 3.

Performance gains are achieved in all CP subsets for the MobileNet backbone and on three of the four subsets on the HRNet backbone.

4.3 Ablation Study

To investigate the influence of each considered factor an ablation study was conducted with the results presented in Table 4. When the bounding box coordinates (o_{cx}, o_{cy}, o_h, o_w) are removed from the weight calculation, the overall performance on the *All* subset decreases only slightly, indicating the rather small influence of this factor. Further, removing the distance (o_d) from weight calculation leads to a slightly more drastic decrease in the overall performance which decreases even further when we remove the contrast measures ($o_{cfull}, o_{cmean}, o_{cedge}$) from weight calculation. When only the visible pixel count (o_{vp}) is used for weight calculation, i.e. removing the occlusion (o_{ocl}), the overall performance gain compared to the Baseline is still around 1% which gives a clear indication for the visible pixel count being the most influential performance factor.

4.4 Comparison with sampling losses

We compare our DIW loss with the two widely applied sampling losses for two-stage detectors: online hard example mining (OHEM) [39] and IoU balanced negative sampling [35]. Results are shown in Table 5. Again a pre-trained HRNetV2p backbone in combination with the Cascade R-CNN detector were trained and evaluated.

Our DIW loss outperforms both considered sampling losses on three subsets by at least 1% laMR while it suffers only a 0.4% decrease in performance on the *Heavy* subset. Both sampling losses lead to a slightly worse performance compared to regular training.

4.5 Training Performance

Figure 7 depicts the training progress when evaluated on the four CP subsets. While the *Reasonable* and *Reasonable Small* subsets performance improve measured by decreasing laMR, the *Heavy* subset sees a strong incline in laMR and with it the performance on the *All* subset as well.

The strong incline on the *Heavy* subset, and overall worse performance on this subset, can be attributed to the pre-filtering of the synthetic dataset to only contribute pedestrian objects at pixel height above 33, occlusion rates below 0.8 and no pedestrian instances with an overlap greater than 0.5 to one another to prevent NMS causing missed predictions. This leaves us with a subset more similar to the *Reasonable* and the *Reasonable Small* subsets and

Table 1: We can show that our approach improves the state-of-the-art on three of the four subsets on the CityPersons test set benchmark.

Model	Pre-trained	laMR % *Reasonable* ↓	*Reasonable Small* ↓	*Heavy* ↓	*All* ↓
Adapted Faster R-CNN [49]	×	12.97	37.24	50.47	43.86
OR-CNN [50]	×	11.32	14.19	51.43	40.19
MGAN [36]	×	9.29	11.38	40.97	38.86
Cascade R-CNN [3]	×	11.62	13.64	47.14	37.63
APD [47]	×	8.27	11.03	35.45	35.65
APD-Pretrain [47]	✓	7.31	10.81	28.07	32.71
Pedestron [19]	✓	7.69	9.16	**27.08**	28.33
Cascade R-CNN & DIW loss	✓	**6.23**	**7.36**	28.37	**26.45**

Table 2: Performance can be improved on pre-trained and non-pre-trained networks.

Dataset	Pre-trained	laMR % *Reasonable* ↓	*Reasonable Small* ↓	*Heavy* ↓	*All* ↓
$\mathcal{D}_{CP}^{train}$	×	12.92	16.24	46.08	37.24
$\mathcal{D}_{CP}^{DIW}$	×	12.39	14.63	44.60	35.23
$\mathcal{D}_{CP}^{train}$	✓	7.55	8.55	**27.47**	26.89
$\mathcal{D}_{CP}^{DIW}$	✓	**6.51**	**7.10**	28.39	**25.35**

Table 3: Measured performance increase by our method for different backbones.

Dataset	Backbone	laMR % *Reasonable* ↓	*Reasonable Small* ↓	*Heavy* ↓	*All* ↓
$\mathcal{D}_{CP}^{train}$	MobileNet v2	10.24	11.91	*36.03*	31.48
$\mathcal{D}_{CP}^{DIWloss}$	MobileNet v2	*9.60*	*11.71*	36.62	*30.81*
$\mathcal{D}_{CP}^{train}$	HRNetV2p-W32	7.55	8.55	**27.47**	26.89
$\mathcal{D}_{CP}^{DIWloss}$	HRNetV2p-W32	**6.51**	**7.10**	28.39	**25.35**

therefore samples from these subsets will receive higher weights than from the *Heavy* subset. Another influence on this behavior are the impairment factor estimations for the real-world data, these tend to be imprecise the less visible, i.e., smaller or more occluded, the pedestrian gets. Therefore, high-quality annotations and metadata, such as from the synthetic dataset, are a key for our method to work optimally.

Our Training results ultimately suggest that the overall best training method can be found by utilization of a pre-trained backbone on ImageNet, pre-training the detector on a different automotive pedestrian detection dataset, i.e., ECP, and then fine-tune for a few epochs on the weighted target dataset.

Further visualizations and also evaluation results regarding influence of the used backbone, as well as the influence of pre-training a network can be found in the Appendix. We show that for both cases our loss still increases the detection performance independent of these factors. Monitoring the detection performance in the course of the training leads to overall recommendation of fine-tuning with our method for few epochs to receive best results.

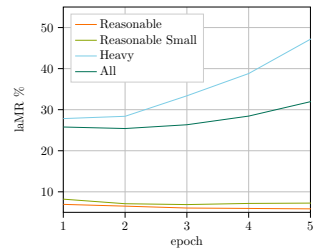


Figure 7: Training progress measured by evaluation on the *Reasonable, Reasonable Small, Heavy* and *All* subset.

5 CONCLUSIONS

In this paper, we show by defining and extracting detection impairing factors, i.e., occlusion rate, number of visible pixels, distance to the camera, three different contrast measures, and the bounding box coordinates per object, can be used to implement a new form of a sample weighting loss. Utilizing synthetic data for rich and precise metadata extraction, a new loss is implemented by weighting the samples of a real-world dataset CityPersons (CP) according to their distance of extracted detection impairment factors to the false negatives ones of objects on the synthetic dataset. Showing the extraction of detection impairment factors is also possible on real-world data, but it has to rely on several approximations. Last, we evaluated the performance gain of our method after training on the weighted CP automotive pedestrian detection dataset on the CP benchmark and could improve the current state-of-the-art on 3 out of 4 detection subsets. We analyzed the performance gain of our method on different backbones and with pre-trained or non-pre-trained backbones and could show that for each of these backbones the performance improved. An ablation study of our detection impairment factors showed the most influential factor is the number of visible pixels of an object. Last, we investigated the training

Table 4: Performance on the validation set decreases when the CNN is trained with lesser factors for the distance calculation.

										laMR %			
o_{cx}	o_{cy}	o_h	o_w	o_d	o_{cfull}	o_{cmean}	o_{cedge}	o_{ocl}	o_{vp}	*Reasonable* ↓	*Reasonable Small* ↓	*Heavy* ↓	*All* ↓
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**6.51**	**7.10**	**28.39**	**25.35**
				✓	✓	✓	✓	✓	✓	6.67	7.49	28.78	25.41
					✓	✓	✓	✓	✓	6.85	7.15	29.52	25.62
								✓	✓	6.95	7.14	29.93	25.82
									✓	7.01	7.22	30.10	25.93
										7.55	8.55	27.47	26.89

Table 5: Comparison of our DIW loss with other sampling losses on the CP val set.

Method	laMR % *Reasonable* ↓	*Reasonable Small* ↓	*Heavy* ↓	*All* ↓
OHEM [39]	7.61	8.75	**27.99**	26.70
IoUBalanced Negative Sampling [35]	7.66	8.63	28.40	26.84
DIW loss	**6.51**	**7.10**	28.39	**25.35**

performance and recommend our method as a fine-tuning method to get the best results.

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] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* 29 (2016).

[2] Markus Braun, Sebastian Krebs, Fabian B. Flohr, and Dariu M. Gavrila. 2019. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1. https://doi.org/10.1109/TPAMI.2019.2897684

[3] Zhaowei Cai and Nuno Vasconcelos. 2019. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] 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*.

[7] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 304–311. https://doi.org/10.1109/CVPR.2009.5206631

[8] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. 2011. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence* 34, 4 (2011), 743–761.

[9] Qi Dong, Shaogang Gong, and Xiatian Zhu. 2017. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 1851–1860.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[12] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[13] Sujan Gannamaneni, Sebastian Houben, and Maram Akila. 2021. Semantic Concept Testing in Autonomous Driving by Extraction of Object-Level Annotations from CARLA. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1006–1014.

[14] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[16] 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

[17] 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

[18] 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

[19] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. 2021. Generalizable pedestrian detection: The elephant in the room. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11328–11337.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[21] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems* 31 (2018).

[22] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*

abs/1704.04861 (2017). arXiv:1704.04861 http://arxiv.org/abs/1704.04861

[23] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[24] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*. PMLR, 2304–2313.

[25] Herman Kahn and Andy W Marshall. 1953. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America* 1, 5 (1953), 263–278.

[26] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems* 29, 8 (2017), 3573–3587.

[27] M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems* 23 (2010).

[28] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).

[29] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1910–1918.

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[31] Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. 2017. Self-paced cotraining. In *International Conference on Machine Learning*. PMLR, 2275–2284.

[32] Prasanta Chandra Mahalanobis. 1936. On the generalized distance in statistics. National Institute of Science of India.

[33] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. 2011. Ensemble of exemplar-svms for object detection and beyond. In *2011 International conference on computer vision*. IEEE, 89–96.

[34] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. *Advances in neural information processing systems* 26 (2013).

[35] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. 2019. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 821–830.

[36] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. 2019. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4967–4975.

[37] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* (2014).

[38] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*. PMLR, 4334–4343.

[39] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 761–769.

[40] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* 30 (2017).

[41] Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

[42] Kai Ming Ting. 2000. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer.

[43] Arash Vahdat. 2017. Toward robustness against label noise in training deep discriminative neural networks. *Advances in Neural Information Processing Systems* 30 (2017).

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. 2019. Deep High-Resolution Representation Learning for Visual Recognition. *TPAMI* (2019).

[45] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. 2018. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7774–7783.

[46] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2636–2645.

[47] Jialiang Zhang, Lixiang Lin, Jianke Zhu, Yang Li, Yun-chen Chen, Yao Hu, and Steven C. H. Hoi. 2021. Attribute-Aware Pedestrian Detection in a Crowd. *IEEE Transactions on Multimedia* 23 (2021), 3085–3097. https://doi.org/10.1109/TMM.2020.3020691

[48] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. 2016. How Far Are We From Solving Pedestrian Detection?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[49] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. 2017. CityPersons: A Diverse Dataset for Pedestrian Detection. In *CVPR*.

[50] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. 2018. Occlusion-aware R-CNN: detecting pedestrians in a crowd. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 637–653.

7.7 Publication 7

SynPeDS – A Synthetic Dataset for Pedestrian Detection in Urban Traffic Scenes

Thomas Stauner, Frédérik Blank, Michael Fürst, Johannes Günther, Korbinian Hagn, Philipp Heidenreich, Markus Huber, Bastian Knerr, Thomas Schulik, Karl Leiss

Published in

2022 Proceedings ACM Computer Science in Cars Symposium. [SBF+22]

DOI: 10.1145/3568160.3570230

SynPeDS: A Synthetic Dataset for Pedestrian Detection in Urban Traffic Scenes

Thomas Stauner
thomas.stauner@bmw.de
BMW Group
Germany

Frédérik Blank
frederik.blank@de.bosch.com
Robert Bosch GmbH
Germany

David Michael Fürst
david_michael.fuerst@dfki.com
DFKI Kaiserslautern
Germany

Johannes Günther
johannes.guenther@intel.com
Intel Corporation
Germany

Korbinian Hagn
korbinian.hagn@intel.com
Intel Corporation
Germany

Philipp Heidenreich
philipp.heidenreich@stellantis.com
Opel Automobile GmbH
Germany

Markus Huber
m.huber@accenture.com
Accenture Song Content GmbH
Germany

Bastian Knerr
bastian.knerr@qualityminds.de
QualityMinds GmbH
Germany

Thomas Schulik
thomas.schulik@zf.com
ZF Friedrichshafen AG
Germany

Karl-Ferdinand Leiß
leiss@movex.de
BIT Technology Solutions GmbH
Germany

ABSTRACT

We introduce the Synthetic Pedestrian Dataset (SynPeDS) which was designed to support a systematic safety analysis for pedestrian detection tasks in urban scenes. The dataset was generated synthetically with a real-time and a physically-based rendering pipeline and provides camera frames and in part associated LiDAR point clouds. It contains ground truth for semantic segmentation, instance segmentation, 2D and 3D bounding boxes, and in part, pose information and bodypart segmentation. In particular, it comes with a large amount of meta information for in-depth performance and safety analysis, e.g. addressing semantic properties of the pedestrians and their environment in the frames. Some scenarios were specifically designed to systematically cover certain safety-relevant or performance-reducing dimensions of the input space, defined in project *KI Absicherung*. The dataset does not claim to be complete or free of bias, but to support coverage and data distribution studies.

CCS CONCEPTS

• **Software and its engineering → Software safety**; • **Computing methodologies → Computer vision**; *Vision for robotics.*

KEYWORDS

Automated driving, Pedestrian detection

CSCS '22, December 8, 2022, Ingolstadt, Germany

ACM ISBN 978-1-4503-9786-5/22/12.
https://doi.org/10.1145/3568160.3570230

ACM Reference Format:
Thomas Stauner, Frédérik Blank, David 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 *Computer Science in Cars Symposium (CSCS '22), December 8, 2022, Ingolstadt, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3568160.3570230

1 INTRODUCTION

The ability to ensure safety of AI based computer vision functions is a central prerequisite for automated driving. One of the key requirements to reach this goal, is to analyze and to mitigate possible generalization insufficiencies of an AI-based function, especially for safety critical situations by using training and assurance data that targets at covering the domain in which it is to be used. Therefore the publicly funded project *KI Absicherung* has created the Synthetic Pedestrian Dataset (SynPeDS)[1] that explicitly targets at supporting performance and safety analyses for pedestrian detection in urban traffic scenarios. It is going to be published under a license that allows usage by academic and commercial parties.

While we think that currently synthetic data is only complementary to real data when it comes to development of environment perception for automated vehicles, it offers a set of important features. Being based on 3D scene models, firstly variations, coverage and bias of the data can be controlled explicitly. Secondly, it allows for in-depth "performance to metadata correlation analyses". Thirdly, it allows to simulate new sensor variants and mounting positions before they become available and fourthly compliance with the General Data Protection Regulation (GDPR) is easily controllable. W.r.t. safety, the ability to construct safety-relevant corner

[1] https://www.ki-absicherung-projekt.de/

cases not easily obtainable in reality is a great benefit. Furthermore, it is possible to vary single scene parameters, which can be semantic parameters like spatial object distribution and physical parameters like illumination, and to compute data for any combination of parameters. For safety assurance this is important in order to systematically analyse performance limiting influence factors. A further vital property is that high-quality ground truth and rich meta information can be automatically generated for synthetic data. Examples are light properties of the scene, the occluded-by relation and fine-grained attributes of objects, like appearance and pose of pedestrians.

The differentiating property of our dataset is its focus on metadata and scenes supporting safety analyses. It contains over 200k frames. Ground truth includes semantic segmentation, instance segmentation, 2D and 3D bounding boxes for all data. For parts of the data, pose information and bodypart segmentation, respectively, are available. A large part of the pedestrian poses was collected via motion capturing. Sensor data is provided for a front camera and, in parts of the dataset, for LiDAR. Moreover, rich meta information is available. For instance, knowing the sun direction and elevation in the scene and the contrast to background of every pedestrian allows a detailed analysis of the influence of lighting conditions on pedestrian detection. The data has been created by two different toolchains: (1) real-time rendering with Epic's Unreal Engine 4 (UE4) [7], see example in Fig. 1, and (2) physically-based rendering (PBR) with Blender Cycles [8], see example in Fig. 2. Within the project a third toolchain with PBR with OSPRay [28] has been developed that uses a realistic camera and LiDAR model. Providing data from this toolchain is future work in the sister project *KI Data Tooling*. With the publication of the dataset we want to stimulate research on methods for safe AI and for safety assurance of AI in computer vision.

The paper is structured as follows. Section 2 provides an overview of related work. In Section 3 we explain the ground truth format and meta information the dataset provides. Section 4 describes main design principles of the dataset, the tooling used for its production and essential aspects of its structure. The quality of the dataset in relation to other datasets and an example application from safety assurance are given in Section 5.

2 RELATED WORK

Synthetic data generation received broad interest in the last years. The Synthia dataset [24] is based on real-time rendering in Unity [26] and provides semantic segmentation for 13k single frames and a large further amount of frames from four sequences of driving through a virtual city. [23] introduce an approach of using the GTA V video game as basis to mine camera frames as well as corresponding semantic segmentation information. The dataset contains 25k frames of game engine quality. VKITTI [9] also uses Unity and models five scenes from the KITTI real dataset [10]. Standards assets from the Unity library are used. Pedestrians are not included in this dataset. In [30] the authors introduce the Synscapes dataset consisting of 25k procedurally generated single frames and rendered with PBR. The dataset shows that scene variance and rendering can be addressed as separate problems. Synscapes provides semantic segmentation, 2D and 3D bounding boxes. It also contains additional meta information and gives an example on how it can be used to study influence factors on the ML algorithm. Our dataset is partially created with real-time rendering and partially with PBR. Procedural generation is only applied to parts of the scene layout, which is based on five different junction geometries. While the dataset offers ground truth beyond that of related work, its main point of distinction is its focus on pedestrians and safety assurance. A large part of the pedestrian poses is based on motion capturing that we conducted and comprehensive meta information to support safety analysis of deep neural networks (DNNs) is available.

W.r.t. synthetic LiDAR [17] extends [23] by calculating pseudo-LiDAR data based on instance segmentation and depth information. [5] uses the CARLA simulator [6] to generate a synthetic KITTI-like dataset that also contains LiDAR point clouds. The sensor physics is approximately considered in the sense that simulated objects move further during rotation of the sensor, which is broken down into 100 simulation steps. Our dataset provides pseudo-LiDAR point clouds similar to [17] for a part of the dataset.

Recent GAN-based approaches offer to augment and thereby advance existing real and synthetic datasets to increase its size and degree of variation [29]. Note that in this work, we have focused on synthetic data generation only, and have not considered GAN-based approaches, due to the high degree of controllability needed for safety analysis.

3 GROUND TRUTH & META INFORMATION

The dataset provides RGB camera images in the OpenEXR and png file formats. Images have a resolution of 1920×1280 pixels and represent a camera system with a $60°$ horizontal field-of-view, except for few selected sequences that additionally contain different camera parameters for domain adaption and domain gap analysis. The LiDAR point clouds are produced corresponding to a Velodyne HDL-64E model and stored in a PCD format.

Each sequence contains a ground truth directory with annotations and meta information in a JSON file format, both on a per frame and per sequence level. Specifications and characteristics of the annotations and meta information are described in Sections 3.1 and 3.2.

3.1 Ground truth

The annotation format for our dataset is a super set derived from widely used datasets like CityScapes [3], KITTI [10], COCO [19] and OpenPose [1]. By using a super set, deriving the annotation format of prior datasets is possible. However, our data contains more detailed annotations, e.g. even fully occluded objects are annotated pixel-accurate and provide a value of occlusion and how many pixels are visible. Similarly for human pose annotations even occluded joints are annotated correctly allowing models to learn correct priors for occluded joints. Using the advantage of perfect ground truth in simulated data, the annotations in our dataset annotate every single instance pixel-accurate. For example, in groups of pedestrians each pedestrian is annotated – not only the group as a whole. This allows us to follow various object detection benchmark protocols. When required group annotations can be derived from individual annotations automatically.

Figure 1: SynPeDS example frame and ground truth information from Accenture Song Content GmbH: camera sensor image with 2D bounding boxes, semantic segmentation, depth channel, and skeletal information. ©Accenture Song Content GmbH

The overall structure of the annotations is to split them into task specific JSON files (2D bounding boxes, 3D bounding boxes, etc.). In each of the files is a dictionary mapping InstanceIDs to the annotation per frame. By giving each annotated instance in the scene a unique ID, which is constant over time, annotations can be combined between tasks and frames easily, thus allowing for flexible use and extension of the annotations. An exception form pixel-based annotations, i.e. semantic segmentation, instance segmentation, and depth. These are stored in image files, with the InstanceIDs in the instance segmentation matching the InstanceIDs in the JSON files.

A full specification of the annotation format and the file structure is provided together with the dataset.

3.2 Meta information

High-quality labeling and meta information are key enablers for DNN training and testing as well as for in-depth data and coverage analyses. Moreover, they are a vital prerequisite for producing data-related evidences for a safety argumentation. In *KI Absicherung*, an ontology [16] has been developed systematically to structure the input space, to identify performance relevant factors and to define an operational design domain. The meta information which the dataset provides has been deduced from the ontology.

As shown in Fig. 3, in SynPeDS, we use a meta information concept based on five key IDs to uniquely describe a pedestrian and its context: ImageID, InstanceID, AssetID, MoCapID, and SensorID. Each pedestrian in an image has a unique InstanceID, which is linked to an AssetID from the asset catalogue describing the phenological appearance of an asset. The MoCapID in combination with a time stamp describe the motion and pose of an asset in the image based on a recorded motion sequence. The SensorID identifies the ego-sensor and its characteristics as well as applicable coordinate transformations.

The meta information contains more than 50 additional entries to further characterize each pedestrian and his/her context. These have been either aggregated directly, such as the pedestrian position or orientation in terms of hip or head direction, or are the result of a post-processing, such as the RGB contrast of the pedestrian to the background. Other meta information regarding visibility include the total degree of occlusion, the number of visible and occluded joints, or the type of occluding objects. We have found that a simplified, but useful dimension to represent pedestrian pose is the difference

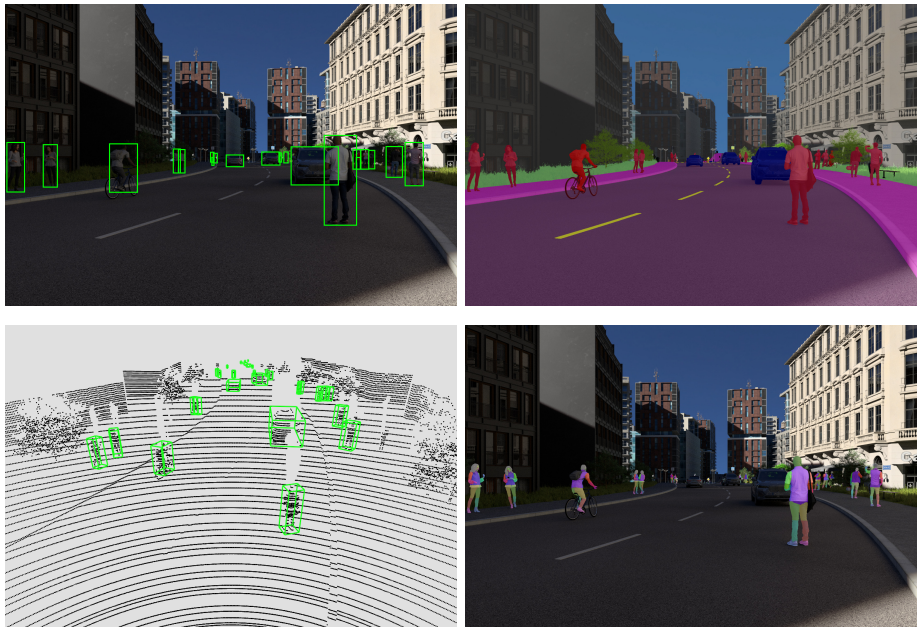


Figure 2: SynPeDS example frame and ground truth information from BIT Technology Solutions GmbH: camera sensor image with 2D bounding boxes, semantic segmentation, LiDAR point cloud with 3D bounding boxes, bodypart segmentation. ©BIT Technology Solutions GmbH

of height between the shoulder and toe joints. The direction and elevation of the sun as well as weather and road wetness conditions further allow to study the influence of lighting. We additionally provide the pedestrian position in the car coordinate system and in a semantic map, which allows to assess the pedestrian relevance for a potential automated breaking function.

Fig. 4 shows some exemplary images for four pedestrian meta information dimensions and their possible value ranges.

4 DATASET GENERATION

The dataset has been produced in an interative manner in order to incorporate learnings and refined user requirements in later data deliveries. During the creation of the dataset, the tool capabilities have been continuously extended and a large number of new assets was introduced.

4.1 Design principles

After a starting phase, assets have been selected according to the ontology used in the project, including both public libraries as well as self-created assets – the latter to satisfy specific requirements, e.g. regarding pose animation or specific asset attributes. Some assets only appear in the test data in order to enable experiments on detection of assets unknown to the neural network. In the progress of the project, scenes have been extended iteratively and new tool-features were added, thereby addressing user requirements on complexity and variance. This also included composition of the scenes, e.g. regarding pedestrian or vehicle distribution and variation. Frame-to-frame variations were introduced early in the project in order to further boost variance. These variations concerned multiple features within the scene design which are composed according to the user needs and include e.g. the variation of the clothing of pedestrians, their orientation, or sky and sun properties.

The iteratively produced data tranches correspondingly have different characteristics, as listed in Table 1. In order to raise synergies between toolchains, the glTF file format [13] has been selected as common format for the assets produced. The amount of person assets has been continuously increased from 13 to 89 (52, if different clothing is not considered). Despite this continuous improvement,

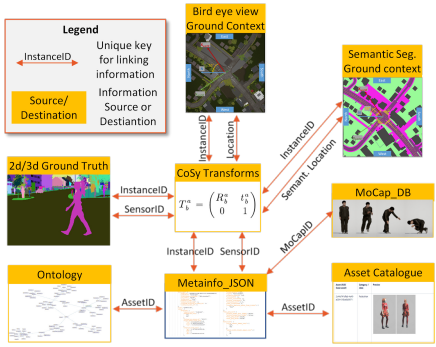


Figure 3: Linking of data sources and destinations to obtain meta information.

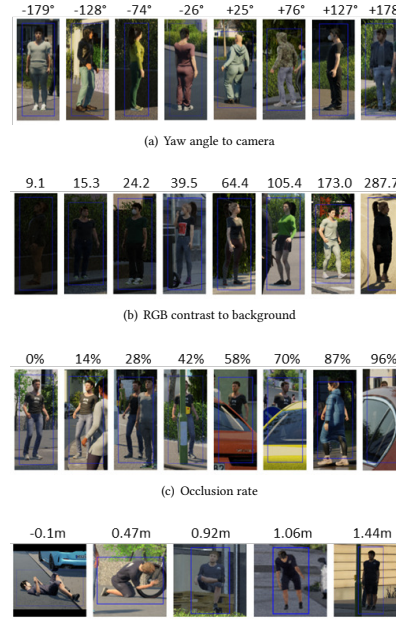
Table 1: Features added per data tranche and data pipeline (PBR: physical-based rendering pipeline, RT: real-time engine based pipeline). Tranches 1, 2 and 7 (PBR) are not part of the published dataset.

Tranche	New Features	PBR	RT
1, 2, 3	Preperation for large-scale data production	×	×
4	Frame-to-frame variations	×	×
	Meta information on AssetIDs	×	×
	Bodypart segmentation	×	
	Procedural sun model		×
5	Sensor noise as post-processing	×	
	Procedural clouds model		×
	Ground truth for pose estimation		×
	Meta information on occlusion		×
6	Environmental effects: wetness and sun glare		×
	Out-of-distribution assets		×
	Variatios of camera sensor parameters		×
7	Camera and LiDAR sensor models using PBR with OSPRay and different LiDAR sensor parameters	×	
	Meta information on AnimationID		×
	Environmental effects: fog, vignetting		×
8	Night scenes with artificial light		×
9	Specific user requests for contrast or material		×

there is a common specification for sensor data, annotations, and meta information to achieve a consistent dataset for all users.

4.2 Data generation with a real-time engine

As it can be witnessed in modern video games, the underlying engines enable the creation of realistic and highly complex virtual worlds – a requirement for datasets for automated driving applications. State-of-the-art game engines offer high quality lighting, powerful material systems, animation tools, and flexible application programming interfaces (APIs). Thus, as another toolchain for data generation we chose UE4. Because real-time frame rates as in computer games are not required we select high-quality settings



(a) Yaw angle to camera

(b) RGB contrast to background

(c) Occlusion rate

(d) Shoulder-toe height difference

Figure 4: Example images for four pedestrian meta information dimensions and their possible value ranges.

for rendering, including (starting with tranche 6) ray tracing for global illumination, shadows, reflections, and translucency.

Based on an initial scene generation, scenarios are automatically generated by predefined or randomized parameters. We also implemented an interface to load scenario definitions with JSON files that allow for setting parameters on a frame level, see Section 5.3 for an application.

Natural lighting is one of the key elements in virtual environments. For this, UE4 already features a flexible, procedural sun and sky model that allows real-world settings for location, time and date, and a realistic simulation of atmosphere. For increased realism and more variety in light scenarios, we created a procedural volumetric cloud model based on 3D noise textures.

Furthermore, additional environmental and surface effects are used: a simple fog model and shader-based wet materials. This results in not only different surface appearances but also introduces effects such as reflections of light, pedestrians, or objects and puddle formation on the road. With the last two data tranches, there are also individual sequences in a night environment with artificial light sources.

As stated in Section 1, a large amount of typical and untypical pedestrian motions were recorded using inertial motion capture. The motion capture data is applied to the character assets using Unreal Engines skeletal meshes. This allows usage for animations such as walking or running motions on predefined trajectories with avoiding obstacles as well as automated assignment of special or challenging poses.

4.3 Data generation with physically-based rendering

It is currently unclear what level of fidelity of synthetic image generation is needed to provide adequate data for training and validation of pedestrian detection systems. To rule out those uncertainties we decided that one toolchain should strive for an accurate simulation of light transport within the virtual scene. Therefore we selected the open source software Blender [8] as rendering core, which supports accurate soft and hard shadows, correct reflections and transparency, indirect illumination, and complex materials and light sources. Such a ray-tracing based PBR system also provides the base for accurate simulation of sensors (here LiDAR and camera, where images are post-processed by the sensor module from [14]).

The generation of ground truth and meta information is largely straightforward, it can be computed and derived from the internal hierarchical scene structure or renderer internals. However, some details are worth discussing. Many different objects can be "visible" in a single pixel, for example due to antialiasing, depth of field effects and motion blur. Therefore we disable those effects when computing ground truth segmentation or 2D bounding boxes. Constructing most-tight oriented 3D bounding boxes is a non-trivial optimization problem and is also not wanted, because that may lead to discontinuous jumps in orientation between frames. Instead, we choose the orientation beforehand and then compute the axis-aligned bounding box in a accordingly rotated, local coordinate system. For rigid objects the orientation is given by its placement into the scene, for pedestrians the bounding box is additionally tilted to be perpendicular to the ground.

As outlook, further work on the PBR data generation pipeline is pursued in the project *KI Data Tooling*: For the PBR render engine we are currently switching to the open source, scalable ray tracing engine Intel OSPRay [28] to improve rendering performance and to add volumetric effects to the scenes to handle different weather conditions. Furthermore, the use of the application OSPRay Studio [25] allows us fine control of the generation of the ground truth and the meta information, and an easy integration of advanced sensor modules. Based on OSPRay's thinlens camera model with support for motion blur from a global or rolling shutter as well as a panoramic 360 degree camera the integration of the following new components show promising early results:

(1) A camera module from Robert Bosch GmbH, which uses pre-simulated, up to four-dimensional point-spread functions (PSF) to compute response of the optical system based on lens/image position, depth, and color/wavelength.
(2) A LiDAR module from Valeo Schalter und Sensoren GmbH, which produces realistic and idealized point clouds (including associated meta information) based on the properties of the IR receivers of the simulated LiDAR device; the interaction of moving objects with the scanning nature of the LiDAR are correctly captured by the rolling shutter effect.

Table 2: Cross-domain generalization performance results of DeeplabV3+ models trained on our *SynPeDS*, synthetic GTAV, SYNS and datasets, evaluated with the mIoU on real-world datasets and on synthetic datasets. Higher values indicate better generalization performance, bold are highest and underlined second-highest. Performance results for CS are added as reference.

Evaluated Dataset mIoU [%]		Trained model GTAV	SYNS	SynPeDS	CS
Synthetic	GTAV	-	28.71	**40.72**	38.74
	SYNS	48.83	-	**60.00**	73.04
	SynPeDS	**50.41**	50.25	-	69.30
Real-World	A2D2	34.69	22.67	**45.76**	51.35
	BDD100K	42.56	25.00	**44.73**	59.18
	CS	39.07	**59.33**	55.94	81.27
	IDD	**45.17**	29.59	43.87	61.72
	MV	48.30	35.03	**55.15**	64.69

5 QUALITY OF THE DATASET

5.1 Comparison to other datasets

To demonstrate the quality of our synthetic dataset we conducted several cross-domain performance analyses with other real-world automotive and synthetic datasets. This cross-domain performance analysis is also commonly referred to as generalization distance. Our experiments consider the task of semantic segmentation with a DeeplabV3+ [2] segmentation model, utilizing a ResNet101 [15] for low-level feature extraction, pre-trained on the ImageNet [4] dataset. We trained a DeeplabV3+ model on our SynPeDS dataset, i.e., tranches 1 to 7, as well as for the GTA V [23] and Synscapes (SYNS) [30] dataset. Next, we evaluated the segmentation performance on the same synthetic datasets and further on real-world datasets A2D2 [11], BDD100K [31], Cityscapes (CS) [3], India Driving Dataset (IDD) [27] and Mapillary Vistas (MV) [22]. For these real-world and synthetic datasets we have to train the segmentation network on a subset of 11 labels per dataset to ensure consistency of classes across all datasets. These labels are road and sidewalk which incorporate the road-markings and the curb respectively. Further, the building, sky, car and truck classes which are consistent across these datasets. Pole, traffic light and traffic sign classes are mapped from similar sub-classes in the datasets, such as utility pole in MV. The vegetation class consists of the CS subclasses 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, i.e. pedestrians, riders etc.

Measuring the performance of the segmentation model is done by calculating the mean intersection over union (mIoU) over all considered classes on the image predictions. The mIoU cross-domain generalization results over all classes are listed in Table 2.

Overall the SynPeDS performance reaches highest mIoU values on the real-world datasets A2D2, BDD100K and MV as well as

Table 3: Cross-domain generalization performance of class person results of DeeplabV3+ models trained on our SynPeDS dataset, synthetic GTAV and Synscapes (SYNS) datasets evaluated on real-world datasets. Higher values indicate better generalization performance. Bold marked are highest and underlined second-highest performance values. Performance results for CS are added as reference.

Evaluated Dataset Person mIoU [%]		Trained model GTAV	SYNS	SynPeDS	CS
Synthetic	GTAV	-	4.29	**16.91**	22.37
	SYNS	66.57	-	**72.33**	78.70
	SynPeDS	**76.15**	69.24	-	71.83
Real-World	A2D2	**47.26**	15.04	42.88	45.75
	BDD100K	**47.04**	21.47	35.98	46.68
	CS	53.71	**73.41**	73.31	78.51
	IDD	**76.67**	43.52	58.65	71.77
	MV	**60.69**	20.60	53.03	64.95

on the synthetic datasets GTAV and Synscapes. Additionally, the model trained on SynPeDS reaches second-highest mIoU values on CS and IDD datasets with only a few percent distance to the top performing model on these datasets. Compared to a CS trained model the remaining generalization distance on the A2D2 and MV datasets is as low as 9%. The GTAV trained model performs only marginally better on the IDD dataset. As expected, the performance on the CS dataset is highest with the Synscapes trained model, as the Synscapes images are targeted to closely resemble the CS images.

As our datasets targets the safety-related aspects of pedestrian detection the person class generalization performance is of increased interest. Results of our cross-domain performance analysis on the person class can be seen in Table 3.

Again, performance on real-world datasets is second-highest for our synthetic dataset and highest for all cross-evaluated synthetic datasets. The models trained on the synthetic datasets perform on par and better than a model trained on the CS dataset. We attribute the high performance of the GTAV dataset on the overall higher availability of person assets compared to our dataset even though the fidelity of the assets is higher in our synthetic dataset.

To emphasise the importance of this pedestrian asset diversity on generalization performance we conducted further cross-domain pedestrian class performance analysis experiments on the CS dataset. Training the segmentation network with a combination of tranche 1 and tranche 2 data and adding data tranches to the training set until we reach the overall SynPeDS dataset. Before new data is added to the training, the cross-domain person class performance is evaluated. The results of this experiment are depicted in Fig. 5.

We found with increasing unique person assets in the dataset tranches, the cross-domain performance on the CS dataset continuously increases from 40.78% and 13 unique person assets with tranches 1 and 2 combined to 73.31% and 89 unique person assets with the overall SynPeDS dataset. The remaining difference to the

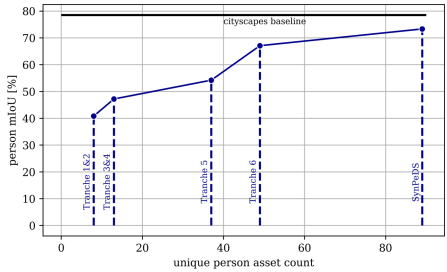


Figure 5: Cross-domain pedestrian class performance on the Cityscapes dataset in relation to the number of unique person training assets.

DeeplabV3+ CS trained baseline which reaches 78.51% mIoU on the person class, are only 5.21%.

Summarizing the semantic-segmentation cross-domain generalization performance, we deliver a dataset that yields good real-world as well as very good synthetic domain segmentation performance. The dataset suits for real-world and synthetic automotive detection or segmentation experiments as well as for pre-training before fine-tuning on the target automotive domain. Note that further studies and measures to increase transferability and decrease the domain gap to real-world data are outside the scope of this paper and are addressed in [14].

5.2 Pose analysis

Since complete coverage of the pose space is important, we conducted a study to evaluate the coverage of the pose variance. To achieve this, we clustered the annotated poses using k-means clustering and analysed the frequency of pose assignments to the cluster centers.

Our analysis shows large variance in the poses and a comparison to the results of the same analysis on the public real world dataset PedX [18] with human pose annotations confirms the superior variance in overall poses of our dataset (see Fig. 6). While the figure indicates that our dataset has lower variance in upper body articulation, specifically arm movement, than PedX, there are many poses such as lying, crawling and sitting in our dataset which our analysis could not find in PedX. We consider the existence of such poses that occur rarely in real drives as specifically relevant for safety analysis.

Studying the performance influence of poses in a real-world dataset comes with difficulties, since other effects such as the contrast of the pedestrian, the relative size within the image or the degree of occlusion also have large effects on the detection performance. We therefore designed a subset of SynPeDS to specifically support the isolated evaluation of the effect of different pedestrian poses. This pose subset consists of a training set (≈4000 images) and two test sets (≈5300 images). To isolate the influence of each pose, parameters were defined and varied around specific values. The

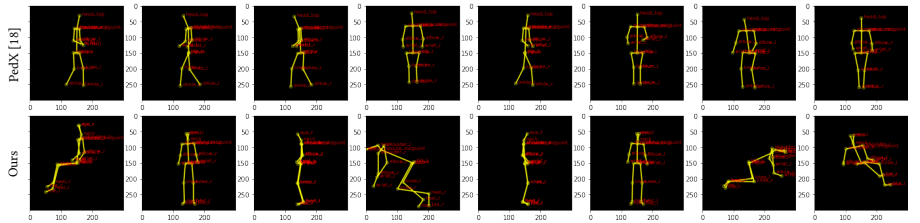


Figure 6: Pose clusters for our synthetic dataset (bottom) show more variance then real poses from PedX (top).

subset contains all combinations of the parameters in individual images. Following parameters were varied:

- pedestrian asset or clothing
- pose of the asset
- rotation of the asset to the ego camera
- position of the asset in the base context
- sun position with regards to the relative azimuth and elevation
- position of the ego camera within the base context

5.3 Usage for systematic testing of AI

For advanced driver assistance systems, the Euro NCAP specifications are an important reference. These scenarios are deemed representative and reflect actual accident statistics. We focus especially on the specification for autonomous emergency braking for vulnerable road users (AEB VRU test protocol). Even though these scenarios are defined for and assessed on a vehicle system level, they also rely on a well-functioning perception. On the basis of this test protocol, several safety-related Euro NCAP-like scenarios were defined and simulated as part of this dataset. These include the following:

- Pedestrian crosses the street between two cars
- Pedestrian crosses a four-lane road occluded by a car
- Pedestrian and cyclist cross the street
- Pedestrian crosses the street when a car turns left
- Pedestrian crosses the street when a car turns right

We focus on the pedestrian perception based on single images, so we simulate single relevant pedestrians in each image, and systematically vary selected discrete dimensions which affect the pedestrian detection performance. The structural concept of the first Euro NCAP-like scenario and the variation of selected discrete dimensions is illustrated in Fig. 7.

To be precise, we use the following 13 dimensions for test data variation in the first Euro NCAP-like scenario: the position of the ego car with the camera, the position, pose, and rotation of the pedestrian, the pedestrian asset, the type and color of the first parked car, the type, color, and position of the second parked car, the illumination, and the direction and elevation of the sun. These dimensions were each discretized into a set of discrete states using a Zwicky box approach [12, 16]. This structure was further used to create a 3-wise combinatorial test plan, resulting in more than 16k

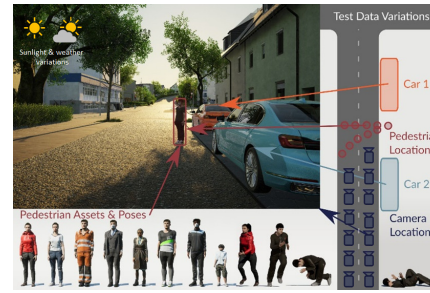


Figure 7: Euro NCAP-like scenario definition and test data variations.

independent frames. The process of systematically generating appropriate combinations of variations of the descriptive dimensions and optimizing the number of images to produce, was automated using the software tool PICT [21] and a self-programmed wrapper in Python.

In *KI Absicherung*, extensive experiments have been carried out with the 16k test frames of the Euro NCAP-like scenarios. To this end, mainly an SSD [20] has been used, which constitutes a traditional anchor-based network for object detection with known limitations. Important insight has been generated to better understand the performance of pedestrian detectors and their optimization in safety-related scenarios. In particular, w.r.t. the presented examples in Figure 4, it could be verified that the contrast between the pedestrian and its background influences the detection performance, and that it is very crucial which part of the pedestrian is visible, either due to occlusion, or special poses or orientation. Here, the availability of meta information, as described in Section 3.2, has proven very valuable for a detailed analysis. During the project, we were able to use this knowledge to iteratively improve the performance of the SSD. We conclude that using Euro NCAP-like scenarios in combination with rich meta information can be used, on the one hand, to carefully design a training dataset so that a pedestrian detector performs satisfactory in rare critical events. On the other hand, it

can be used for systematic testing and can thereby contribute to a safety assessment of an automated driving component.

6 CONCLUSION AND FUTURE WORK

We have introduced the SynPeDS dataset which offers a wide range of ground truth and comprehensive meta information that supports safety analysis in pedestrian detection tasks. Among others, the meta information in particular supports to systematically examine a DNN's performance under different, controlled lighting conditions, for different pedestrian attributes, like poses or body size, and occlusion degrees and occlusion situations. The generalization performance of the dataset is similar or better to other synthetic datasets (Sec. 5.1). Due to the construction from captured motions, the poses in the dataset are more diverse than in real-world datasets (Sec. 5.2). With the pose analysis example of Sec. 5.3 we demonstrated how it can be used for safety analysis. The dataset is intended to serve as a basis for future research on techniques that contribute to the safety assessment of pedestrian detection in automated driving.

From the data generation experience we note that the real-time and PBR pipelines have different focus and strengths. Real-time game engines support many visual effects and allow for fast data generation, but require carefully optimized scene assets. PBR enables highly accurate sensor simulations and can easily handle a huge number and size of assets, but need more computation time. Since game engines more and more apply ray tracing and physically-based rendering techniques and since PBR production renderers are increasingly ported to GPU to take advantage of hardware-accelerated ray tracing the gap in feature set between the pipelines may shrink in the future.

ACKNOWLEDGMENTS

The research leading to these results was funded by the German Federal Ministry for Economic Affairs and Climate Action 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.

Further thanks go to Christopher Hauck, Christian Zilliken (both Accenture Song Content Germany GmbH), Tom Stone (BMW Group), Michael Schuldes (Institute for Automotive Engineering, RWTH Aachen), Oliver Grau (Intel Corporation), Christoph Gladisch, Christian Heinzemann, Martin Herrmann, Falko Matern, Ulrich Seger (all Robert Bosch GmbH), and Christian Witt (Valeo Schalter und Sensoren GmbH).

REFERENCES

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 833–851.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] 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*.

[5] Jean-Emmanuel Deschaud. 2021. KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator. *arXiv preprint arXiv:2109.00892* (2021).

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16. http://carla.org/

[7] Epic Games. 2022. Unreal Engine 4. https://www.unrealengine.com Accessed: 2022-04-23.

[8] Blender Foundation. 2022. blender.org - Home of the Blender project - Free and Open 3D Creation Software. https://www.blender.org/. Accessed: 2022-04-23.

[9] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. 2016. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4340–4349.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* (2013).

[11] 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. 2020. A2D2: Audi Autonomous Driving Dataset. arXiv:2004.06320 [cs.CV]

[12] Christoph Gladisch, Christian Heinzemann, Martin Herrmann, and Matthias Woehrle. 2020. Leveraging Combinatorial Testing for Safety-Critical Computer Vision Datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[13] The Khronos® 3D Formats Working Group. 2021. glTF™ 2.0 Specification. https://www.khronos.org/registry/glTF/specs/2.0/glTF-2.0.html Accessed: 2022-04-24.

[14] 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

[15] 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

[16] Martin Herrmann, Christian Witt, Laureen Lake, Stefani Guneshka, Christian Heinzemann, Frank Bonarens, Patrick Feifel, and Simon Funke. 2022. Using Ontologies for Dataset Engineering in Automotive AI Applications. In *Proceedings of the 2022 Conference & Exhibition on Design, Automation & Test in Europe (DATE '22)*. European Design and Automation Association, 526––531.

[17] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. 2019. Precise synthetic image and lidar (PreSIL) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2522–2529.

[18] Wonhui Kim, Manikandasriram Srinivasan Ramanagopal, Charles Barto, Ming-Yuan Yu, Karl Rosaen, Nick Goumas, Ram Vasudevan, and Matthew Johnson-Roberson. 2019. Pedx: Benchmark dataset for metric 3-d pose estimation of pedestrians in complex urban intersections. *IEEE Robotics and Automation Letters* 4, 2 (2019), 1940–1947.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[21] Microsoft. 2022. GitHub - microsoft/pict: Pairwise Independent Combinatorial Tool. https://github.com/microsoft/pict Accessed: 2022-07-06.

[22] 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

[23] 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.

[24] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3234–3243.

[25] Isha Sharma, Dave DeMarle, Alok Hota, Bruce Cherniak, and Johannes Günther. 2021. OSPRay Studio: Enabling Multi-Workflow Visualizations with OSPRay. In *VisGap - The Gap between Visualization Research and Visualization Software*, Christina Gillmann, Michael Krone, Guido Reina, and Thomas Wischgoll (Eds.). The Eurographics Association. https://doi.org/10.2312/visgap.20211086

[26] Unity Technologies. 2022. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. https://unity.com/ Accessed: 2022-04-24.

[27] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. 2019. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1743–1751.

[28] Ingo Wald, Greg P. Johnson, Jefferson Amstutz, Carson Brownlee, Aaron Knoll, Jim Jeffers, Johannes Günther, and Paul Navrátil. 2017. OSPRay - A CPU Ray Tracing Framework for Scientific Visualization. *IEEE Transactions on Visualization and Computer Graphics* (2017). https://www.ospray.org

[29] Garrett Wilson and Diane J Cook. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 5 (2020), 1–46.

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

[31] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2636–2645.

Bibliography

[ACP21] Rob Ashmore, Radu Calinescu, and Colin Paterson. "Assuring the machine learning lifecycle: desiderata, methods, and challenges". In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–39. DOI: 10.1145/3453444.

[ADG+16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent". In: *Advances in neural information processing systems* 29 (2016). DOI: 10.5555/3157382.3157543.

[AGG+21] Stephanie Abrecht, Lydia Gauerhof, Christoph Gladisch, Konrad Groh, Christian Heinzemann, and Matthias Woehrle. "Testing deep learning-based visual perception for automated driving". In: *ACM Transactions on Cyber-Physical Systems (TCPS)* 5.4 (2021), pp. 1–28. DOI: 10.1145/3450356.

[BB95] Wilhelm Burger and Matthew J. Barth. "Virtual reality for enhanced computer vision". In: *Virtual Prototyping: Virtual environments and the product design process*. Boston, MA: Springer US, 1995, pp. 247–257. ISBN: 978-0-387-34904-6. DOI: 10.1007/978-0-387-34904-6_19.

[BBL+19] Jan-Aike Bolte, Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. "Towards corner case detection for autonomous driving". In: *2019 IEEE Intelligent vehicles symposium (IV)*. IEEE. 2019, pp. 438–445. DOI: 10.1109/IVS.2019.8813817.

[BGH17] Simon Burton, Lydia Gauerhof, and Christian Heinzemann. "Making the case for safety of machine learning in highly automated driving". In: *Computer Safety, Reliability, and Security*. Ed. by Stefano Tonetta, Erwin Schoitsch, and Friedemann Bitsch. Cham: Springer International Publishing, 2017, pp. 5–16. ISBN: 978-3-319-66284-8. DOI: 10.1007/978-3-319-66284-8_1.

[Bis95] C. M. Bishop. “Training with noise is equivalent to tikhonov regularization”. In: *Neural Computation* 7.1 (1995), pp. 108–116. DOI: 10.1162/neco.1995.7.1.108.

[BKF+19] Markus Braun, Sebastian Krebs, Fabian B. Flohr, and Dariu M. Gavrila. “Eurocity persons: a novel benchmark for person detection in traffic scenes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2897684.

[BKS+21] Simon Burton, Iwo Kurzidem, Adrian Schwaiger, Philipp Schleiss, Michael Unterreiner, Torben Graeber, and Philipp Becker. “Safety assurance of machine learning for chassis control functions”. In: *International Conference on Computer Safety, Reliability, and Security*. Springer. 2021, pp. 149–162. DOI: 10.1007/978-3-030-83903-1_10.

[BRP+15] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. “Sliced and radon wasserstein barycenters of measures”. In: *Journal of Mathematical Imaging and Vision* 51.1 (2015), pp. 22–45. DOI: 10.1007/s10851-014-0506-3.

[BSA+18] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. “Demystifying MMD GANs”. In: *International Conference on Learning Representations*. 2018. DOI: 10.48550/arXiv.1801.01401.

[BSS+21] Johannes Bernhard, Thomas Schulik, Mark Schutera, and Eric Sax. “Adaptive test case selection for dnn-based perception functions”. In: *2021 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE. 2021, pp. 1–7. DOI: 10.1109/ISSE51541.2021.9582499.

[BWL20] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020). DOI: 10.48550/arXiv.2004.10934.

[CBH+02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. "Smote: synthetic minority oversampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357. DOI: 10.5555/1622407.1622416.

[CCN+16] Gabriel B. Paranhos da Costa, Welinton A. Contato, Tiago S. Nazaré, João do E. S. Batista Neto, and Moacir Ponti. "An empirical study on the effects of different types of noise in image classification tasks". In: *CoRR* abs/1609.02781 (2016). DOI: 10.48550/arXiv.1609.02781. arXiv: 1609.02781.

[CKL21] Chih-Hong Cheng, Alois Knoll, and Hsuan-Cheng Liao. "Safety metrics for semantic segmentation in autonomous driving". In: *arXiv preprint arXiv:2105.10142* (2021). DOI: 10.1109/AITEST52744.2021.00021.

[CLP+13] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. "What is a good evaluation measure for semantic segmentation?." In: *Bmvc*. Vol. 27. 2013. 2013, pp. 10–5244. DOI: 10.5244/C.27.32.

[CNH+18] Chih-Hong Cheng, Georg Nührenberg, Chung-Hao Huang, Harald Ruess, and Hirotoshi Yasuoka. "Towards Dependability Metrics for Neural Networks". In: *Proceedings of the ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*. Beijing, China, Oct. 2018, pp. 43–46. DOI: 10.5555/3343872.3343877.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "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, June 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[CPK+17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[CSV+18] Alexandra Carlson, Katherine A. Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. "Modeling camera effects to improve visual learning from synthetic data". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. Sept. 2018. DOI: 10.1007/978-3-030-11009-3_31.

[CSV+19] Alexandra Carlson, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. "Sensor transfer: learning optimal sensor effect image augmentation for sim-to-real domain adaptation". In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2431–2438. DOI: http://dx.doi.org/10.1109/LRA.2019.2896470.

[CV19] Zhaowei Cai and Nuno Vasconcelos. "Cascade r-cnn: high quality object detection and instance segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). DOI: https://doi.org/10.1109/TPAMI.2019.2956516.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Cham: Springer International Publishing, 2018, pp. 833–851. ISBN: 978-3-030-01234-2. DOI: 10.1007/978-3-030-01234-2_49.

[DBK+21] Alexey Dosovitskiy et al. "An image is worth 16x16 words: transformers for image recognition at scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. DOI: 10.48550/arXiv.2010.11929. URL: https://openreview.net/forum?id=YicbFdNTTy.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[Del+34] Boris Delaunay et al. “Sur la sphere vide”. In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800 (1934), pp. 1–2.

[DG18] Werner Damm and Roland Galbas. “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. 2018, pp. 39–46. DOI: 10.1145/3194085.3194086.

[DGZ17] Qi Dong, Shaogang Gong, and Xiatian Zhu. “Class rectification hard mining for imbalanced deep learning”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1851–1860. DOI: https://doi.org/10.1109/ICCV.2017.205.

[DKF20] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. “Meta-sim2: learning to generate synthetic datasets”. In: *ECCV*. virtual conference, 2020. DOI: 10.1007/978-3-030-58520-4_42.

[DRC+17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the Conference on Robot Learning CORL*. Mountain View, CA, USA, Nov. 2017, pp. 1–16. DOI: 10.48550/arXiv.1711.03938.

[DWS+09] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. “Pedestrian detection: a benchmark”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 304–311. DOI: 10.1109/CVPR.2009.5206631.

[DWS+11] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. “Pedestrian detection: an evaluation of the state of the art”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2011), pp. 743–761. DOI: 10.1109/TPAMI.2011.155.

[ELV07] Andreas Ess, Bastian Leibe, and Luc Van Gool. “Depth and appearance for mobile scene analysis”. In: *2007 IEEE 11th international conference on computer vision*. IEEE. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4409092.

[EM94] Herbert Edelsbrunner and Ernst P Mücke. "Three-dimensional alpha shapes". In: *ACM Transactions on Graphics (TOG)* 13.1 (1994), pp. 43–72. DOI: 10.1145/174462.156635.

[FMW+18] E. Fernandez-Moral, R. Martins, D. Wolf, and P. Rives. "A new metric for evaluating semantic segmentation: leveraging global and contour accuracy". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1051–1056. DOI: 10.1109/IVS.2018.8500497.

[Fou] Blender Foundation. *Blender*. https://www.blender.org/.

[FS97] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139. DOI: 10.1007/3-540-59119-2_166.

[Gam] Epic Games. *Unreal engine 4*. https://www.unrealengine.com.

[GDD+14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81.

[GHA21] Sujan Gannamaneni, Sebastian Houben, and Maram Akila. "Semantic concept testing in autonomous driving by extraction of object-level annotations from carla". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1006–1014. DOI: 10.1109/ICCVW54120.2021.00117.

[GHP+20] Lydia Gauerhof, Richard David Hawkins, Chiara Picardi, Colin Paterson, Yuki Hagiwara, and Ibrahim Habli. "Assuring the safety of machine learning for pedestrian detection at crossings". In: *SAFECOMP 2020 (39th International Conference on Computer Safety, Reliability and Security)*. York. 2020. DOI: 10.1007/978-3-030-54549-9_13.

[GHS22] Oliver Grau, Korbinian Hagn, and Qutub Syed Sha. "A variational deep synthesis approach for perception validation". In: *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*. Ed. by Tim Fingscheidt, Hanno Gottschalk, and Sebastian Houben. Cham: Springer International Publishing, 2022, pp. 359–381. ISBN: 978-3-031-01233-4. DOI: 10.1007/978-3-031-01233-4_13.

[Gir15] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[GKM+20] Jakob Geyer et al. *A2d2: audi autonomous driving dataset*. 2020. DOI: 10.48550/arXiv.2004.06320. arXiv: 2004.06320 [cs.CV].

[GL15] Yaroslav Ganin and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189. DOI: 10.5555/3045118.3045244.

[GMB18] Lydia Gauerhof, Peter Munk, and Simon Burton. "Structuring validation targets of a machine learning function applied to automated driving". In: *Computer Safety, Reliability, and Security*. Ed. by Barbara Gallina, Amund Skavhaug, and Friedemann Bitsch. Cham: Springer International Publishing, 2018, pp. 45–58. ISBN: 978-3-319-99130-6. DOI: 10.1007/978-3-319-99130-6_4.

[HCG+18] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. "The apolloscape dataset for autonomous driving". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 954–960. DOI: 10.1109/CVPRW.2018.00141.

[HG21] Korbinian Hagn and Oliver Grau. "Improved sensor model for realistic synthetic data generation". In: *Computer Science in Cars Symposium*. CSCS '21. Ingolstadt, Germany: Association

for Computing Machinery, 2021. ISBN: 9781450391399. DOI: 10.1145/3488904.3493383.

[HG22a] Korbinian Hagn and Oliver Grau. "Increasing pedestrian detection performance through weighting of detection impairing factors". In: *Proceedings of the 6th ACM Computer Science in Cars Symposium*. CSCS '22. Ingolstadt, Germany: Association for Computing Machinery, 2022. ISBN: 9781450397865. DOI: 10.1145/3568160.3570225.

[HG22b] Korbinian Hagn and Oliver Grau. "Optimized data synthesis for dnn training and validation by sensor artifact simulation". In: *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*. Ed. by Tim Fingscheidt, Hanno Gottschalk, and Sebastian Houben. Cham: Springer International Publishing, 2022, pp. 127–147. ISBN: 978-3-031-01233-4. DOI: 10.1007/978-3-031-01233-4_4.

[HG23] Korbinian Hagn and Oliver Grau. "Validation of pedestrian detectors by classification of visual detection impairing factors". In: *Computer Vision – ECCV 2022 Workshops*. Ed. by Leonid Karlinsky, Tomer Michaeli, and Ko Nishino. Cham: Springer Nature Switzerland, 2023, pp. 476–491. ISBN: 978-3-031-25072-9. DOI: 10.1007/978-3-031-25072-9_33.

[HGD+17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969. DOI: 10.1109/ICCV.2017.322.

[HMW+18] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. "Using trusted data to train deep networks on labels corrupted by severe noise". In: *Advances in neural information processing systems* 31 (2018). DOI: 10.5555/3327546.3327707.

[HRU+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "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. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6629–6640. ISBN: 9781510860964. DOI: 10.5555/3295222.3295408.

[HSR+20] Maximilian Henne, Adrian Schwaiger, Karsten Roscher, and Gereon Weiss. "Benchmarking uncertainty estimation methods for deep learning with safety-related metrics." In: *SafeAI@ AAAI*. 2020, pp. 83–90. DOI: 10.24406/publica-fhg-407174.

[HTP+18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. "CyCADA: cycle-consistent adversarial domain adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1989–1998. DOI: 10.48550/arXiv.1711.03213.

[HZC+17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: efficient convolutional neural networks for mobile vision applications". In: *CoRR* abs/1704.04861 (2017). DOI: 10.48550/arXiv.1704.04861.

[HZR+16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[Int] Intel. *Intel ospray*. https://www.ospray.org/.

[ISO18] ISO. *ISO 26262: Road vehicles - Functional Safety*. International Standards Organisation (ISO). Geneva, Dec. 2018.

[ISO21a] ISO. *ISO/AWI PAS 8800: Road vehicles - Safety and artificial intelligence*. International Standards Organisation (ISO). Geneva, Sept. 2021.

[ISO21b] ISO. *ISO/DIS 21448: Road vehicles - Safety of the intended functionality*. International Standards Organisation (ISO). Geneva, Apr. 2021.

[JMZ+15] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. "Self-paced curriculum learning". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015. DOI: 10.5555/2886521.2886696.

[JWK+18] Philipp Junietz, Walther Wachenfeld, Kamil Klonecki, and Hermann Winner. "Evaluation of different approaches to address safety validation of automated driving". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 491–496. DOI: 10.1109/ITSC.2018.8569959.

[JZL+18] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. "Mentornet: learning data-driven curriculum for very deep neural networks on corrupted labels". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2304–2313. DOI: 10.48550/arXiv.1712.05055.

[KHB+17] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. "Cost-sensitive learning of deep feature representations from imbalanced data". In: *IEEE transactions on neural networks and learning systems* 29.8 (2017), pp. 3573–3587. DOI: 10.1109/TNNLS.2017.2732482.

[KM53] Herman Kahn and Andy W Marshall. "Methods of reducing sample size in monte carlo computations". In: *Journal of the Operations Research Society of America* 1.5 (1953), pp. 263–278. DOI: 10.1287/opre.1.5.263.

[KP16] Nidhi Kalra and Susan M Paddock. "Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability?" In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 182–193. DOI: 10.1016/j.tra.2016.09.010.

[KPK10] M Kumar, Benjamin Packer, and Daphne Koller. "Self-paced learning for latent variable models". In: *Advances in neural information processing systems* 23 (2010). DOI: 10.5555/2997189.2997322.

[KRF+16] Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, Kristen Grauman, et al. "Crowdsourcing in computer vision". In: *Foundations and Trends® in computer graphics and Vision* 10.3 (2016), pp. 177–243. DOI: 10.1561/0600000071.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: single shot multibox detector". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 21–37. ISBN: 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_2.

[LCW+15] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. "Learning transferable features with deep adaptation networks". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 97–105. DOI: 10.5555/3045118.3045130.

[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 international conference on computer vision*. 2017, pp. 2980–2988. DOI: 10.1109/TPAMI.2018.2858826.

[LGH+21a] Maria Lyssenko, Christoph Gladisch, Christian Heinzemann, Matthias Woehrle, and Rudolph Triebel. "From evaluation to verification: towards task-oriented relevance metrics for pedestrian detection in safety-critical domains". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 38–45. DOI: 10.1109/CVPRW53098.2021.00013.

[LGH+21b] Maria Lyssenko, Christoph Gladisch, Christian Heinzemann, Matthias Woehrle, and Rudolph Triebel. "Instance segmentation in carla: methodology and analysis for pedestrian-oriented synthetic data generation in crowded scenes". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 988–996. DOI: 10.1109/ICCVW54120.2021.00115.

[LLF+20a] Z. Liu, T. Lian, J. Farrell, and B. A. Wandell. "Neural network generalization: the impact of camera parameters". In: *IEEE Access* 8 (2020), pp. 10443–10454. DOI: 10.1109/ACCESS.2020.2965089.

[LLF+20b] Zhenyi Liu, Trisha Lian, J. Farrell, and B. Wandell. "Neural network generalization: the impact of camera parameters". In: *IEEE Access* 8 (2020), pp. 10443–10454. DOI: 10.1109/ACCESS.2020.2965089.

[LUT+17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40 (2017). DOI: 10.1017/S0140525X16001837.

[LWZ+16] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. "Crowdsourced data management: a survey". In: *IEEE Transactions on Knowledge and Data Engineering* 28.9 (2016), pp. 2296–2319. DOI: 10.1109/TKDE.2016.2535242.

[LYS+17] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. "Learning from noisy labels with distillation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1910–1918. DOI: 10.1109/ICCV.2017.211.

[Mah36] Prasanta Chandra Mahalanobis. "On the generalized distance in statistics". In: National Institute of Science of India. 1936. DOI: 10.1007/s13171-019-00164-5.

[MBM18] T. Menzel, G. Bagschik, and M. Maurer. "Scenarios for development, test and validation of automated vehicles". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. June 2018, pp. 1821–1827. DOI: 10.1109/IVS.2018.8500406.

[MGE11] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. "Ensemble of exemplar-svms for object detection and beyond". In: *2011 International conference on computer vision*. IEEE. 2011, pp. 89–96. DOI: 10.1109/ICCV.2011.6126229.

[MMX+17] Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. "Self-paced co-training". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2275–2284. DOI: 10.5555/3305890.3305916.

[NCC+18] Tiago S. Nazaré, Gabriel B. Paranhos da Costa, Welinton A. Contato, and Moacir Ponti. "Deep convolutional neural networks and noisy images". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by Marcelo Mendoza and Sergio Velastín. Cham: Springer International Publishing, 2018, pp. 416–424. ISBN: 978-3-319-75193-1. DOI: 10.1007/978-3-319-75193-1_50.

[NDR+13] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. "Learning with noisy labels". In: *Advances in neural information processing systems* 26 (2013). DOI: 10.5555/2999611.2999745.

[NOR+17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. "The mapillary vistas dataset for semantic understanding of street scenes". In: *International Conference on Computer Vision (ICCV)*. 2017. DOI: 10.1109/ICCV.2017.534. URL: https://www.mapillary.com/dataset/vistas.

[PEZ+20] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. "Contrastive learning for unpaired image-to-image translation". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 319–345. ISBN: 978-3-030-58545-7. DOI: 10.1007/978-3-030-58545-7_19.

[PJX+20] Ishaan Paranjape, Abdul Jawad, Yanwen Xu, Asiiah Song, and Jim Whitehead. "A modular architecture for procedural generation of towns, intersections and scenarios for testing autonomous vehicles". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 162–168. DOI: 10.1109/IV47402.2020.9304625.

[RAK22] Stephan R. Richter, Hassan Abu Al Haija, and Vladlen Koltun. "Enhancing photorealism enhancement". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–1. DOI: 10.1109/TPAMI.2022.3166687.

[RBK+21] Julia Rosenzweig, Eduardo Brito, Hans-Ulrich Kobialka, Maram Akila, Nico M Schmidt, Peter Schlicht, Jan David Schneider, Fabian Hüger, Matthias Rottmann, Sebastian Houben, et al. "Validation of simulation-based testing: bypassing domain shift with label-to-image synthesis". In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE. 2021, pp. 182–189. DOI: 10.1109/IVWorkshops54471.2021.9669248.

[RF18] Joseph Redmon and Ali Farhadi. "Yolov3: an incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018). DOI: 10.48550/arXiv.1804.02767.

[RHG+15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. DOI: 10.1109/TPAMI.2016.2577031.

[RHK17] S. R. Richter, Z. Hayder, and V. Koltun. "Playing for benchmarks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2232–2241. DOI: 10.1109/ICCV.2017.243.

[RLA+14] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. "Training deep neural networks on noisy labels with bootstrapping". In: *arXiv preprint arXiv:1412.6596* (2014). DOI: 10.48550/arXiv.1412.6596.

[RSM+16a] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. "The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3234–3243. DOI: 10.1109/CVPR.2016.352.

[RSM+16b] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. "The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes". In: *Proceedings of the IEEE conference on*

computer vision and pattern recognition. 2016, pp. 3234–3243. DOI: 10.1109/CVPR.2016.352.

[RST+20] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. "Lgsvl simulator: a high fidelity simulator for autonomous driving". In: *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*. IEEE. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294422.

[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*. 2019, pp. 658–666. DOI: 10.1109/CVPR.2019.00075.

[RV19] Suman V. Ravuri and Oriol Vinyals. "Seeing is not necessarily believing: limitations of biggans for data augmentation". In: *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*. New Orleans, LA, USA, June 2019, pp. 1–5.

[RVR+16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. "Playing for data: ground truth from computer games". In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 102–118. ISBN: 978-3-319-46475-6. DOI: 10.1007/978-3-319-46475-6_7.

[RZY+18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. "Learning to reweight examples for robust deep learning". In: *International conference on machine learning*. PMLR. 2018, pp. 4334–4343. DOI: 10.48550/arXiv.1803.09050.

[SAS+18] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. "Effective use of synthetic data for urban scene semantic segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 84–100. DOI: 10.48550/arXiv.1807.06132.

[SBF+22] Thomas Stauner, Frederik Blank, Michael Fürst, Johannes Günther, Korbinian Hagn, Philipp Heidenreich, Markus Huber, Bastian Knerr, Thomas Schulik, and Karl-Ferdinand Leiß. "Synpeds: a synthetic dataset for pedestrian detection in urban traffic scenes". In: *Proceedings of the 6th ACM Computer Science in Cars Symposium*. CSCS '22. Ingolstadt, Germany: Association for Computing Machinery, 2022. ISBN: 9781450397865. DOI: 10.1145/3568160.3570230.

[SDF12] Hao Su, Jia Deng, and Li Fei-Fei. "Crowdsourcing annotations for visual object detection". In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012. DOI: 10.1145/3387168.3387242.

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. "Training region-based object detectors with online hard example mining". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 761–769. DOI: 10.1109/CVPR.2016.89.

[SGH20] Qutub Syed Sha, Oliver Grau, and Korbinian Hagn. "Dnn analysis through synthetic data variation". In: *Computer Science in Cars Symposium*. CSCS '20. Feldkirchen, Germany: Association for Computing Machinery, 2020. ISBN: 9781450376211. DOI: 10.1145/3385958.3430479.

[SGZ+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016), pp. 2234–2242. DOI: 10.5555/3157096.3157346.

[SKR+21] Franziska Schwaiger, Maximilian Henne Fabian Küppers, Felippe Schmoeller Roza, Karsten Roscher, and Anselm Haselhoff. "From Black-box to White-box: Examining Confidence Calibration under Different Conditions". In: *Proceedings of the AAAI Conference on Artificial Intelligence Workshops*. virtual conference, Feb. 2021. DOI: 10.48550/arXiv.2101.02971.

[SLA+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897. DOI: 10.48550/arXiv.1502.05477.

[SQC17] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. "An analysis of iso 26262: using machine learning safely in automotive software". In: *arXiv preprint arXiv:1709.02435* (2017). DOI: 10.4271/9780768002683.

[SS20] Gesina Schwalbe and Martin Schels. "Concept Enforcement and Modularization As Methods for the ISO 26262 Safety Argumentation of Neural Networks". In: *Proceedings of the European Congress Embedded Real Time Software and Systems (ERTS)*. Jan. 2020. DOI: 10.20378/irb-47276.

[SSH20] Timo Sämann, Peter Schlicht, and Fabian Hüger. "Strategy to increase the safety of a dnn-based perception for had systems". In: *arXiv preprint arXiv:2002.08935* (2020). DOI: 10.48550/arXiv.2002.08935.

[Stu14] Peter Sturm. "Pinhole camera model". In: *Computer Vision: A Reference Guide*. Ed. by Katsushi Ikeuchi. Boston, MA: Springer US, 2014, pp. 610–613. ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_472.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations (ICLR)*. 2015. DOI: 10.48550/arXiv.1409.1556.

[THS+17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial discriminative domain adaptation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2962–2971. DOI: 10.1109/CVPR.2017.316.

[THS+18] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker. "Learning to adapt structured output space for semantic segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7472–7481. DOI: 10.1109/CVPR.2018.00780.

[Tin00] Kai Ming Ting. "A comparative study of cost-sensitive boosting algorithms". In: *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer. 2000. DOI: 10.5555/645529.657944.

[TP12] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[Vah17] Arash Vahdat. "Toward robustness against label noise in training deep discriminative neural networks". In: *Advances in Neural Information Processing Systems* 30 (2017). DOI: 10.48550/arXiv.1706.00038.

[VSN+18] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C V Jawahar. *Idd: a dataset for exploring problems of autonomous navigation in unconstrained environments*. 2018. DOI: 10.48550/arXiv.1811.10200.

[VSN+19] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. "Idd: a dataset for exploring problems of autonomous navigation in unconstrained environments". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1743–1751. DOI: 10.1109/WACV.2019.00190.

[WEG+] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. *Torcs, the open racing car simulator*. URL: http://torcs.sourceforge.net.

[WKM+19] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[WPC20] Mingyun Wen, Jisun Park, and Kyungeun Cho. "A scenario generation pipeline for autonomous vehicle simulators". In: *Human-centric Computing and Information Sciences* 10 (2020), pp. 1–15. DOI: 10.1186/s13673-020-00231-z.

[WSC+20] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. "Deep high-resolution representation learning for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* (2020). DOI: 10.1109/TPAMI.2020.2983686.

[WU18] Magnus Wrenninge and Jonas Unger. *Synscapes: a photorealistic synthetic dataset for street scene parsing*. 2018. DOI: 10.48550/arXiv.1810.08705.

[WXJ+18] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. "Repulsion loss: detecting pedestrians in a crowd". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7774–7783. DOI: 10.1109/CVPR.2018.00811.

[YCW+20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. "Bdd100k: a diverse driving dataset for heterogeneous multitask learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645. DOI: 10.1109/CVPR42600.2020.00271.

[ZBO+16] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. "How far are we from solving pedestrian detection?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016. DOI: 10.1109/CVPR.2016.141.

[ZBS17] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. "Citypersons: a diverse dataset for pedestrian detection". In: *CVPR*. 2017. DOI: 10.1109/CVPR.2017.474.

[ZWB+18] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. "Occlusion-aware r-cnn: detecting pedestrians in a crowd". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 637–653. DOI: 10.1007/978-3-030-01219-9_39.

[ZXW+19] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z Li, and Guodong Guo. "Widerperson: a diverse dataset for dense pedestrian detection in the wild". In: *IEEE Transactions on Multimedia* 22.2 (2019), pp. 380–393. DOI: 10.1109/TMM.2019.2929005.